Math INDUSTRY

Aerium Analytics

AI for road surface scanning

Amit Jha, Avleen Kaur, Aysa Fakheri Tabrizi, Hamid Hamidi, Parham Hamidi, Xiaowei Li

Industry mentor: Heather Vooys

Abstract

This analysis is to develop a machine learning tool that detects cracks on road pavements by processing RGB aerial images of the road. The data considered is unlabelled. Thus supervised and unsupervised learning approaches are applied to design an intelligent detector. The results are combined to produce optimal feedback.

1. Introduction

Artificial intelligence (AI) has been a valuable mechanism for the growth of many industries, such as education, health care, lifestyle, transportation, web search engines, etc. It can be applied to enforce road safety by monitoring the condition of the roads, which is also the goal of this project. Since roads are the primary mode of transportation, they are essential for everyone. The materials mainly used for the construction of roads are concrete and asphalt. The road pavements can develop defects such as cracks and potholes due to fatigue from excessive usage, extreme changes in weather temperatures, water accumulation, etc. Such defects are treated according to their severity to ensure safe travel. Otherwise, they could degrade the quality of roads and cause more irreversible damage. Traditionally, the inspection of pavements is done by a team physically. Or a human expert analyzes the aerial images of road pavements for surveillance.Since there could be numerous pictures for a particular province, having a human expert go over each one of the images is a laborious task. It could also pose a financial strain on the government.

This project develops a machine learning or computer vision tool, which reads the images of road pavements. It highlights the defects that are present on the pavements in the output images. The data provided for training and testing the machine learning model consists of RGB aerial images of road pavements of a parking lot and other roads. Such a model is highly beneficial for the government, as it reduces the need for human efforts. Consequently, the model is economical and is a much faster option. Time and money are two key factors that affect the performance of a task in the real world. On reducing these stressors, the government gains more resources to enforce measures for the correction of road pavements efficiently, see [1], [2].

The data provided for designing this model is unlabelled. It leads to two possibilities. The first one is labelling the data manually for designing this project, which is a supervised learning approach. Otherwise, we could train the model by using unlabelled data itself. It is called an unsupervised learning approach. We explore both of these approaches in this project to build a coherent tool. However, there are only fourteen images in the data set. We generate more images for increasing the data by using the techniques such as image segmentation and windowing, which split each image into several parts. The first stage creates several sub-images for the input image, and then each one of them is processed by the model. The resultant output images of each sub-image are combined back together to give the input image back in its original form at the end of the processing. Since the images are aerial, they contain several other elements such as cars, markings on the roads, poles, trees, shadows, etc. They all contribute as noise in data, which is the biggest challenge in developing this model. Moreover, the defects can be of a random shape, so it is difficult to train the model to detect such defects in the pavements. Some similar problems on detecting cracks have been studied in [3], [4], [5], [6].

This report is organized as follows. Firstly, section 2 presents a supervised learning model that highlights cracks on aerial images of road pavements is presented. The second approach, an unsupervised learning technique is depicted in section 4. The combination of the two techniques is described in section 5, Finally, section 6 discusses the results and conclusions of this study.

