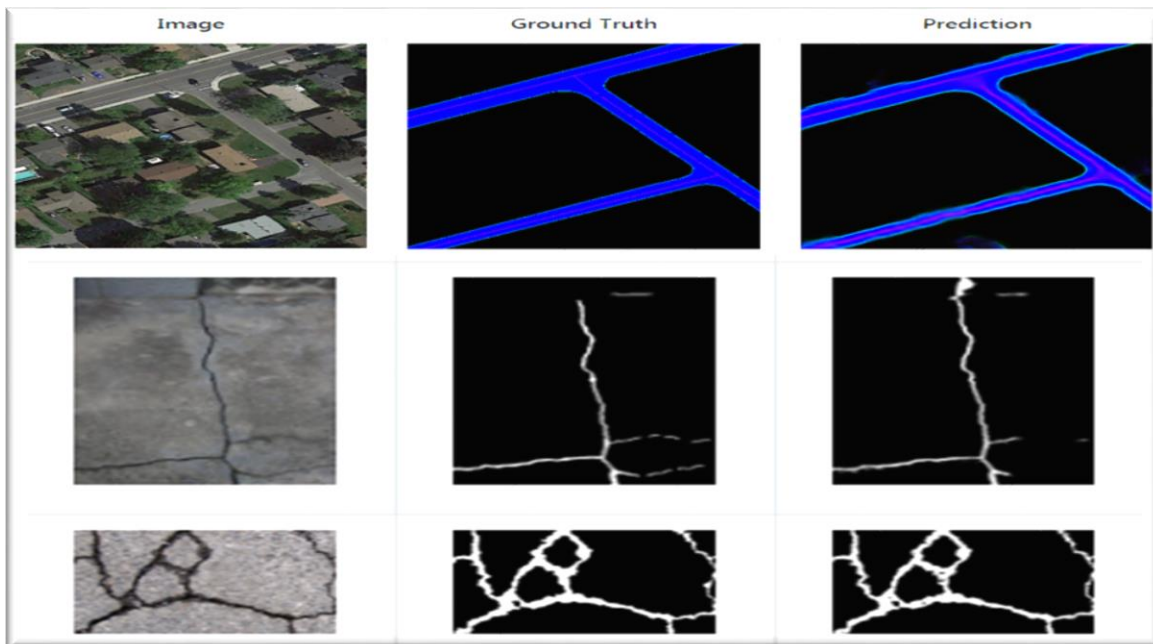


Automatic Road Cracks Detection



Road Cracks

Detection & Segmentation



Agilytics Technologies Pvt Ltd.

Copyright © 2020 by Agilytics

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

Table of Contents

INTRODUCTION.....	3
METHODOLOGY	3
DATASET.....	4
INSTALLATION.....	5
BALANCING WEIGHTS.....	5
TRAINING	5
Crack Detection.....	5
Road Detection	6
TESTING.....	6
Crack Detection.....	6
Road Detection	6

