Municipal Infrastructure Anomaly and Defect Detection

David Abou Chacra
Systems Design Engineering Department
University of Waterloo
Waterloo, Canada
d2abouch AT uwaterloo.ca

John Zelek
Systems Design Engineering Department
University of Waterloo
Waterloo, Canada
jzelek AT uwaterloo.ca

***Abstract*—Road quality assessment is a key task in a city's duties as it allows a city to operate more efficiently. This assessment means a city's budget can be allocated appropriately to make sure the city makes the most of its usually limited budget. However, this assessment still relies largely on manual annotation to generate the Overall Condition Index (OCI) of a pavement stretch. Manual surveying can be inaccurate, while on the other side of the spectrum a large portion of automatic surveying techniques rely on expensive equipment (such as laser line scanners). To solve this problem, we propose an automated infrastructure assessment method that relies on street view images for its input and uses a spectrum of computer vision and pattern recognition methods to generate its assessments. We first segment the pavement surface in the natural image. After this, we operate under the assumption that only the road pavement remains, and utilize a sliding window approach using Fisher Vector encoding to detect the defects in that pavement; with labelled data, we would also be able to classify the defect type (longitudinal crack, transverse crack, alligator crack, pothole ... etc.) at this stage. A weighed contour map within these distressed regions can be used to identify exact crack and defect locations. Combining this information allows us to determine severities and locations of individual defects in the image. We use a manually annotated dataset of Google Street View images in Hamilton, Ontario, Canada. We show promising results, achieving a 93% F1-measure on crack region detection from perspective images.**

***Index Terms*—Road Quality; Computer Vision; Machine Learning; Pattern Recognition**

I. Introduction

Road infrastructure is often a main concern for any city, province, or country and maintaining it is critical. The American Society of Civil Engineers (ASCE) calculated that $91 billion a year is invested in road infrastructure in the United States alone [1]. However, it's estimated that despite the large sums of money invested on road maintenance, road quality is actually predicted to decline in major countries such as Canada [2] and the United States [1]. Cities often have asset management plans in place that allow them to monitor the condition of their roads and maintain them. In the case of Canadian municipalities, 71% perform data collection on their roads at least once every five years [2]. This allows cities to allocate their limited funds optimally.

Road quality assessment is currently done in one of two ways. The first is via surveyors who drive along the roads, check their condition, and note defects either manually or using a specialized device. The second is a semi-automated method utilizing specialized equipment attached to a truck or van. Both of these methods can be costly and come with their own drawbacks. This makes road quality assessment an expensive and time-consuming task, especially in larger municipalities. In addition, different countries, states, or even cities use different standards for their assessments, which can mean that methods that work in one instance won't transfer directly to another.

Fig. 1: Sample final result of our algorithm. The overall defect severity is shown in the blue box, while its pixel-wise extent and severity is color coded and overlaid.

We can forgo the problem of gathering data by tapping into street view(SV) databases which are updated regularly. Google Street View, for example, contains large amounts of data and is updated approximately every two years for large cities. These kinds of databases are accessible to the public, and we can automate mining of street view data to assess municipal assets and infrastructure. When coupled with image processing on these images, this can prove to be the first step to a low-cost and efficient solution.

Our focus, then, is to be able to perform road quality assessment on street view images. Road texture can be a powerful cue for this problem, since good quality road and damaged road can be differentiated using texture descriptors. As such, we built upon the describable texture dataset (DTD) methods which achieve the state-of-the-art in texture detection [3]. In brief, our algorithm works by first detecting the visible road in the image, see [4]. Then, we densely sample windows from the detected road in the input image, and calculate SIFT descriptors in those windows. These descriptors are then

ISBN 978-90-827970-1-5

encoded using the Fisher vector formulation [5]. The encoded features are then classified using an SVM classifier and separated into "good quality" and "damaged" road. This is followed by a window-by-window voting scheme to generate a proposed region of poor quality road. This algorithm is described fully in our previous work [4]. Specific cracks are detected within damaged regions by utilizing an ultrametric contour map (UCM) [6], a weighted edge map, inside damaged road regions. For this work, we build upon the region and contour proposals to generate pixel-wise annotations for defects along with their severities, as well as to classify defect instances and find their overall severity. We also explore the use of deep learning to solve this problem in a scarce-data domain. Deep learning techniques are likely to outperform any method using hand-crafted features when data is readily availiable, however, in our case, our data was limited. We employed a modified U-net [7] architecture and measured its performance on the task at hand.