2. Approach 1: Supervised Learning

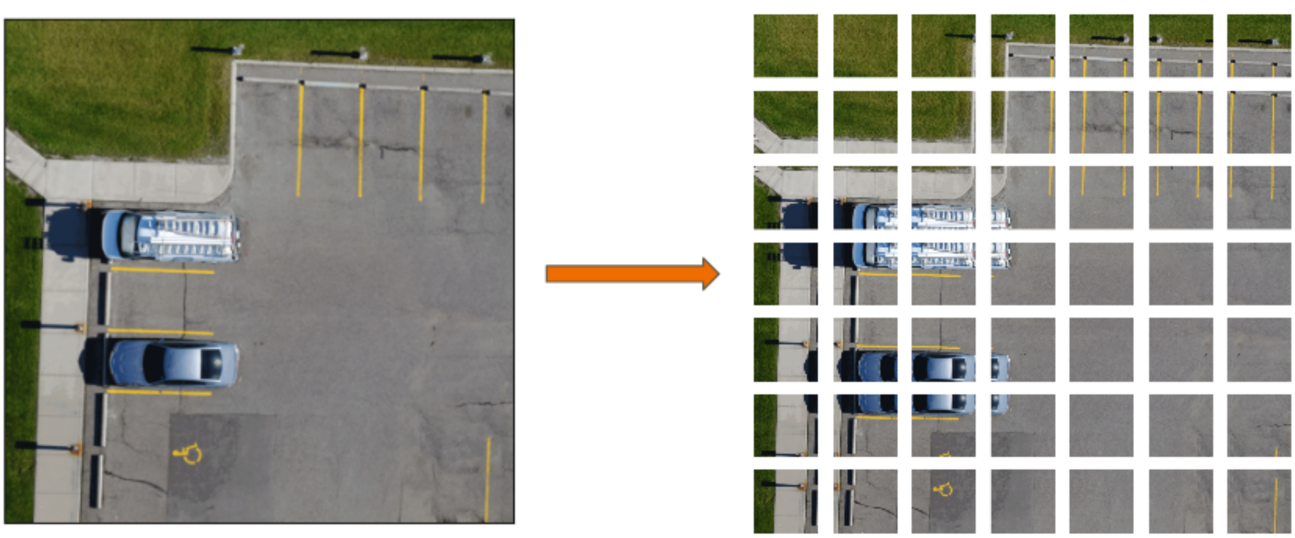
Background

Convolutional Neural Network (CNN) is known for its exceptional preformance on image recongnition. Our idea here is to train a CNN to "identify" the cracks in pictures. However, typical data sets for training CNNs usually contains tens of thousands of samples with labels. Here we have only 14 images and no "label" for the cracks. Moreover, a few images should be left out of training to use as test set for the trained model. In order to increase the sample size and label them, we do some pre-processing on original data set as detailed below.

Methodology

Creating tiles and overlapping patches

The original dimension of the pictures were mostly 3724 by 3724 pixels (with a few pictures having even more pixels). We turned each picture into overlapping tiles/patches with dimension 224 by 224 pixels. Overlapping tiles helped us capture each crack better (in case the boundry of a tile falls on a crack). Moreover, it helped us increase the data for validation and training.



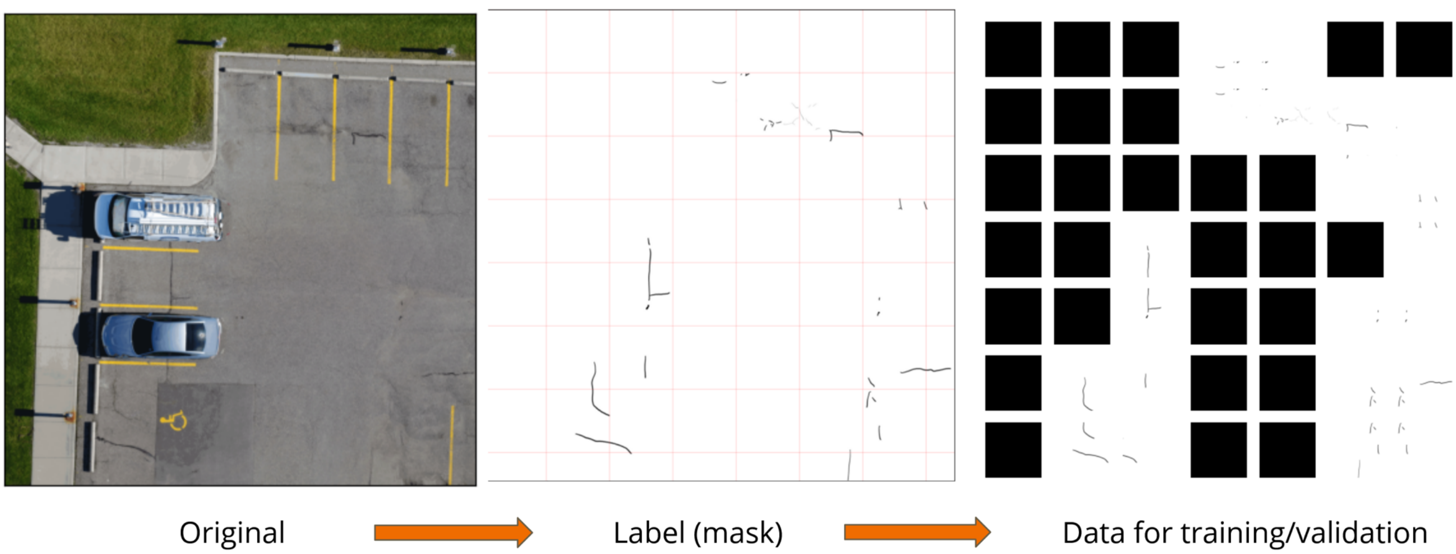
Labelling

The data that was provided to us by Aerium Analytics was not labelled. This is often the case in the industry. To train a Neural network with supervised learning, we needed to label our raw data. Our first task was to identify the cracks which we want to use for training our Neural network.

