Inspeção de *Waysides* usando Processamento de Imagens e *Deep Learning*

Bruna Reis Lyra [*] **Clebeson Canuto** [**] **Raquel Frizera Vassallo** [*]

[*] *Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, ES, (e-mail: brunarlyra@gmail.com, raquel.vassalo@ufes.br).*
[**] *ISVision - Soluções Inteligentes, ES, (e-mail: clebeson.canuto@gmail.com)*

Abstract: The advances on deep learning techniques made computer vision tasks more accurate and faster by relying on convolutional neural networks and more powerful hardware. In the industry, automatic inspection supported by these methods may ensure constant maintenance and avoid railway acidentes. Thereby, this work proposes the application of deep learning and image processing methods for automatic inspection of train wagons wheelsets. More specifically, the size of wheels and the thickness of bandage are measured, in addition to locating the bearing fixing screws. The neural network built performs semantic segmentation on photographs provided by the mining company Vale. Using a U-Net architecture with ResNet50 as backbone, the network was able to reach 92.50% in mIoU and 97.52% in mPA, which are the adopted metrics for evaluating this proposal. The post-processing step retrieved the screws and improved evaluation metrics, indicating the success of the proposed inspection.

Resumo: O avanço das técnicas de aprendizado profundo tornou as tarefas de visão computacional mais precisas e rápidas ao contarem com redes neurais convolucionais e hardware mais potentes. Na indústria, a inspeção automática apoiada por esses métodos é capaz de oferecer manutenção constante e evitar acidentes em ferrovias. Dessa forma, este trabalho propõe a aplicação de *deep learning* e métodos de processamento de imagens para realizar a inspeção automática de rodeiros em vagões de trem. Mais especificamente, são medidos o tamanho da roda e a espessura da bandagem, além de localizar os parafusos de fixação dos rolamentos. A rede neural construída realiza segmentação semântica em fotografias fornecidas pela empresa de mineração Vale. Utilizando uma arquitetura U-Net, com a ResNet50 como *backbone*, a rede foi capaz de atingir 92,50% em mIoU e 97,52% em mPA, métricas adotadas para avaliação desta proposta. A etapa de pós-processamento recuperou os parafusos e aprimorou as métricas de avaliação, indicando o sucesso da inspeção proposta.

Keywords: Deep learning; Computer vision; Semantic segmentation; Railway; Inspection.
Palavras-chaves: Aprendizado profundo; Visão computacional; Segmentação semântica; Ferrovias; Inspeção.

1. INTRODUÇÃO

O processamento visual humano é rápido e complexo, permitindo a execução das mais variadas tarefas. A ideia de fazer uma máquina atingir um nível de interpretação da informação visual semelhante ao patamar humano impulsiona o estudo da inteligência artificial no ramo da visão computacional.

Algoritmos de aprendizado profundo têm transformado esses estudos desde 2012, com o surgimento da rede AlexNet (Krizhevsky et al., 2012), que colocou as redes neurais convolucionais (CNNs, do inglês *Convolutional Neural Networks*) como a principal abordagem para tarefas de visão computacional (Chollet, 2018).

O avanço das tecnologias associadas a hardware dedicado a processamento possibilitou o desenvolvimento das redes convolucionais e elevou o desempenho de tarefas de classificação de imagem e, posteriormente, detecção de objetos e segmentação semântica. Especificamente, a detecção e identificação de itens de diferentes classes em imagens digitais e vídeos apresenta aplicações em diversas áreas, como segurança, transporte, militar e médica (Jiao et al., 2019). Já a segmentação semântica, que classifica cada pixel da imagem, é aplicada em tarefas envolvendo veículos autônomos, detecção de pedestres, plano de tratamento médico e diagnóstico computadorizado (Hao et al., 2020).

Na indústria, essas tecnologias possibilitam a automatização de diversos processos de detecção, reconhecimento, monitoramento e inspeção. Alguns exemplos de aplicação industrial são os trabalhos de Machado (2020), de Franca and Vassallo (2021) e de Luchi and Adami (2020), que tratam, respectivamente, da leitura automática de calado de navios, da classificação de dormentes e defeitos de superfície e da inspeção de eixos veiculares. No âmbito ferroviário, a inspeção automática de componentes de vagões,

trilhos e rodeiros torna o sistema mais seguro e eficiente. A análise automática pode auxiliar os profissionais da área, mitigar possíveis divergências entre técnicos e eliminar erros devido a estresse, distração ou fadiga. (Rocha et al., 2017)

Dessa forma, este trabalho visa desenvolver um sistema de inspeção de rodeiros, baseado em visão computacional e processamento de imagens, associado ao *wayside* [1] nas ferrovias. Para delimitar o escopo do trabalho apresentado, apenas alguns elementos do *wayside* serão abordados. O foco estará na medição das espessuras, em pixel, das rodas dos vagões, bem como a contagem e a localização dos parafusos de fixação do rolamento correspondente.

