Concrete Crack Pixel Classification Using an Encoder Decoder Based Deep Learning Architecture

Umme Hafsa Billah, Alireza Tavakkoli, and Hung Manh La(✉)

University of Nevada, Reno, NV 89557, USA
hla@unr.edu
https://ara.cse.unr.edu

Abstract. Civil infrastructure inspection in hazardous areas such as underwater beams, bridge decks, etc., is a perilous task. In addition, other factors like labor intensity, time, etc. influence the inspection of infrastructures. Recent studies [11] represent that, an autonomous inspection of civil infrastructure can eradicate most of the problems stemming from manual inspection. In this paper, we address the problem of detecting cracks in the concrete surface. Most of the recent crack detection techniques use deep architecture. However, finding the exact location of crack efficiently has been a difficult problem recently. Therefore, a deep architecture is proposed in this paper, to identify the exact location of cracks. Our architecture labels each pixel as crack or non-crack, which eliminates the need for using any existing post-processing techniques in the current literature [5,11]. Moreover, acquiring enough data for learning is another challenge in concrete defect detection. According to previous studies, only 10% of an image contains edge pixels (in our case defected areas) [31]. We proposed a robust data augmentation technique to alleviate the need for collecting more crack image samples. The experimental results show that, with our method, significant accuracy can be obtained with very less sample of data. Our proposed method also outperforms the existing methods of concrete crack classification.

Keywords: Crack detection · Pixel labeling · Deep learning architecture · Data augmentation

1 Introduction

Modern transportation system consists of a variety of civil infrastructures such as roads, bridge decks, highways, etc. Potential defects or deterioration in these structures can lead to unwanted situation e.g., road accidents. Civil infrastructure inspection is an essential for ensuring a well performing transportation system. The main element of these structures, is concrete. Concrete is composed of an aggregate mixture of various type of rocks, limestone, clay, and water. This liquid paste is altogether known as cement. The cement hardens over time as

G. Bebis et al. (Eds.): ISVC 2019, LNCS 11844, pp. 593–604, 2019.
https://doi.org/10.1007/978-3-030-33720-9_46

water evaporates from the mixture. This leads to various type of deterioration such as cracking, spalling, abrasion, etc. Cracking in concrete occurs more frequently than any other defects. The cement of concrete suffers from cracking mostly in comparison to other defects. Shrinkage in concrete elements, expansion, overloading, heaving, chemical exposure, corrosion with metals infused in concretes, improper drying, etc is some of the most common reasons for concrete cracking.

The longevity and performance of civil infrastructure are immensely affected by the defects of the concrete. Moreover, rebuilding infrastructure is time-consuming, uneconomic as well as time-consuming. Thus, continuous inspection of concrete health is necessary for maintaining a fully functional transportation system.

Earlier civil infrastructure inspection was performed manually by experts. However, the time consuming and labor-intensive manner of such method necessitated autonomous inspection. Autonomous inspection robotic systems [11,16,20] integrated with various sensors e.g., non-destructive evaluation (NDE) sensors [13] and camera can access infrastructure location to collect NDE data and sample images, which are further processed for defect identification. Autonomous inspection systems can reach dangerous areas where it is unsafe for humans to reach. The less error-prone nature of the autonomous systems also made them a convenient technique to adopt for maintenance [10,14,15].

One of the main challenges in identifying concrete defect autonomously is to provide the machine with enough knowledge of unhealthy concrete. Therefore, the need for an extensive amount of concrete samples is undeniable. In addition, an efficient architecture that can extract distinguishing features of concrete defect is also necessary. Recently many studies have proposed many architectures or methods for extracting such features. In the remainder of this section we have elaborated about the recent works of concrete crack inspection, the challenges associated and our contribution to the research problem.

