SURFACE CRACK DETECTION

MD. Saimom Islam (160204011@aust.edu)
Rimon Datta (160204013@aust.edu)

1.INTRODUCTION:

Crack plays a critical role in the field of evaluating the quality of concrete structures, which affects the safety, applicability, and durability of the structure.Crack is one of important damages on a real concrete surface.There are different crack detections like surface ,road, building etc.

The motivation of this project is to monitor structural health and ensure structural safety.The manual process of crack detection is painstakingly time-consuming and suffers from subjective judgments of inspectors. Manual inspection can also be difficult to perform in case of high rise buildings and bridges. In this blog, we use deep learning to build a simple yet very accurate model for crack detection. Furthermore, we test the model on real world data and see that the model is accurate in detecting surface cracks in concrete and non concrete structures, for example roads,buildings etc.

Cracks in building structures

However, it remains a challenging task due to the intensity inhomogeneity of cracks and complexity of the background, e.gThe low contrast with surrounding pavement and possible shadows with similar intensity.the extracting of features is certainly necessary when image processing techniques detect cracks in an image. As a result, the usage of image processing techniques is also limited, since images taken on real concrete surfaces are influenced by some noises caused by lighting, blur, and so on.

2.Review of Related Works:

In recent years, researchers have begun to use neural networks to detect cracks automatically. The following is a brief overview of the application of different networks in crack detection.

Zhang et al. proposed a crack detection method based on deep learning, which seems to be one of the earliest works applying CNN to road crack detection. The pavement pictures are taken by smartphones, and the network model is built on the Caffe DeepLearning (DL) framework. By comparing with traditional machine learning classifiers such as support vector machines (SVM) and boosting methods, the author proved the effectiveness of deep learning methods.

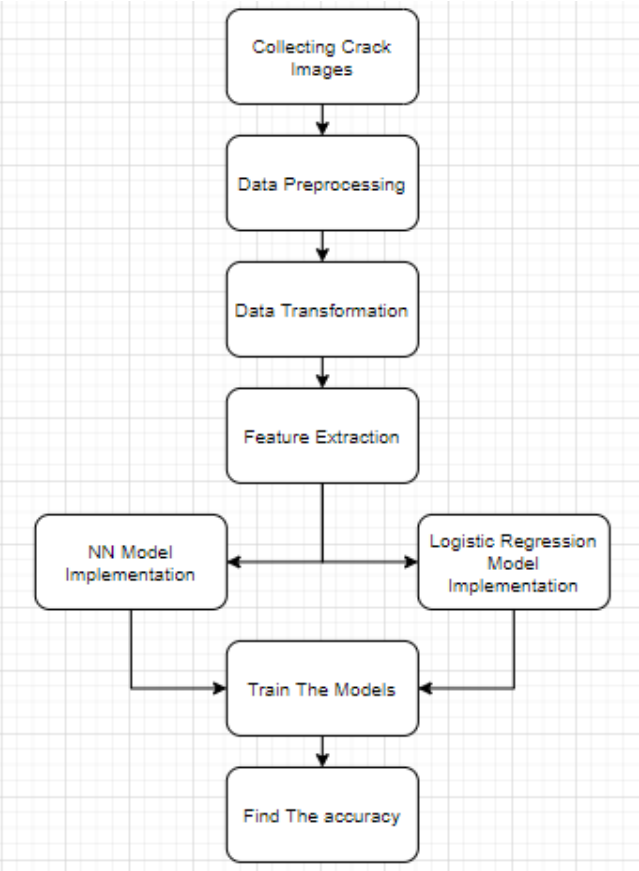
Pauly et al. studied the influence of CNN depth and the position change between training dataset and test dataset on pavement crack detection accuracy. The results show that increasing the network depth can improve the network performance, but when the image position changes, the detection accuracy will be greatly reduced.

