

DETECÇÃO E QUANTIFICAÇÃO AUTOMATIZADAS DE TRINCAS EM PAVIMENTOS DE RODOVIAS

Jorge Destri Junior

Laboratório de Transportes e Logística
Universidade Federal de Santa Catarina

Nicolas Pereira Borges

Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica

Nataniel Pereira Borges Junior

Centro de Segurança da Informação CISPA Helmholtz, Alemanha

Strauss Carvalho Cunha

Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica

Amir Mattar Valente

Departamento de Engenharia Civil
Universidade Federal de Santa Catarina

RESUMO

Um dos defeitos mais comuns no pavimento das rodovias são as trincas. Sua presença é um grande indicador de necessidade de ações de restauração. Atualmente, o DNIT utiliza vídeos-registros e processamento manual e automatizado para detectar a condição do pavimento e de outros elementos das rodovias. Porém, localizar e quantificar a presença de trincas ainda é um processo manual, que é lento, custoso e tendencioso. Nesse contexto, este trabalho propõe um método automatizado para detectar trincas e estimar o seu percentual, por quilômetro, em vídeos-registros, com o auxílio de técnicas de processamento de imagem e de aprendizado de máquina. Foram combinadas duas abordagens, uma que visa descartar regiões cuja probabilidade de ter trincas é baixa, e outra que se certifica de que as demais regiões têm trincas ou não. O modelo proposto foi validado em um conjunto de 138.909 imagens, obtendo-se acurácia de 0,90, sensibilidade de 0,89 e precisão de 0,75.

ABSTRACT

One of the most common defects in road pavements is cracks and its presence is an excellent indicator that restoration actions are necessary. Currently, DNIT uses semi-automated video-records processing techniques to aid the extraction of some elements of federal highways, such as traffic signs and road geometry. Cracks are, however, manually detected being a laborious, biased, and time-consuming task. In this context, this work proposes a combination of image processing and machine learning techniques to automatically detect cracks and to estimate their percentage in video segments. We use image processing to filter out regions with a low probability of having cracks and machine learning to determine if the regions have cracks or not. Our experiments with 138,909 images obtained a 0.75 precision, 0.90 accuracy and 0.89 sensitivity.

1. INTRODUÇÃO

O Brasil tem cerca de 74.000 km de rodovias federais pavimentadas (Brasil, 2019), e, desse total, cerca de 80% estão sob jurisdição do Departamento Nacional de Infraestrutura de Transportes (DNIT). Além disso, a distribuição modal do transporte de cargas é altamente distorcida, onde o modo rodoviário responde por, aproximadamente, 61% da movimentação total (CNT, 2019).

Essa alta concentração modal aumenta a importância de se ter o pavimento em bom estado de conservação. O pavimento em mau estado afeta o custo logístico dos transportadores, pois, dentre outros fatores, aumenta o tempo de viagem e o custo de manutenção dos veículos. Em relação à segurança viária, paradoxalmente, o pavimento ruim diminui a quantidade e a gravidade dos acidentes, pois, nos trechos em que ele ocorre, a velocidade dos veículos tende a ser bem menor (CNT, 2018).

A manutenção do pavimento, para deixá-lo em bom estado, tem um custo estimado de 80 a 150 mil para estratégias de conservação, e cerca de 2 milhões se for necessária uma restauração (Brasil, 2018).

Considerando a extensão da malha rodoviária sob sua responsabilidade e os recursos financeiros envolvidos, o DNIT se utiliza de várias metodologias para avaliação dos pavimentos, dentre elas: o Índice de Gravidade Geral (IGG); o Índice Gravidade Individual (IGI); o Índice de Condição de Pavimentos Flexíveis (ICPF); o Índice Internacional de Irregularidade, *International Roughness Index* (IRI); e o Índice de Condição da Manutenção (ICM).

O ICM prioriza os trechos que necessitam de algum tipo de intervenção de manutenção. Com ele, são avaliados três parâmetros associados à conservação da via: roçada, sinalização e os elementos de drenagem; e três elementos associados ao pavimento: o número de painéis, o número de remendos e a área com trincamento. O levantamento desses parâmetros é realizado a cada quilômetro, com um veículo percorrendo a via e com um técnico registrando os dados através de um aplicativo específico do próprio órgão (Brasil, 2017). Essa forma de obtenção de dados apresenta alguns problemas, dentre eles: a necessidade de treinamento para os técnicos que vão a campo; a subjetividade de cada técnico em determinar o grau de conservação dos elementos ou da área com trincamento; e, devido à velocidade desenvolvida pelo veículo, a necessidade de observar e de registrar de forma adequada todos os parâmetros solicitados.

Atualmente, já existem tecnologias e procedimentos que permitem detectar, classificar e avaliar, de forma automatizada ou semiautomatizada, elementos viários a partir de imagens georreferenciadas, extraídas de vídeos. Um exemplo disso é o *software* Restituídor *as Built*, que permite extrair informações sobre a presença de alguns tipos de sinalização horizontal e vertical (advertência e regulamentação), presença ou não de acostamento e largura de faixa (UFSC, 2012). Esse *software* tem como dado de entrada os vídeo-registros fornecidos pelo DNIT, porém, não contempla identificação de defeitos no pavimento.

