

Pavement Crack Segmentation using a U-Net based Neural Network

Raido Lacorte Galina, Thadeu Pezzin Melo, Karin Satie Komati
Programa de Pós-graduação em Computação Aplicada (PPComp)
Campus Serra do Instituto Federal do Espírito Santo (IFES)
raido.l.g@hotmail.com, tpezzin@gmail.com, kkomati@ifes.edu.br

Resumo—Cracks on the concrete surface are symptoms and precursors of structural degradation and hence must be identified and remedied. However, locating cracks is a time-consuming task that requires specialized professionals and special equipment. The use of neural networks for automatic crack detection emerges to assist in this task. This work proposes one U-Net based neural network to perform crack segmentation, trained with the Crack500 and DeepCrack datasets, separately. The U-Net used has seven contraction and seven expansion layers, which differs from the original architecture of four layers of each part. The IoU results obtained by the model trained with Crack500 was 71.03%, and by the model trained with DeepCrack was 86.38%.

Index Terms—CNN, Crak500, DeepCrack, crack segmentation.

I. INTRODUÇÃO

As fissuras e as trincas, ou popularmente rachaduras, são manifestações patológicas das edificações observadas em construções de alvenaria e estruturas de concreto, geralmente causadas por tensões dos materiais. Se os materiais sofrerem um esforço maior que sua resistência, acontece a falha, provocando uma abertura. Rachaduras presentes na superfície do concreto são sintomas e precursores da degradação das estruturas [1]. Elas representam informações indicativas no que diz respeito ao estado da construção civil quanto à segurança e à durabilidade [2].

No processo de manutenção das construções é necessário fazer a localização e o reparo das rachaduras. Ainda que a detecção dessas falhas seja uma etapa essencial, quando é feita manualmente é uma tarefa tediosa, demorada e que requer *expertise* [3]. Outro problema que surge na avaliação das rachaduras de estruturas de concreto é que muitas vezes elas se localizam em lugares de difícil acesso [4]. A inspeção, neste caso, exige equipamento especial, equipe especificamente treinada para a atividade, e muitas vezes o fechamento da estrutura.

Neste contexto, sistemas automáticos têm sido propostos por diversos autores ao longo dos anos para auxiliar na detecção de rachaduras [5], [6]. Uma das formas mais simples de se considerar o problema de detecção de rachaduras é efetuando uma classificação binária das imagens, indicando se há ou não uma rachadura na imagem [7], [8]. Uma outra forma de abordar o problema da detecção de rachaduras é fazer a

segmentação das rachaduras nas imagens, isto é, definir o contorno das rachaduras [9], [10].

A detecção ou a segmentação de rachaduras é uma tarefa menos complicada quando o primeiro plano da imagem contém apenas a rachadura e o plano de fundo possui uma textura simples. A Fig. 1a apresenta essa característica, de se ter uma rachadura sobre uma textura simples. A Fig. 1c apresenta uma superfície com uma textura mais complexa, com maior granularidade. Além disso, a imagem pode apresentar outras informações, ruídos do ambiente externo, que dificultam a tarefa [11], tais como manchas de óleo ou tinta (como na Fig. 1b), ou que existam sombras com a mesma intensidade das rachaduras [3], como na Fig. 1d. Outra dificuldade é que as rachaduras apresentam diversas estruturas topológicas em que não é possível distinguir uma regularidade ou continuidade. Todos os problemas citados dificultam o processo de identificação automática das falhas.

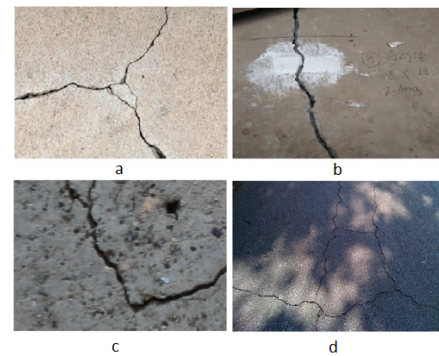


Figura 1. Exemplos de imagens de rachaduras. Adaptado de [2].

