IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium | 978-1-7281-6374-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IGARSS39084.2020.9323936

LEVEE-CRACK DETECTION FROM SATELLITE OR DRONE IMAGERY USING MACHINE LEARNING APPROACHES

Aditi Kuchi [2], Md Tamjidul Hoque[1, 2,], Mahdi Abdelguerfi[1, 2], Maik C. Flanagin[3]*

***Email**: {askuchi, thoque, mabdelgu}@uno.edu, maik.c.flanagin@usace.army.mil*

[1]Canizaro/Livingston Gulf States Center for Environmental Informatics, University of New Orleans, New Orleans, LA 70148
[2]Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA.
[3]US Army Corps of Engineers, New Orleans District, LA, USA.
**Correspondence: thoque@uno.edu; Tel.: +1-504-280-2406*

ABSTRACT

This paper describes and compares different approaches for detecting cracks in the concrete toe, other general areas of levees and dams using satellite or drone images. The dataset was sourced from real drone flight data and manually collected and annotated as needed. We compare old and modern algorithms alike to determine which ones perform best in this case. We also explain the reasonings for a particularly interesting case of the viola jones algorithm, where our calculated accuracy is 100%. We study stacking (85% accuracy), and the latest deep learning techniques (90.90% accuracy) as well. This research hopes to help the U.S. Army Engineers Corps integrate the model into drones to better monitor the levee areas prone to disaster.

Index Terms— Object detection, Machine Learning, Deep Learning, Stacking,

1. INTRODUCTION

Levees are structures constructed along natural water bodies to stop the flooding of low-lying areas. The presence of cracks in the concrete of these structures can indicate impending structural failure. Breakage of levees can result in catastrophic damage to property and life [1]. The monitoring of these problems is currently done manually or at best by flying drones only to collect images [2, 3].

However, we aim to automate this process by testing and selecting the best machine learning model that can most accurately detect cracks near these structures so that the person trying to identify these problem areas can be more targeted in their approach. The most significant issue we face while trying to detect cracks is the actual collection of images. Most models train on a curated dataset and then are tested on a smaller and significantly different test dataset. Any slight change in the lighting conditions, angles, etc. will cause the training data to vary vastly. In this experiment, however, we have taken adequate care to modify and augment the collected data suitably.

Once the dataset is created and annotated (12,800 images), we move on to the selection of models. We have chosen to use one of the oldest algorithms for object detection, Viola-Jones [4, 5], one of the latest deep learning models, Single Shot MultiBox Detector [6], some more commonly used non-deep learning methods such as Support Vector Machines, Gradient Boosting Classifier, etc. along with a Stacking [7-11] approach to compare their performances.

In this paper, we primarily focus on using the data from a custom-built dataset from drone images near levees, testing each of these methods, provide reasons as to why and how they work for our current dataset. Then we further study the methods and select the best one that might work best for real-time detection of cracks. We then evaluate the performance and compare its results.

2. BACKGROUND AND RELATED WORK

Before proceeding with collecting the required data from the drone images and applying various machine learning methods, we first study the significance of this issue and try to grasp the key points that need to be addressed in such situations so that we can pose the correct questions and extract meaningful observations from our data.

Levees that protect vast areas of commercial and residential properties are prone to degradation over time. This could be the result of severe weather, subsidence of land, seepage, development of cracks, sand boils, etc. Further published data indicates that coastal Louisiana lost approximately 16 square miles of land between 1985 and 2010 [12]. New Orleans' history records the worst flooding in the area due to hurricane Katrina in 2005. There were over 50 failures of levees and floodwalls protecting New Orleans, Louisiana, and its surrounding suburbs, following the passage of Hurricane Katrina. These failures caused 80% of the flooding that affected New Orleans. Such events show us that levee and dam failures can be catastrophic. There is enormous potential for significant property damage, loss of life, damages of vegetation, and land.

