Multi-Scale Deep Pixel Distribution Learning for Concrete Crack Detection

Xuanyi Wu[1], Jianfei Ma[2], Yu Sun[3], Chenqiu Zhao[4], Anup Basu[5]

Dept. of Computing Science, University of Alberta, Canada

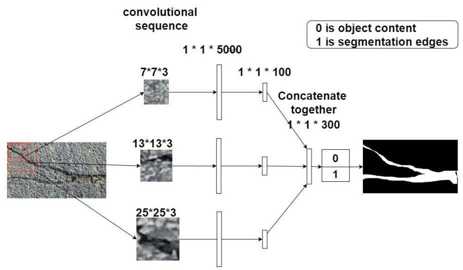
INTRODUCTION

A number of roads, bridges, and buildings built in the last century have already reached their intended lifetime. For maintenance purposes, people need to inspect these structures regularly. In this paper, we propose the MS-DPDL(multi scale deep pixel distribution learning) model for crack detection, which not only learns fast on limited number of training data, but also shows good transferability.

In particular, patches with different scales are initialized. For each patch, the pixels are randomly permuted and sampled to be the final input to the network. Patches with different scales will go to different convolutional sequences. By learning from the concatenated the outputs from those layers, the network predicts the label for the center pixel.

The experiment is done on two different datasets. First, the model is trained only on 20 images on the dataset *CRACK500*. Three different MS-DPDL network implementations shows similar results, which shows the strong learning ability for the multi-scale structure.

Then the pre-trained model is tested directly on another dataset *Concrete Crack Images for Classification*. The outstanding performance on a totally new dataset demonstrates the good transferability of the proposed method.



Proposed Method – PART 1

STEP 1: SHUFFLE THE IMAGE & CAPTURE THE FEATURE

Generate Random Permutation of Spatial pixels (RPoSP features). Let us denote the given concrete image as I(x, y), where x and y is the location of pixels. Each pixel has a label, (e.g., 0 for background and 1 for crack).

First, randomly shuffle the images. Second, generate groups of patches centered at (x, y) under several different scales and subtract the intensity of the center pixel. Last but not least, to avoid over-fitting, the pixels in the patches are we randomly permuted and sampled.

Mathematically, the above steps can be summarized as follows:

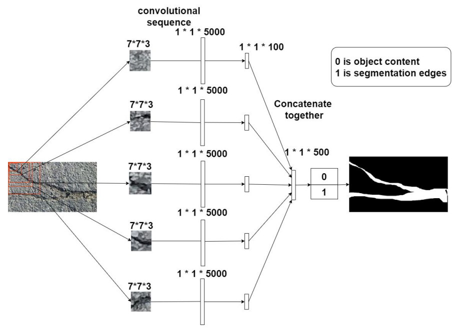
$$\mathrm{RPoSP}_{x,y}(m,n:R_i,R_o) = I(x,y) - I(x+r(m), y+r(n))$$
$$m,n \in [1,R_i], \quad r(m), r(n) \in [1,R_o]$$

where $\mathrm{RPoSP}_{x,y}(m,n:R_i,R_o)$ denotes the RPoSP feature extracted from the pixel located at (x,y). m, n are the indices of an entry in a patch and r() is random permutation step. R_i and R_o are the parameters to control the size of RPoSP features under multiple scales. In particular, R_o is the radius of patches under different scales, R_i is the radius of the RPoSP features. Specifically, a patch with radius R_o is captured and randomly permuted, then downsample the patch with radius R_i.

Proposed Method – PART 2

STEP 2: FEED INTO CNN

Following the above steps, the groups of patches are fed into a CNN. This is quite different from the implementations of existing CNN networks, which use the whole images as the input. Patches with different scales go into different convolutional sequences as shown in Figure. The output is the predicted label (0 or 1) for the central pixel in the input patch.



CNN procedure for MS-DPDL

By training with only crack images and ground truth images, our proposed MS-DPDL network can learn the pixel distribution for both crack pixels and non-crack pixels.

EXPERIMENT

Crack Segmentation Evaluation:

MS-DPDL models with 3 different network structures are trained on crack segmentation dataset *CRACK500* for comparison. The result shows that with limited training data(only 20 images are used for training), all the three implementations get similar results.

Crack Classification Evaluation:

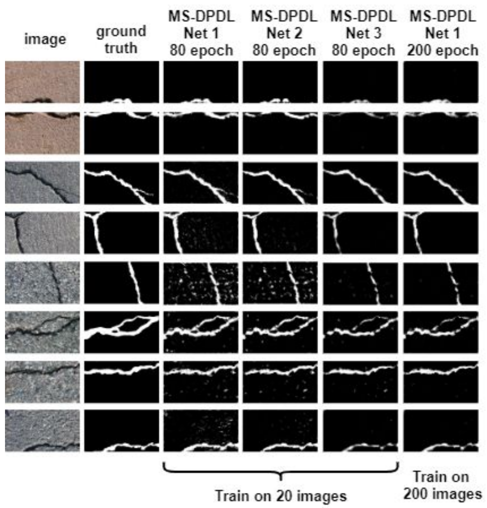
To show the good transferability of our proposed method, the pre-trained MS-DPDL Net 1 model (80 epoch) is used directly to test on another dataset *Concrete Crack Images for Classification*. Recall, Precision, F-measure and accuracy are used to evaluate the result for crack classification task.

$$precision = \frac{n_{tp}}{n_{tp}+n_{fp}} \quad recall = \frac{n_{tp}}{n_{tp}+n_{fn}} \quad F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$IoU = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}} \quad accuracy = \frac{n_{tp}+n_{tn}}{n_{tp}+n_{tn}+n_{fp}+n_{fn}}$$

n_{tp} = number of true positive, n_{fp} = number of false positive, n_{tn} = number of true negative, n_{fn} = number of false negative.

SEGMENTATION EVALUATION



CLASSIFICATION EVALUATION

A QUANTITATIVE COMPARISON OF DIFFERENCE MODELS' CLASSIFICATION PERFORMANCE.

Measurements	Recall	Precision	Fm	Accuracy
Proposed MS-DPDL	**0.9916**	**0.9918**	**0.9916**	**0.9920**
CNN(Sitara) [21]	0.94	0.95	0.94	0.99
VGG16 [21]	0.92	0.93	0.92	0.96
VGG19 [21]	0.73	0.80	0.76	0.81
Inception ResNet [21]	0.93	0.93	0.93	0.98
SVM [22] [23]	0.7333	0.6875	0.7096	0.7187
CNN [22] [24]	0.7802	0.8875	0.8304	0.8187
FCN(Manjurul) [22]	0.941	0.913	0.927	0.928
CNN-AT(Rui) [16]	0.9992	0.9992	0.9992	0.9992

PROPOSED MS-DPDL RESULT ON 40K IMAGES, K = 40.

Measurements	crack	non-crack	overall
Re	NA	NA	0.9733
Pr	NA	NA	0.97711
Fm	NA	NA	0.97520
Accuracy	**0.9733**	**0.9772**	**0.97525**

As shown, the proposed MS-DPDL method gets an classification accuracy of 0.992 among the straightly sampled 10K testing images, and an accuracy of 0.9725 among the entire 40K images. Even though it is totally trained on another dataset, it still has a high placing.

ICPR 2020