CONCRETE CRACK DETECTION USING DEEP LEARNING TECHNIQUES

Rahul Nabera M[1], Aravindhan K[2], Srikanth Kini[3], Priyadharshini J[4]

School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India

Email-ID: { mrahul.nabera2015[1], aravindhan.k2015[2], psrikanth.kini2015[3], priyadarshini.j[4]}@vit.ac.in

1. ABSTRACT

Buildings tend to expand or contract based on the environment they are in and that leads to cracks on to the buildings and they can be a serious threat to the people using it and more often than not these movements are too small to be observed and so often go unnoticed. Movement can be caused by defects, movement of the ground, foundation failure, decay of the building fabric, and so on. If a structure is unable to accommodate this movement, cracking is likely to occur and its highly dangerous to the safety of the building. Only upon the identification of the cracks, it can be subjected to treatment and the existing manual methods of sketching the crack patterns are much subjective to the person analysing them, and are often bounded by high costs, equipment and tools availability and is highly time consuming. In this paper, we provide with a comparative study of various Deep Neural Networks to classify the image as one with the presence or absence of cracks and thereby suggesting a state of art binary classifying Neural Network that best suits the goal of Crack Detection.

2. INTRODUCTION

A Crack is a line on the surface of something (a concrete structure, here) along which it has a split without yet breaking apart. Hairline, Stepped, Vertical, Horizontal are various kinds of cracks having its own impact on the structure under consideration and can arise due to different kinds of factors like drought, weak foundation, uneven load distribution, ground movement, structure deformation under load, expansion or contraction of underlying material etc. Though there are various means to rectify the cracks once it has been found, timely identification of cracks, per se, poses a challenge. In the age of computing, it becomes absolutely necessary to automate the process to have an accurate analysis of the detection of various cracks and the literature provides with various methods starting from a basic image processing analysis to a complex Deep Learnt Models. Since the ultimatum being the accuracy we have explored a bunch of neural-net models to provide with an accurate means of identifying the cracks.

3. LITERATURE SURVEY

The current state of art proposes various analytical method to solve the problem and below are few standard works driven in the direction using the techniques of image analysis and processing attached with intelligence to solve in an effective way.

[1] A method based on computer vision uses images captured using high magnification image acquisition system, a 2-D electric cradle and laser ranging device which works in unison to mark the cracks in its observing coordinate system which is further mapped to observation coordinates and thus the spatial location is of measured cracks are obtained irrespective of positioning of the device. It is reported that the system works within an average of about 16s with a deviation of

no more than 0.07° of crack locating. This is of great convenience provided the devices and tools are readily available and test time of 16s though is on the higher side the required accuracy level is reached using the method.

[2] A method based on multi-scale enhancement and visual features is developed to detect the cracks. Firstly, to avoid the hindrances due to low contrast a multi-scale enhancement method using guided filter and gradient information is used which is then followed by adaptive thresholding algorithms to obtain a binary images to a combination of morphological processing and visual features is applied to purify cracks. The experimental results with different images of real concrete surface demonstrate a validity of the developed technique, in which the average TPR can reach 94.22%.

[3] A method of crack image processing for concrete bridge bottom crack inspections is proposed to solve this problem. A machine vision system based on this method, which could detect cracks in real time.

4. ABOUT THE DATASET

The dataset contains concrete images having cracks. The data is collected from various METU Campus Buildings. The dataset is divided into two as negative and positive crack images for image classification. Each class has 20000 images with a total of 40000 images with 227 x 227 pixels with RGB channels. The dataset is generated from 458 high-resolution images (4032x3024 pixel) with the method proposed by Zhang et al (2016). High resolution images have variance in terms of surface finish and illumination conditions. No data augmentation in terms of random rotation or flipping is applied. [4]