Considerando a importância da presença de trincas no processo de gestão do pavimento, este trabalho propõe apresentar um procedimento de detecção automatizado de trincas no pavimento, de modo que possa ser utilizado no processo do cálculo do ICM, assim como para outras aplicações, como o Restituídor *as Built*.

Inicialmente, serão apresentados alguns conceitos sobre trincas. Em seguida, será descrito o método proposto e como a literatura tem abordado essa problemática. Feito isso, o texto se dirigirá à análise dos resultados da aplicação do método para alguns vídeos coletados em rodovias estaduais e federais do estado de Santa Catarina.

2. TRINCAS

Segundo o DNIT (2003), uma fenda é qualquer descontinuidade na superfície do pavimento, que conduza a aberturas de menor ou de maior porte, apresentando-se sob diversas formas. Dentre elas, há a fissura, que é de largura capilar, posicionada longitudinal, transversal ou obliquamente ao eixo da via e é somente perceptível, à vista desarmada, de uma distância inferior a 1,50 m. Elas não causam problemas funcionais ao revestimento, não sendo, assim,

consideradas quanto à gravidade nos métodos atuais de avaliação das condições de superfície. Outra forma é a trinca, que tem como características ser facilmente visível à vista desarmada e ter abertura superior à da fissura. Apresenta-se sob as formas isolada (transversal, longitudinal ou de retração) ou trinca interligada (“couro de jacaré”, “bloco”), conforme exemplos exibidos na Figura 1.



Figura 1: Exemplos de trincas

As trincas podem surgir por diversas causas, dentre elas: fadiga; envelhecimento; reflexão; temperatura; umidade; recalque de fundação; e ruptura de aterros.

O DNIT (2005) aponta que as trincas são um defeito na superfície que enfraquece o revestimento, permitindo a entrada da água, o que provoca um enfraquecimento adicional da estrutura. Uma vez iniciado, o trincamento tende a aumentar sua extensão e sua severidade conduzindo, eventualmente, à desintegração do revestimento, o que, normalmente, leva à formação de panelas, defeito este mais crítico para a segurança viária. Por isso, ao longo do tempo, a presença de trincas tem sido o critério importante para a deflagração de intervenções de restauração de pavimentos.

3. DETECÇÃO AUTOMATIZADA DE TRINCAS

O modelo proposto neste trabalho recebe como entrada um vídeo-registro e um arquivo contendo o georreferenciamento dos *frames* do vídeo, conforme o seu *Frames Per Second* (FPS). O objetivo é calcular a média de trincas por quilômetro. Nesse contexto, a Figura 2 apresenta um diagrama com a visão geral do sistema. Cada um dos blocos será detalhado na sequência.

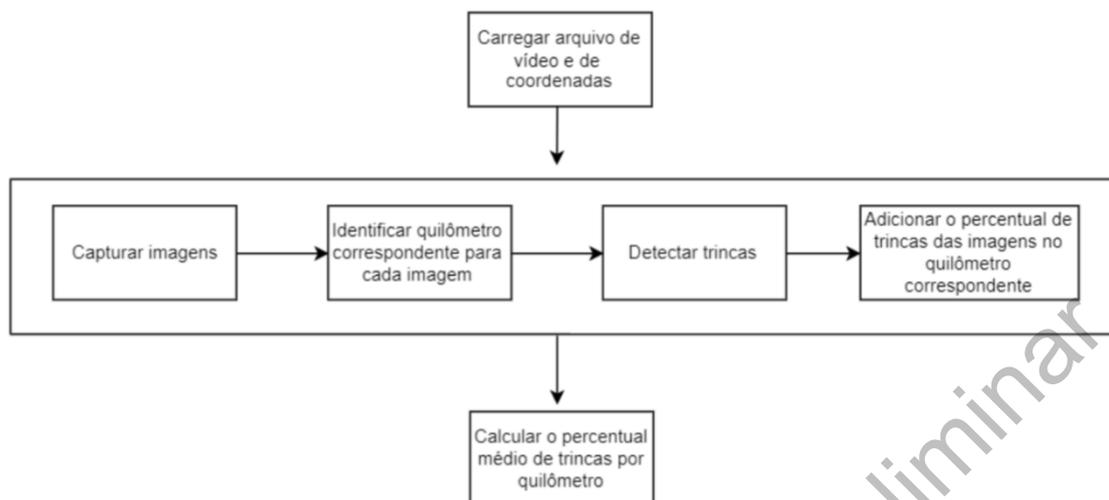


Figura 2: Visão geral do modelo

Abaixo, estão descritas as características dos blocos.