Due to the described adverse effects of levee failure, close monitoring of cracking near the levee's concrete and surrounding area is of the utmost importance [3, 13]. In this research, we will focus on the detection of any such cracks and try to predict the exact area they are present in so that they can be monitored closely.

There have been similar research papers focusing on the detection of cracks, for example, in concrete pipes [14], crack detection on pavements [15], etc. with high accuracies. However, our paper deals with the discovery of cracks in a niche area, which requires a precise collection of data and pre-processing. Another novel contribution is the introduction of the stacking method to detect cracks in levees.

Object detection using machine learning is a popular approach to locate regions of interest in an image. The usage of object detection methods for cracks is particularly well-suited because of the characteristic features of a crack that a machine learning model can pick up on. For example, the area of contrast between the brighter concrete and the dark portion of the crack.

3. DATA AND METHODS

The entire dataset was created from scratch using images photographed from drones flying over levees in the New Orleans areas. We sliced the images so that the most relevant part, i.e., cracks, are most prominent. Since the number of images with cracks were limited, we opted to augment the images to synthetically create more positive samples by superimposing the cracks on related images. These images were collected from ImageNet and OpenStreet Maps [16].

The methods used include Viola-Jones detection which uses Haar features to classify the presence of cracks and generate appropriate bounding boxes, non-deep learning techniques such as SVM, GBC, kNN, etc., Stacking and the Single Shot MultiBox Detector (SSD). In order to use the images for deep learning, they need to be manually annotated.

To use these images with non-deep learning techniques, custom features must be extracted from each image and then passed onto the individual models. To do this, we chose a combination of several different features that provide the best results in combination with each other. They are Hu moments (7 features), Haralick features (13 features), Histograms (32 features), Histogram of Oriented Gradients (HOG) (648 features), Canny features, and Gabor features. Since Canny and Gabor features are defined as matrices, we chose to derive additional features such as the sum along axes, and total sum from them and include them as individual features. All implementations of models such as SVM, etc. used Python and related machine learning libraries like Scikit-Learn, Mahotas, Pillow, etc. For implementing Haar cascades, we use OpenCV in conjunction with Python. The code to calculate the accuracy of the Viola-Jones Detector is custom and is based on the categorization of the bounding box results, as explained in figures 1, 2, and 3. SSD uses MobileNetV2 as its feature extraction layer. We use a Pytorch based network and tune it further to help with accuracy.

4. RESULTS AND DISCUSSIONS

4.1. Viola-Jones object detector

Viola-Jones object detector can be implemented using OpenCV's Haar Cascade training. It uses Haar features and boosting to determine which of the input images are positive and quickly rejects all the sub-windows that appear not to have cracks in them. We train the cascade for 25 steps with the training data and use the generated XML files to test their object detection capabilities. Upon testing with multiple XML files from each stage, the accuracy of this model turns up to be nearly 100%. The reason behind this phenomenon is the inherent characteristic of the haar feature itself.

The Haar feature is a rectangular image consisting of rectangular black and white regions that are superimposed on each image to check for the presence of contrast regions. For example, in the case of a face recognition problem, a Haar cascade considers it a face, if, for example, three regions near the eyes, nose, and mouth are dark/light as per the model. Similarly, in the case of cracks, every part of the image that contains any region of dark versus light areas is classified as cracks. In the later stages, these classifications become a lot more refined. But unless we introduce unnaturally occurring images such as that of cats or dogs out of context, the model will continue to predict most cracks successfully. The models were tested on two different test datasets. One with entirely positive images containing multiple cracks to see the actual detection of bounding boxes, and a second test set with a mix of both positive and negative images. The accuracy was calculated to be very close to 100% in both cases.

The border-crack problem: The detections resulting from the Viola-Jones method present a unique challenge that we shall refer to as the border-crack problem.