Hoang (2018a,b, pp. 1–4) almost achieves this by treating this as multi-class problem using SVM (Support Vector Machine) algorithm to detect "longitudinal crack","transverse crack","diagonal crack","spall damage","intact wall"(Hoang 2018a,b,p. 1). SVM creates a hyperplane that is a line which separates each class based on their characteristics ensuring that separation between classes is at the maximal distance possible. Hoang (2018a,b, p. 1) attained a test accuracy of 85.33% (14.67% error rate) using 100 image samples per class. Even though Hoang's work in Hoang (2018a,b,4 D. B. Agyemang and M. Bader pp. 1–4) method treated the problem as a multi-class task, the researcher did not think about training their model to detect if the image is not a surface at all as the CNN cannot recognise if an image does not belong to its classes

After analyzing a few related works, we add logistic regression models with different settings and also use deep neural networks for getting better accuracy.

3.Project Objectives:

The section summarizes the whole process of the proposed crack detection method. The figure is the general flow of NN model and logistic model based surface crack detection. But before training the models, a dataset should be established to generate training and testing sets. In the databank, a series of large crack images are cropped into small images with 28*28 pixels, and then classified into images with cracks and without cracks manually. The training set and testing set are picked up from those small images randomly. After extracting the features ,the models are implemented .Through training the models using the training and testing sets, the NN and logistic classifier for crack detection can be obtained accordingly. To verify the effectiveness of the trained models, a process of testing is implemented. And finally, the accuracy is calculated.

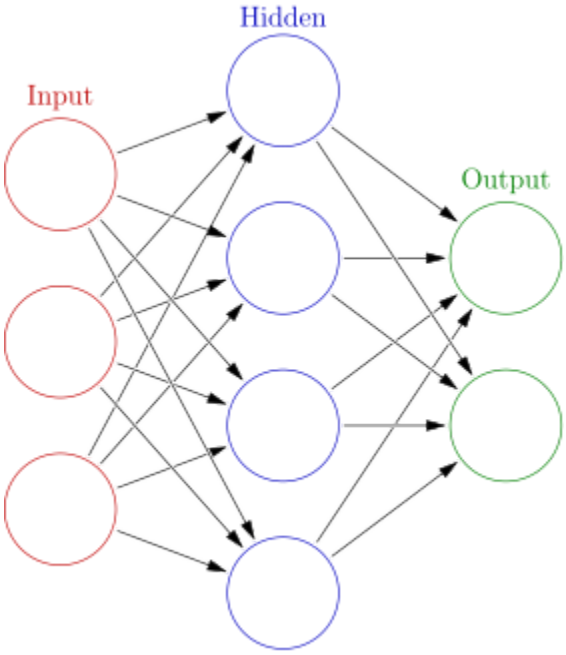


Flowchart of the proposed system

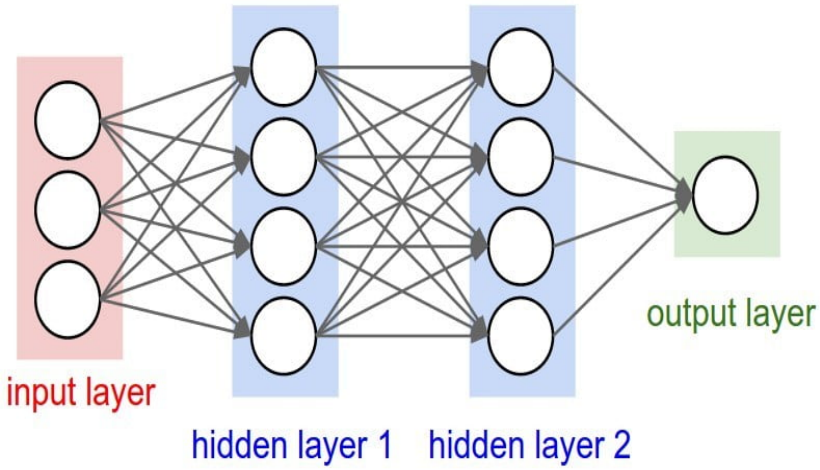
4.Methodology:

4.1 Model

a)**Deep Neural Network:** At its simplest, a neural network with some level of complexity, usually at least two layers, qualifies as a deep neural network (DNN), or deep net for short. Deep nets process data in complex ways by employing sophisticated math modeling. And we use single layer and multiple layers in different settings.