Our damaged region detection algorithm achieves an F-measure (combination of precision and accuracy) of over 93%. The algorithm is tested on a subset of Google Street View Images which were manually collected and annotated, to demonstrate the method's effectiveness on a commercial SV database. The algorithm proves to be accurate and properly detects cracks and damaged road, as seen in Figure 1. On the other hand, the deep learning approach was not able to learn the feature space with the limited amount of data it was given and resulted in and F1 score of 55%, however it exhibited interesting behaviour by maximizing its activations around specific defects within labelled areas.

II. Road Quality Assessment

The frequency of performing road quality assessment led to a good deal of work aimed at automating it. The prevalent automation methods involve attaching an apparatus to a truck that also drives along the city [8], [9]. This apparatus can be attached to several locations on the vehicle, for example in tow, on the top or bottom of the vehicle. The main sensing portion contains a combination of one or more cameras pointed orthogonal to the road, LIDAR or LASER scanners, GPSs, IMU's, and accelerometers, among others. Notable among these is the Laser Crack Measurement System (LCMS) produced by Pavemetrics. The information from these sensing apparati is aggregated and different vision and signal processing methods are used to determine the pavement quality. These methods are quite powerful and yield highly precise results. CrackIT [9] merges multiple preprocessing techniques and clustering algorithms (K-means, Gaussian Mixture Models, among others) to detect cracks and their types and also yields favourable results with a 93.5% F-measure for their best performing algorithm. The LCMS comes with proprietary software to aid in PCI generation, it utilizes all aspects of the LCMS, from the texture and LIDAR information (to detect and measure crack depth), to the IMU information that detects rutting. The main caveat is that the input sensory data can be expensive and tedious to procure. The sensing apparatus has a limited scope of vision, specifically 4 meters wide for the Pavemetrics devices. It is usually "looking down" at the road (pointing orthogonal to the road), so the vehicle still needs to drive up and down every road lane in the area to be checked. This also means parts of the road that are not surveyed by the vehicle will not be sensed, such as portions with parked cars, or portions blocked off, or even road shoulders. Sensing requires high resolution cameras (almost 1 pixel per millimetre of road in the case of the LRIS), and expensive LIDAR scanners, so there are usually only a few sensing vehicles that can do the sensing, which would require more time in large areas.

A few other methods utilize street view images for road quality assessment. In their work, Varadharajan et al. use a dashboard camera and drive around the city of Pittsburgh, Pennsylvania to collect their own data, essentially similar to street view data [10]. They detect the ground plane, then over-segment it into superpixels and generate various descriptors for these superpixels which give way to a binary classifier used to identify cracks inside the superpixels. A different approach filters the input image rigorously by background subtraction followed by wavelet-based de-noising, then, locally thresholds patches from the image (by utilizing Otsu's method to quickly find these local thresholds) resulting in a binary image [8]. A survey of older methods [11] highlights the fact that these crack detection methods operate locally and can be fooled by the low signal-to-noise ratio (SNR) in pavement images. The work in [12] utilizes Google Street View specifically, and its authors generate a database of over 700K images coupled with their rating(poor, fair, or good) which was inferred from the public records of the New York City Department of Transportation. They utilize a method similar to [3] to classify whole images as containing poor, fair, or good roads.

Since we're looking to detect individual defects in street-view images, we resorted to creating our own database by manually annotating pixles in street view images as either damaged road or good quality road. This prevented us from being able to compare our results to any of the methods presented in this review as either the datasets they tested on are not publicly available, or their datasets do not provide adequate fine-grained information for image segmentation [12]. With this in mind, our work focuses on finding any and all road defects in an input image, with the knowledge that classifying these defects can be done per the methods mentioned here.