- **Carregar arquivo de vídeo e o arquivo de coordenadas:** consiste em abrir o arquivo de vídeo e o arquivo de coordenadas.
- **Capturar imagens:** consiste em extrair os *frames* do vídeo, um de cada vez.
- **Identificar quilômetro correspondente a cada imagem:** cada imagem é georreferenciada com suas latitude e longitude. Sendo assim, dado o índice de uma imagem, é possível verificar a distância dela para com a coordenada do primeiro *frame*. De posse dessa distância, verifica-se a quilometragem a que a imagem pertence. É importante ressaltar que o modelo atual assume que o vídeo sempre irá começar a ser processado pelo quilômetro 1.
- **Detectar trincas:** esta etapa visa calcular o percentual de trincas identificadas na imagem. O detalhamento desse processo se dará na sequência.
- **Adicionar o percentual de trincas das imagens no quilômetro correspondente:** uma vez identificado o quilômetro correspondente a uma determinada imagem e calculado seu percentual de trincas, esse percentual é, então, inserido na lista do quilômetro correspondente.
- **Calcular o percentual médio de trincas por quilômetro:** após o processamento de todos os *frames* de um quilômetro, calcula-se a média aritmética dos percentuais de trincas correspondentes. Como resultado, tem-se o percentual médio de trincas no quilômetro. Esse cálculo, aliás, é feito para todos os quilômetros do vídeo.

3.1 Método para detectar trincas

Uma abordagem comum para identificação de objetos em imagem são modelos baseados em análise de textura. Para o âmbito de trincas, um trabalho que se destaca é o de Chatterjee e Yi-Chang (2018), que descreve um modelo para detecção de trincas baseado em segmentação da imagem. O algoritmo consiste nas seguintes etapas: pré-processamento de imagem; segmentação da imagem para minimizar falsos negativos; geração e conexão das possíveis áreas de trincas identificadas, de forma que sejam removidos falsos positivos; e refinamento da segmentação por meio de um procedimento baseado em busca de caminho mínimo. Uma deficiência dessa abordagem, contudo, é que ela apresenta uma baixa capacidade de detecção

de trincas quando a imagem possui elementos externos ao pavimento, como, por exemplo, alta incidência de sujeira, de veículos, etc. Além disso, em coletas de vídeo, a imagem tende a ficar menos nítida, uma vez que o veículo se encontra em deslocamento.

Outra abordagem comum para identificação de objetos em imagem é através de aprendizado de máquina. Nesse cenário, Cha, Choi e Büyükoztürk (2017) apresentaram uma solução baseada em uma arquitetura de redes neurais convolucionais, *Convolutional Neural Network* (CNNs), para detecção de trincas. O mais interessante dessa abordagem é que CNNs são capazes de aprender automaticamente as características da imagem, logo, o método proposto funciona sem a necessidade de serem aplicados complexos filtros de processamento de imagem, de forma a se obterem informações de textura, iluminação, dentre outras. Por essa razão, as CNNs têm sido amplamente utilizadas na área de Processamento de Imagem Digital, em diversas áreas, como, por exemplo, reconhecimento facial (Bhosle et.al, 2018), área médica para identificação de alguns tipos de câncer (Raghavendra et.al, 2018; Rakhlin et.al, 2018).

Diante da deficiência do método de Chatterjee e Yi-Chang (2018), o presente trabalho o combinou com a ideia de verificar trincas por intermédio de CNNs, descrita (Cha, Choi e Büyükoztürk, 2017). A Figura 3 mostra um diagrama com as etapas para detecção e cálculo do percentual de trincas. Na sequência serão detalhadas estas etapas.

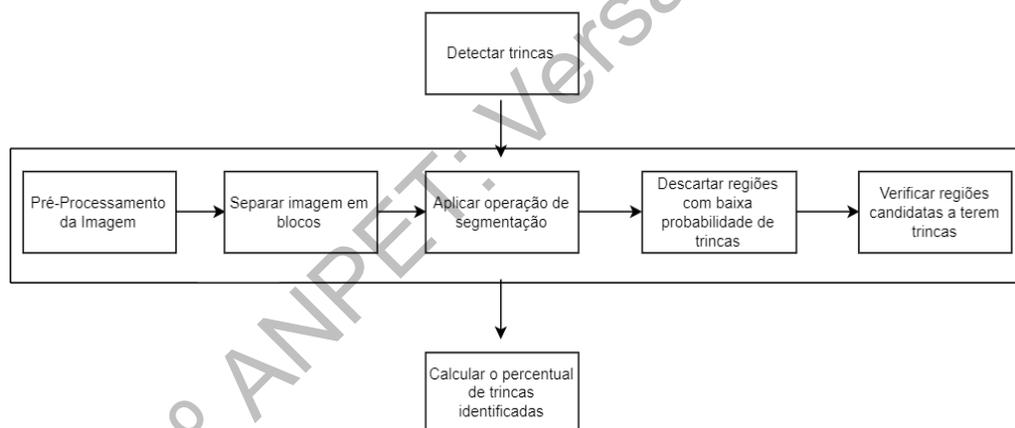


Figura 3: Modelo para detecção de trincas

3.1.1 Pré-Processamento da imagem

Conforme ocorre em qualquer aplicação real de processamento de imagem, inicialmente, definiu-se qual será o objeto de estudo. No caso deste trabalho, o foco é a detecção de trincas na faixa em que o veículo está trafegando. Sendo assim, regiões fora dessa faixa não precisam ser analisadas. Além disso, os vídeos possuem 48 FPS, logo, existe sobreposição de regiões em *frames* sequenciais. Portanto, somente a região inferior da imagem precisa ser analisada. Essa região é denominada Região de Interesse (ROI). A Figura 4 apresenta a ROI em destaque em um *frame* dos vídeos coletados.