1.1 Literature Review

Concrete defect classification and localization have enticed the attention of the research community very recently. The state of the art image processing techniques e.g., thresholding [19,24,29], morphological operations [2,7,17,22,28,30] and edge detection algorithms [21,25,32], have been proven efficient for defect localization [5]. The most significant advantage of these approaches is the extraction of both local and global feature of crack location with less computational complexity. However, the extensive amount of noise (crack like areas) present in real-world images induces extraction of unnecessary feature points. 'These unnecessary feature points increase the number of falsely identified crack pixels. Moreover, perfect parameter selection is also a conundrum for these type of methods.

Since the machine learning architectures are adaptive to real-world situations, it is intuitive to use them for defect classification and localization. In this paper, we categorize the machine learning architectures into two divisions such as (a) shallow architectures and (b) deep architectures. Some of the shallow machine learning architectures in the literature, for defect classification are Support Vector Machines (SVM) [8,25,28], Adaboost [25], Multi-Layer Perceptron (MLP) [4]. The conundrum of selecting perfect parameter set is solved by these architectures. An accurate classification using these architectures, require a balanced data-set having uniform instances of both healthy and defected areas of concrete. However, recent studies show that defected pixels occur only 10% of the time in an image [31]. Thus, the scarcity of well-balanced instances of both classes significantly drops accuracy. On the other hand, the recent advent of deep architectures [18] in solving classification problem,

Since the deep architectures [18] achieved significant performance gain in image classification, some studies [3,5,6,11,12,27] used them for crack image classification. These approaches apply image processing techniques on the extracted crack blocks, to localize the cracks. However, the rate of false classification and parameter selection of image processing techniques affects the performance of crack localization. Moreover, the computational complexity, as well as the number of parameters of deep architectures increases as the network, goes deeper. The growing number of parameters degrades with a deeper layer which, drops the performance of the network. In addition, the deep architectures require an extensive amount of data to train the networks on. Collection and processing of such data-set are time-consuming and memory intensive.

1.2 Contributions

From the above discussion we postulate that the main challenges in concrete defects detection in previous studies are :

1. Generating distinguishing feature maps for crack and non-crack pixels with a deep architecture.
2. A deep architecture that is less affected by parameter degradation problem of deep architecture.
3. Generation of a balanced data-set with enough instances of both crack and non-crack pixels.

In this paper, we proposed a robust deep network architecture alleviating the effect of parameter degradation. Our architectures use a series of encoder and decoder to generate distinguishing feature descriptor for crack and non-crack class. Lastly, our data augmentation technique enables to generate a substantial amount of instances for training. The rest of the paper is organized as follows: Sect. 2 elaborately explains our Proposed architecture. In Sect. 3, we discuss our training process, data augmentation and results. Lastly, we conclude in Sect. 4.

2 Methodology

The deep network architectures eliminated the need for feature extraction using traditional image processing techniques. In this paper, we introduce an encoder-decoder based deep network architecture to extract definitive features of crack and non-crack pixels. There exists, a number of different encoder-decoder based deep convolutional network architectures in literature, such as, UNet [26], SegNet [1], DeconvNet [23], FCN [12] etc. These networks are designed for semantic segmentation of generic object classes in natural images, and therefore, are not quite suitable for crack detection. To address the generic nature of these networks, we have designed the Proposed network architecture inspired by the architecture of SegNet [1]. In this section, we briefly reviewed the architecture of SegNet and then discussed our proposed network architecture.

2.1 SegNet

The SegNet architecture is composed of five encoders and five decoders. The encoders consist of a series of convolution layers, followed by ReLu, Batch Normalization, and max-pooling layer. The first two encoders have three consecutive convolution layers with 64 and 128 filters of size 3×3 respectively. The remaining encoders are composed of two convolution layers with 256, 512 and 512 filters of size 3×3 respectively. The convolution operations are performed with 3×3 filters. Each max-pooling layer uses a 2×2 window with stride 2 to obtain translation in-variance as mentioned in [1]. The max pooled indices are stored in memory to use in their respective decoder. Each decoder consists of a bilinear upsampling layer, followed by the convolution, ReLu, Batch Normalization layers using the same convention as their corresponding encoder layers. The upsampling operations are performed using memorized pool indices from the max-pooling layers. The SegNet architecture does not add a fully connected layer at the end of the network. Alternatively, it maps the output of the decoder layer to the softmax layer and assigns a label to each pixel.