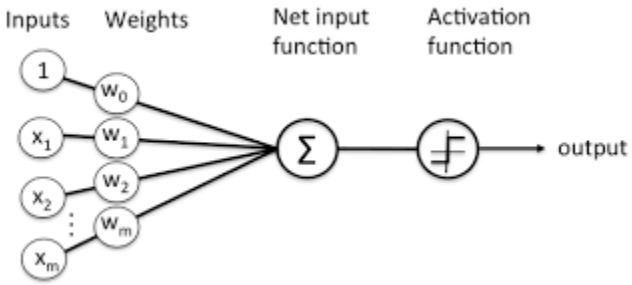


DNN with single hidden layer



DNN with Multiple hidden layer

b)**Logistic Regression:** Logistic regression is a binary classification method. It can be modelled as a function that can take in any number of inputs and constrain the output to be between 0 and 1. This means, we can think of Logistic Regression as a one-layer neural network. Simple. Logistic regression takes an input, passes it through a function called ***sigmoid function*** then returns an output of probability between 0 and 1. This sigmoid function is responsible for classifying the input.And Logistic regression has no hidden layer.



Logistic Regression

4.2 Activation Function:

a)**Relu**: ReLU stands for rectified linear activation unit and is considered one of the few milestones in the deep learning revolution. It is simpler yet really better than its predecessor activation functions such as sigmoid or tanh.ReLU function is its derivative both are monotonic. The function returns 0 if it receives any negative input, but for any positive value x, it returns that value back. Thus it gives an output that has a range from 0 to infinity. It uses this simple formula to transform its input:

```
f(x)=max(0,x)
```

b)**Leaky Relu:** Leaky ReLU function is an improved version of the ReLU activation function. As for the ReLU activation function, the gradient is 0 for all the values of inputs that are less than zero, which would deactivate the neurons in that region and may cause dying ReLU problems. Leaky ReLU is defined to address this problem. Instead of defining the ReLU activation function as 0 for negative values of inputs(x), we define it as an extremely small linear component of x. Here is the formula for this activation function

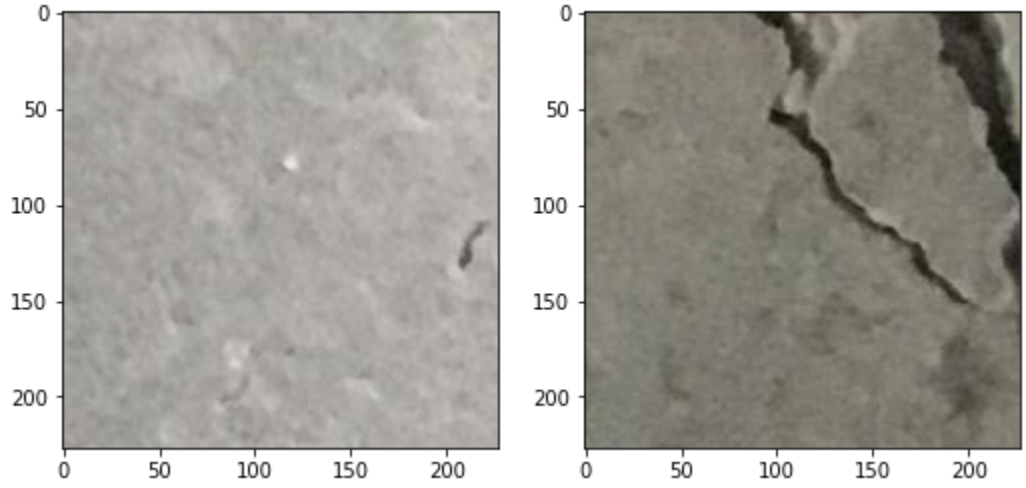
```
f(x)=max(0.01*x , x).
```

This function returns x if it receives any positive input, but for any negative value of x, it returns a really small value which is 0.01 times x.

5.Experiments

5.1 Dataset: We are using the Publicly available Surface Crack Detection. This data set was made publicly available in kaggle.