Este projeto é parte de uma parceria entre o laboratório de pesquisa LabVISIO (Laboratório de Visão Computacional e Robótica), localizado no Centro Tecnológico da Universidade Federal do Espírito Santo (UFES), e a empresa mineradora Vale, que possui milhares de quilômetros de malhas ferroviárias no Brasil, utilizadas diariamente para o transporte de minério (Vale, 2020). Atualmente, o gerenciamento de manutenção de locomotivas e vagões monitora apenas alguns componentes e é desenvolvido por empresas estrangeiras. A Vale tem interesse em nacionalizar o processo e incluir novas funcionalidades, de forma a influenciar a solução final.

2. TRABALHOS RELACIONADOS

O levantamento feito por Pacheco and Pereira (2018) evidencia vinte publicações que exemplificam aplicações de DL (do inglês *Deep Learning*) em diferentes áreas do conhecimento, mostrando a amplitude de alcance das redes neurais para solucionar diversos problemas práticos. No contexto da inspeção de vagões, Rocha et al. (2017) realizam a inspeção de *pads* [2] em vagões, utilizando um detector de histograma de gradientes orientados (HOG, do inglês *Histogram of Oriented Gradients*) e quatro diferentes abordagens de CNNs para o classificador. Gonçalves et al. (2019) abordam o problema da detecção de componentes do vagão também a partir de um detector HOG, mas usando um classificador com Máquinas de Vetores-Suporte (SVM, do inglês *Support Vector Machines*). Ao final do trabalho, sugere a introdução de estratégias de DL para aumento da eficiência computacional e precisão. Por fim, Rocha (2020) faz uma revisão sistemática de técnicas de DL e de *machine learning* aplicadas à inspeção e à classificação de componentes de vagões, na qual conclui a aplicabilidade das CNNs como técnica de DL para automatização da inspeção visual, além de indicar superioridade de seus resultados frente a métodos clássicos de *machine learning*.

Entretanto, em nenhuma das pesquisas citadas foi proposta uma rede neural que estime a espessura da roda, avaliando seu desgaste, ou que identifique os parafusos de fixação dos rolamentos, verificando sua presença. Além disso, o problema de inspeção dos vagões também pode ser tratado com técnicas de DL diferentes da classificação proposta pelos trabalhos citados, como detecção de objetos ou segmentação semântica.

Assim, este trabalho visa preencher essas lacunas, desenvolvendo um sistema de identificação de parafusos e marcação de bandagem [3] que possibilite a realização de uma inspeção automática em rodeiros de vagões de trem. Para tal, será usada uma técnica de DL baseada em segmentação semântica, acompanhada de métodos clássicos de processamento de imagens.

A avaliação do desgaste da bandagem contribui com os procedimentos de inspeção de rodeiros, evitando que esses equipamentos sejam mantidos em funcionamento indevidamente, o que aumentaria as chances de acidentes nas ferrovias. A identificação de parafusos, por sua vez, caracteriza o tipo de fixação utilizado no rolamento em questão. Além disso, um sistema capaz de identificar a presença de parafusos pode ser uma ferramenta importante para alertar sobre sua ausência, o que também comprometeria a integridade dos rodeiros em funcionamento.

3. REFERENCIAL TEÓRICO

Segundo Haykin (2009), uma rede neural artificial é projetada para simular o funcionamento do cérebro humano em uma tarefa em particular. Unidades de processamento simples são chamadas de neurônios e dispostas em camadas, nas quais cada neurônio é conectado às camadas anteriores e posteriores. O processo de aprendizagem de máquina ocorre por meio de um algoritmo que permite a modificação dos pesos atribuídos em cada unidade de processamento, e o desempenho da rede neural está diretamente ligado à sua capacidade de generalização.

3.1 CNNs

Para tratar dados multidimensionais com elevada correlação espacial, como imagens, são comumente utilizadas redes neurais convolucionais (Machado, 2020). Basicamente, as CNNs são redes neurais profundas que combinam estruturas de percepção simples hierarquicamente para formar uma estrutura de percepção complexa (Marim, 2019). Nas camadas convolucionais, núcleo da CNN, operações de convolução são realizadas entre a imagem e um filtro de pesos ajustáveis, para a extração do chamado mapa de características. Além disso, nas camadas mais profundas, as características são extraídas do mapa fornecido pela camada anterior, resultando na extração e seleção de características mais complexas e abstratas (Rocha, 2020).

Os principais tipos de problemas que utilizam CNN no campo de visão computacional são: classificação, detecção de objetos e segmentação semântica. Classificação consiste em rotular imagens inteiras em uma ou mais categorias, como mostrado na Figura 1(b). A detecção de objetos identifica e localiza objetos na imagem, determinando uma caixa delimitadora contendo o elemento de interesse (Figura 1(c)). A segmentação semântica, por sua vez, consiste em classificar cada pixel da imagem, oferecendo informação semântica e espacial dos objetos (Figura 1(d)) (Hao et al., 2020).

[1] Sistema de monitoramento e inspeção do material rodante, posicionado na lateral da via ferroviária.

[2] Peça instalada nos quadros laterais do truque ferroviário, para proteção do rolamento contra impactos e correção do movimento lateral após trechos de curvas.