Figura 4: Definição da Região de Interesse

Uma vez delimitado o ROI, o próximo passo consistiu na aplicação de filtros morfológicos na imagem. Inicialmente, reduziu-se o número de canais de cores da imagem de três (vermelho, verde e azul) para um (1) (cinza), por intermédio da operação de conversão em tons de cinza (Saravan, 2010). Na sequência, viu-se a necessidade de se implementar um método de suavização, que tornasse mais uniformes as regiões onde não houvessem trincas, consequentemente dando destaque a regiões com trincas. Nesse cenário, métodos de suavização através de processos de difusão são frequentemente utilizados. O método adotado, neste trabalho, foi o filtro de Difusão Anisotrópica, descrito no trabalho de Perona, Pietro e Malik (1990). A Figura 5 apresenta o resultado da aplicação desse filtro.



Figura 5: Exemplo de filtro de difusão anisotrópica

3.1.2 Separar imagem em blocos

Dada a natureza não uniforme das trincas, buscar uma trinca em uma região pequena e de tamanho controlado é uma tarefa mais simples do que em um grande espaço de busca. Por essa razão, autores como Chatterjee e Yi-Chang (2018) realizam processamento por blocos. A ideia dessa abordagem se baseia em um paradigma da área de computação conhecido como divisão e conquista. Nele, a imagem é dividida em diversos blocos, todos do mesmo tamanho. Cada bloco é avaliado para saber se tem trincas ou não, de forma independente. Após avaliar todos os blocos e obter, para cada um destes, a solução local, esses blocos são combinados de forma a se achar a solução global. Para o presente trabalho, foram considerados blocos de tamanho 25×25 pixels. Este valor foi adotado através de calibração manual.

3.1.3 Aplicar operação de segmentação

O procedimento de segmentação adotado neste trabalho visa converter a imagem para a forma binária, isto é, cada *pixel* pode admitir somente duas cores: branco (255, 255, 255) ou preto (0, 0, 0). A abordagem de segmentação utilizada foi a limiarização (Sezgin e Sankur, 2004), que se caracteriza por ser um processo de segmentação de imagens que se baseia na diferença dos níveis de cinza que compõem diferentes objetos de uma imagem. A partir de um limiar estabelecido de acordo com as características dos objetos que se quer isolar, a imagem pode

ser segmentada em dois grupos: o grupo de *pixels* com níveis de cinza abaixo do limiar; e o grupo de *pixels* com níveis de cinza igual ou acima do limiar. Em uma imagem limiarizada, atribui-se um valor fixo para todos os *pixels* de mesmo grupo. Para se obter esse limiar, neste trabalho, utilizou-se a abordagem descrita em Chatterjee e Yi-Chang (2018).

Seja S_i o conjunto de *pixels* em uma imagem i , o primeiro passo para determinar o limiar consiste em encontrar os valores mínimos e máximos de i , segundo a Equação 1 e a Equação 2.

$$v(i)_{mean} = \frac{\sum_{j \in S_i} v_j}{|S_i|} \quad (1)$$

$$v(i)_{min} = \min_{j \in S_i} v_j \quad (2)$$

Em que:

v_j é o valor do *pixel* j ;

$v(i)_{mean}$ é o valor médio dos *pixels* da imagem i ;

$v(i)_{min}$ é o valor mínimo dos *pixels* da imagem i ;

Uma vez calculados esses valores, o limiar $t(i)$ pode ser calculado, seguindo-se a Equação 3:

$$t(i) = \max(0, \min(i)_{min} + \alpha(v(i)_{mean} - v(i)_{min}) - \beta, \gamma) \quad (3)$$

$$0 \leq \alpha \leq 1, 0 \leq \beta \leq 0, 0 \leq \gamma \leq 255$$

Em que: α , β e γ são parâmetros ajustáveis.

O parâmetro α controla o valor inicial do limiar como um ponto entre o valor mínimo e o valor médio da imagem i , enquanto que o parâmetro β objetiva reduzir o valor do limiar para um valor constante. No caso de imagens sem trincas, essa redução empurra o limiar para um valor inferior ao valor mínimo, evitando-se, assim, que qualquer *pixel* seja segmentado como *pixel* de trinca. Por fim, o parâmetro γ é responsável por delimitar o limite superior do limiar. A Figura 6 apresenta o resultado de segmentação em duas imagens, uma sem trincas (A) e outra com trincas (B). É possível notar que o procedimento de segmentação conseguiu separar, com sucesso, regiões com trincas daquelas nas quais as trincas estão ausentes.

