Detecção de Fissuras utilizando Redes Neurais Convolucionais

*Raianny Proença C. de Oliveira, *Claudio Roberto M. Mauricio, †Valéria Nunes dos Santos e *Fabiana Frata F. Peres
*Universidade Estadual do Oeste do Paraná
Ciências da Computação - Centro de Engenharias e Ciências Exatas
Foz do Iguaçu, Paraná - Brasil
Emails: raiannyproenca1@gmail.com, crmmauricio@gmail.com, fabiana.peres@unioeste.br
†Fundação Parque Tecnológico Itaipu
Foz do Iguaçu, Paraná, Brasil
Email: valeria.valsnfw@gmail.com

***Resumo*—Fissuras em concreto representam manifestações patológicas e ocorrem por diversos motivos, mesmo que haja boas práticas na fase de construção. Em estruturas de grande porte, como pontes, túneis e barragens é exigido que, com certa periodicidade, ocorra inspeções visuais com objetivo de detectar, diagnosticar a causa e quando possível, reparar a fissura. Nos casos que não é possível reparar a fissura, se deve acompanhar o seu comportamento. Muitas técnicas computacionais para a detecção de fissuras têm sido propostas mas suas aplicações são limitadas pois as imagens de fissuras tendem a variar muito e neste caso, extrair informações como a localização da fissura em uma imagem requer que seja realizada uma segmentação a nível de pixel. Neste contexto, esse trabalho apresenta uma proposta utilizando o Detectron2, inspirado na rede neural convolucional Mask R-CNN, que oferece suporte para detecção de objetos, segmentação de instâncias, segmentação de panorâmica, e segmentação de semântica.**

***Abstract*—Cracks in concrete represent pathological manifestations and occur for various reasons, even if there are good practices in the construction phase. In large structures, such as bridges, tunnels and dams, it is required that, at certain intervals, visual inspections are carried out with the objective of detecting, diagnosing the cause and, when possible, repairing the crack. In cases where it is not possible to repair the crack, its behavior must be monitored. Many computational techniques for crack detection have been proposed but their applications are limited because crack images tend to vary a lot and in this case, extracting information such as crack location in an image requires segmentation at the pixel level. In this context, this work presents a proposal using Detectron2, inspired by the Mask R-CNN convolutional neural network, which offers support for object detection, instance segmentation, panoramic segmentation, and semantic segmentation.**

keywords - reconhecimento visual, Detectron2, Mask R-CNN

I. Introdução

Estudos apontam fissuras em obras de concreto como patologias. Demonstram sinais de desgaste estrutural [1] e caracterizam-se como aberturas com direções e formatos variados, colocando em risco a estrutura. Sua ocorrência é muito comum e muitas vezes passam despercebidas, principalmente em obras de grande porte como pontes, túneis e barragens. As fissuras podem ocorrer por diversos motivos e seus aspectos auxiliam a identificar suas causas. Muitas vezes representam o problema total ou pode significar um problema maior [2]. A detecção e o diagnóstico adequado da fissura é de suma importância. A realização de reparos ou o acompanhamento do seu compartamento, nos casos onde o reparo não é possível, minimizam danos ou resolvem o problema como um todo. Caso contrário, a fissura poderá ser fonte de outras complicações que contribuirão com a deterioração da estrutura [2].

(a) Falta de sombreamento (b) Textura de fundo (c) Ruídos

Figure 1. Exemplos dos desafios encontrados em imagens com fissuras ou trincas.

Para tanto, é necessário que sejam realizados monitoramentos e inspeções regularmente para manutenção e prevenção de falhas. Geralmente, essas inspeções são realizadas manualmente por um especialista ou técnico experiente, no entanto essa é uma tarefa custosa e trabalhosa [1]. Para auxiliar nessa tarefa, muitas técnicas de detecção de fissuras têm sido propostas, mas suas aplicações ainda são bem limitadas, pois as imagens capturadas das estruturas que apresentam fissuras tendem a variar muito [3] e geralmente são afetadas por diversos fatores como ruídos, irregularidades da superfície, sombreamento, iluminação, textura de fundo, objetos entre outros [1] [4], como apresenta a Figura 1.