In this project, we use 40,000 photos of the dataset, and the number of crack and non-crack images are set to equal.The data set consists of 20,000 images of concrete structures with cracks and 20,000 images without cracks. The dataset is generated from 458 high-resolution images (4032x3024 pixel). Each image in the data set is a 227 x 227 pixels RGB image. Some sample images with cracks and without cracks are shown below:



Negative Crack **Positive Crack**

We shuffle and split the data into train and test. The data downloaded will have 2 folders one for Positive and one for Negative. We need to split this into train and test. We randomly shuffle the data into the train and rest into val where the train set has 35000 images and the test set have 5000 images.

5.2 Evaluation metric: The DNN and logistic regression are used in this project to detect the surface crack.
All the experiments in this paper are performed on TensorFlow in Windows system: hardware settings: CPU: Intel (R) Core (TM) i7CPU@3.20 GHz, RAM: 16G, and GPU: NVIDIA GTX1080Ti.

And we use custom model to train the data.The specific experimental steps are as follows:

Step 1: data loading
Importing concrete surface crack images. A batch of data is randomly loaded from the training set (batch size: 20) for subsequent data processing.

Step 2: image preprocessing
We use a built-in function in TensorFlow to adjust the size of the input image to the fixed size of the model. Then, do data augmentation via resize, translation and other operations. It is worth noting that, due to the use of TensorFlow's built-in function, a large number of pictures generated by data augmentation will not be saved to the local computer.

Step 3: define the structure of the crack detection model
We use custom models in this project . Single hidden layer is used in First DNN settings where the activation function we use is Leaky Relu. And in the settings2 we use multiple layer with the Relu activation function.In the last settings for DNN,multiple layers is used where the activation function is Leaky Relu.
We use Logistic Regression Model in the last settings and the activation function is Softmax.

Step 4: compile the model and start training
Before training the model, it is necessary to specify the hyperparameters related to the network structure and select the appropriate optimization strategy. In this experiment, The batch training is 20. The dataset is randomly shuffled before each epoch of training to ensure that the same batch of data in each epoch of model training is different, which can increase the rate of model convergence. The learning rate plays a significant role in the

training of model. Choosing an appropriate learning rate can speed up the model's convergence speed; on the contrary, it may cause the loss value of objective function to explode.

Step 5: test the performance of the model.
Finally,we calculate the accuracy on test sets where the values are differ from one to another ,Because we use different model with different settings.

5.3Results:

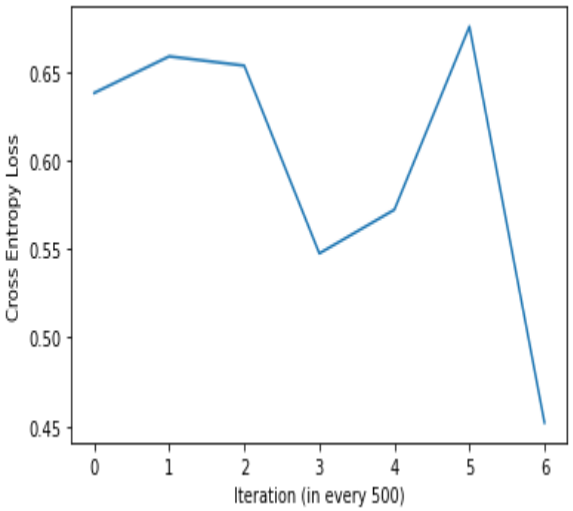
The results for different Models are given below:

	Accuracy	Lost
DNN(single layer&leakyrelu)	78.92	0.4519
DNN(mulitple layer&Relu)	50.52	0.69
DNN(multiple layer& leakyRelu)	84.74	0.38
Logistic(Softmax)	87.76	0.16