2.2 Proposed Architecture

SegNet architecture uses a 3×3 window for the convolution operations in the encoder layers. Performing convolution operation with smaller windows causes significant boundary information loss. Since the pixels in a crack location are connected, such information loss has an enormous effect on classification. For example, if a $256 \times 256 \times 3$ (196,608 pixels)image is convolved with eight filters of size 3×3, a feature map of size $128 \times 128 \times 8$ (131,072 pixels)is generated. The reduced feature map on this particular operation loses the information of 65,536 pixels. The deeper the network goes, the more pixel information is lost. Our proposed network architecture accumulates for this lost pixel values in SegNet.

Since significant feature information is lost during the convolution process of each encoder, we have added a side output function [31] to each encoder. The side output function accommodates for the feature loss in each encoding operation. The side output function takes the output feature map of its corresponding encoder layer and up-samples it to original size. To up-sample the feature maps to its original size, we use a transposed convolution operation, followed by sigmoid activation. Since the transposed convolution includes learn-able parameters, it feeds the network with more information about the feature space. Our Proposed

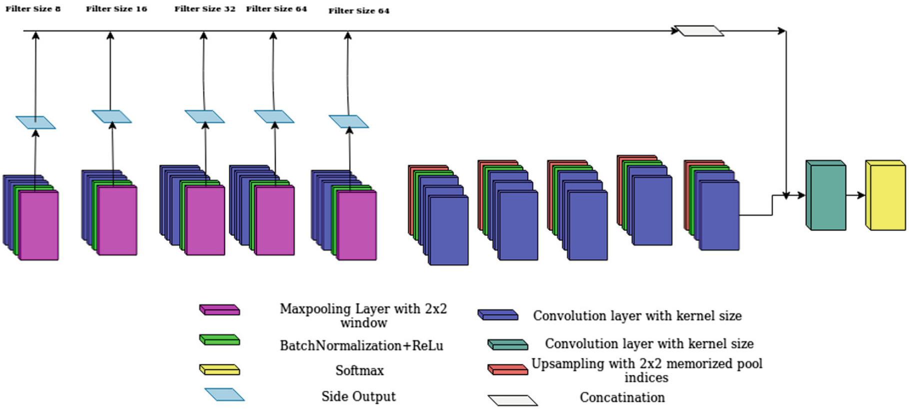

Fig. 1. Proposed architecture for detecting pixels

network architecture is shown in Fig. 1. The side outputs from each encoder layer are concatenated together in the final layer. The output of the merged layer is passed to a convolution layer with filter size 1. The number of filters in the final convolution layer is the number of desired classes. A softmax layer is added at the end of the convolution layers.

2.3 Data Augmentation

The deep network architectures require a balanced data-set with enough instances of each class. In this section, we discuss our data augmentation technique which alleviates the need to collect the huge number of data instances.

The deep network architecture proposed in this paper labels each pixel as crack or non-crack. Thus, a pixel map of each training sample from the original data set is generated manually. We define a pixel map as a binary image of the same size as the training sample images. Each pixel in the pixel map is assigned a value. The crack pixels in the original image are assigned the value 1 in the pixel map. The non-crack pixels are assigned the value 0.

Our data augmentation method generates thousands of sub-sample from a single image sample and its pixel map. We take each image randomly from our data set. At first, A random center position (x, y) is generated for that image. Then we sub-sample an image and its corresponding pixel map of height h and width w from the large image. After that, it is randomly decided if the image and its pixel map should be flipped (horizontally or vertically). Lastly, we perform gamma correction on the image randomly.