[3] Espessura da região da roda que entra em contato com o trilho, sofrendo desgaste.

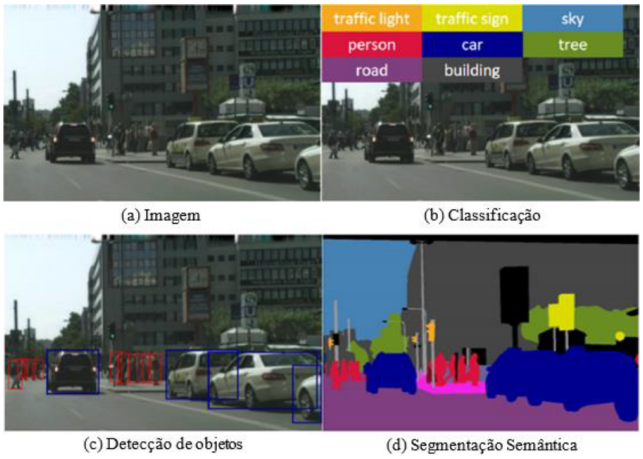


Figura 1. Exemplos de aplicação de CNNs. (Alterada de (Hao et al., 2020))

A capacidade da segmentação semântica em delinear contornos a torna ideal para a delimitação da bandagem das rodas neste trabalho. Além disso, como a detecção de parafusos também pode ser realizada pelo método da segmentação semântica, esta foi a abordagem adotada para o desenvolvimento da solução apresentada neste artigo.

3.2 Arquitetura

Para realizar uma segmentação semântica, Long et al. (2015) propuseram uma rede totalmente convolucional (FCN, do inglês *Fully Convolutional Network*), inspirados pelo sucesso das CNNs. Enquanto a CNN possui uma camada densa de saída, que ignora informações locais e oferece um rótulo como saída da rede, a FCN substitui essa camada por mais uma camada convolucional, produzindo mapas de características como resultado. Como a saída teve suas dimensões reduzidas por processos de subamostragem, sua ampliação pode ser feita via interpolação, consistindo em uma convolução com passo fracionário (o inverso do passo da convolução), que recebe o nome de convolução transposta. O filtro de convolução transposta pode ter seus parâmetros aprendidos dentro da rede, possibilitando o aprendizado de ampliações não-lineares e resultando em um método rápido e efetivo para predição densa.

Até então, o sucesso das CNNs estava limitado à disponibilidade de um grande conjunto de dados, como foi o caso da AlexNet. Conjuntos de dados extensos, combinados a redes maiores e mais profundas, tornavam o treinamento lento e custoso computacionalmente. A fim de modificar a arquitetura de uma FCN para funcionar com poucas imagens de treinamento, Ronneberger et al. (2015) construíram a U-Net (Figura 2). A rede consiste em um caminho de contração, ou codificador, no qual as camadas convolucionais capturam contexto, e um caminho de expansão, ou decodificador, que recupera detalhes de contorno. Enquanto o codificador reduz o tamanho das imagens, o decodificador o recupera por meio de convoluções transpostas e combina os resultados com os mapas de características do codificador. A ideia é permitir à rede propagar informação de contexto para as camadas de maior resolução, adicionando grande número de canais de características às camadas de expansão. Assim, o caminho

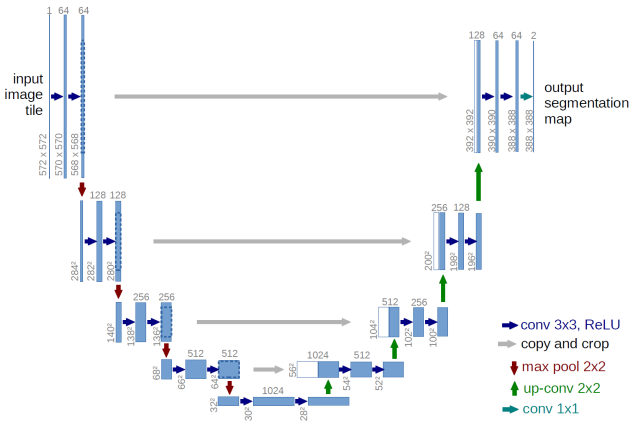


Figura 2. Arquitetura U-Net. (Ronneberger et al., 2015)

de expansão torna-se aproximadamente simétrico ao de contração, fazendo com que a arquitetura tome a forma da letra U (Figura 2).

3.3 Redes Pré-Treinadas

A estrutura codificador-decodificador consiste em abordagem padrão para redes de segmentação semântica, utilizando uma rede pré-treinada para a extração de características no codificador (Zhang et al., 2020). Redes treinadas em extensos conjuntos de imagens, como ImageNet (Deng et al., 2009), possuem um melhor desempenho no aprendizado de características (Shorten and Khoshgoftaar, 2019).

As arquiteturas para extração de características propostas até 2016 focavam em aumentar a profundidade do modelo para melhorar a acurácia, causando problemas devido à redução do gradiente durante o algoritmo de *backpropagation*. Em 2016, a ResNet (He et al., 2016) superou humanos em tarefas de classificação nos dados ImageNet, ao propor uma solução para o problema de desaparecimento do gradiente. Implementando o chamado aprendizado residual, as conexões de atalho adicionadas não acrescentam parâmetros de treinamento e evitam a degradação do gradiente.