Before we start labelling, we had to decide what are the cracks we wanted our Neural network to detect. To do this, we looked at some of the previous worked that has been done, as well as a pavement distress identification manual [7]. Some of the cracks in our data were miniature and hence we decided that they are irrelevant for our training of crack detection.

After we came to consensus about the cracks we wanted to consider, we used a free graphics editor, called GIMP, to mask over the cracks. This was done by drawing over the sensible cracks with a digital pen. Then, the masks were turned into an appropriate file format (PNG) to be read by OpenCV.

We also formulate the problem into a binary classification. Using the masks we draw, each tile was labelled as positive or negative depending on whether the tile contains more than 800 pixels of cracks (`PIXEL_THRESHOLD=800`).



We note here that the actual data for traning and validation is different than the above figures. The above figures are only for demonestation of the process.

Training and validation set

We took 11 out of 14 images and created 4169 patches with about 15% overlap between adjacent tiles, among them 842 were labelled positive (i.e., containing cracks). Note that this is an unbalanced data set (about 20% samples are positive) and we will later address this issue in the loss function for training. We further randomly divided these 4169 patches into training set and validation set with a ratio of 6:4. During training, the training set is also augmented with a random horizontal and/or random vertical flip to obtain up to 4 times training samples.

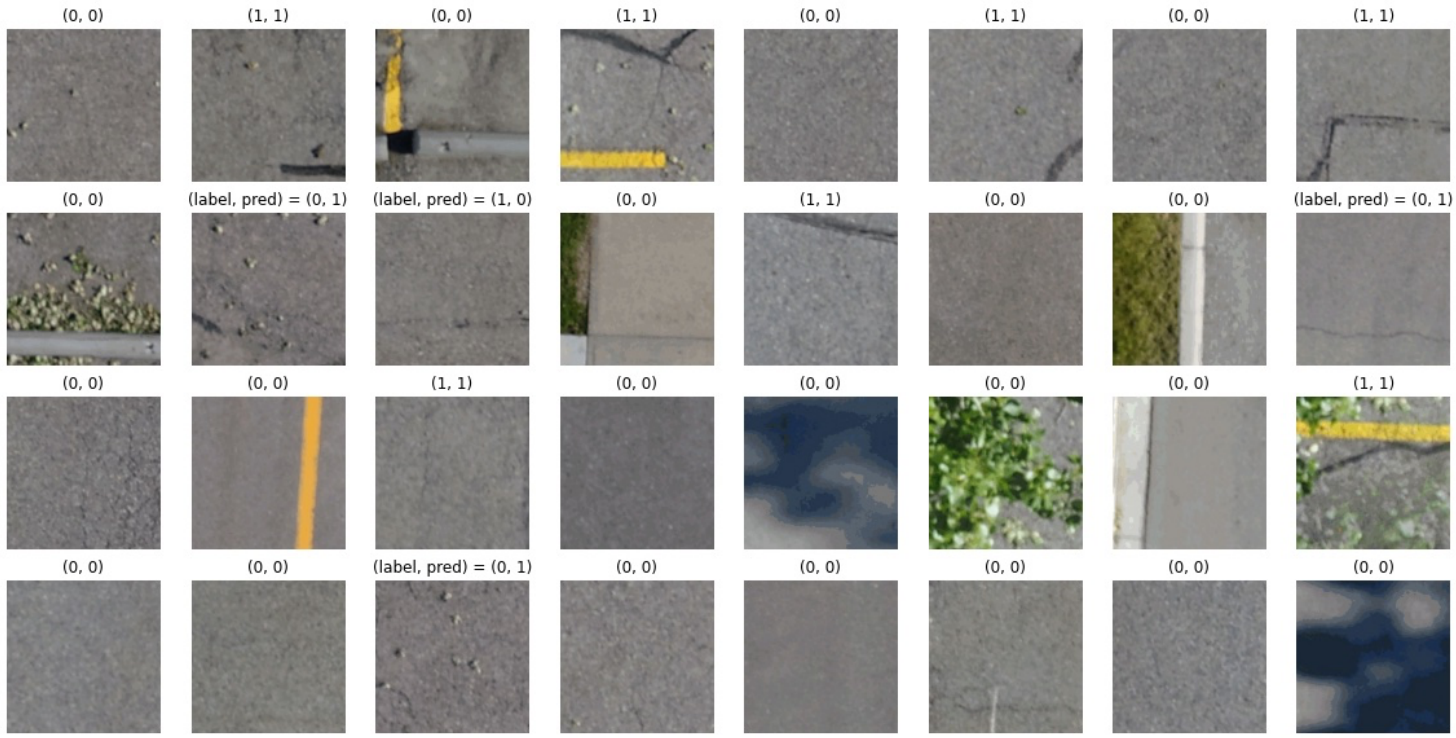
Test set

For the remaining 3 images, we turned them into patches of size 224 by 224 pixels with (almost) zero overlap. They will be used to show our results.

Training CNNs with transfer learning

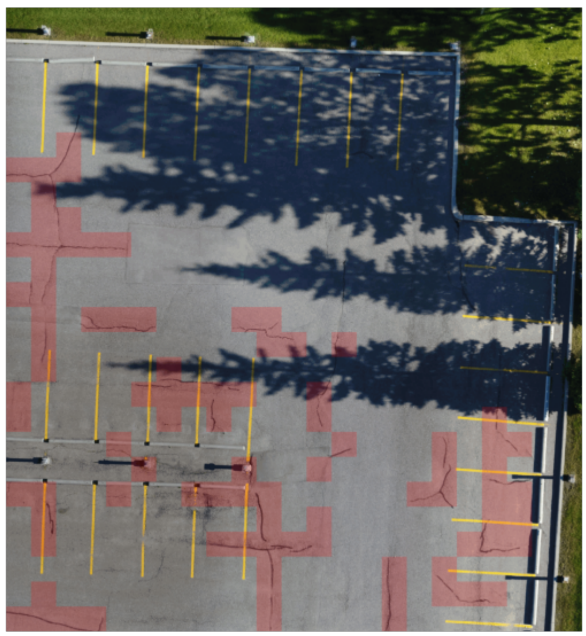
To further combat the issue with small data set, we use pre-trained networks as the backbone of our classifier. In particular, we took ResNet50 [8] and modified its last layer (with an added `Dropout=0.4`) to perform binary classification. The training process started with pre-trained weights (except last layer) and 'fine-tuned' the entire network. The loss function is the standard binary cross entropy, but with 4 times the weight on positive samples to 'balance' the data set. The optimizer is stochastic gradient descent with momentum. We ran this for up to 20 epochs and evaluate the accuracy on validation set. At last, we obtained a classifier with about 94% accuracy on the validation set, and the number of false positive and false negative samples on validation are 50 and 39 respectively. The training is done using Google Colab and the details can be found in this Colab notebook.