3 Experiment Results

We performed multiple experiments to train and evaluate our method. We compared our method with recent crack detection architectures in literature. In this section, we elaborately discussed our experimental setup, comparison with different methods and result interpretation.

3.1 Experimental Setup and Parameters

We optimized our proposed network architecture with Adam optimizer with a constant learning rate of 0.001. Since we have only two classes, a binary cross-entropy loss was used. It was trained for 100 episodes on a 1080Tx GPU.

We collected our data using an autonomous robotic system with NDE sensor fusion method [11] from various roads and bridges. These images contain many noises such as oil spilling, paint, stones, strips from tire screeching and many more. To generate the pixel map of each image we used Gimp software. To train our method we have generated pixel maps 33 large resolution images of $2304 \times 3456 \times 3$ (height $\times$ width $\times$ channel).

Using our data augmentation technique we have generated 5000 sub-sampled images of size 512×512 (height $\times$ width) for training and 1000 images of size 512×512 (height $\times$ width) for validation from the large resolution images of the database. In each episode of training, a different set of 6000 images (training + validation) is generated. We evaluated the performance of our network with recent deep architectures for crack classification. At first, we compared our results with the image block classification method [9,11] to demonstrate the nature of the segmentation problem affecting crack detection. Since the proposed network architecture labels pixel and uses an encoder-decoder architecture, we have also compared our results with recent encoder-decoder architecture for pixel-level labeling such as Unet [26], SegNet [1].

3.2 Comparative Analysis

For evaluation, we generated 200 images of size $1024 \times 1024 \times 3$ (height $\times$ width $\times$ channel) from our validation data set using our data augmentation technique. In this section, we elaborately explained the quantitative and qualitative results on different architectures based on our data-set.

Quantitative Comparisons. For evaluating our Proposed network architecture, we have taken into account different state-of-the-art statistical measurements such as true positive, false positive, true negative, false negative, accuracy, error rate, specificity, precision, recall, and F-1 score. We defined the crack pixels as the positive class and non-crack pixels as the negative class.

Thus, true positive (TP) is defined as the number of correctly detected crack pixels. True negative (TN) is defined as the number of pixels detected as non-crack that are labeled non-crack pixels in the ground truth. False-positive (FP) is the number of pixels that are erroneously detected as crack pixels. Finally, False negative (FN) is the number of crack pixels detected as non-crack pixels. A summary of the evaluation results performed on different methods are shown in Table 1.

Table 1. Quantitative Comparisons of the Proposed Network Architecture with Existing Crack Detection Methods.

Method	TP%	FP%	TN%	FN%	Acc.$^\diamond$	E.R.*	Spc.§	Prec.†	Rec.‡	F-1*
Gibbs [11]	25.2	23.0	77.0	25.2	76.9	23.1	77.0	0.004	0.25	0.007
Unet [26]	53.5	1.3	98.7	46.5	98.6	1.4	98.7	12.9	53.5	20.8
SegNet^1 [1]	55.4	1.3	98.7	44.6	98.5	1.5	98.7	13.2	55.4	21.3
SegNet^2 [1]	55.1	1.3	98.7	44.9	98.6	1.4	98.7	13.5	55.1	21.7
Proposed^1	56.1	**1.1**	**98.9**	43.9	**98.7**	**1.3**	**98.9**	**15.3**	56.1	**24.1**
Proposed^2	**57.2**	1.3	98.7	**42.8**	98.5	1.5	98.7	13.1	**57.2**	21.4

1: Conv-Layer Sizes (8-16-32-64-64)

2: Conv-Layer Sizes (16-32-64-128-128)

TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative

$\diamond$: Accuracy=$\frac{TP+TN}{TP+FP+TN+FN}$ $\quad$ $*$: Error Rate=$\frac{FP+FN}{TP+FP+TN+FN}$

$\S$: Specificity=$\frac{TN}{TN+FP}$ $\quad$ $\dagger$: Precision= $\frac{TP}{TP+FP}$