A localização, a abertura e o comprimento das fissuras são informações de alto nível que requerem uma segmentação a nível de pixel, isto é, segmentação semântica [3]. A segmentação semântica detecta todos os objetos presentes em uma imagem no nível de pixel, e então agrupa os pixels de forma semântica, produzindo regiões com diferentes classes ou

objetos [5]. É possível extrair informações com a segmentação semântica através do uso de redes neurais convolucionais (CNN). As redes neurais convolucionais encontram-se dentro do universo *Deep Learning* e foram desenvolvidas para classificar imagens pois são capazes de analisar constantemente os dados e reconhecer padrões. O *Deep Learning*, é uma subárea da inteligência artificial que além de fazer com que a máquina seja capaz de aprender por meio de algoritmos e dados fornecidos previamente e se torne apta a tomar decisões sozinha, é capaz de desenvolver funções mais complexas como reconhecimento visual e de fala [6].

Este trabalho utilizou a Detectron2 [7], que é uma rede neural convolucional que possui um ramo adicional para predições de máscaras de segmentação e utiliza uma abordagem de segmentação semântica, oferecendo suporte para extrair as informações de alto nível necessárias.

II. Referencial Teórico

A. Fissuras

A consequência da instabilidade do concreto relacionado às interações de seus elementos constituintes com agentes externos podem prejudicar a estrutura, causando deformidades e deterioração [8], [9]. Uma vez que a estrutura esteja comprometida devido às irregularidades, tais como fissuras e trincas, essas são vistas como patologias e são tratadas como tais. Portanto, é necessário realizar um diagnóstico adequado e compreender as características das fissuras, tais como causas e origens [8] para que haja o reparo correto. Os principais processos que contribuem com a deterioração do concreto estão relacionados às propriedades físico-químicas do concreto e sua exposição. As fissuras são um dos principais resultados da deterioração. Uma de suas causas são as variações abruptas de temperaturas que contribuem com contração e/ou expansão do concreto, que caso seja restrito e a tensão de tração seja maior que a resistência do concreto, pode acarretar em fissuras [8].

O fenômeno da retração pode ocorrer no concreto em duas situações, no estado plástico e endurecido [8], denominados respectivamente como retração plástica que ocorre quando o concreto está sujeito a uma redução considerável de água provocada pela umidade relativa do ar e o vento na superfície, já as fissuras no concreto endurecido são provocadas por retração por secagem, tensões térmicas, reação química, intemperismo, entre outros [2]. Para manutenção e prevenção de falhas, são realizados regularmente monitoramento e inspeções das fissuras. No entanto, esta é uma tarefa muito custosa e limitada à experiência do especialista [1]. Tornar o processo de detecção de fissuras automático é uma tarefa desafiadora. Inicia-se por entender quais técnicas e softwares são adequados de utilizar para o tratamento das imagens e qual será o método da extração das informações importantes como localização e larguras das fissuras, pois as imagens das estruturas de concreto com geralmente variam muito e são afetadas por vários fatores, tais como irregularidades, sombreamento, iluminação, ruídos, textura de fundo, vegetação, entre outros que podem ser confundidos com fissuras [1], [4].

B. Redes Neurais Convolucionais

As redes neurais convolucionais (*Convolutional Neural Network/ CNN/ ConvNet*) foram inspiradas no funcionamento do córtex visual para classificação de imagem [10]. A CNN pode receber uma imagem de entrada e atribuir filtros, que podem ser aprendidos, aos objetos presentes na imagem e ser capaz de diferenciar um do outro [11]. O pré-processamento exigido pela CNN é muito menor em comparação aos outros algoritmos de classificação, tornando o processamento de imagens computacionalmente gerenciável, pois ao receber as imagens, a rede as reduz, sem perder informações que são críticas para uma boa previsão. Isso ocorre porque o número de parâmetros envolvidos é menor e há a reutilização de pesos, fazendo com que a arquitetura execute um ajuste melhor ao conjunto de dados da imagem [11].