III. Algorithm and Methodology

Our crack detection algorithm, based on [4], is visualized and described in Figure 2. In brief, our algorithm densely samples windows from the detected road in the input image then finds SIFT descriptors in those windows. These descriptors are then encoded using the Fisher Vector formulation. The encoded features are then classified using an SVM classifier into "good quality" and "damaged" road. This is followed by a window-by-window voting scheme utilizing Gaussian windows to generate a segmentation that detects damaged road regions. Note that roads are initially detected via a

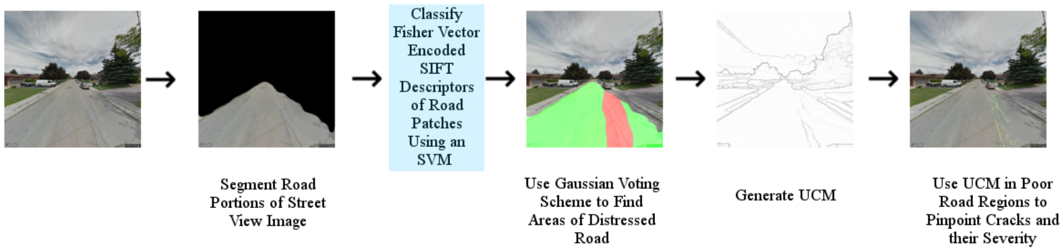

Fig. 2: A visualization of our crack detection algorithm. We first segment the image into 'road' and 'background' using the method of [4]. We then single out the Fisher vectors of the road pixels only. The Fisher vectors are passed through an SVM to classify the vectors as belonging to 'good road' or 'distressed road'. A voting scheme using patches of Gaussians is employed to generate a segmentation. An ultrametric contour map [6] is computed for the image, and cracks are detected by finding locations of high UCM response within 'distressed road' regions. These cracks are further distinguished by isolating them and classifying them according to severity which is also deduced from the UCM response.

similar method which means the encoding has to be done only once, and we simply classify "road" and "background" with a different SVM classifier than the one used for classifying road quality.

We then utilize an ultrametric contour map (UCM), to detect defects precisely inside damaged road regions. This "structured forests" [6] method uses learned low level features and a decision forest to detect and weight specific pixels belonging to defects in an image and produces a UCM. The UCM goes beyond a regular edge map by weighting edges, thereby giving a likelihood that the pixel in question corresponds to a real defect. When combined with the segmentation found earlier we can further improve our road quality detection by generating probability maps of where cracks exist in the road. This can be done by detecting strong UCM responses inside regions classified as 'distressed road' to focus on the cracks present in the road. The weight of the UCM at a pixel is found to be reflective of its defect severity, where larger defects usually illicit a high UCM weight.

Finally, we isolate disconnected cracks into separate distress instances, and based on the the severities generated by the UCM, we calculate an overall severity rating for this defect. This step comes with the advantage of having adjustable weights that can accommodate for varying rating systems across the world by simply changing the cut-off thresholds for different severities.

To measure the performance of a deep learning approach to this task, we utilized a modified U-Net. U-Nets, originally described in [7], are known to train well with limited data, which is why we settled on them as opposed to other deep learning architectures. The U-Net [7] architecture employs an encoder-decoder framework, doubling the feature map depth with each max-pool and uses skip connections between encoder feature maps and the corresponding decoder feature maps to generate a segmentation map for every class in the training set. U-Nets originally were used on medical images, and required few training examples, however they are usable with natural images as well. We opted to modify a U-Net for our work, as we desired a network employing encoder-decoder architecture, as well as one that can be trained rapidly and using a relatively small dataset.

IV. Results and Discussions

We tested our crack detection algorithm on the subset of images from Google Street View, obtained by querying the Google Street View API in the Hamilton region. We annotated a total of 250 images using LibLabel [13], and used 150 of the images for training, 50 for validation, and 50 for testing. We manually annotated the road in the images into regions of 'good', 'medium', and 'poor' road. 'Good' road is defect free, whereas 'medium' road contains small and medium cracks and defects, and 'poor' road contains potholes and larger cracks and defects. We noticed only a few samples contained 'poor' road, which was insufficient for training an SVM, and decided to merge 'medium' and 'poor' road labels into one 'distressed' road label.