Segundo Zhang et al. (2020), a ResNet tornou-se um dos melhores *backbones* [4] existentes para segmentação semântica. Dessa forma, este projeto propõe o uso da ResNet50 como codificador da arquitetura U-Net, seguindo a abordagem de Sadad et al. (2021), que define a técnica como bem conhecida para tarefas de segmentação.

4. METODOLOGIA

Para a realização deste trabalho, um conjunto de 2.742 imagens foi disponibilizado pela Companhia Vale S.A. Tratam-se de fotografias tiradas por uma câmera posicionada na lateral dos trilhos, apontando perpendicularmente para o sentido de passagem das rodas. Tal câmera dispara automaticamente ao detectar a aproximação de um vagão. São imagens em tons de cinza, de resolução 1024×1024, com diferentes níveis de iluminação, diferentes espessuras de bandagem e com três diferentes configurações para fixação do rodeiro. Exemplos de imagens presentes no conjunto de dados podem ser vistos na Figura 3.

[4] Rede neural usada como base para a extração de características.

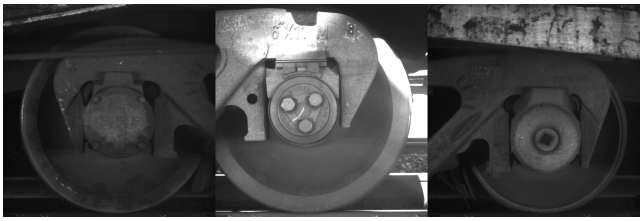


Figura 3. Exemplos de imagem do conjunto de dados.

Figura 4. Ferramenta de rotulação desenvolvida para este trabalho.

4.1 Rotulação

Antes de iniciar o treinamento da rede, as imagens precisam ser rotuladas, indicando a localização dos parafusos e os contornos da bandagem. A fim de viabilizar e simplificar a tarefa, foi desenvolvida uma ferramenta de rotulação específica para o projeto (Figura 4). Apesar de existirem ferramentas prontas para o uso, as opções disponíveis costumam fazer aproximações por polígonos, não oferendo a melhor precisão para as bordas circulares (Labelbox, 2021; Scalabel, 2021; Rectlabel, 2021).

Para uma delimitação mais precisa das circunferências da bandagem, a solução desenvolvida armazena as coordenadas de seu centro e seu raio, representando perfeitamente um círculo. Além disso, para que o sistema identifique os raios interno e externo da roda mais facilmente, a marcação deve ser composta por circunferências completas, ignorando as regiões onde os contornos não são visíveis. A marcação dos parafusos também é feita a partir de círculos indicando as posições correspondentes.

Por meio de *mouse* e teclado, o usuário interage com a interface, seguindo as instruções. Técnicas de processamento de imagens são aplicadas para auxiliar cada etapa da rotulação. Antes de finalizar qualquer etapa, o usuário pode realizar um ajuste fino na marcação. O resultado final da anotação é um documento de texto, no formato CSV (do inglês *Comma Separated Values*), com vetores correspondentes à marcação de cada imagem. Por fim, para gerar o *ground truth*, um algoritmo lê o arquivo CSV e desenha os círculos indicados, fazendo as marcações pertinentes. Na Figura 5, pode ser visto o resultado da rotulação aplicada a cada uma das imagens da Figura 3.

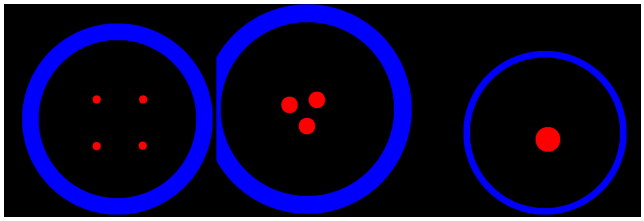


Figura 5. Exemplos de *ground truth*.

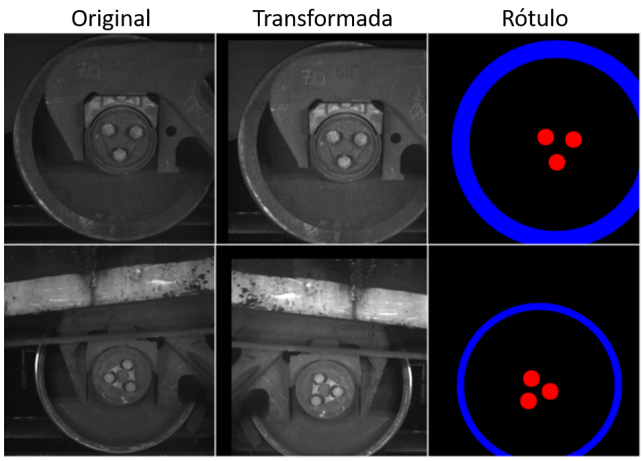


Figura 6. Exemplos de transformações aplicadas para *data augmentation*.