Below are some examples for this classifier's prediction on validation set.



Results

Below we show the predictions of our classifier on the 3 test images. Here red shaded patches are the ones our classifier thinks there is a crack in them. For the first image, there are no cracks in the image and no cracks are predicted. For the second image, the classifier is able to identify almost all patches with cracks, aside from a few false positives. For the third image, the classifier again identified most of the cracks. However, it missed almost all of the cracks in the shadow of trees. We believe this phenomenon is mostly due to there being very little such samples in traning set, and the model can be further improved in this scenario with more related training samples.



Further Remarks

- Overfitting
  During training, the ResNet50 model was able to reach an accuracy of above 98% on the training set while having about 94% accuracy on validation set. We also tried the same training approach on ResNet18 and ResNet101, and obtained similar accuracy on training and validation set. However, as we can see from the prediction examples on the validation set (e.g., last picture in the second row), our labelling itself is not 100% accurate. Thus we suspect there is some overfitting in the model(s) at this point.
- Freeze all but last layer
  We also tried freezing all the pre-trained network parameters and only re-train the last layer. In this case, the trained model performs a few percentage worse on the validation set when compared to the 'fine-tunning' model above. We believe one reason for this could be that there is not enough 'free' parameters to capture the complexity of training set. One future direction may be only to freeze some of the bottom layers and 'fine-tune' the upper layers. This could also help with the overfitting issue since now there are fewer parameters to train.
- Different patches sizes
  We also tried larger patch sizes (e.g., 448 by 448 pixels, 600 by 600 pixels, etc.) and found similar accuracy on training and validation set. In the end, we choose the smaller patch size because it can give more accurate location about where the cracks are.