As redes neurais convolucionais baseadas em região (R-CNN), inicialmente apresentada em [12], é uma abordagem para detecção de objeto que divide a imagem em 2000 regiões propostas, então a CNN é aplicada e avaliada em cada uma dessas regiões de forma independente. A Fast R-CNN, é uma outra abordagem proposta que em vez de aplicar a CNN em cada região, aplica apenas uma vez por imagem e um mapa de recursos é gerado [13]. Um outro algoritmo proposto para detecção de objetos foi o Faster R-CNN. Ele é composto por dois módulos: um que utilizando uma Rede Totalmente Convolucional (*Fully Convolutional Network*/FCN) para modelar uma *Region Proposal Network* (RPN), é capaz de receber uma imagem de qualquer tamanho como entrada e gerar um conjunto de detecção recomendadas, ou seja, esse primeiro módulo propõe regiões e o segundo módulo corresponde ao detector da Fast R-CNN utilizar as regiões propostas [14].

C. Mask R-CNN

Mask R-CNN é uma rede neural convolucional extensão da Faster R-CNN, mas com um ramo adicional para predição de máscaras de segmentação para cada região de interesse [15]. Ela utiliza uma abordagem de Segmentação de Instância (*Instance Segmentation*), que integra a tarefa de detecção de objetos com a tarefa de Segmentação Semântica. Admite-se que seu processo é dividido em duas fases: uma para detecção de objetos, usando uma arquitetura semelhante ao Faster R-CNN [14] e outra para a Segmentação Semântica, utilizando-se FCN (*Fully Convolution Network*).

A Segmentação Semântica detecta todos os objetos presentes em uma imagem a nível de pixel; agrupa os pixels de forma semântica produzindo regiões com diferentes classes ou objetos. Já a segmentação de instância identifica cada instância de objeto para cada objeto conhecido em uma imagem atribuindo um rótulo para cada pixel da imagem. A segmentação de instância requer a detecção de todos os objetos presentes na imagem, e então classifica os objetos individuais e localiza cada instância de objeto utilizando *bounding box*, depois segmenta cada uma dessas instância a fim classificar cada pixel em um conjunto fixo de categorias sem distinguir instâncias de objetos [5].