4.2 Data Augmentation

Quando os dados disponíveis para treinamento são insuficientes, o modelo enfrenta dificuldades para generalizar a aprendizagem para novos dados, resultando no que é chamado de *overfitting*, ou sobreajuste. De acordo com Chollet (2018), *data augmentation* é uma técnica poderosa para mitigar o *overfitting* e quase universalmente utilizada em modelos de DL aplicados a imagens. A abordagem consiste em expandir o conjunto de treinamento, gerando novos dados a partir dos originais, por meio de transformações aleatórias que resultem em imagens ainda não vistas pela rede.

É importante que as imagens geradas, apesar de inéditas, preservem consistência quanto sua aparência convincente, ou seja, devem ser imagens realistas e fiéis ao contexto do conjunto de imagens original. Por isso, tanto nas imagens quanto nos rótulos, foram aplicadas transformações aleatórias de deslocamento linear, tanto vertical quanto horizontal, com limite máximo de 10% da dimensão correspondente, além de um espelhamento horizontal. Também foi realizado um pequeno ajuste aleatório no brilho das imagens, sem alterar os rótulos. Exemplos de possíveis transformações a serem aplicadas aos dados de treinamento são apresentadas na Figura 6.

O *data augmentation* descrito é aplicado durante o treinamento, de forma on-line, a cada lote de dados retirado do conjunto original, antes de ser inserido na rede. Assim, a cada época de treinamento, a rede tem acesso a um conjunto de imagens modificado, favorecendo a qualidade de generalização e limitando as possibilidades de sobreajuste.

4.3 Arquitetura Proposta

Este trabalho propõe o uso da arquitetura U-Net adaptada para uma tarefa multiclasse, utilizando a ResNet50, pré-treinada em ImageNet, como codificador. Para isso, sua camada final densamente conectada é removida, restando cinco blocos de convolução. Os quatro primeiros blocos enviam seus mapas de características resultantes para a recuperação de contexto no decodificador, por meio das conexões de atalho típicas da U-Net. Cada bloco do decodificador expande o resultado do bloco anterior, por meio de uma convolução transposta, e o concatena com o mapa de características recebido do codificador. Ainda dentro do bloco, o resultado da concatenação passa por dois conjuntos, cada um composto por uma camada convolucional, uma camada de normalização de lote e uma camada de ativação ReLU (*Rectified Linear Unit*).

A camada de normalização de lote não foi sugerida na primeira versão da U-Net, proposta por Ronneberger et al. (2015), mas se tornou a prática mais comum em aplicações de segmentação semântica, para solucionar o fenômeno do deslocamento de covariância interna (do inglês *Internal Covariate Shift*)(Zhou and Yang, 2019). Esse fenômeno é descrito como a mudança de distribuição da rede provocada por alterações dos parâmetros durante o treinamento. Em outras palavras, mudanças nos mapas de características mais superficiais acumulam ao longo das camadas mais profundas, obrigando-as a se adaptarem a essas mudanças em detrimento do aprendizado pretendido. Proposta por Ioffe and Szegedy (2015), a normalização por lotes reduz o deslocamento de covariância interna, acelerando o treinamento de redes neurais profundas. Além disso, é uma técnica de regularização que, assim como a utilização de redes pré-treinadas e técnicas de *data augmentation*, contribui com a capacidade de generalização da rede (Shorten and Khoshgoftaar, 2019).

Ao utilizar a ResNet50 como codificador, Tomar (2021) adiciona um bloco decodificador que recebe a imagem da entrada por conexão de atalho. Essa modificação possibilita retornar uma saída do mesmo tamanho da entrada (512×512), além de contribuir para o delineamento de contornos. Por fim, um classificador multiclasse softmax gera as pontuações de classificação para cada pixel. A Figura 7 resume a arquitetura proposta.

4.4 Métricas de Desempenho

Segundo Hao et al. (2020), as métricas mais comumente usadas para a avaliação de redes neurais de segmentação semântica são: PA (do inglês *Pixel Accuracy*); mPA (do inglês *Mean Pixel Accuracy*); IoU (do inglês *Intersection over Union*); e mIoU (do inglês *Mean Intersection over Union*). Cada métrica é descrita a seguir e tem sua equação apresentada. Para tal, considere n_{ij} como o número de pixels da classe i, classificados como classe j, e k como o número de classes excluindo o fundo, de modo que o total de classes seja $k+1$.