$\ddagger$: Recall=$\frac{TP}{TP+FN}$ $\quad$ $\star$: F-1 Measure=$2 \times \frac{Pre.\times Rec.}{Pre.+Rec.}$

In Table 1 we compared our results with both block-wise crack detection and pixel-wise crack detection techniques from the recent literature. The block detection method [11] under-performs than all the other existing pixel labeling method. The false-positive rate of this method is much higher than the other methods. Moreover, each detected crack block is a mixture of crack and non-crack pixels. This makes the block detection method to suffer from unacceptably high false positive, false negative and error rates compared to pixel labeling methods. It is evident from these measures, that a model with such a high error rate is not feasible for any classification.

We also compared our method with recent pixel classification methods from the literature, such as SegNet [1] and UNet [26]. For uniform comparison, we have employed two filter banks of size (8-16-32-64-64) and (16-32-64-128-128) for SegNet and our Proposed Architecture. The Unet architecture is designed with a regular number of filter banks of VGG-16, which is (64-128-256-512-512).

The SegNet^1 architecture uses a fewer number of filters compared to Unet, which reduces the number of learning parameters. The SegNet^1 architecture achieves more true positive rate, precision, recall and F1 score in comparison to Unet [26]. The false-positive rate, true negative rate, specificity measures are similar in both SegNet^1 and UNet. It under-performs than Unet in case of false-negative rate and error rate. Therefore, we can deduce that the SegNet^1 architecture acceptable in terms of less computational complexity and slightly better result than Unet.

The SegNet^2 model uses filter bank of size (16-32-64-128-128), which is larger than SegNet^1 but smaller than Unet. SegNet^2 achieves higher true positive rate, precision and F1 score than both Unet and SegNet^1. It has a similar false-positive rate, true negative rate, specificity measures with both SegNet^1 and

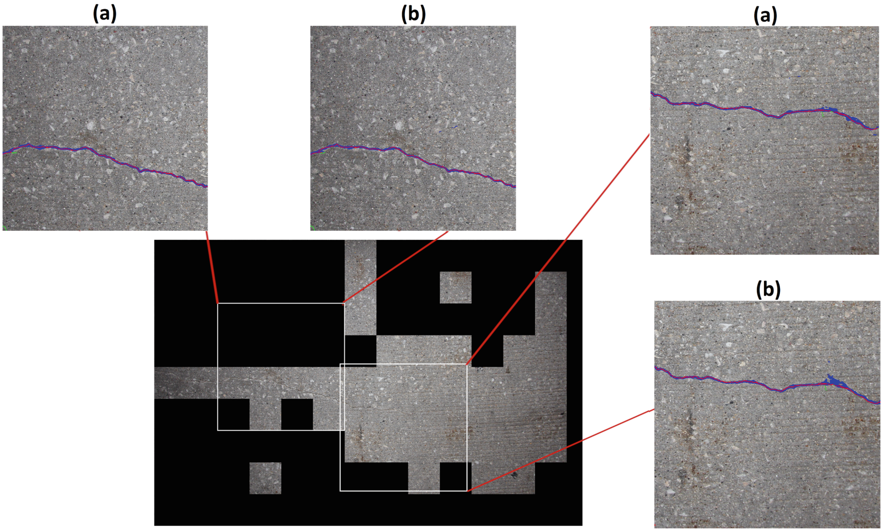

Fig. 2. Qualitative comparison of the proposed method with Gibbs [11] block detection architecture. The result shown in the middle is the crack detection using the Gibbs method, while (a) shows the pixels-wise crack detection results of the Proposed^1 architecture, and (b) shows the results of the Proposed^2 architecture on two random 1024×1024 sub-regions of the original image.