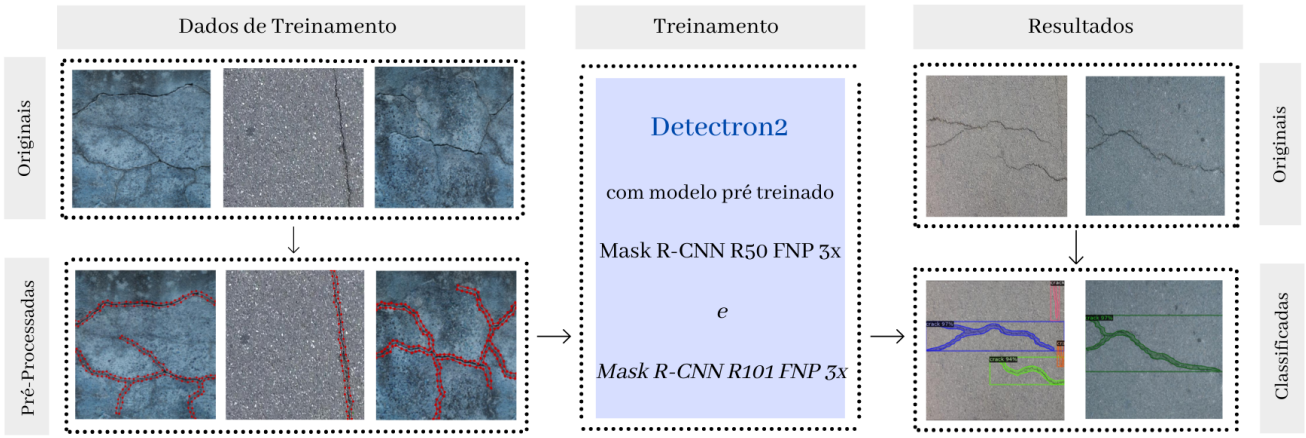


Figure 2. Fluxograma do funcionamento do trabalho

Para calcular as Regiões de Interesse (RoI - *Region of Interest*) calcula-se a Interseção sobre a União (IoU/ *Intersection over Union*), com as caixas *Ground-truth*, para todas as regiões prevista [16]. O IoU é a intersecção divida pela divisão, ou seja, o numerador apresenta a sobreposição entre a caixa delimitadora verdadeira e a prevista; Já o denominador é a união que abrange toda a área das duas caixas delimitadoras.

A região de interesse é considerada positiva se tiver IoU maior ou igual 0.5 e negativa caso contrário. O ROI é o conjunto de regiões no qual o IoU foi maior que 0.5 [15]. Com esse procedimento realizado, adiciona-se o ramo para predição de máscara [16]. Uma máscara é prevista para cada RoI utilizando uma FCN. A máscara codifica o *layout* espacial de um objeto de entrada, e a extração da estrutura espacial das máscaras pode ser tratada pela correspondência pixel a pixel fornecida pelas camadas convolucionais. Onde são previstas máscaras para todos os objetos na imagem.

III. Materiais e Métodos

A. Dataset

O dataset utilizado contém 2000 imagens [17]–[25]. No qual foram separadas 1500 para compor o conjunto de treinamento e validação e 500 para o conjunto de teste. Todas as imagens possuem tamanho de 448x448.

B. Labelme

Todas as imagens do conjunto de treino foram rotuladas manualmente através da ferramenta *LabelMe*. Após serem rotuladas, cada imagem gerou um arquivo json, que contém as coordenadas de cada polígono (ponto vermelho) e características das imagens, que será posteriormente utilizado como entrada da rede Detectron2. Um exemplo das imagens rotuladas pode ser visto na Figura **??**.

C. Detectron2

É um sistema de software desenvolvido por Facebook AI Research que implementa algoritmos de detecção de objetos. O Detectron2 se originou do benchmark Mask R-CNN [15] e oferece suporte para detecção de objetos, segmentação de instâncias, segmentação de panorâmica, e segmentação de semântica [7].

Figure 3. Exemplo de imagens no Conjunto de teste.

Figure 4. Exemplo de imagens no Conjunto de treino

D. Metodologia

A metodologia utilizada para a elaboração deste artigo foi composta por 5 principais etapas em que a primeira e a segunda abordam a fase de pré-processamento, a terceira

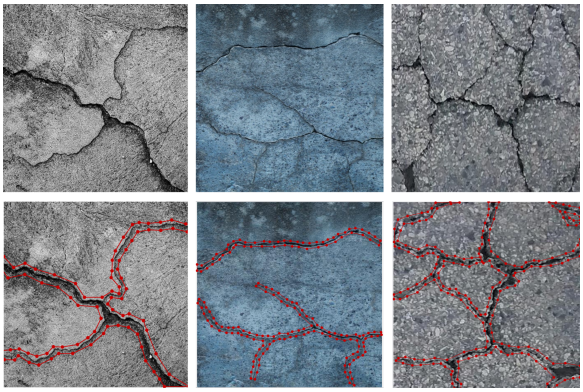


Figure 5. Exemplo da rotulação com a ferramenta LabelMe.

refere-se a fase de treinamento, a quarta e a quinta são as fases de teste e de avaliação respectivamente.