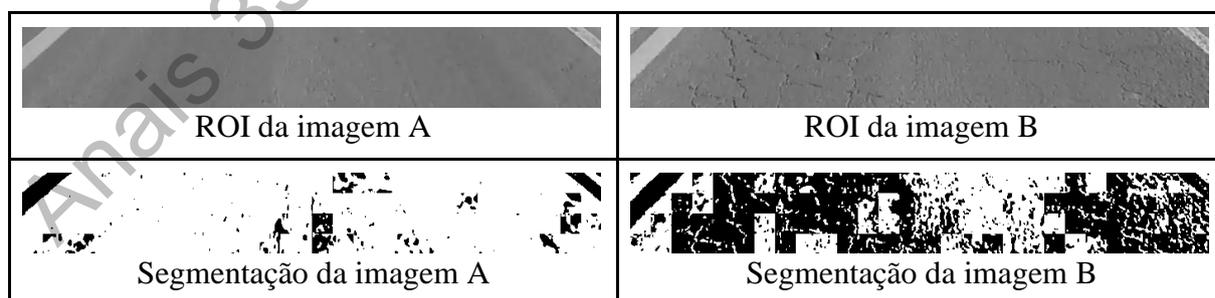


Figura 6: Aplicação da operação de segmentação

3.1.4 Descartar regiões com baixa probabilidade de trincas

Tendo em vista que o procedimento de segmentação realiza somente uma transformação da imagem, faz-se necessária a utilização de um procedimento para descartar blocos com baixa probabilidade de trincas, baseado no resultado da segmentação. Sendo assim, para cada bloco, caso este tenha um percentual de *pixels* brancos maior que um limite (foi utilizado 40%), esse

bloco é considerado sem trinca, caso contrário passa a ser um candidato a ter trincas. A Figura 7 mostra o resultado da política de descartar regiões com baixa probabilidade de trincas. É possível notar que a maior parte dos blocos sem trincas é descartada, de forma a minimizar o erro de uma possível classificação errada na sequência.

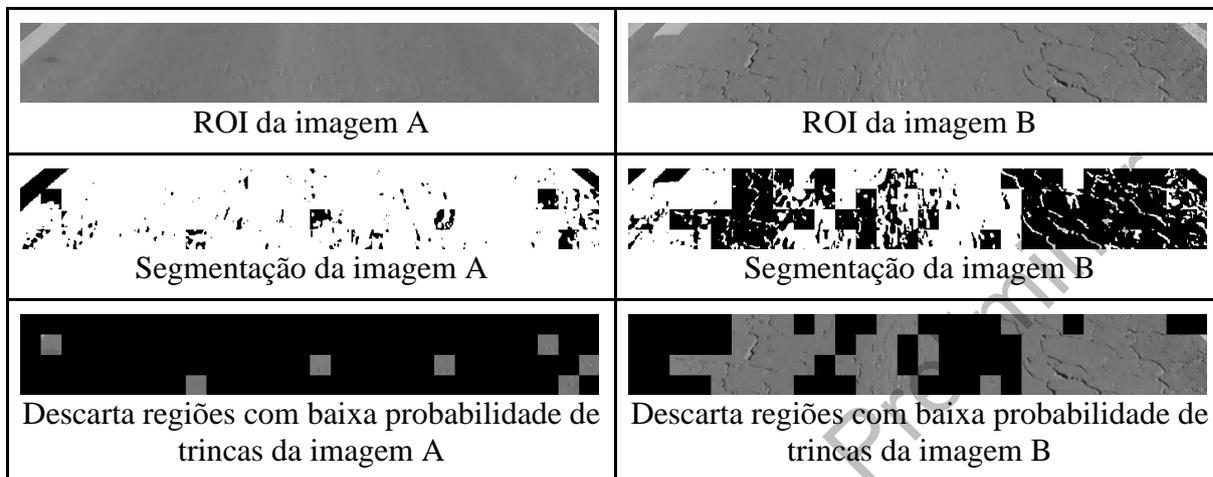


Figura 7: Aplicação do método para descartar blocos com baixa probabilidade de terem trincas

3.1.5 Verificar regiões candidatas a terem trincas

Após descartar os blocos com baixa probabilidade de terem trincas, é necessário verificar, dentre os demais blocos, quais são aqueles que realmente têm trincas. Para isso, fez-se o uso da abordagem de CNN, baseada no trabalho descrito em Cha, Choi e Büyüköztürk (2017). Como treinamento, foi utilizado um total de 553.669 imagens 25×25 pixels. Uma imagem foi considerada como tendo trincas, caso o percentual de regiões de trincas desta fosse superior a 25%. Nesse contexto, obteve-se um índice de 69% de imagens com trincas. Para serem obtidas essas imagens, um conjunto de imagens, cujo *ground-truth* tenha sido identificado manualmente, foi pós-processado. A Figura 8 mostra um exemplo de imagem e seu respectivo *ground-truth*.