Unet. It reduces the error rate of SegNet^1, which is higher than Unet. In addition, it improves the accuracy in comparison to SegNet^1. However, the higher false-negative rate and lower recall rate in comparison to SegNet^1 represent the effect of gradient degradation. Though the overall method shows higher F1 score than the previous methods, this method is not suitable for crack classification considering the degradation of gradient problem.

The Proposed^1 architecture shows significant performance gain in all of the statistical measure. It achieves more true positive rate, false-positive rate, true negative rate, accuracy, specificity, precision, F-1 score and less error rate, the false-negative rate in comparison to SegNet^1, SegNet^2 and Unet.

To reveal the effect of parameter degradation in our architecture we have experimented our architecture by increasing the filter bank to (16-32-64-128-128). The Proposed^2 architectures achieve more true positive rate, recall rate and less false-negative rate than all the previous methods in Table 1. However, because of parameter degradation the error rate increases as well as the accuracy drops. Therefore, we can deduce that the Proposed^1 architecture is more suitable for crack pixel classification in comparison to other recent methods. Moreover, the Proposed^2 architecture is less affected by the gradient degradation problem in comparison to SegNet^2.

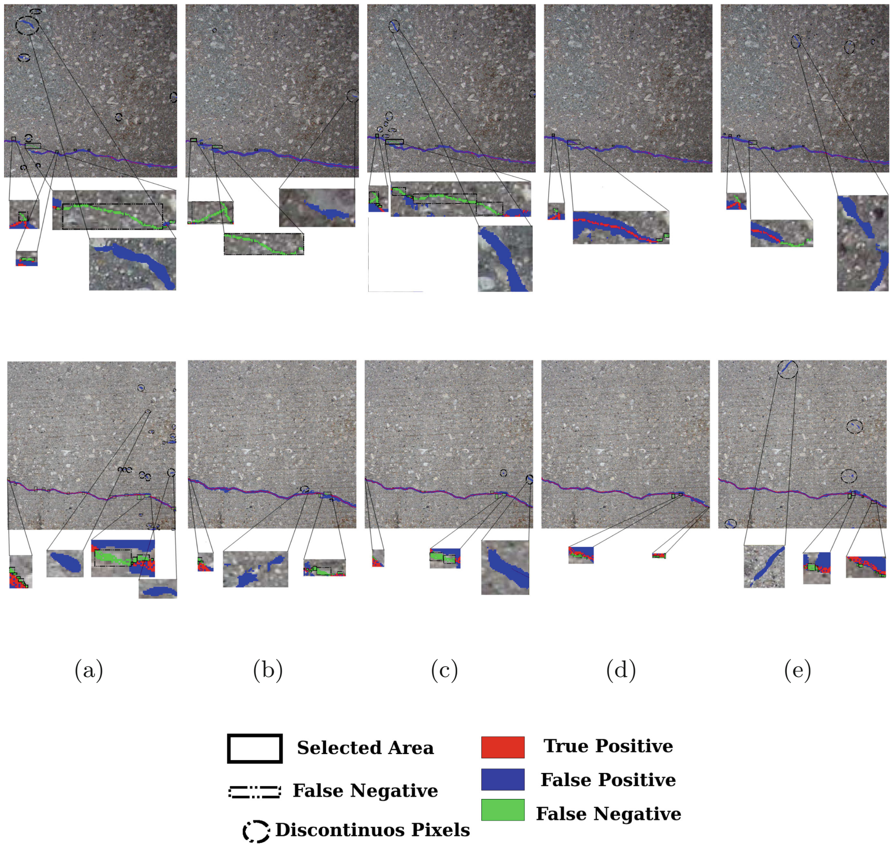

Fig. 3. Qualitative comparison of the proposed method with the state-of-the-art: (a) Unet [26]. (b) Segnet[1]. (c) Segnet[2]. (d) Proposed[1] Architecture. (e) Proposed[2] Architecture

It is worth noting that crack pixels exist in such small numbers, considering the vast majority of pixels in an image being normal. This makes the evaluation result biased toward the effective detection of non-crack pixels. To accommodate for this bias, we do not solely, rely on true positive rates. Instead, better measures of the unbiased performance of the crack detection methods are the accuracy, and the Error Rate, shown in Table 1. As observed from the table, both of the proposed methods outperform the existing techniques from the literature, while the Proposed[1] shows the highest Accuracy and lowest Error Rate.