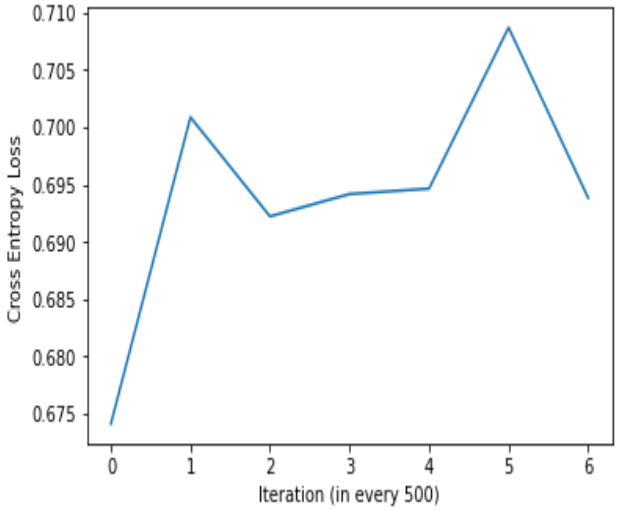
After every 3500 iterations the accuracy and loss is calculated for each setting. We can see the difference from the table. Using Multiple layers with Relu function is not good for detection and accuracy is only 50.52 .But if we use Multiple layer with Leaky relu function in the Deep neural network ,the accuracy is shown 84.74% which is far better than the 2nd settings.

But if we use logistic regression for detection of the surface crack ,the accuracy rate is higher than all of that which is 87.76. Because there is no hidden layer in this model so that there is no chance of overfitting in the model.

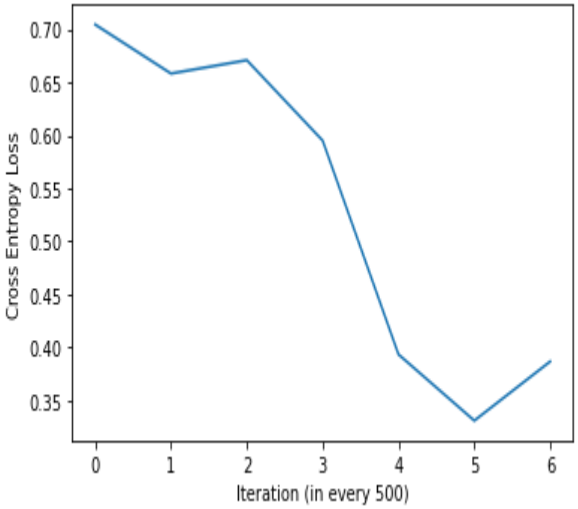
The loss graph for each settings is given below:



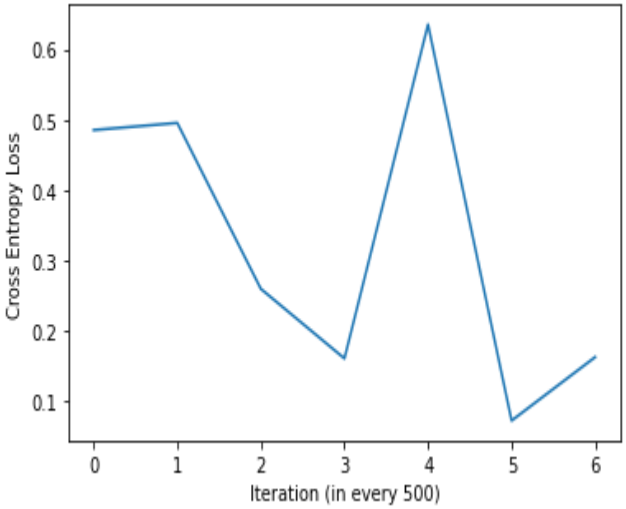
DNN(Single layer)



DNN(Multiple layer)



DNN(Multiple Layer & Leaky Relu)



Logistic Regression Model

6.Conclusion:

This project shows how easy it has become to build real world applications using deep learning and open source data.The benefit of the project is that users such as surveyors will be able to record cracks at rapid rate as they would not need to manually record the conditions. The reason why that benefit is important is that users such as surveyors are required to write a report on a building which is a long process, but if the surface crack

detection was used they would be able to quickly take pictures of the building, send the classification results to themselves and go back to the office to elaborate on the classification made by the application for their report. This would improve the work flow of surveyors.

Reference:

1.Priya Dwibedi,"*Detection of Surface Cracks in Concrete Structures using Deep Learning*"

2.Hoang, D.N," *Image processing-based recognition of wall defects using machine learning approaches and steerable filters. Comput. Intell. Neurosci. 1 (2018b*)"

3.L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "*Road crack detection using deep convolutional neural network*," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, IEEE, Phoenix, AZ, USA, September 2016.