PA é a razão entre os pixels classificados corretamente e o total (1). Conceitualmente, é a métrica mais simples, mas pode não representar adequadamente o desempenho da rede se as classes forem desbalanceadas. Uma extensão da PA é a mPA (2), que é a média entre as PAs calculadas

Caminho	Bloco	Atalhos U-Net	Tamanho Saida	Camadas Convolucionais	Ativação
Entrada	input	-	512 x 512 x 3	-	-
Codificador	conv1 (ResNet50)	-	256 x 256 x 64	1 x 7 x 7	ReLU
Codificador	conv2 (ResNet50)	-	128 x 128 x 256	1 x 1 3 x 3 x 3 1 x 1	ReLU
Codificador	conv3 (ResNet50)	-	64 x 64 x 512	1 x 1 4 x 3 x 3 1 x 1	ReLU
Codificador	conv4 (ResNet50)	-	32 x 32 x 1024	1 x 1 6 x 3 x 3 1 x 1	ReLU
Ponte	conv5 (ResNet50)	-	16 x 16 x 2048	1 x 1 3 x 3 x 3 1 x 1	ReLU
Decodificador	d1	conv4	32 x 32 x 1024	2 x 3 x 3	ReLU
Decodificador	d2	conv3	64 x 64 x 512	2 x 3 x 3	ReLU
Decodificador	d3	conv2	128 x 128 x 256	2 x 3 x 3	ReLU
Decodificador	d4	conv1	256 x 256 x 64	2 x 3 x 3	ReLU
Decodificador	d5	input	512 x 512 x 3	2 x 3 x 3	ReLU
Saida	output	-	512 x 512 x 3	-	Softmax

Figura 7. Resumo da arquitetura da rede proposta.

para cada classe individualmente. Uma alternativa mais robusta e utilizada como métrica padrão em trabalhos de segmentação semântica (LONG; SHELHAMER; DARRELL, 2015) é a mIoU (3), que representa a média entre as IoUs (4) de cada classe. A IoU representa a interseção sobre a união, sendo a razão entre os pixels classificados corretamente e a união entre os pixels pertencentes à classe e os classificados como sendo da classe.

$$PA = \frac{\sum_{i=0}^{k} n_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} n_{ij}} \tag{1}$$

$$mPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{n_{ii}}{\sum_{j=0}^{k} n_{ij}} \tag{2}$$

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{n_{ii}}{\sum_{j=0}^{k} n_{ij} + \sum_{j=0}^{k} n_{ji} - n_{ii}} \tag{3}$$

$$IoU = \frac{\sum_{i=0}^{k} n_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} n_{ij} + \sum_{i=0}^{k}\sum_{j=0}^{k} n_{ji} - \sum_{i=0}^{k} n_{ii}} \tag{4}$$

Para a avaliação de desempenho da rede deste trabalho, é construída a matriz de confusão dos dados de teste, representando a classificação por pixel realizada pela segmentação semântica. Assim, uma matriz 3×3 apresenta, em suas posições, o número de pixels da classe i classificados como da classe j. A partir da matriz de confusão, são calculadas e apresentadas as métricas de (1), (2), (3) e (4).

4.5 Pós-Processamento

Neste trabalho, o pós-processamento é responsável por interpretar as imagens de saída da rede, convertendo-as em vetores similares aos resultantes da tarefa de rotulação, descrita na Seção 4.1. A partir do vetor gerado, é possível

verificar a espessura da roda numericamente, bem como a presença e o posicionamento de cada parafuso. Alarmes e relatórios automáticos podem ser configurados com base nesse formato de saída.

As imagens RGB resultantes da rede neural têm seus canais de cor separados, de modo que o canal vermelho contenha os parafusos e o azul, a bandagem. Uma operação morfológica de abertura, usando filtro elíptico, é aplicada a cada um dos canais para eliminação de pequenos objetos e suavização das bordas. Para os parafusos, o filtro aplicado possui dimensões 7×7, enquanto o filtro para a bandagem é de tamanho 9×9. Em seguida, buscam-se os contornos do canal de cor dos parafusos em forma de lista e é determinada a circunferência circunscrita a cada um deles. As coordenadas do centro das circunferências são as localizações dos parafusos na imagem.

Como a bandagem ocupa uma parte maior da imagem e a delimitação de seu contorno é importante, busca-se a elipse que melhor contenha o contorno externo do canal de cor da bandagem. Pequenas deformações no contorno da roda, na imagem de saída da rede, aumentariam excessivamente o raio da circunferência circunscrita, tornando a abordagem por elipse mais robusta do que o método adotado para localizar os parafusos. A elipse molda-se às deformações e a aproximação garante maior correspondência com o rótulo da imagem. Depois, os pixels têm sua intensidade logicamente invertida (0 torna-se 255 e 255 torna-se 0) e, com uma delimitação de região, é encontrada a elipse correspondente ao contorno interno. Apenas ao final, as duas elipses são aproximadas para circunferências concêntricas, marcando a bandagem. Toda essa manipulação de imagens utiliza ferramentas disponíveis na biblioteca OpenCV.

5. EXPERIMENTOS E RESULTADOS

5.1 Treinamento

A base de dados foi dividida para treinamento e teste de forma aleatória, na proporção de 80% para treinamento e 20% para teste (Tabela 1). Durante o treinamento, os dados separados foram agrupados em lotes de 4 imagens aleatoriamente, redimensionadas para uma resolução 512×512 pixels, e passaram pelo procedimento de *data augmentation* descrito na Seção 4.2. Apesar da limitação de capacidade computacional imposta ao tamanho dos lotes, reduções maiores na resolução das imagens não foram consideradas, devido à possível perda de precisão da informação visual.