Qualitative Comparisons. In this section, we have discussed the comparative results on sample images from our validation data set. We have shown comparative results from both block detection methods and pixel classification methods.

In Fig. 2 the comparative result of block detection method and our proposed method is shown. The Gibbs [11] architecture, divides an image into smaller sub-blocks of size 256×256. Each block is identified as crack and non-crack. The non-crack blocks are labeled as black in the image. Moreover, we have shown some results from the Proposed architecture on the same block. It is evident from Fig. 3, that our Proposed^1 architecture outperforms Gibbs [11] method in case of crack localization.

Figure 3 shows the result of different pixel-level classification methods. The number of crack pixels occurs are significantly lower than non-crack pixels. Therefore, we zoomed out some of our crack pixel location to represent a more clear result. The first row in Fig. 3, shows that the number of false-negative pixels in the same area is higher in Unet. It is reduced in both SegNet^1 and SegNet^2 architecture. However, in the same area, our Proposed^1 architecture significantly reduces the number of false-negative pixels in comparison to SegNet^1, SegNet^2 and Unet. The Proposed^2 have more false-negative pixels than Proposed^1 but it has less false negative pixels than the other previous architectures.

One of the most important observations in this study is, the pixel labeling method is affected by pixel discontinuity. We define pixel discontinuity is the detection of the anomalous length of pixels (not continuous pixels as crack) as cracks. These discontinuous pixels are the effect of gradient degradation. The discontinuous pixels are shown by the circled area in Fig. 3.

Our results show that the Proposed^1 architecture removes most of these discontinuous pixels significantly. From Figure 3, we can see that the Unet architecture have many discontinuous pixels. SegNet^1 and SegNet^2 reduces the length and number of these pixels. On the other hand, the discontinuous pixels affect the Proposed^1 architecture very little. Moreover, the Proposed^2 architecture have some discontinuous pixels (last image of row 2). However, the length of the discontinuous pixels is reduced in comparison to Unet. In addition, it has more true-positive pixels in comparison to Unet and SegNet architecture.

4 Conclusions and Future Work

We presented in this paper, a deep convolutional network architecture while accommodating for the feature loss. Our architecture shows that with a limited number of filters we can alleviate the feature loss. The main motivation of this work is to design an efficient crack detection system for civil infrastructure inspection. Our method also shows significant improvement over existing methods. Moreover, we presented data augmentation techniques, which can significantly improve the performance of any architecture. On the contrary, our architecture consumes more memory because of its large number of parameters. In the future, we would optimize the memory structure as well as achieve better performance.