Para o problema da segmentação de rachaduras, a técnica de Aprendizado Profundo (*Deep Learning*) tem sido usada como solução em artigos recentes [1], [2], [7], [8], [10]. Mais especificamente, a arquitetura de rede neural baseada na U-Net tem sido usada para segmentação de rachaduras [9], dada a vantagem de que necessita de poucas imagens de treinamento [12]. A definição da arquitetura de uma rede neural ainda é um desafio [13]. Um questionamento seria: qual é a quantidade de camadas da U-Net que seria mais adequada para a segmentação de rachaduras?

O objetivo deste artigo é avaliar os resultados de uma arquitetura baseada na U-Net, com mais camadas, para o problema

da segmentação semântica de rachaduras. A quantidade de camadas foi inspirado no trabalho de [14], que usa 7 camadas de contração e 7 camadas de expansão na arquitetura U-Net, ao invés das 4 originais. A comparação será feita com trabalho de Lau *et al.* [9] que usa a U-Net original com a base de dados Crack500, logo, uma contribuição deste trabalho é promover a comparação de diferentes quantidades de camadas da U-Net na segmentação de rachaduras. Nos experimentos, foi usada uma segunda base de dados, o DeepCrack, e comparando os resultado com o artigo de origem da base [2]. As métricas de avaliação usadas são a acurácia, a precisão, a revocação, a medida-F (*F-score*) e a interseção sobre a união.

O resto do artigo está organizado da seguinte forma. A Seção II descreve alguns trabalhos correlatos. A Seção III versa sobre os materiais e métodos utilizados. A Seção IV apresenta os resultados obtidos. Por último, a Seção V conclui o artigo e apresenta direções para próximos trabalhos.

II. TRABALHOS CORRELATOS

Nesta seção são detalhados os trabalhos cujos resultados são comparados ao deste artigo. O trabalho de Liu *et al.* [2] que deu origem à base de dados DeepCrack, o trabalho de Yang *et al.* [3] que deu origem à base de dados Crack500 e o trabalho de Lau *et al.* [9] que usou uma proposta baseada na rede U-Net.

No trabalho de Liu *et al.* [2], é proposto um método de segmentação de rachaduras utilizando uma rede neural convolucional hierárquica profunda, chamada de DeepCrack. A DeepCrack é inspirada em uma rede de detecção de borda e consiste na combinação das redes totalmente convolucionais estendidas (FCN, do inglês *Fully Convolutional Network*) e das redes profundamente supervisionadas (DSN, do inglês *Deeply-Supervised Nets*). Durante o treinamento, o modelo elaborado aprende e agrega recursos de várias escalas e níveis, desde as camadas convolucionais baixas até as camadas convolucionais de alto nível. Na arquitetura proposta, são utilizados as 13 primeiras camadas convolucionais da VGG16. Foram testadas algumas variações da rede DeepCrack na base de dados DeepCrack, e todas alcançaram medida-F1 superiores a 79%, chegando a 86,5%, sendo superiores às outras arquiteturas comparadas.

O trabalho de Yang *et al.* [3] propôs uma nova arquitetura, denominada Feature Pyramid and Hierarchical Boosting Network (FPHBN), para detecção de rachaduras em pavimentos. A rede integra informações de contexto de baixo nível em uma forma de pirâmide de características. O método foi avaliado em cinco conjuntos de dados e comparado com métodos de detecção de bordas e segmentação semântica. De acordo com o critério da melhor medida-F no conjunto de dados para uma escala fixa (ODS), para a base de dados Crack500, o ODS foi de 0,604 para o FPHBN, melhor que os métodos comparados.