Na etapa 1, houve a seleção e tratamento de imagens, Foram selecionadas e tratadas 1500 imagens do *dataset* definido para treino, e seleção e tratamento de mais 500 imagens do *dataset* definido para teste. O tratamento incluiu rotação, organização e padronização nos nomes das imagens. Foi priorizada a seleção de imagens mais nítidas, distintas e que não houvessem marcas d'água. Na etapa 2, houve a rotulação das imagens do conjunto de treino.

A etapa 3 é a fase de treinamento. Com o objetivo de verificar a influência do número de camadas convolucionais e o impacto do tamanho do conjunto de treino nos resultados finais. A fase de treinamento foi dividida em duas fases: Na primeira fase, houve o treinamento com o Mask R-CNN utilizando como *backbone* ResNet-50 (50 camadas) e na segunda fase de treinamento utilizou-se o ResNet-101 (101 camadas). Para facilitar o entendimento, cada uma dessas fases podem ainda ser divididas em 3 sub etapas, no qual aqui será referenciado como T1, T2 e T3 para treinamento 1, treinamento 2 e treinamento 3 respectivamente.

Isto é, no treinamento 1 (T1), a CNN foi treinada com 500 imagens. No segundo treinamento (T2), utilizaram-se 1000 imagens, sendo 500 imagens do T1 e outras 500 novas imagens. No treinamento 3 (T3) seguiu-se essa lógica, utilizando as 1000 imagens do treinamento anterior com mais 500 novas imagens, totalizando 1500 imagens no conjunto de treino. Na etapa 4 houve a fase de teste, onde foram utilizadas as imagens do conjunto de teste sobre a CNN treinada. E por fim, a etapa 5, que consistiu na rotulação das imagens do conjunto de teste para serem comparadas com o resultado devolvido pelas CNN. A partir dessa fase que é obtém-se as métricas. As métricas utilizadas neste artigo são as métricas padrões COCO [26].

IV. Resultados

Inicialmente, a CNN foi treinada com a ResNet-50 FPN, variando o número de épocas (32 e 64) para cada um dos treinos (T1, T2 e T3), esse processo será denominado como primeira fase do treinamento. Posteriormente, o mesmo procedimento foi realizado com a rede configurada com o modelo pré-treinado ResNet-101 FPN e será referenciado como segunda fase do treinamento.

Os resultados obtidos na primeira fase de treinamento estão expostos nas Tabelas I, II, III e IV. As Tabelas I e II apresentam as precisões média AP (*Average Precision*) da classificação do *Bounding Box* nas imagens.

Table I
Resultados do AP do Bounding Box utilizando Mask R-CNN ResNet-50 FPN com 32 épocas.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	32	38.75	59.84	42.60
T2	1000	32	53.65	74.75	60.31
T3	1500	32	56.28	77.10	61.80

Observa-se que, com 32 épocas, o AP melhorou subindo de 38.75 obtido no T1 para 56.28 no T3, assim também o AP 50 de 59.84 no T1 para 77.10 no T3. Estes resultados mostram que houve uma melhora na precisão, consequente ao aumento do número de imagens no conjunto de treino.

Table II
Resultados do AP do Bounding Box utilizando Mask R-CNN ResNet-50 FPN com 64 épocas.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	64	40.29	58.75	43.58
T2	1000	64	47.82	67.82	52.20
T3	1500	64	56.43	76.24	61.55

O mesmo pode ser observado na Tabela II, analisando os resultados obtidos do treinamento com 64 épocas, onde o AP aumentou de 40.29 no T1 para 56.43 no T3 e o AP50 que foi 58.75 em T1 para 76.24 em T3. As Tabelas III e IV por sua vez, apresentam os resultados da precisão média de acerto da máscara de segmentação.

Table III
Resultados do AP para Máscara de Segmentação utilizando Mask R-CNN ResNet-50 FPN com 32 épocas.