Tabela 1. Divisão do conjunto de dados.

Base de Dados	Treinamento	Teste
2742	2193	549
100%	80,0%	20,0%

A função de perda escolhida foi a entropia cruzada categórica ponderada, que permite corrigir o desbalanceamento entre as classes, problema recorrente em tarefas de segmentação semântica. Os pixels de todas as imagens de treinamento foram contados para cada classe e, então, foram calculadas as estimativas iniciais para os pesos na função de perda. Entretanto, após observação dos resultados nas imagens de validação, ainda se manteve uma tendência da rede em classificar os parafusos como bandagem. Assim, o peso inicialmente atribuído à classe da bandagem foi reduzido pela metade, resolvendo o problema e permitindo um melhor delineamento de contornos. Como os lotes são relativamente pequenos, com apenas 4 imagens, e passam por um processo de *data augmentation* antes de serem inseridos na rede, os pesos inicialmente definidos podem não representar precisamente as frequências de classes em cada lote. A Tabela 2 resume o procedimento para o balanceamento das classes.

Arquitetura	U-Net
Backbone	ResNet50
Perda	Entropia Cruzada Ponderada
Pesos por classe	[2,8 61,4 1,0]
Otimizador	Adam com decaimento exponencial
Taxa de Aprendizado Inicial	10^{-4}
Decaimento exponencial	$0{,}9^{step/2200}$
Lote	4
Épocas	30

Figura 8. Resumo das configurações da rede.

Tabela 2. Balanceamento de classes.

Classe	Pixels	Pesos	Pesos Corrigidos
Bandagem	84.674.137	5,7	2,8
Parafusos	7.859.107	61,4	61,4
Fundo	482.348.548	1,0	1,0

Com 2193 imagens distribuídas em lotes de 4, cada época de treinamento contou com 550 passos e o modelo foi treinado por 30 épocas. A taxa de aprendizado foi inicializada em 10^{-4}, utilizando o otimizador Adam (Kingma and Ba, 2015) em conjunto com um decaimento exponencial a uma taxa de 0,9 a cada 2200 iterações [5]. A Figura 8 resume os ajustes de configuração da rede.

5.2 Desempenho da Rede

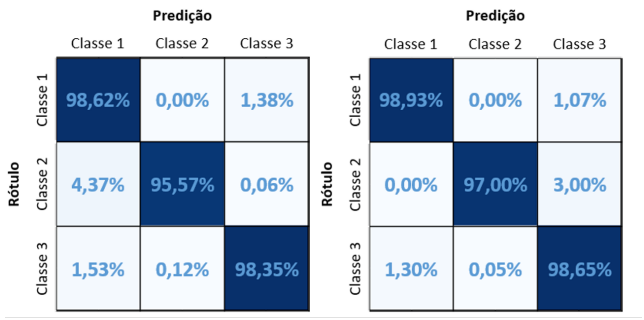
Como visto na Tabela 1, o conjunto de teste contém 549 imagens ainda não vistas pela rede. Para avaliar o desempenho da segmentação semântica, a rede neural treinada realizou a predição das imagens de teste e salvou as imagens de saída. Comparando as imagens preditas com seus respectivos *ground truths*, construiu-se a matriz de confusão da rede normalizada por classe (Figura 9(a)). A classe 1 representa a bandagem, a classe 2 refere-se aos parafusos e a classe 3, ao fundo. A partir da matriz e das equações apresentadas na Seção 4.4, calcularam-se as métricas de desempenho da rede (Tabela 3).

Tabela 3. Métricas de desempenho da rede.

	PA	mPA	IoU	mIoU
Classe 1	98,62%	-	90,22%	-
Classe 2	95,57%	-	89,17%	-
Classe 3	98,35%	-	98,12%	-
Global	**98,35%**	**97,52%**	**96,76%**	**92,50%**

Observa-se que a bandagem nunca foi classificada como parafuso e o erro mais significativo ocorreu na classificação de parafusos como sendo da classe bandagem. As métricas apontam qualidade na precisão da rede e evidenciam uma pequena dificuldade em encontrar os parafusos na imagem, em relação às outras classes.

[5] Equivalente a quatro épocas de treinamento.



(a) Saída da rede (b) Pós-processamento

Figura 9. Matrizes de confusão normalizadas.

Como a classe 2 possui menor ocorrência, as métricas PA e IoU globais representam pouco o desempenho da rede, sendo mais interessantes na avaliação de cada classe individualmente. A mIoU de 92,50% e a mPA de 97,52% representam mais adequadamente a qualidade da inferência como um todo, sendo as medidas de desempenho mais relevantes para a avaliação da rede neural construída.

Observando a matriz de confusão e confirmando nas imagens de saída da rede (Figura 10), percebe-se que o erro mais relevante ocorre quando a classe 2 (parafuso) é classificada como classe 1 (bandagem). Em algumas situações o parafuso foi quase completamente marcado como bandagem. Assim, para recuperar a localização dos parafusos, utilizaram-se as coordenadas do centro da roda e o raio interno da bandagem para determinar a região provável dos parafusos. Baseando-se no conhecimento prévio de que essa região não contém bandagem, assumiu-se que os pixels que não pertenciam ao fundo faziam parte da classe dos parafusos. Assim, a etapa descrita foi inserida antes da abertura morfológica aplicada no canal dos parafusos (Seção 4.5) e garantiu a recuperação de todos os parafusos do conjunto de teste.