O trabalho de Lau *et al.* [9] apresenta uma técnica de aprendizagem profunda baseada em uma rede neural convolucional para realizar a tarefa de segmentação de rachaduras de pavimento. O artigo propõe uma arquitetura baseada na U-Net

com um codificador de ResNet-34 pré-treinado e implementa um cronograma de treinamento de “um ciclo” para acelerar a convergência. A abordagem consegue atingir valores de medida-F de 96% para o conjunto de dados CFD e de 73% para o conjunto de dados Crack500. O artigo alcançou os melhores resultados na Crack500, quando comparado a outros 4 trabalhos, sendo 2 deles baseados na U-Net.

III. MATERIAIS E MÉTODOS

A. Materiais

Para realizar os experimentos para avaliar o desempenho da abordagem proposta neste trabalho serão utilizadas duas bases de dados contendo imagens de rachaduras. Cada imagem presente em ambas bases de dados dispõe de uma máscara, que é uma anotação da rachadura, o *benchmark*, e possui a mesma dimensão da imagem original.

1) *Banco de imagens Crack500*: O conjunto de dados Crack500 foi proposto por [3] e contém imagens obtidas com telefones celulares ao redor do campus principal da Temple University. Consistiu inicialmente de 500 imagens de tamanhos variados, em torno de 2.000×1.500 píxeis. Foi dividida em 250 amostras de treinamento, 200 testes e 50 amostras de validação.

Cada imagem foi recortada em 16 regiões não sobrepostas e apenas as regiões que contêm mais de 1.000 píxeis de rachadura foram mantidas. Desta forma, a base processada ficou com 1.896 imagens de treinamento, 348 imagens de validação e 1.124 imagens de teste, num total de 3.368 imagens. Os dados de validação são usados para escolher o melhor modelo durante o processo de treinamento para evitar *overfitting*. Uma vez que o modelo é definido, é testado com os dados de teste para avaliação de generalização. As imagens estão em um formato jpg, e os mapas binários, as máscaras, em formato png. Na Fig. 2 há dois exemplos de imagens de rachaduras na primeira linha, e as suas respectivas máscaras na segunda linha.

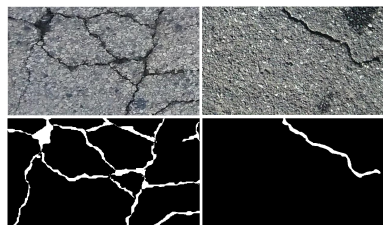


Figura 2. Exemplos de imagens do conjunto Crack500. Adaptado de [3].

2) *Banco de imagens DeepCrack*: O conjunto de dados DeepCrack foi proposto por [2] e contém 537 imagens RGB com segmentações de rachadura manualmente anotadas. O total de imagens foi dividido em dois sub-conjuntos. O primeiro, de treinamento, contém 300 imagens. O segundo, de teste, contém 237. Todas as imagens possuem tamanho 544×384 píxeis. As larguras da rachadura variam de 1 a 180 píxeis. As imagens estão em formato jpg e as máscaras de segmentações em formato png. Uma informação dada pelo artigo de origem

é que a quantidade de píxeis de rachadura correspondem à 3,54% do total de píxeis, demonstrando o quanto as classes dos píxeis são desbalanceadas.

Os dois tipos de superfície presentes na base de dados é o concreto e o asfalto, que são comumente usados em construções. Na Fig. 3, é possível observar exemplos de rachaduras, na primeira linha da figura estão as imagens originais e na segunda linha, suas respectivas máscaras de segmentações. Pode-se perceber nas imagens a presença de diferentes características nos fundos, com diferentes texturas quanto à coloração, granularidade e rugosidade, além de ângulos distintos de captura da imagem e rachaduras de larguras variáveis. Há imagens com manchas, sombras e pequenos objetos.



Figura 3. Exemplos de imagens do conjunto DeepCrack, adaptado de [2].

B. Método Proposto

A U-Net é uma rede neural convolucional (CNN, do inglês *convolutional neural network*) que foi desenvolvida por [12]. Ela foi originalmente projetada para segmentação de imagens biomédicas, com um foco específico na segmentação de estruturas celulares com membranas finas. Posteriormente, outros artigos também usaram a U-Net em outras tarefas de segmentação [15].