3. Approach 2: Unsupervised Learning and Image Processing

Background

As the number of samples is limited, we considered unsupervised approaches simultaneous to the supervised approach. We applied unsupervised techniques consisting of two independent sections. One is the classical image processing techniques used in finding edges in images, including but not limited to filtering, morphological operations, and edge detection. However, the cracks with the edges of other objects such as cars are detected in our image processing steps. To overcome this problem, we tried another approach whic is clustering pixels using machine learning techniques. We used two different clustering algorithms to identify objects and remove unwanted parts of the image. In addition to object detection, clustering could be used to group pavements' pixels as a unique cluster among others. In the following, the classical and machine learning approaches would be described in detail.

Image processing techniques

The images contain pixel information as the values in different channels such as RGB, standing for Red, Green, and Blue, respectively. The pixel values in each channel could take a value between 0 to 255, which would be the intensity of that specific channel. To identify a crack in an image based on the channels' intensities, we have to suppress other features or highlight the feature of interest, the cracks, to an extent where it is easily identifiable by the edge detection method.

Image processing techniques mainly consists of 4 steps to delineate crack patterns:

1. Size reduction
2. Noise reduction and filtering
3. Edge detection
4. Morphological transformation

Size reduction

In our dataset, the images have the size of order 3000×3000. To reduce memory consumption and perform our operations faster, we reduced the sizes of the images to 800×800

Edge detection

The points (pixels) that have discontinuities in brightness are detected by mathematical operations as the edges in the image. Here, we applied the Canny edge detector to find the patterns of pavements in our dataset.

Morphological transformation

In the edge detection process, some points can be missed and cracks would be shown as discontinuous patterns. Therefore, morphological operations such as closing and erosion have been used in the last step of crack detection using image processing techniques.

Shadow removal

This is a complex problem that depends on many factors and camera settings. We tried implementing the paper by [9] which talks about shadow removal. The shadows can be removed if the log response of camera sensors is projected along an illumination invariant vector. The grayscale results were not very promising. We tried to improve the results provided at Stack..

Machine learning approache

We used two different algorithms for clustering the pixels, K-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

In the following, steps of both these appraoches and their results will be explained.

Methodology and Results

Noise reduction and filtering

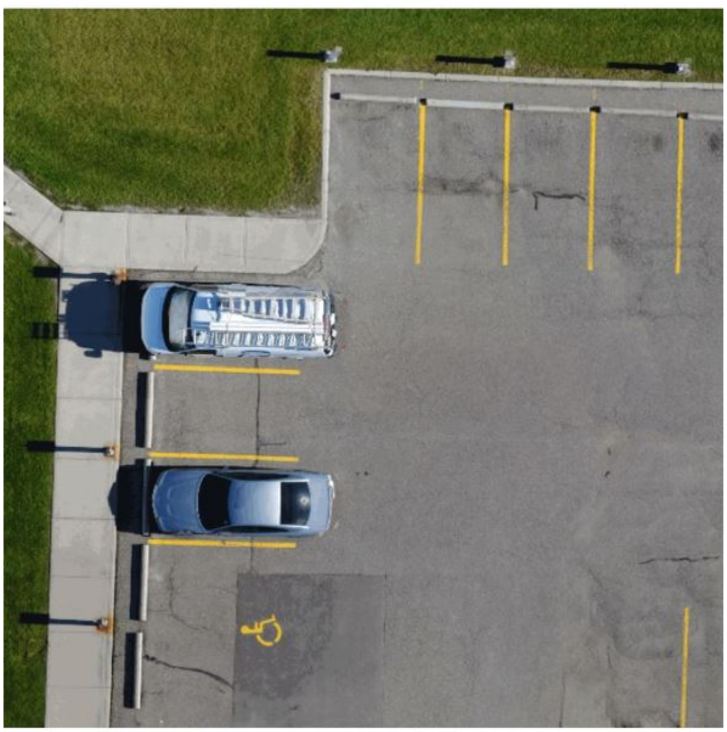
This is the first step in removing unwanted information from the image. Filters are mainly used to smooth images without losing edge precision. Here, we used distinctive filters that cut the high frequency by convoluting a kernel over the pixels. We used bilateral, median, Gaussian, and uniform blurring filters on images. Bilateral and gaussian blur preserve edges if appropriately applied.