Because of the way a Haar feature functions, and because of the way these synthetic samples have been generated, there arose a different kind of problem in detections. The border-crack problem occurs when there is a stark contrast between the superimposed image and the negative background image. These borders are then mistaken for being cracks by the detector. This contributes to the high accuracy parameters. This problem can be clearly seen in figures 1, 2, and 3, where each of them is explained in detail.

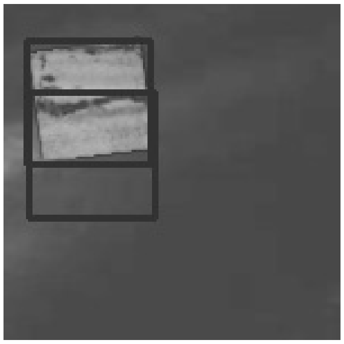

Figure 1: A clear example of the border-crack problem where the detector mistakes the border of the superimposed image to be a crack. The actual crack is also detected. Such images are classified as true positives for the purpose of calculation of accuracy.

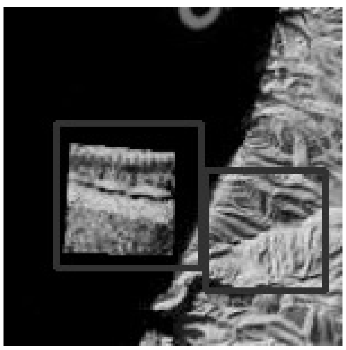

Figure 2: An example of correctly detected cracks. This is another example of a true positive detection. The crack on the land nearby has also been successfully detected.

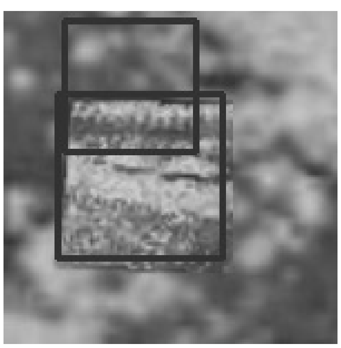

Figure 3: Another example of the border-crack problem where the detection occurs near the border of the superimposed image. Note that the actual crack has also been successfully detected.

4.2. Single Shot MultiBox Detector

SSD is a deep learning method that does not need any manual extraction of features. We use a PyTorch implementation of the network to train and test on the dataset. Upon running it for 400 epochs, the accuracy was measured to be 90.90%. On a small dataset with few variations, this was a high result. The only drawback in this method is that of manually annotating all the images in the dataset and have it ready in the VOC-format.

4.3. Non-deep learning methods and Stacking

Below, we show the results for the overall accuracy of the individual methods.

Random Decision Forest –	70.022%
Extra Tree Classifier –	65.86%
K-Nearest Neighbors –	58.58%
Logistic Regression –	58.52%
Xtreme Gradient Boosting –	71.83%
Gradient Boosting Classifier –	**76.12**%
Support Vector Machine –	73.52%

Bagging – 70.33%

Stacking based machine learning is an ensemble approach that obtains information from multiple models and aggregates them to form a new, generally improved model. We ensure that the selected methods that perform well but are least correlated to each other. Below are a few results from stacking approaches.

GBC, kNN, SVM as base, GBC as meta –	76.22%
RDF, GBC, kNN, SVM as base, GBC as meta –	79.54%
RDF, GBC, kNN, SVM as base, SVM as meta –	**84.58**%

5. CONCLUSIONS AND FUTURE WORK

In this paper, we compared some of the best and latest object detection algorithms to detect cracks near levees. We developed a stacking-based machine learning method that is capable of detecting cracks by prediction. In comparison, the deep learning method performed best at 90.90%. The stacking method performed comparably well at around 85%. Despite the high performance of the Viola-Jones detector at 100%, it might not be the best possible one to use for a real-world scenario since it categorizes most high contrast surfaces in the context of dams and levees as a crack. Using such a method would only be beneficial in especially high-risk cases to detect all and any possible cracks. In cases where these models must run in real-time on the small architecture of a drone, the lower the computational overhead, the better. Deep learning processes take up a lot more computational power than the stacking method does. Therefore, without sacrificing too much accuracy, the stacking method would work better on smaller devices. The code and data of our proposed model are freely available here http://cs.uno.edu/~tamjid/Software/crack/code_data.zip

6. REFERENCES

[1] F. E. M. Agency, “Evaluation and Monitoring of Seepage and Internal Erosion,” *Interagency Committee on Dam Safety (ICODS)*, 2015.