O nome provém do formato em ‘U’ da visualização da arquitetura proposta pelo artigo original. A arquitetura consiste em três seções: a contração (*encoder*, lado esquerdo da Fig. 6), o gargalo (*bottleneck*) e a seção de expansão (*decoder*, lado direito da Fig. 6). O caminho de contração segue a arquitetura típica de uma rede convolucional e é composta por vários blocos. Cada bloco recebe uma entrada e aplica duas camadas de convolução 3x3, seguida por uma unidade linear retificada (ReLU), seguida por uma camada de agrupamento máximo (*max pooling*) de 2x2. O número de *kernels*, ou mapa de características, dobra após cada bloco para que a arquitetura possa aprender as estruturas complexas de maneira eficaz. A camada mais inferior, *bottleneck*, faz a mediação entre a camada de contração e a camada de expansão. Ele usa duas camadas convolucionais 3x3 seguidas por uma camada 2x2 para cima (*upsampling*).

Semelhante à camada de contração, a expansão também consiste em vários blocos. Cada bloco tem duas camadas convolucionais 3x3 seguidas por uma camada de *upsampling* 2x2. Além disso, após cada bloco, o número de mapas de recursos diminui para a metade, mantendo a simetria com a seção de expansão. Em cada bloco de expansão, concatenam-se o mapa de características da camada de contração corres-

pondente. Essa ação garante que as características aprendidas durante a contração da imagem sejam usadas para reconstruí-la. O número de blocos de expansão é igual ao número de blocos de contração.

Na Fig. 6 é apresentada a arquitetura da U-Net utilizada neste trabalho. Ela difere da rede proposta por [12] no que diz respeito ao tamanho da imagem de entrada, que neste caso é 256x256 e na arquitetura original é de 572x572. Além disso, a quantidade de operações de *max pooling* e *upsampling* na rede usada neste trabalho foi de 7 de cada, enquanto que na U-Net original foi 4 de cada. Com isso, foram 41 camadas convolucionais totais neste modelo de U-Net, sendo que no original eram 23.

Os modelos de segmentação foram implementados usando um *framework* do Keras. Os modelos foram treinados usando o serviço de *notebooks* hospedados do Jupyter, o Colab, rodado no navegador Chrome na versão 93.0.4577.82 (Versão oficial) 64 bits. As GPUs disponíveis no Colab incluem Nvidia K80s, T4s, P4s e P100s.

C. Treinamento

Para o treinamento dos modelos, as imagens passaram por uma etapa de pré-processamento. No conjunto DeepCrack, as imagens possuem tamanho 544 x 384 píxeis, enquanto no conjunto Crack500 têm tamanho de 640 x 360 píxeis. As imagens de ambas as bases de dados passaram por um redimensionamento para que ficassem com o tamanho de 256 x 256 píxeis.

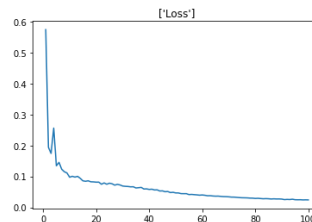


Figura 4. Evolução da função perda para o modelo treinado com o conjunto de dados Crack500.

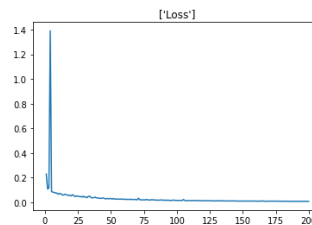


Figura 5. Evolução da função perda para o modelo treinado com o conjunto de dados DeepCrack.