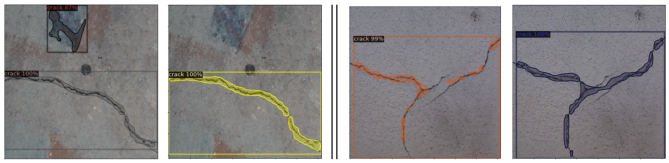
Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	32	06.58	27.90	00.45
T2	1000	32	11.42	42.37	00.72
T3	1500	32	17.88	54.23	04.94

Demonstrando para o AP75 do treinamento com 32 épocas, valores significantemente baixos, o mesmo pode ser observado em AP. Já em AP50 os resultados foram melhores, principalmente em T3, passando de 50.

A Figura 6 apresenta as classificações previstas pela rede neural treinada, resultantes da primeira fase do treinamento, comparando os resultados de T1 e T3. Observa-se em 6a), que inicialmente a CNN reconheceu um ruído como fissura, porém já melhorou o reconhecimento em 6b), identificando a fissura com 100% de precisão. Também houveram melhorias de 6c) T1 para 6d) T3, mesmo que ainda alguns pontos das fissuras continuaram sem ser reconhecidos

Table IV
RESULTADOS DO AP PARA MÁSCARA DE SEGMENTAÇÃO UTILIZANDO MASK R-CNN RESNET-50 FPN COM 64 ÉPOCAS.

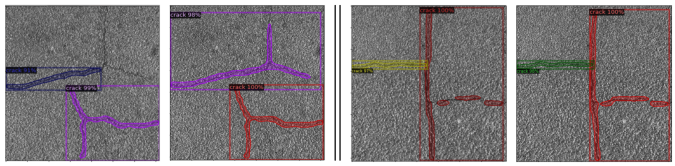
Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	64	06.4	27.34	11.41
T2	1000	64	11.41	42.60	11.55
T3	1500	64	16.52	56.86	01.22



a) T1 b) T3 c) T1 d) T3

Figure 6. Resultado do treinamento Mask ResNet-50 FPN com 32 épocas onde T1 possui 500 imagens e T3 1500 imagens.

A Figura 7 apresenta os resultados obtidos após a primeira fase do treinamento com 64 épocas. Algumas regiões continuaram sem ser detectadas em 7 a) e b), mas ainda assim houve uma melhora no T3, pois inicialmente a CNN só estava reconhecendo 91% (rótulo azul), já no último treino, ele reconheceu 98%.



a) T1 b) T3 c) T1 d) T3

Figure 7. Resultado do treinamento Mask ResNet-50 FPN com 64 épocas onde T1 possui 500 imagens e T3 1500 imagens.

Na segunda fase do treinamento, a configuração do *backbone* do Mask R-CNN foi alterada para ResNet-101, com 101 camadas. Os resultados, seguindo os mesmos critérios que a primeira fase, podem ser observados nas Tabelas V, VI, VII e VIII. Resultados positivos podem ser vistos logo nos primeiros treinamentos, nas Tabelas V e VI, onde com 32 épocas o AP50 é 74.14 e com 64 épocas o AP50 é de 83.15. Observa-se que o AP50 em T1 com 500 imagens e 64 épocas, obteve o maior valor considerando o *bounding box*, mas comparado com AP50 de T2 com 32 épocas, tem apenas uma pequena diferença.

Table V
RESULTADOS DO AP DO BOUNDING BOX UTILIZANDO MASK R-CNN RESNET-101 FPN COM 32 ÉPOCAS.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	32	51.98	74.14	56.50
T2	1000	32	62.74	82.86	69.72
T3	1500	32	62.53	80.02	67.63

Já nas Tabelas VII e VIII, demonstra-se os resultados obtidos da precisão média sobre a máscara de segmentação.