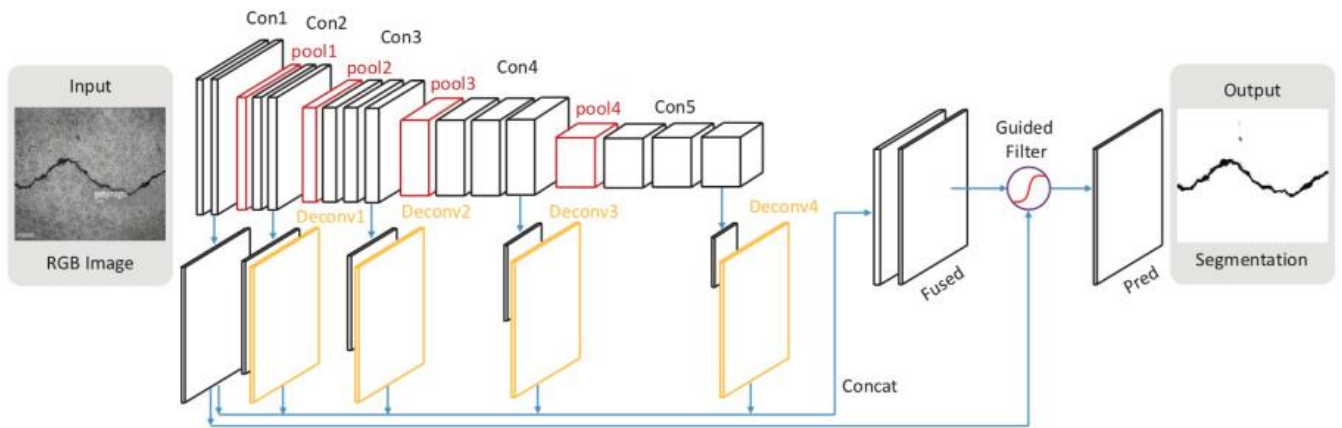


INTRODUCTION

Automatic crack detection from images of various scenes is a useful and challenging task in practice. In this project, we have created a Deep Learning architecture to detect the road-cracks automatically and also categorize them same. We used **PyTorch** open source machine learning library.

METHODOLOGY

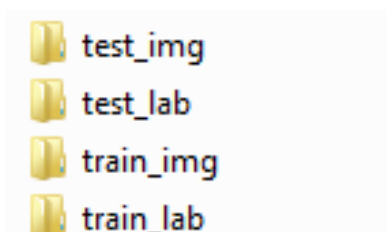
In this project, we have created a deep hierarchical Convolution Neural Network (CNN), to predict pixel-wise crack segmentation in an end-to-end method. It consists of the extended Fully Convolutional Network (FCN) and the Deeply-Supervised Nets (DSN). During the training, the designed model learns and aggregates multi-scale and multi-level features from the low convolutional layers, which is different from the standard approaches of only using the last convolutional r - layer. DSN provides integrated direct supervision for features of each convolutional stage. We applied both guided filtering and Conditional Random Fields (CRFs) methods to refine the final prediction results. A benchmark dataset consisting of 537 images with manual annotation maps have been built to verify the effectiveness of the model. Our method achieved state-of-the-art performances on the benchmark dataset (mean I/U of 85.9, best F-score of 86.5, and 0.1 s per image).



In this architecture there are no any fully connected layers. The side-output layers are inserted after the convolution layers and then the deep supervision is applied at each side-output layer, followed by concatenation to form a final fused output. By following this methodology the final output layer acquires multi-scale and multi-level features as the plane size of the input of side-output layers becomes smaller and the receptive field size becomes larger. The fused prediction is refined by guided filtering with the first side-output layer.

DATASET

We established a basic dataset with cracks in multiple scales and scenes to evaluate the crack detection systems. All of the crack images in our dataset are manually annotated.



Folder	Description
train_img	RGB images for training
train_lab	binary annotation for training images
test_img	RGB images for testing
test_lab	binary annotation for testing images

INSTALLATION

The configuration method is via Conda system, and one can create a new Conda environment using the command:

```
conda env create -f environment.yml
```

BALANCING WEIGHTS

We used the Median Frequency Balancing method, using the command

```
python3 ./tools/calculate_weights.py --data_path <path_to_segmentation>
```

TRAINING

The benchmark dataset, as described above, is used to train the model: -

5

Crack Detection

```
sh ./scripts/train_deepcrack.sh <gpu_id>
```





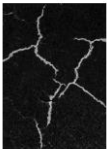
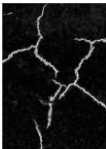
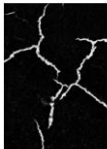
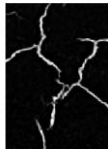


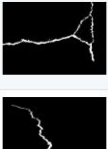
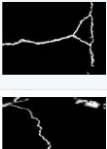
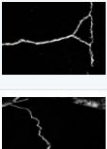
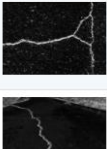

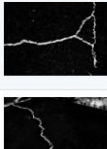
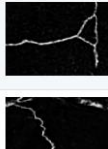





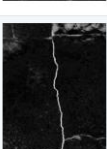
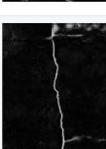
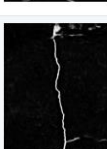