O modelo do conjunto Crack500 foi treinado com 100 épocas o que levou aproximadamente 180 minutos, e o modelo do conjunto DeepCrack foi treinado com 200 épocas, o que levou aproximadamente 40 minutos. Na Fig. 4 está a evolução da função perda com as épocas para o modelo treinado com

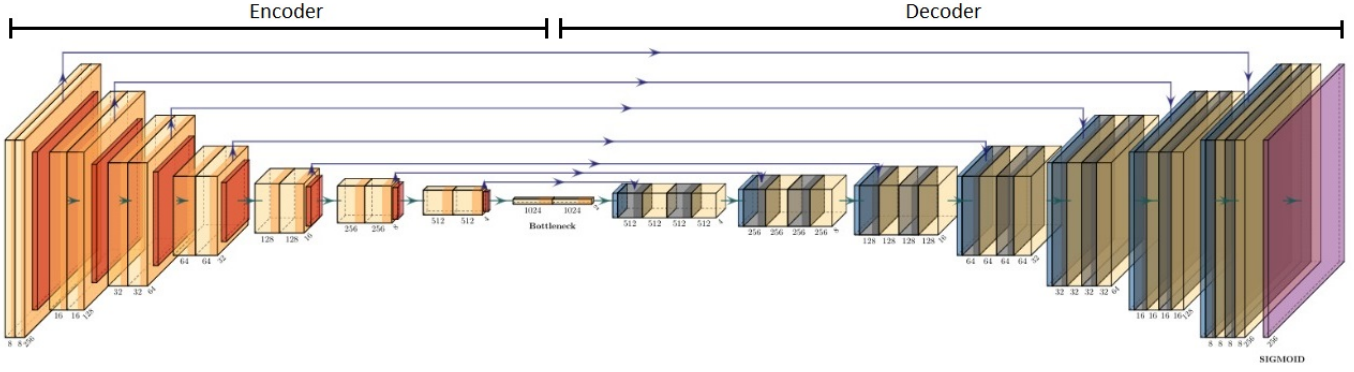


Figura 6. Adaptação da U-Net proposta em [14], figura criada usando biblioteca PlotNeuralNet.

o conjunto de dados Crack500, e na Fig. 5, está a evolução da função perda com as épocas para o modelo treinado com o conjunto de dados DeepCrack. É possível observar que o modelo treinado com o conjunto DeepCrack convergiu mais rapidamente que o modelo treinado com o conjunto Crack500.

D. Métricas de Comparação

Para a avaliação quantitativa da segmentação são usadas seis métricas. Três métricas são: a precisão (P), a revocação (R) e o medida-F (F). A Eq. 1 e a Eq. 2 definem a precisão e a revocação, respectivamente, onde VP é o número dos verdadeiros positivos, VN é o número dos verdadeiros negativos, FP é o número dos falsos positivo e FN é o número dos falsos negativos. A Eq. 3 define a medida-F:

$$\text{Precisão} = \frac{VP}{(VP + FP)} \quad (1)$$

$$\text{Revocação} = \frac{VP}{(VP + FN)} \quad (2)$$

$$\text{medida-F} = \frac{2 \times (\text{Precisão} \times \text{Revocação})}{\text{Precisão} + \text{Revocação}} \quad (3)$$

Ademais, outras três métricas também são utilizadas: a acurácia global (G), a acurácia média de classe (C) e a média da interseção sobre a união (IoU, do inglês *Intersection over Union*). A Eq. 4 define a acurácia global, que calcula a porcentagem de píxeis corretamente predita, a Eq. 5 define a acurácia média de classe, que é a acurácia predita de todas as classes e a Eq. 6 define a média do IoU. Nas equações, n_{ij} é o número de píxeis da classe i preditos como classe j , onde existem $n_{classes}$ diferentes (no caso deste trabalho, são 2), e $t_i = \sum_j n_{ij}$ é o número total de píxeis da classe i (tanto verdadeiros positivos e falsos positivos estão incluídos).

$$\text{Acurácia global} = \frac{\sum_i n_{ii}}{\sum_i t_i} \quad (4)$$

$$\text{Acurácia média de classe} = \frac{1}{n_{classe}} \sum_i \frac{n_{ii}}{t_i} \quad (5)$$

$$\text{IoU}_{\text{média}} = \frac{1}{n_{classe}} \frac{\sum_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \quad (6)$$

IV. RESULTADOS

Nesta seção, os resultados qualitativos e quantitativos são apresentados.