[2] P. C. Jay N. Stateler, etc., *United States Society on Dams - Monitoring Levees*, 2016.

[3] R. a. A. Nobrega, James and Gokaraju, Balakrishna and Mahrooghy, Majid and Dabbiru, Lalitha and O'Hara, Chuck, “Mapping weaknesses in the Mississippi river levee system using multi-temporal UAVSAR data,” *Brazilian Journal of Cartography, Photogrammetry and Remote Sensing,* vol. 65(4), pp. 681-694, 2013.

[4] P. Viola, and M. J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision,* vol. 57, no. 2, pp. 137–154, 2004.

[5] Y.-Q. Wang, “An Analysis of the Viola-Jones Face Detection Algorithm,” *Image Processing On Line* vol. 4, pp. 128–148, 2014.

[6] D. A. Wei Liu, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD : Single Shot MultiBox Detector,” *CoRR*, 2015.

[7] A. Mishra, P. Pokhrel, and M. T. Hoque, “StackDPPred: A Stacking based Prediction of DNA-binding Protein from Sequence,” *Oxford Bioinformatics,* vol. 35, no. 3, pp. 433–441, 2019.

[8] S. Iqbal, and M. T. Hoque, “PBRpredict-Suite: A Suite of Models to Predict Peptide Recognition Domain Residues from Protein Sequence,” *Oxford Bioinformatics* 2018

[9] M. Flot, A. Mishra, A. S. Kuchi, and M. T. Hoque, "StackSSSPred: A Stacking-Based Prediction of Supersecondary Structure from Sequence," *Protein Supersecondary Structures. Methods in Molecular Biology*, K. A, ed., pp. 101-122, New York, NY: Humana Press, 2019.

[10] D. M. Alawad, A. Mishra, and M. T. Hoque, “AIBH: Accurate Identification of Brain Hemorrhage using Genetic Algorithm based Feature Selection and Stacking,” *Mach. Learn. Knowl. Extr., MDPI.,* vol. 2, no. 2, pp. 56-77, 2020.

[11] S. G. Gattani, A. Mishra, and M. T. Hoque, “StackCBPred: A Stacking based Prediction of Protein-Carbohydrate Binding Sites from Sequence,” *Carbohydrate Research, Elsevier.*, 2019.

[12] B. R. Couvillion, J. A. Barras, G. D. Steyer, W. Sleavin, M. Fischer, H. Beck, N. Trahan, B. Griffin, and D. Heckman, "Land Area Change in Coastal Louisiana from 1932 to 2010," *U.S. Geological Survey Scientific Investigations Map 3164, scale 1:265,000, 12 p. pamphlet.*, 2011.

[13] J. M. O. L. A. Schaefer, Timothy & Robbins, Bryant, “Assessing the Implications of Sand Boils for Backward Erosion Piping Risk.,” pp. 124-136, 2017.

[14] F. P. W. Sinha. S. K, “Automated detection of cracks in buried concrete pipe images,” *Automation in Construction, Elsevier*, 2006.

[15] M. S. Salman.M, Kamal.K, Rahman.M, “Pavement crack detection using the Gabor Filter,” *16th International IEEE Conference on Intelligent Transportation Systems* 2013.

[16] A. S. Kuchi, M. T. Hoque, M. Abdelguerfi, and M. Flanagin, “Machine Learning Applications in Detecting Sand Boils from Images,” *Array, Elsevier,* vol. 3-4, 2019.