For our deep learning experiments, we employed a 4-layer U-Net (4 layers of convolution, followed by 4 layers of transposed convolution) with a starting filter bank size of 64. We explored several other configurations, but this was the highest performing one. We used an ADAM optimizer with a fixed learning rate of 0.001, and a mini-batch size of 10 training for 500 epochs on our training set. Due to GPU memory requirements images were resized from 640x640 into 320x320 images for training. We trained two separate modes, one for detecting roads only and one for detecting defects only. While detecting roads is unnecessary for this problem, we decided to measure performance on this task as a potential input to our main algorithm.

The metrics used for evaluation are precision, recall, false-positive rate (FPR), and F-measure. The metrics are calculated in a pixel-wise manner using true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Positive samples are the damaged road samples, whereas negative samples are the good road samples. The metrics are calculated as $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$, $\text{FPR} = \frac{\text{FP}}{\text{FP+TN}}$, $\text{F} - \text{measure} = \frac{\text{Precision} \times \text{Recall}}{(1-\alpha)\times\text{Precision}+\alpha\times\text{Recall}}$, with $\alpha = 0.5$.

Sample results of our main algorithm are shown in Figure 4, and our numerical results are shown in Tables I and II. We show a sample of the U-Net results in Figure 3.

While Table II cites results by other authors, a direct comparison of our method with theirs is not possible. The lack of a standardized dataset for road quality assessment results in every work being tested on a custom dataset, independent of other work in the field. Moreover, metrics are measured differently throughout various works. For example, the works in [8], [10], [12], [14] don't report the results of crack detection, but an overall road quality assessment for a whole image, making comparing metrics between our method and theirs inconclusive. While a majority of researchers that work on crack detection report their metrics on cracks in a manner similar to ours [15], [16], since their images are in a different image space (mainly top-down images, as opposed to perspective images as with our work) these metrics can also be skewed since the metrics are based on percentages of correctly/incorrectly labelled pixels. With the exception of [15], the methods we reviewed did not offer any source code to experiment on our dataset with, and we could only extract the results cited in their work.

We argue that the results in Table I are more useful. These results demonstrate the ability of Fisher vector encoded features and an SVM in detecting damaged road in perspective images with high accuracy. This proves that a road assessment system based off of street view data is indeed viable, and points to the fact that we can, and have, emulated a crucial portion of a surveyor's process of road assessment. The SIFT features coupled with the Fisher vector encoding have transformed the image space into one where road defects are distinguishable from good quality road using an SVM. This feature space is highly correlated with image "texture" features as explored in [3], and we believe it is an effective feature space for the problem of road quality assessment.

Qualitatively, Figure 4 paints a clear picture of what the algorithm can do. The algorithm detects defects and its measurement of "severity" peaks around the center of a defect. A simple non-maximum suppression step allowed us to shrink the large areas into skeletons of single-pixel thickness. This skeleton is the true fruit of our results, as it can be used to classify the defect as a longitudinal or transverse crack and to measure crack length and width. Furthermore, an easily detectable pattern of cracks represented by the skeleton can be deduced to be alligator cracking, while a circular pattern is likely to be a pothole.

The deep learning architecture suffered from lack of data, and was unable to properly learn the class of "defective road". This was more than likely due to the common problem of class-imbalance where the network was more interested in minimizing its loss by perfecting the common "background" class over the rarer "defective road" class. A future step in mitigating this could be experimenting with a modified loss function to penalize the network more for missing a defect. However, there was a slight upside to the network being very particular about what it labels as "defective road" in that the network learned to hone in on the specific defect at even finer levels than hand-labelled data. In a sense the network was treating the human labels as "weak labels", which indeed they were as a the labelled regions often included good quality road surrounding the defect. This could mean that annotating future datasets for this kind of architecture can be made easier since annotators don't have to follow an individual defect precisely, but simply annotate a rectangle or loose shape around it.

Fig. 3: The original image(top) and the output activations of our trained U-Net overlaid over the image in the jet colorspace, red implies a larger activation, while blue is a smaller activation. A notable trait of the U-Net was its ability to maximize its activations around the defect itself.