5. PROPOSED WORK

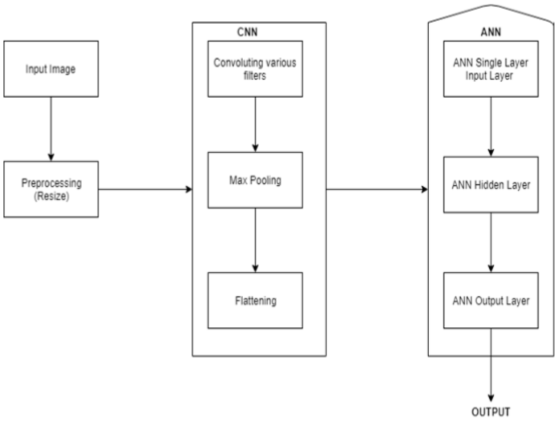
Neural networks are used for the purpose of training and capturing the features of the dataset. A neural network is a combination of neurons which fires a positive or negative signal upon its activations from the inputs, weights and biases. The learning part concentrates on determining the weights and biases, given the architecture of the neural network as a hyper-parameter. A neural network network has a bunch of these neurons connected in various possible combination making a small scale mimic of the brain itself having an input layer, various hidden intermediate layers and final output layer, where the decision is given as the result.

The neural network has a loss function to measure of how good a prediction model does in terms of being able to predict the expected outcome, an error function to correct its mistake and a backpropagation algorithm to propagate the error measured.

Neural networks are notoriously known for their capabilities of overfitting and in order to overcome the issue, we have designed system architecture. The considered dataset[4] was expanded from given set, by applying standard image transformations like scaling, skewing, rotations in order to avoid overfitting.

Convolution Neural Network: A Convolution Neural Network is designed to work especially on images to capture the features. Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a

ConvNet have neurons arranged in 3 dimensions: width, height, depth(color



dimension).

Fig 5.1

As shown in the above figure, the neural network, taken in a n-dimension image (3 in case of RGB image) and tries to extract few features sending them to the next level retaining maximum information in minimal form before flattening everything out in the last but one level upon which the final transformation is applied to get the final decision output.

In the design of our system we have applied, two different activation functions, relu function for the hidden layer and sigmoid for the output layer and we use binary cross-entropy as our loss function and adam as our optimizer and standard back propagation algorithm with single hidden layer to learn the dataset.

Recurrent Neural Network: A recurrent neural network (RNN) is a class of neural network which exhibits temporal dynamic behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs and thereby making a better approximation of the neuron's understanding of the information it is supposed to process. It has its application in handwriting and speech recognition. The principle of the RNN is used in crack detection.

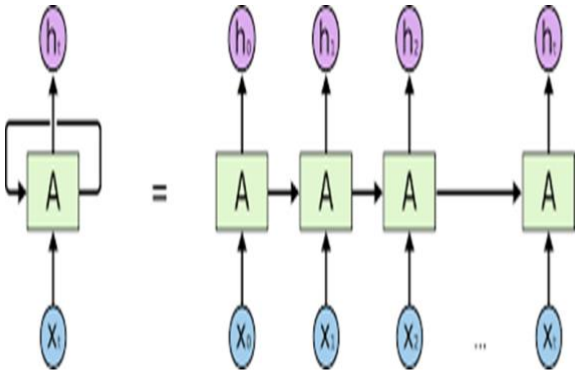


Fig 5.2

The recurrent neural network is represented as shown in the above figure. Each node at a time step takes an input from the previous node and this can be represented using a feedback loop. At each time step, we take an input x_i and a_i-1(output of the previous node) and perform computation on it and produce an output h_i. This output is taken and given to the next node. This process continues until all the time steps are evaluated.

Let a_t represent the ouput from the previous node

$$a_t = f(h_{t-1}, x_t)$$

$$g(x) = \tanh x$$

$$a_t = g(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t)$$

$$a_t = \tanh W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t$$

$$h_t = W_{hy} \cdot a_t$$

Backpropagation in recurrent neural networks occurs in the opposite direction of the arrows drawn in Fig 5.2 Like all other back propagation techniques, we evaluate a loss function and obtain gradients to update our weight parameters. The interesting part of backpropagation in RNN is that backpropagation occurs from right to left. Since the parameters are updated from final time steps to initial time steps, this is termed as backpropagation through time.