References

1. Badrinarayanan, V., Handa, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293 (2015)
2. Bai, X., Zhou, F., Xue, B.: Multiple linear feature detection based on multiple-structuring-element center-surround top-hat transform. Appl. Opt. **51**(21), 5201–5211 (2012)
3. Billah, U.H., La, H.M., Tavakkoli, A., Gucunski, N.: Classification of concrete crack using deep residual network. In: 9th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-9), pp. 1–6, August 2019
4. Bray, J., Verma, B., Li, X., He, W.: A neural network based technique for automatic classification of road cracks. In: 2006 International Joint Conference on Neural Networks, IJCNN 2006, pp. 907–912. IEEE (2006)
5. Cha, Y.J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Comput. Aided Civil Infrastruct. Eng. **32**(5), 361–378 (2017)
6. Dinh, T.H., Ha, Q., La, H.M.: Computer vision-based method for concrete crack detection. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1–6. IEEE (2016)
7. Elbehiery, H., Hefnawy, A., Elewa, M.: Surface defects detection for ceramic tiles using image processing and morphological techniques (2005)
8. Gavilán, M., et al.: Adaptive road crack detection system by pavement classification. Sensors **11**(10), 9628–9657 (2011)
9. Gibb, S., La, H.M., Louis, S.: A genetic algorithm for convolutional network structure optimization for concrete crack detection. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, July 2018
10. Gibb, S., Le, T., La, H.M., Schmid, R., Berendsen, T.: A multi-functional inspection robot for civil infrastructure evaluation and maintenance. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2672–2677, September 2017
11. Gibb, S., La, H.M., Le, T., Nguyen, L., Schmid, R., Pham, H.: Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection. J. Field Rob. **35**(6), 988–1004 (2018)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
13. La, H.M., Gucunski, N., Kee, S.H., Nguyen, L.: Data analysis and visualization for the bridge deck inspection and evaluation robotic system. Vis. Eng. **3**(1), 1–16 (2015)
14. La, H.M., Gucunski, N., Kee, S.-H., Yi, J., Senlet, T., Nguyen, L.: Autonomous robotic system for bridge deck data collection and analysis. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1950–1955, September 2014
15. La, H.M.: Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation. IEEE/ASME Trans. Mechatron. **18**(6), 1655–1664 (2013)
16. La, H.M., Gucunski, N., Dana, K., Kee, S.H.: Development of an autonomous bridge deck inspection robotic system. J. Field Rob. **34**(8), 1489–1504 (2017)

17. Landstrom, A., Thurley, M.J.: Morphology-based crack detection for steel slabs. IEEE J. Sel. Top. Signal process. **6**(7), 866–875 (2012)
18. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
19. Li, Q., Liu, X.: Novel approach to pavement image segmentation based on neighboring difference histogram method. In: 2008 Congress on Image and Signal Processing, CISP 2008, vol. 2, pp. 792–796. IEEE (2008)
20. Lim, R.S., La, H.M., Shan, Z., Sheng, W.: Developing a crack inspection robot for bridge maintenance. In: 2011 IEEE International Conference on Robotics and Automation, pp. 6288–6293, May 2011
21. Lim, R.S., La, H.M., Sheng, W.: A robotic crack inspection and mapping system for bridge deck maintenance. IEEE Trans. Autom. Sci. Eng. **11**(2), 367–378 (2014)
22. Maode, Y., Shaobo, B., Kun, X., Yuyao, H.: Pavement crack detection and analysis for high-grade highway. In: 2007 8th International Conference on Electronic Measurement and Instruments, ICEMI 2007, pp. 4–548. IEEE (2007)
23. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)
24. Oliveira, H., Correia, P.L.: Automatic road crack segmentation using entropy and image dynamic thresholding. In: 2009 17th European Conference on Signal Processing, pp. 622–626. IEEE (2009)
25. Prasanna, P., Dana, K., Gucunski, N., Basily, B.: Computer-vision based crack detection and analysis. In: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2012, vol. 8345, p. 834542. International Society for Optics and Photonics (2012)
26. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Sun, Y., Salari, E., Chou, E.: Automated pavement distress detection using advanced image processing techniques. In: 2009 IEEE International Conference on Electro/Information Technology, EIT 2009, pp. 373–377. IEEE (2009)
29. Sy, N., Avila, M., Begot, S., Bardet, J.C.: Detection of defects in road surface by a vision system. In: 2008 14th IEEE Mediterranean Electrotechnical Conference, MELECON 2008, pp. 847–851. IEEE (2008)
30. Tanaka, N., Uematsu, K.: A crack detection method in road surface images using morphology. MVA **98**, 17–19 (1998)
31. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1395–1403 (2015)
32. Zhao, H., Qin, G., Wang, X.: Improvement of canny algorithm based on pavement edge detection. In: 2010 3rd International Congress on Image and Signal Processing. vol. 2, pp. 964–967. IEEE (2010)