A. Resultados Qualitativos

Alguns exemplos dos resultados da Crack500 são apresentados na Fig. 7. Cada conjunto de 3 imagens da Fig. 7 apresenta, na ordem: a máscara de segmentação, imagem de entrada e o resultado da segmentação do modelo treinado. Os resultados da coluna à esquerda são os que consideramos de boa qualidade e os da coluna direita, os que apresentaram problemas. Em geral, as segmentações ruins deste conjunto de dados ficaram fragmentadas. As imagens da base Crack500 são mais homogêneas que as da base DeepCrack, mas possuem um contraste menor com o fundo quando comparadas com as rachaduras presentes no conjunto Deepcrack.

A Fig. 8 contém exemplos de resultados do conjunto de dados DeepCrack. Da mesma forma que a figura anterior, cada conjunto apresenta, na ordem: a máscara de segmentação, imagem de entrada e o resultado da segmentação do modelo treinado. E da mesma forma, a coluna à esquerda são os que consideramos de boa qualidade e os da coluna direita, os que apresentaram problemas. Na primeira coluna da Fig. 8 é possível observar diferentes texturas como plano de fundo que obtiveram uma boa segmentação. Entretanto, na segunda coluna são mostradas imagens que possuíram uma segmentação com uma qualidade inferior. A primeira linha da segunda coluna mostra uma rachadura cuja segmentação ficou fragmentada. A segunda linha mostra uma rachadura mais grossa que, em seu canto inferior direito, nota-se que houve uma lacuna na segmentação. A terceira linha mostra um resultado em que há a tentativa da rede neural de segmentar a parte superior da textura como uma rachadura, acredita-se que foi pela alta granularidade dos pontos escuros da textura do fundo. E a quarta linha, a rede neural segmentou a sombra da flor vermelha como uma rachadura.

Todas as imagens da Fig. 1 (na Seção de Introdução) são da base de dados DeepCrack, e os resultados com o modelo treinado com a DeepCrack são apresentados na Fig. 9. A Fig. 1 contém um exemplo de rachadura com alto contraste contra

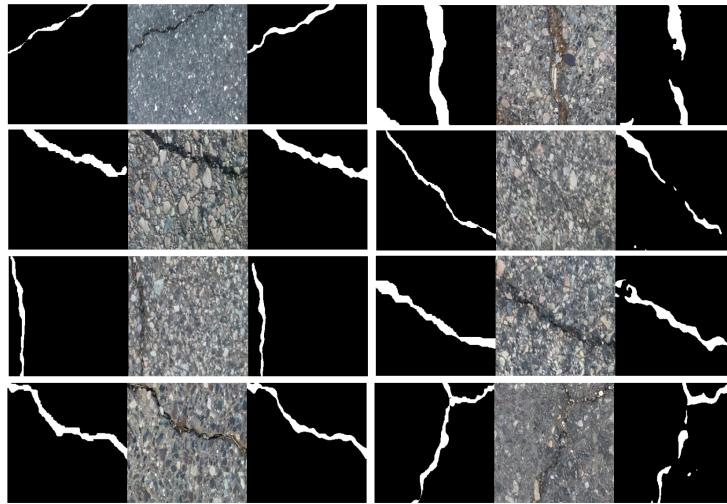


Figura 7. Exemplos de resultados da U-Net na base de dados Crack500. Cada conjunto de imagens, da esquerda para direita: máscara, imagem de teste e resultado da segmentação.



Figura 8. Exemplos de resultados da U-Net na base de dados DeepCrack. Cada conjunto de imagens, da esquerda para direita: máscara, imagem de teste e resultado da segmentação.