Table VI
RESULTADOS DO AP DO BOUNDING BOX UTILIZANDO MASK R-CNN RESNET-101 FPN COM 64 ÉPOCAS.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	64	63.65	83.15	68.96
T2	1000	64	61.17	80.14	65.72
T3	1500	64	64.52	81.63	70.16

O melhor resultado foi obtido utilizando 1500 imagens no conjunto de treino com 32 épocas, sendo o AP50 no valor de 63.31.

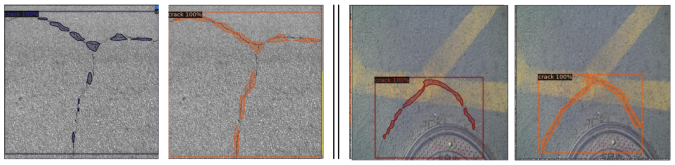
Table VII
RESULTADOS DO AP PARA MÁSCARA DE SEGMENTAÇÃO UTILIZANDO MASK R-CNN RESNET-101 FPN COM 32 ÉPOCAS.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	32	05.21	23.38	00.12
T2	1000	32	12.01	44.09	00.80
T3	1500	32	21.93	63.31	08.93

Table VIII
RESULTADOS DO AP PARA MÁSCARA DE SEGMENTAÇÃO UTILIZANDO MASK R-CNN RESNET-101 FPN COM 64 ÉPOCAS.

Treino	Imagens	Épocas	AP	AP50	AP75
T1	500	64	13.72	49.09	1.54
T2	1000	64	10.47	41.16	00.61
T3	1500	64	19.06	61.51	4.43

Observa-se na Figura 8a) que a CNN reconheceu a fissura de forma mais fragmentada do que em 8b), pode-se notar que este reconhecimento foi otimizado em T3, pois a rede delineou a fissura quase que integralmente. Já em 8c) e d) observa-se a presença de ruídos, como a tampa do bueiro e faixa amarela, que poderiam ser confundidos como fissuras. A CNN se demonstrou capaz em fazer a distinção e demarcar somente o alvo de interesse.

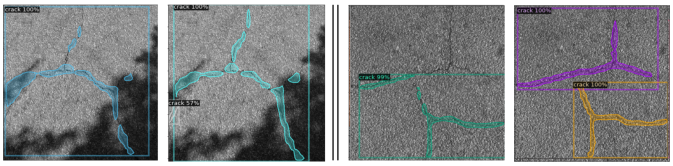


a) T1 b) T3 c) T1 d) T3

Figure 8. Resultado do treinamento Mask ResNet-101 FPN com 32 épocas onde T1 possui 500 imagens e T3 1500 imagens.

Na Figura 9, comparando a) e b), nota-se que a rede neural pôde ignorar o sombreamento, visto como ruído, melhorando a máscara de segmentação de T1 para o T3. Em 9c) e d) observa-se novamente a melhora no reconhecimento da CNN, que foi capaz de identificar fissuras que inicialmente não haviam sido.

A avaliação da rede é feita através do IoU, que utiliza a área verdadeira do objeto na imagem marcada manualmente e a



a) T1 b) T3 c) T1 d) T3

Figure 9. Resultado do treinamento Mask ResNet-101 FPN com 64 épocas onde T1 possui 500 imagens e T3 1500 imagens.

área prevista pela CNN, isto é, o que a rede deveria reconhecer e o que foi reconhecido de fato. A máscara de segmentação apresentou valores significantemente menores se comparado ao *bounding box*, devido a sua grande margem de erro, já que depende diretamente da marcação manual.

V. Conclusão

Através dos estudos e testes realizados, observou-se que as redes neurais convolucionais são de fatos os algoritmos mais adequados para essa tarefa de detecção. Principalmente o Mask R-CNN, que através da segmentação de instância devolve não só uma caixa delimitadora com grau de confiança do reconhecimento, como também uma máscara de segmentação que destaca o alvo de interesse, facilitando assim sua identificação. A aplicação desta CNN foi facilitada, graças ao Detectron2, assim como sua descrição previa.