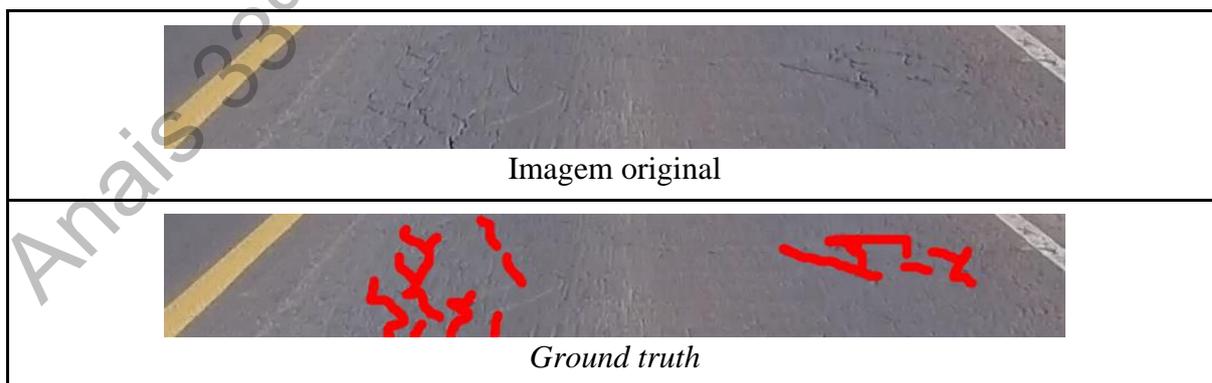


Figura 8: Exemplo de imagem utilizada para treinamento da CNN

No que diz respeito às suas características, a CNN foi criada com o auxílio da biblioteca Keras (Python 3.6). A rede implementada possui sete camadas, sendo que as camadas de um a seis têm as respectivas quantidades de nós: 32, 64, 128, 256, 256, 2. A sétima camada recebe a denominação *Flaten* (Jin *et.al*, 2014) e tem o objetivo de conectar a convolução com as

camadas densas. Como função de ativação se utilizou de *Rectified Linear Units* (ReLU), que é uma das funções de ativação não linear mais utilizadas em CNNs (Ararap, 2018; LeCun *et.al*, 2015), e, no que se refere ao ajuste dos pesos sinápticos e limiares da rede neural, foram utilizadas 30 épocas. Por fim, utilizou-se, como otimizador, uma variação do gradiente descendente, denominado Otimizador de Adam, que possui como característica o uso da taxa de aprendizado adaptativa. A Figura 9 apresenta todas as etapas para detecção de trincas, desde a segmentação até a detecção de trincas com CNN. É possível notar que a CNN consegue descartar a maior parte dos blocos sem trincas, que eram candidatos a terem trincas. Por fim, a Figura 10 apresenta o resultado do cálculo de percentual de trincas dessa mesma imagem.

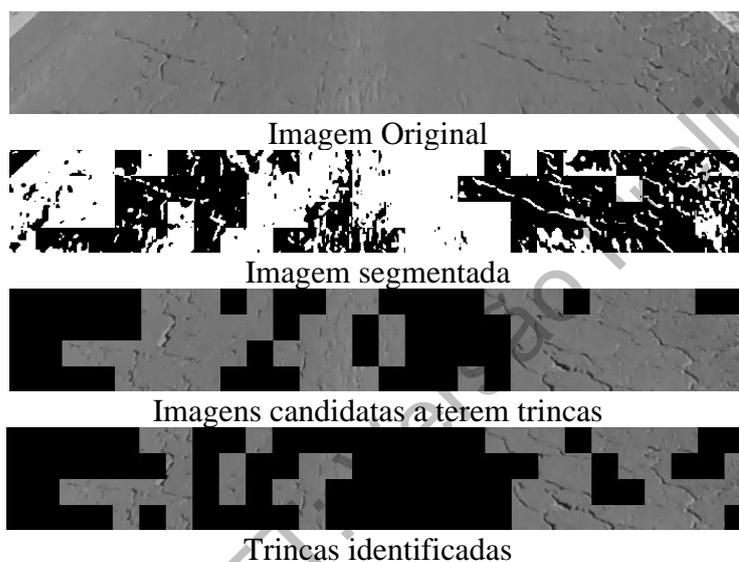


Figura 9: Trincas identificadas

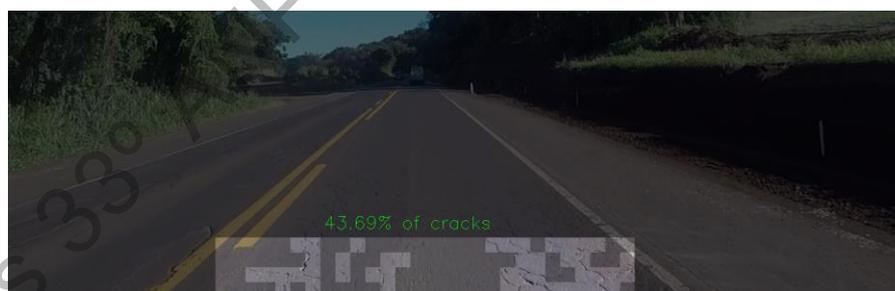


Figura 10: Percentual de trincas

4. AVALIAÇÃO DO MODELO

A avaliação do modelo se deu pela análise da CNN. Para isso, foram analisados um total de 138.909 imagens 25×25 pixels. Foi utilizada uma matriz de confusão, juntamente com métricas de avaliação. Essa matriz de confusão apresenta quatro valores, as quais são descritas a seguir.

- **Verdadeiros Positivos (TP):** indica que determinada condição existe, quando realmente ela existe.
- **Falsos Positivos (FP):** indica que existe uma determinada condição, quando a condição não existe.

- **Falsos Negativos (FN):** indica que não existe uma determinada condição, quando a condição existe.
- **Verdadeiros Negativos (TN):** indica que determinada condição não existe, quando realmente ela não exista.

A Tabela 1 apresenta o resultado da matriz de confusão. É possível observar que a CNN está com um índice de assertividade alto. Aproximadamente nove imagens, em cada dez, são classificadas corretamente, o que é um valor bastante razoável, considerando todos os empecilhos resultantes de uma coleta com vídeo.