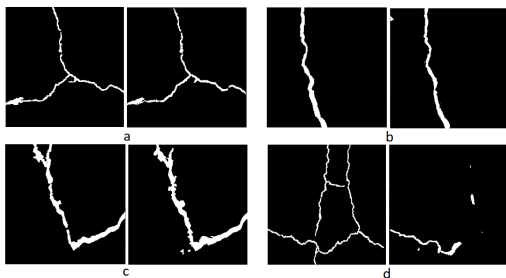


Figura 9. Segmentação dos exemplos de imagens de rachaduras da Fig. 1. A imagem à esquerda é a máscara e a imagem à direita é o resultado da segmentação.

uma textura simples como pano de fundo e outras três imagens de rachaduras mais complexas. Cada par representa a máscara

original e a segmentação pelo modelo, respectivamente. Nela, é possível observar que a imagem mais simples, a Fig. 9a, obteve uma ótima segmentação comparado à máscara de segmentação. Na Fig. 9b, o modelo foi capaz de reconhecer que a mancha de tinta não faz parte da rachadura. Entretanto, na imagem com o fundo de textura de alta granularidade, a Fig. 9c, a segmentação interpretou algum dos pontos mais escuros que fazem parte da textura como rachadura. E na imagem com sombra, a Fig. 9d, a segmentação ficou incompleta.

B. Resultados Quantitativos

A Tab. II e a Tab. I mostram o resultado das seis métricas para os dois conjuntos de dados Crack500 e DeepCrack. Ambas as redes obtiveram uma acurácia global elevada, acima de 96%, mas isso já era esperado uma vez que no problema de detecção de rachaduras as classes estão desbalanceadas, isto é,

há muito mais píxeis de não rachaduras que de rachaduras, mas elas também obtiveram resultados elevados de acurácia média de classe, onde acontece uma normalização por classe, com valores superiores a 80%. Também alcançaram bons resultados de IoU, com a rede treinada com o conjunto DeepCrack apresentando $\text{IoU}_{\text{média}}$ de 86,38% e a rede treinada com o Crack500 apresentando $\text{IoU}_{\text{média}}$ de 71,03%.

Tabela I
TABELA COMPARATIVA PARA A BASE CRACK500

	P	R	F	G	C	$\text{IoU}_{\text{média}}$
Artigo [9]	74,26%	72,85%	73,27%	–	–	–
este trabalho	96,25%	80,94%	87,93%	96,25%	80,94%	71,03%

Tabela II
TABELA COMPARATIVA PARA A BASE DEEPCRAK

	P	R	F	G	C	$\text{IoU}_{\text{média}}$
Artigo [2]	86,10%	86,90%	86,50%	97,50%	97,00%	80,2%
este trabalho	90,41%	80,43%	85,13%	98,70%	90,01%	86,38%

A Tab. I apresenta os resultados obtidos no artigo de Lau *et al.* [9] e os resultados deste trabalho na base de dados Crack500. A proposta deste artigo apresentou melhores resultados em todas as métricas apresentadas. A proposta de [9] apresenta apenas as métricas de precisão, revocação e medida-F. É importante lembrar que a proposta de Lau *et al.* [9] tem a mesma quantidade de camadas da U-Net original, e teve melhores resultados quando comparado com outros dois trabalhos também baseados na U-Net. Já a comparação com o artigo que propôs a base de dados Crack500 [3], é possível comparar apenas o resultado deste trabalho com o ODS, que é a melhor medida-F para uma determinada escala. O ODS de [3] foi de 0,604, enquanto o resultado deste trabalho foi de 0,8793, isto é, superior ao obtido no artigo original.

Na Tab. II são apresentados os resultados do artigo que propôs a base de dados DeepCrack [2] e os resultados deste trabalho, que foram compatíveis. Os melhores resultados estão marcados em negrito. Quando comparado com os resultados obtidos por este artigo, o artigo original obteve resultado inferior de precisão, mas valores superiores de revocação e medida-F. A proposta deste trabalho alcançou uma acurácia global maior que o artigo original, mas foi pior na acurácia média por classe. E por fim, este trabalho conseguiu uma $\text{IoU}_{\text{média}}$ maior que o artigo original.

V. CONCLUSÃO

Neste artigo, foi avaliado o resultado de uma proposta baseada na U-Net, mas com mais camadas que a versão original, para o problema de segmentação de rachaduras em imagens. Os resultados foram melhores para a base de dados Crack500 e compatíveis para a base de dados DeepCrack.