Observou-se também, que o aumento do número de imagens do conjunto de treino bem como do número de camadas convolucionais influencia nas classificações finais de forma positiva, otimizando o resultado apresentado pela CNN na detecção de fissuras. Já o número de épocas variável trouxe melhoras, quando maior, mas que podem ser consideradas insignificantes, em especial diante de todo o custo computacional adicional agregado ao treinamento completo da rede neural convolucional. Como o Detectron2 fornece suporte para mais modelos pré-treinados e outros tipos de segmentação, espera-se continuar este trabalho testando cada um destes, a fim de alcançar a automação de detecção de fissuras de forma plena e confiável.

Referências

[1] L. Ali, F. Alnajjar, H. A. Jassmi, M. Gocho, W. Khan, and M. A. Serhani, "Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures," *Sensors*, vol. 21, no. 5, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1688

[2] M. N. Abou-Zeid, F. Fouad, R. Leistikow, R. Poston, J. Barlow, D. Fowler, P. A. Lipphardt, R. J. Rhoads, F. Barth, G. T. Halvorsen, E. Nawy, J. W. Roberts, J. Best, W. Hansen, K. Nemati, A. Scanlon, D. Darwin, H. Haynes, K. A. Pashina, A. Schokker, J. F. Duntemann, R. Frosch, and J. West, "Causes, evaluation, and repair of cracks in concrete structures," 2007.

[3] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using u-net fully convolutional networks," *Automation in Construction*, vol. 104, pp. 129–139, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519301244

[4] Ç. F. Özgenel and A. G. Sorguç, "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 35. IAARC Publications, 2018, pp. 1–8.

[5] R. Khandelwal, "Computer vision: Instance segmentation with mask r-cnn," Disponível em: https://towardsdatascience.com/computer-vision-instance-segmentation-with-mask-r-cnn-7983502fcad1. Acesso em: 06 março 2020, 2019.

[6] S. S. Damaceno and R. O. Vasconcelos, "Inteligência artificial: uma breve abordagem sobre seu conceito real e o conhecimento popular," *Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-SERGIPE*, vol. 5, no. 1, p. 11, 2018.

[7] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[8] J. S. Lapa, "Patologia, recuperação e reparo das estruturas de concreto," *Monografia, Especialização em Construção Civil–Universidade Federal de Minas Gerais, Belo Horizonte*, 2008.

[9] F. F. F. Peres, "Estrutura conceitual para integração da auscultação de fissuras em barragens de concreto com realidade aumentada," 2017.

[10] J. D. P. Massucatto, "Aplicação de conceitos de redes neurais convolucionais na classificação de imagens de folhas," B.S. thesis, Universidade Tecnológica Federal do Paraná, 2018.

[11] D. S. Academy, "Deep learning book," 2019.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[13] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

[14] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[15] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[16] P. Sharma, "Computer vision tutorial: Implementing mask r-cnn for image segmentation," Disponível em: https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/. Acesso em: 19 julho 2021, 2019.

[17] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3708–3712.

[18] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.

[19] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection."

[20] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, "How to get pavement distress detection ready for deep learning? a systematic approach." in *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2039–2047.

[21] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *arXiv preprint arXiv:1901.06340*, 2019.

[22] L. Cui, Z. Qi, Z. Chen, F. Meng, and Y. Shi, "Pavement distress detection using random decision forests," in *International Conference on Data Science*. Springer, 2015, pp. 95–102.

[23] L. Yang, B. Li, W. Li, Z. Liu, G. Yang, and J. Xiao, "Deep concrete inspection using unmanned aerial vehicle towards cssc database," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 24–28.

[24] 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.

[25] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "Cracktree: Automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[26] COCO, "Detection evaluation and metric," 2020. [Online]. Available: https://cocodataset.org/#detection-eval