TABLE I: Results for Defect Region Proposal

Method	F-measure	Precision	Recall	FPR
Ours	93.12%	93.48%	92.77%	2.93%
U-Net	55.91%	43.36%	78.68%	6.04%

TABLE II: Results for Individual Crack Detection

Method	F-measure	Precision	Recall	FPR
Our Method	92.84%	86.64%	100%	0.02%
Oliviera [15]	93.5%	92.2%	95.5%	4.5%
Marques [16]	93.07%	88.52%	98.85%	-%

V. Conclusion

Despite not having quantitative results to compare against, we saw that the method was able to single out cracks and poor road regions well, and perform quite favourably. The recall of 100% is an indication of the performance of the initial crack segmentation step. The high precision of the road segmentation algorithm [4] provided us with a reliable segmentation on which to detect cracks. We can also see that the portions of the road the segmentation algorithm missed on, in the distance and on the boundaries, were not detrimental to the crack detection algorithm. The precision could be improved by making the crack detection algorithm stricter, however, we did not deem that necessary after qualitatively assessing the algorithm output. Despite the low quality and coarseness of our manually labelled data, the classifier was able to truly learn what 'distressed' road looks like. Texture was the key cue in detecting poor roads, and it is the one we utilized, and it predictably generated good results. Whether the data is collected from an online database like Google Street View, or on a municipal level, the method is

ISBN 978-90-827970-1-5

Fig. 4: Sample crack detection results of our algorithm. The crack detection scheme displays per-pixel severities of detected defects in blue and green for ‘mild’ defects, orange for ‘medium’ defects and red for ‘severe’ defects. The blue boxes contain information about the severity of the defect instance as a whole, using information collected from the per-pixel severities.

still applicable. As long as the images show these distress artefacts, the algorithm can be trained to detect them. On the other hand, a deep learning algorithm may be difficult to apply in such a data-scarce manner, and would require significantly more data to become practically applicable.

REFERENCES

[1] A. S. of Civil Engineers, “Report card for america’s infrastructure,” ”http://www.infrastructurereportcard.org/a/documents/Roads.pdf”, 2013.

[2] CIRC, “Canadian infrastructure report card,” Canadian Construction Association, Canadian Public Works Association, Canadian Society for Civil Engineering, Federation of Canadian Municipalities, Tech. Rep., 2016.

[3] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, “Describing textures in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3606–3613.

[4] D. A. Chacra and J. Zelek, “Fully automated road defect detection using street view images,” in *14th Conference on Computer and Robot Vision (CRV) 2017*, 2017.

[5] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[6] P. Dollár and L. C. Zitnick, “Structured forests for fast edge detection,” in *IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[7] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[8] G. Bao, “Road distress analysis using 2d and 3d information,” Ph.D. dissertation, The University of Toledo, 2010.

[9] H. J. M. Oliveira, “Crack detection and characterization in flexible road pavements using digital image processing,” Ph.D. dissertation, Universidade de Lisboa - Instituto Superior Tcnico, 2013.

[10] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, “Vision for road inspection,” in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 115–122.

[11] Y.-C. Tsai, V. Kaul, and R. M. Mersereau, “Critical assessment of pavement distress segmentation methods,” *Journal of transportation engineering*, vol. 136, no. 1, pp. 11–19, 2009.

[12] K. Ma, M. Hoai, and D. Samaras, “Large-scale continual road inspection: Visual infrastructure assessment in the wild,” in *Proceedings of British Machine Vision Conference*, 2017.

[13] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3d traffic scene understanding from movable platforms,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[14] C. Mertz, S. Varadharajan, S. Jose, K. Sharma, L. Wander, and J. Wang, “City-wide road distress monitoring with smartphones,” in *Proceedings of ITS World Congress*, September 2014.

[15] H. Oliveira and P. L. Correia, “Automatic road crack detection and characterization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, 2013.

[16] A. Marques and P. L. Correia, “Automatic road pavement crack detection using svm,” *Lisbon, Portugal: Dissertation for the Master of Science Degree in Electrical and Computer Engineering at Instituto Superior Técnico*, 2012.

ISBN 978-90-827970-1-5 © EURASIP 2018