Tabela 1: Matriz de confusão obtida com a CNN

		Valor previsto	
		Com Trincas (T)	Sem Trincas (F)
Classe real	Com trincas (T)	33.537 (TP)	10.313 (FP)
	Sem Trincas (F)	4.090 (FN)	90.968 (TN)

De posse da matriz de confusão, é possível calcular a acurácia, a precisão e a sensibilidade do modelo. A descrição dessas proporções, com conceitos extraídos de Zhu, Zeng e Wang (2010), bem como resultados obtidos são apresentados na sequência.

- **Acurácia:** é a proporção de predições corretas, sem levar em consideração o que é positivo e o que é negativo (Equação 4). **O modelo proposto obteve aproximadamente 0,90 de acurácia.**
- **Precisão:** é a proporção de verdadeiros positivos em relação a todas as predições positivas (Equação 5). **O modelo proposto obteve aproximadamente 0,76 de precisão.**
- **Sensibilidade:** é a proporção de verdadeiros positivos, ou seja, a capacidade de o sistema prever corretamente a condição para casos que realmente a têm (Equação 6). **O modelo proposto obteve aproximadamente 0,89 de sensibilidade.**

$$Acurácia = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precisão = \frac{TP}{TP+FP} \quad (5)$$

$$Sensibilidade = \frac{TP}{TP+FN} \quad (6)$$

Baseado nessas métricas, é possível verificar que o modelo, além de possuir uma acurácia grande, tende a acertar mais a classificação quando uma imagem tem trinca, em detrimento de casos em que esta é inexistente. Isso ocorre de forma proposital, uma vez que essa característica foi induzida pelo alto percentual de imagens com trincas no treinamento da CNN (69%). Essa abordagem é justificada em razão de existir um filtro de pré-processamento da imagem, que visa descartar a maior parte das regiões sem trincas antes mesmo de chegarem ao classificador.

No que diz respeito ao tempo de processamento, para fins de validação, considerou-se um

computador com as seguintes especificações: processador I7-8700, 16GB RAM, placa de vídeo nVidia GTX-1050TI.

O modelo proposto foi capaz de processar, no máximo, 1,85 *frames* por segundo, o que evidencia que os algoritmos utilizados são bastante custosos, no que se refere a tempo de processamento. Em contrapartida, em razão de que se avalia a média de trincas por quilômetro, e que cada *frame* está espaçado a, aproximadamente, 40 cm (considerando que o veículo trafega na velocidade média de 70km/h e que o vídeo possui 48FPS), não se faz necessário processar todos os *frames*. Caso seja processado um (1) *frame* a cada dez metros, seriam processados um (1) *frame* a cada 25 existentes, aproximadamente. A amostragem pode ser parametrizada de acordo com a necessidade, seja esta precisão ou tempo de processamento.

5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou um modelo para estimar o percentual de trincas em vídeos-registros. É considerado um trabalho de extrema relevância, uma vez que uma política preventiva de detecção de defeitos no pavimento possibilita maior agilidade em sua manutenção, reduzindo-se, assim, o custo logístico dos transportadores e o custo de manutenção da rodovia.

A solução proposta baseou-se em duas abordagens comuns na literatura de Processamento de Imagens Digitais, que são detecção baseada em textura e classificação de objetos com o auxílio de redes neurais. Essas duas abordagens foram combinadas, de forma que a maior parte de regiões sem trincas sejam descartadas baseada em textura, enquanto que as demais regiões sem trincas são descartadas com o auxílio de uma CNN.

O modelo de CNN proposto foi validado em um conjunto de 138.909 imagens, obtendo excelentes indicadores de acurácia e de sensibilidade (0,9 e 0,89, respectivamente). A precisão do modelo teve um valor menor (0,75), em razão de que, propositalmente, o modelo tende a atuar com melhor eficácia em imagens com trincas, uma vez que as demais tendem a ser descartadas pela abordagem de textura.

A detecção de trincas em um ambiente com sombras ainda é uma questão em aberto na literatura e pode ser explorada em pesquisas futuras. Além disso, construir um conjunto de dados em que se sabe exatamente o percentual de trincas em cada imagem, para aferir a métrica, em detrimento de somente o modelo de CNN, também pode ser abordado em pesquisas sequentes.

Agradecimentos

Os autores agradecem o apoio financeiro do Laboratório de Transportes e Logística (LabTrans), da Universidade Federal de Santa Catarina (UFSC).

REFERÊNCIAS BIBLIOGRÁFICAS

- Agarap, A. F. (2018) *Deep learning using rectified linear units (relu)*. arXiv preprint arXiv:1803.08375.
- Bhosle, V. K., Supriya, S., Sowmya, M. S., Subramani, V., e Shruthi, G. (2018) Face Recognition with 2D-Convolutional Neural Network. *International Journal of Advanced Research in Computer Science*, v. 9, n. 3, p. 373.
- Brasil. Departamento Nacional de Infraestrutura de Transportes – DNIT (2003) *NORMA DNIT 005/2003 – TER, Defeitos nos pavimentos flexíveis e semi-rígidos – Terminologia*. Brasília, DF. Disponível em: http://ipr.dnit.gov.br/normas-e-manuais/normas/terminologia-ter/dnit005_2003_ter.pdf. Acesso em: 13 jun. 2019.