In the design of our system we have applied, two different activation functions, relu function for the hidden layer and sigmoid for the output layer and we use mean-squared as our error function and adam as our optimizer and standard back propagation algorithm with single long term short memory layer and single hidden layer to learn the dataset.

6. DISTRIBUTED TRAINING

Due to the size of the image dataset to train (2 * 20000 images having 227 x 227 pixels with RGB channels), training the dataset on a single node was incomplete even post 4 days of training. To speed up the training process without compromising on accuracy, we probed on various distributed training architectures. Our literature survey pointed us recent advancements in data and model parallelism, decentralized training, synchronous and asynchronous update models. We also became of the recent open-source project on Horovod and Batch AI. Incorporating these, we built our training models to train across a cluster of 10 GPU nodes to complete the training process without loss in accuracy.

7. PERFORMANCE ANALYSIS

We apply two different Neural networks to see which performs better and the thing we notice is that CNN outperforms the RNN by huge margin because the RNN tries to overfit the image dataset.

Performance in all three scenarios :

Accuracy	CNN	RNN
train	95%	100%
test	95%	50%
predict	95%	45%

8. CONCLUSION

This work introduced the implementation of a machine learning-based model to detect cracks on concrete surfaces. Considering the analysis carried out in our experiments, the use of the convolutional neural network approach proved suitable to train a model with a limited dataset size. The developed model is limited to a binary classification. From a practical perspective, it is important to mention that, while AI constitutes a path towards automated inspection of concrete structures, the classification model mirrors the knowledge used during its training. Hence, human expertise is key in the development of an appropriate tool. Thus, machine learning-based models are likely to be initially deployed as a tool that assists experts to provide a safer, faster and more productive inspection, creating new possibilities for increased effectiveness in infrastructure asset management by making unbiased, periodic structure monitoring and/or damage assessment feasible.

9. REFERENCES

[1] Methods for Long-Distance Crack Location and Detection of Concrete Bridge Structures - Youfa Cai, Xing Fu, Yanna Shang, Jingxin Shi

[2] Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features - Xiangzeng Liu, Yunfeng Ai, Sebastian Scherer

[3] A new image-based method for concrete bridge bottom crack detection - Xuhang Tong, Jie Guo, Yun Ling, Zhouping Yin

[4] Concrete Crack Images for Classification - Lei Zhang , Fan Yang , Yimin Daniel Zhang, and Y. J. Z., Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016)

[5] Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks by Young-Jin Cha* & Wooram Choi.

[6] Concrete Cracks Detection using Deep Learning Image Classification.Wilson Ricardo Leal da Silva and Diogo Schwerz de Lucena.

[7] AAOSHAT (2008), Bridging the Gap—Restoring and Rebuilding the Nation's Bridges, American Association of State Highway and Transportation Officials, Washington DC.

[8] Abdel-Qader, I., Abudayyeh, O. & Kelly, M. E. (2003), Analysis of edge-detection techniques for crack identification in bridges, Journal of Computing in Civil Engineering, 17(4), 255–63.

[9] Horovod Distributed Training: https://github.com/horovod/horovod/tree/v0.9.0

[10] Collaborative Filtering as a Case-Study for Model Parallelism on Bulk Synchronous Systems by Ariyam Das, Ishan Upadhyaya, Xiangrui Meng, Ameet Talwalkar.

[11] Wei Tan, Liangliang Cao, and Liana Fong. 2016. Faster and Cheaper: Parallelizing Large-Scale Matrix Factorization on GPUs. In Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC '16). 219–230

[12] Janis Keuper and Franz-Josef Pfreundt. 2015. Asynchronous Parallel Stochastic Gradient Descent: A Numeric Core for Scalable Distributed Machine Learning Algorithms. In Proceedings of the Workshop on Machine Learning in HighPerformance Computing Environments (MLHPC '15). 1:1–1:11.

[13] Qirong Ho, James Cipar, Henggang Cui, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger, and Eric P. Xing. 2013. More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13). 1223–1231.

[14] David Clarke. 2014. Design and Implementation of Parallel Algorithms for Modern Heterogeneous Platforms Based on the Functional Performance Model. Ph.D. Dissertation. University College Dublin.