Os modelos apresentaram resultados promissores nas métricas em que foram avaliados, e pode-se considerar que o uso de mais camadas que o modelo original gerou melhor resultado. No entanto, a questão de quantas camadas deve ter uma rede neural, ainda é um desafio. Conforme [14], “o

design da arquitetura neural depende muito do conhecimento e experiência anteriores dos pesquisadores”. Portanto, um trabalho futuro é usar as técnicas de Neural Architecture Search para conseguir automaticamente a melhor arquitetura neural.

Outra questão, é que é necessário destacar que cada modelo treinado apresenta bons resultados para aquele conjunto de dados específico. Mudanças nas angulações das imagens, na iluminação e na superfície podem levar a uma diminuição das métricas de avaliação. Por isso, em trabalhos futuros, a proposta será treinar a rede neural com um conjunto de dados misto, agregando outras bases de dados, além de incluir no processo de treinamento exemplos negativos de imagens que não contenham rachaduras, principalmente exemplos de sombras.

REFERÊNCIAS

- [1] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, “Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning,” *Automation in Construction*, vol. 118, p. 103291, 2020.
- [2] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, “DeepCrack: A deep hierarchical feature learning architecture for crack segmentation,” *Neurocomputing*, vol. 338, pp. 139–153, 2019.
- [3] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, “Feature pyramid and hierarchical boosting network for pavement crack detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.
- [4] A. Ellenberg, A. Kotsos, F. Moon, and I. Bartoli, “Bridge related damage quantification using unmanned aerial vehicle imagery,” *Structural Control and Health Monitoring*, vol. 23, no. 9, pp. 1168–1179, 2016.
- [5] A. Mohan and S. Poobal, “Crack detection using image processing: A critical review and analysis,” *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.
- [6] H. Zakeri, F. M. Nejad, and A. Fahimifar, “Image based techniques for crack detection, classification and quantification in asphalt pavement: a review,” *Archives of Computational Methods in Engineering*, vol. 24, no. 4, pp. 935–977, 2017.
- [7] Q. Yang, W. Shi, J. Chen, and W. Lin, “Deep convolution neural network-based transfer learning method for civil infrastructure crack detection,” *Automation in Construction*, vol. 116, p. 103199, 2020.
- [8] M. Stoński, “A comparison of deep convolutional neural networks for image-based detection of concrete surface cracks,” *Computer Assisted Methods in Engineering and Science*, vol. 26, no. 2, pp. 105–112, 2019.
- [9] S. L. Lau, E. K. Chong, X. Yang, and X. Wang, “Automated pavement crack segmentation using U-Net-based convolutional neural network,” *IEEE Access*, vol. 8, pp. 114 892–114 899, 2020.
- [10] H. Li, J. Zong, J. Nie, Z. Wu, and H. Han, “Pavement crack detection algorithm based on densely connected and deeply supervised network,” *IEEE Access*, vol. 9, pp. 11 835–11 842, 2021.
- [11] Y. Wang, K. Song, J. Liu, H. Dong, Y. Yan, and P. Jiang, “Renet: rectangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks,” *Measurement*, vol. 170, p. 108698, 2021.
- [12] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [13] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, “A comprehensive survey of neural architecture search: Challenges and solutions,” *arXiv preprint arXiv:2006.02903*, 2020.
- [14] F. Wang, A. Eljarrat, J. Müller, T. R. Henninen, R. Erni, and C. T. Koch, “Multi-resolution convolutional neural networks for inverse problems,” *Scientific reports*, vol. 10, no. 1, pp. 1–11, 2020.
- [15] D. dos Santos, A. Silva, P. de Faria, B. Travengolo, and M. do Nascimento, “Impacts of color space transformations on dysplastic nuclei segmentation using CNN,” in *Anais do XVI Workshop de Visão Computacional*. SBC, 2020, pp. 6–11.