- Brasil. Departamento Nacional de Infraestrutura de Transportes. Diretoria de Planejamento e Pesquisa. Coordenação Geral de Estudos e Pesquisa. Instituto de Pesquisas Rodoviárias (2005) *Manual de Restauração de pavimentos asfálticos*. 2. ed. Rio de Janeiro.
- Brasil. Departamento Nacional de Infraestrutura de Transportes - DNIT (2017) *Instrução de Serviço/DG nº 10, de 11 de setembro de 2017*. Dispõe sobre a rotina de procedimentos para a realização do levantamento de campo do Índice de Condição da Manutenção (ICM) das rodovias federais brasileiras. Brasília, DF.
- Brasil. Ministério da Transparência e Controladoria-Geral da União (2017) Brasília, DF. Disponível em: <https://auditoria.cgu.gov.br/download/10858.pdf>. Acesso em: 13 jun. 2019.
- Brasil. Departamento Nacional de Infraestrutura de Transportes – DNIT (2018) *Custos médio Gerenciais*. Brasília, DF. Disponível em: http://www.dnit.gov.br/custos-e-pagamentos/custo-medio-gerencial/copy_of_ANEXOIXCUSTOMDIOGERENCIALJULHO2017.pdf. Acesso em: 13 jun. 2019.
- Brasil. Departamento Nacional de Infraestrutura de Transportes – DNIT (2019) *Sistema Nacional de Viação*. Brasília, DF. Disponível em: <http://www.dnit.gov.br/sistema-nacional-de-viacao/sistema-nacional-de-viacao>. Acesso em: 13 jun. 2019.
- Cha, Y.; W. Choi e O. Büyükoztürk (2017) Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, v. 32, n. 5, p. 361-378.
- Chatterjee, A. e Y. Tsai (2018) A Fast and Accurate Automated Pavement Crack Detection Algorithm. In: *Anais do 26TH European Signal Processing Conference (EUSIPCO)*. (p. 2140-2144 – IEEE).
- CNT (2018) *Acidentes Rodoviários e a Infraestrutura*. Disponível em: <https://www.cnt.org.br/pesquisas>. Acesso em: 13 jun. 2019. (Confederação Nacional do Transporte, Brasília, DF).
- CNT (2019) *Boletim Estatístico*. Disponível em: <https://www.cnt.org.br/boletins>. Acesso em: 13 jun. 2019. (Confederação Nacional do Transporte, Brasília, DF).
- Jin, J.; D. Aysegul e C. Eugenio (2014) *Flattened convolutional neural networks for feedforward acceleration*. arXiv preprint arXiv:1412.5474.
- LeCun, Y.; B. Yoshua e H. Geoffrey (2015) Deep learning. *Nature*, v. 521, n. 7553, p. 436.
- Perona, P. e M. Jitendra (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, v. 12, n. 7, p. 629-639.
- Raghavendra, U.; H. Fujita; S. V. Bhandary; A. Gudigar; J. H. Tan e U. R. Acharya (2018) Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. *Information Sciences*, n. 441, p. 41-49.
- Rakhlin, A.; A. Shvets; V. Iglovikov e A. A. Kalinin (2018) Deep convolutional neural networks for breast cancer histology image analysis. In: *Anais do International Conference Image Analysis and Recognition*. (p. 737-744 - Springer, Cham).
- Saravanan, C. (2010) Color image to grayscale image conversion. In: *Anais do Second International Conference on Computer Engineering and Applications*. (Vol. 2, p. 196-199 – IEEE).
- Sezgin, M. e B. Sankur (2004) Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, v. 13, n. 1, p. 146-166.
- UFSC - Universidade Federal de Santa Catarina. LABTRANS - Laboratório de Transportes e Logística. Departamento Nacional de Infraestrutura de Transportes – DNIT (2012) *Estudos, pesquisas e programas de capacitação para desenvolvimento e consolidação de métodos e processos para suporte à gestão de competências da CGPERT vinculadas às áreas de segurança viária e operações rodoviárias: gestão da informação – SGV: relatório parcial*. Florianópolis: LabTrans.
- Yang, F.; L. Zhang; S. Yu; D. Prokhorov; X. Mei e H. Ling (2019) *Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection*. arXiv preprint arXiv:1901.06340.
- Zhu, W.; N. Zeng e N. Wang (2010) Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG proceedings: health care and life sciences*, Baltimore, Maryland, n. 19, p. 6-7.

Jorge Destri Junior (jorge.destri@labtrans.ufsc.br)

Nicolas Pereira Borges (nicolas@ita.br)

Nataniel Pereira Borges Junior (nataniel.borges@cispa.saarland)

Strauss Carvalho Cunha (engsts@gmail.com.br)

Amir Mattar Valente (amir.valente@ufsc.br)

LabTrans/UFSC - Laboratório de Transportes e Logística da Universidade Federal de Santa Catarina.

Campus Universitário - Trindade - Florianópolis/SC - CEP 88040-970 - Caixa Postal 5005