We tried following sequences of operations on raw images.

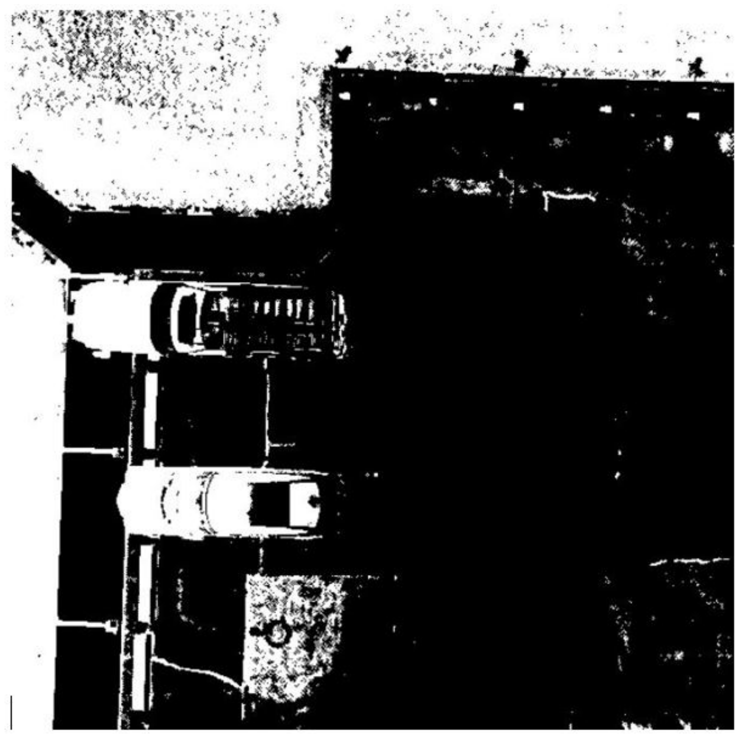
Method 1

We used bilateral blur on gray image followed by convolution with various kernels. After that we applied optimal threshold image binarization Otsu's algorithm. The results are the following:

- *RGB >Gray > Bilateral blur filter > Sharpen convolution > OTSU thrsholding.*

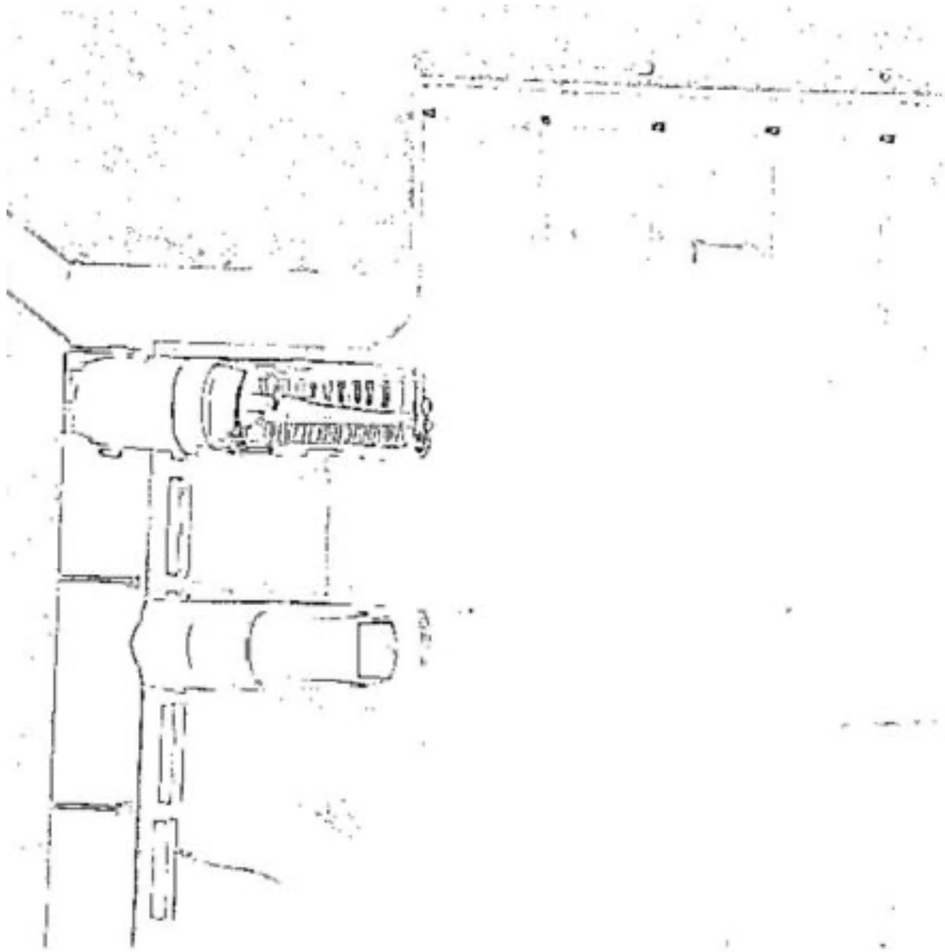


Original image



Bilateral blur + Sharpen + Thresholding

- *RGB >Gray > Bilateral blur filter > Laplacian convolution > OTSU thrsholding.*



Median blur + Laplacian + Thresholding

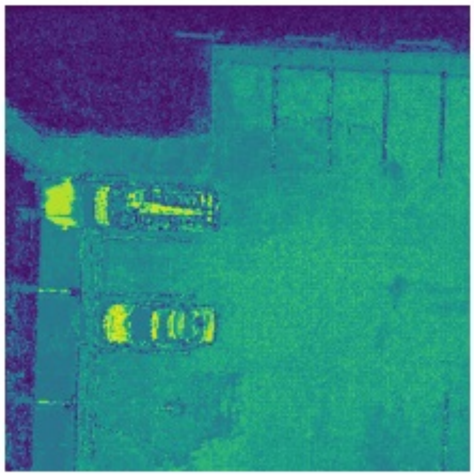
Method 2

We did some and operation value and saturation layers followed by thresholding.

- *RGB > HSV > Bitwise_and(V-Constant,S) > Otsu's thresholding*.

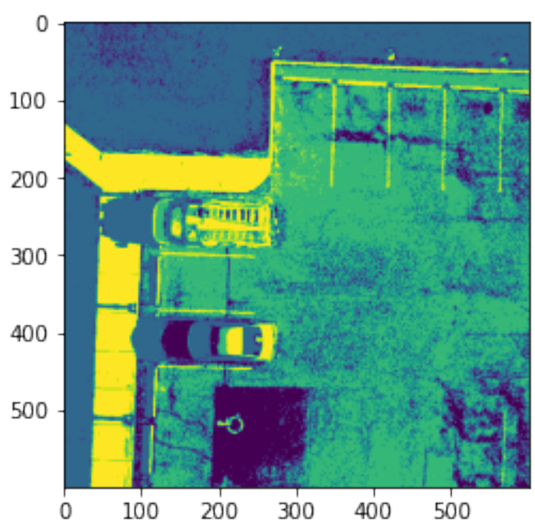
Method 3

We performed DBSCAN on the resized images without applying any filters. As shown in the following figure, the cracks can have a distinctive cluster among other clusters.



Method 4

We performed K-means on the resized images without applying any filters. As shown in the following figure, the cracks can have a distinctive cluster among other clusters. Also, other objects such as cars and grass can have a distinctive cluster.



4. Combined Approach

Based on the nature of supervised and unsupervised learning methods and their application in our problem and based on our observations from the experimental results, each of our two proposed methods has its own advantages and disadvantages in solving the crack detection problem.

Approach 1 advantages:

- Getting most out of the existing data by labeling them and using the labels in training the model.
- Capability of eliminating the redundant objects in the picture such as trees, cars, and curb sides and better performance in distingushing between the shadows and cracks.

Approach 2 advantages:

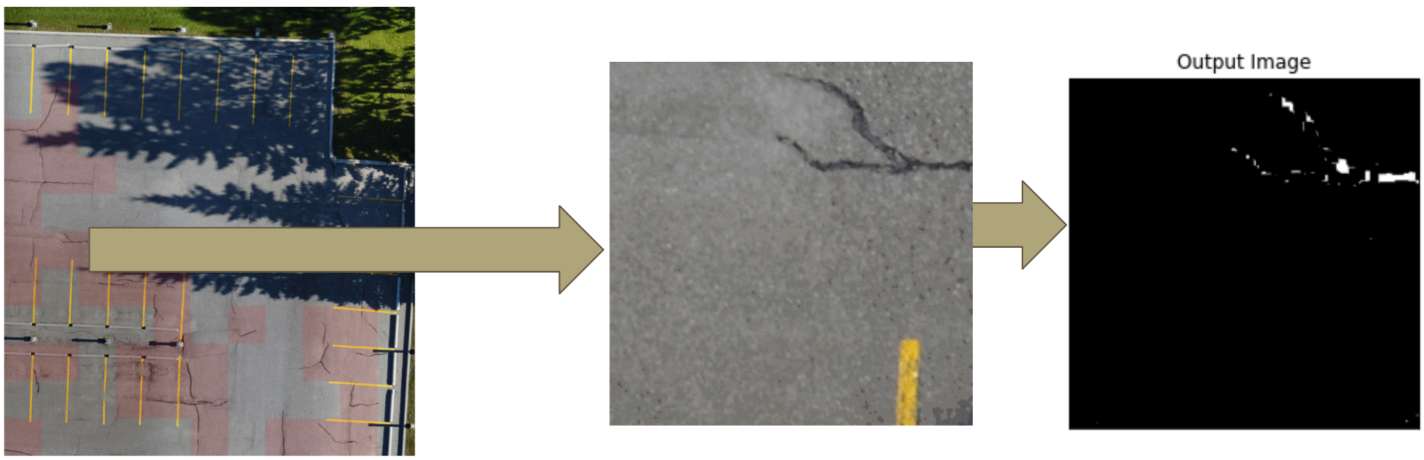
- Ability to detect the exact shape of the cracks vs finding the small patches that contains cracks.

Methodology

In order to benefit from the advantages of both models, we combined our two methodologies and created a framework that first, using our customized supervised learning model, detects small areas in the picture that contain cracks; then the detected area is given to our proposed unsupervised learning crack detector to output the exact location and shape of the crack. By this approach our framework is able to first eliminate the areas that doesn't contain any cracks and output the exact shape of the cracks.

Results

The following figure shows the result of applying our combined method to one of the case studies. The left picture is the output of oursupervised model in which we have detected the patches that contain cracks and highlighted them in red. Each of these red patches are given to our unsupervised model. In the picture one these patches that is predicted as contining crack is selected and is shown in the middle. The supervised model has correctly detected this patch as a patch with crack, however it doesn't output the exact crack shape. The right picture in this figure shows the output of feeding this patch to our unsupervised model. It can be seen that the unsupervised model further refined the output and and eliminated the yellow line and provided the exact shape and location of the crack.



5. Conclusion

In this project, we have developed a machine learning tool that detects cracks on road pavements. We have used unlabbeled data provided by Ariem Analytics and proposed a an intelligent detector that uses supervised and unsupervised learning aand computer vision methods to detect the cracks on the road pavements.

References

[1]. Government of Alberta, Annual Report Transportation 2020-2021. 2021. link

[2]. Transportation Alberta, Highway Maintenance Guidelines and Level of Service Manual, 2000. link

[3]. Yang C, Chen J, Li Z, Huang Y. Structural Crack Detection and Recognition Based on Deep Learning. Applied Sciences. Volume 11, Issue 6:2868, 2021. link

[4]. Suguru Yokoyama, Takashi Matsumoto, Development of an Automatic Detector of Cracks in Concrete Using Machine Learning. Procedia Engineering, Volume 171, 2017, Pages 1250-1255. link

[5]. Arun Mohan, Sumathi Poobal, Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal, Volume 57, Issue 2, 2018, Pages 787-798. link

[6]. Arvydas Rimkus, Askoldas Podviezko, Viktor Gribniak, Processing Digital Images for Crack Localization in Reinforced Concrete Members. Procedia Engineering, Volume 122, 2015, Pages 239-243. link

[7]. Minnesota, Department of Transportation, Office of Materials and Road Research. Pavement Distress Identification Manual. link

[8]. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[9]. Finlayson G.D., S. D. Hordley, C. Lu and M. S. Drew, On the removal of shadows from images. The University of East Anglia, Norwich, UK., and, Simon Fraser University, Vancouver, Canada.

[10]. T. Ahmed Mahgoub Ahmed, Zhangcan Huang, Fan Xi, Liu Hai Ming, Detection crack in image using Otsu method and multiple filtering in image processing techniques, Optik – Int. J. Light Electron Opt. 127 (3) (2016) 1030–1033.