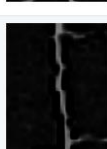









Road Detection

```
sh ./scripts/train_roadnet.sh <gpu_id>
```

TESTING

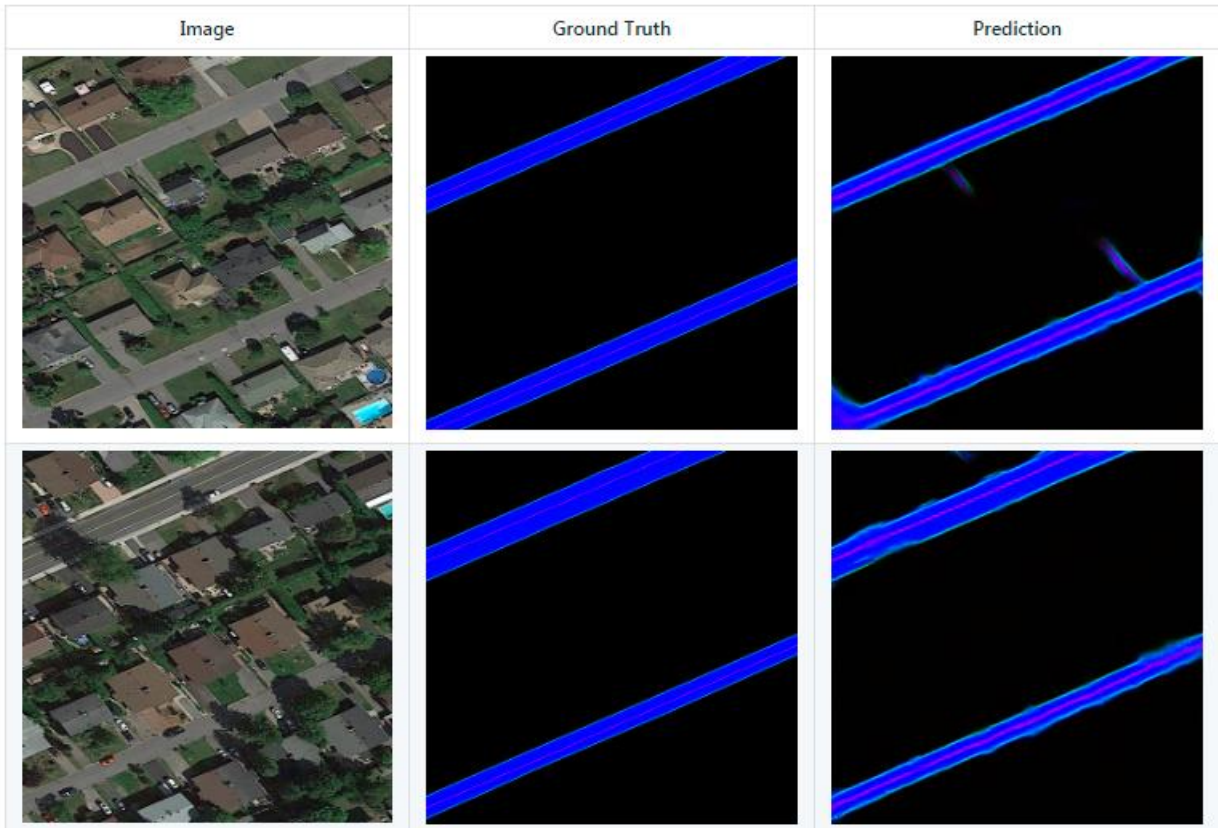
Crack Detection

```
sh ./scripts/test_deepcrack.sh <gpu_id>
```

Image	Ground Truth	GF	fused	side1	side2	side3	side4	side5
								
								
								
								

Road Detection

```
sh ./scripts/test_roadnet.sh <gpu_id>
```



Usage:

```
cd eval
python eval.py --metric_mode prf --model_name deepcrack --output deepcrack.prf
```

CONCLUSION

Automatic road crack detection is useful for the road construction companies for long term economic maintenance of the built roads. Unlike other detection tasks, segmenting the refined crack regions in pixel-wise is better than predicting bounding-boxes in practice.