A partir do arquivo CSV gerado no pós-processamento (Seção 4.5), realizou-se o processo inverso, marcando bandagem e parafusos correspondentes. O método é o mesmo utilizado para a criação dos *ground truths* na etapa de rotulação (Seção 4.1). As métricas utilizadas na avaliação da rede neural foram aplicadas para a avaliação dos resultados após o pós-processamento (Tabela 4). Percebe-se que o processamento possibilitou um aumento na precisão de recuperação dos *ground truths* (Figura 10). A matriz de confusão (Figura 9(b)) mostra que não houve erros na classificação entre as classes da bandagem e dos parafusos, e as três classes foram melhor identificadas após a etapa de pós-processamento.

Tabela 4. Métricas de desempenho do pós-processamento.

	PA	mPA	IoU	mIoU
Classe 1	98,93%	-	91,97%	-
Classe 2	97,00%	-	94,18%	-
Classe 3	98,65%	-	98,42%	-
Global	**98,67%**	**98,19%**	**97,37%**	**94,86%**

A fim de verificar os erros cometidos pelo sistema de inspeção proposto, o arquivo CSV gerado no pós-processamento foi comparado com o arquivo CSV gerado a partir dos *ground truths*. Foram calculadas as distâncias entre as

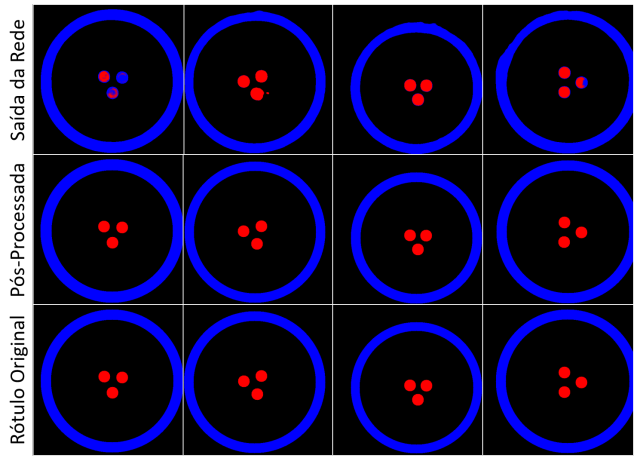


Figura 10. Etapas de resultados.

coordenadas rotuladas e preditas do centro da roda de cada imagem, além da diferença entre os raios rotulados e preditos. Para os parafusos, foram consideradas as distâncias entre as coordenadas rotuladas e preditas para cada parafuso relevante ao tipo de fixação da imagem. A Tabela 5 apresenta a média e a variância, em pixel, dos erros cometidos nas imagens de teste. As médias de erros são próximas a 1 pixel e a variância mostra uniformidade entre as amostras, reafirmando a qualidade da predição.

Tabela 5. Média e variância dos erros.

	Parafusos	Centro	Raio Int.	Raio Ext.
Média [pixel]	1,099	1,213	1,419	0,690
Variância [pixel]	0,836	2,780	0,564	0,516

Ressalta-se que as medidas foram obtidas em pixel, mas podem ser convertidas para a forma métrica se houver informações sobre a calibração da câmera utilizada, ou se a dimensão de algum elemento da imagem for conhecida.

O sistema mostra-se, portanto, adequado à tarefa. A rede treinada recebe imagens sem pré-processamento, realiza a segmentação semântica e é seguida por uma etapa de processamento de imagens. Esse procedimento sequencial identificou todos os parafusos e marcou os contornos das bandagens com sucesso.

6. CONCLUSÕES E TRABALHOS FUTUROS

A inspeção automática de componentes ferroviários substitui a inspeção visual de técnicos, muitas vezes sujeitos a erros por estresse, cansaço ou distração. Enquanto trabalhos anteriores utilizaram redes neurais de classificação para a inspeção de elementos ferroviários, este trabalho oferece uma contribuição ao apresentar uma rede de segmentação semântica capaz de realizar a inspeção de componentes ainda não abordados na literatura.

Sugestões para trabalhos futuros incluem a inspeção de outros componentes ainda não verificados por abordagens de *deep learning*, como molas de suspensão ou sapatas de freio. Projetos futuros também podem incluir a verificação de outros defeitos específicos além do desgaste da bandagem, como calos ou ovalizações de roda. Tarefas de segmentação mostram-se particularmente úteis em problemas de inspeção, quando a delimitação de contornos é importante, como foi o caso da medição da bandagem.

O desempenho da rede neural foi academicamente satisfatório e, após a etapa de pós-processamento, todos os parafusos foram recuperados e as métricas de segmentação mostraram o aprimoramento dos resultados. Admite-se que o conjunto de dados fornecido é relativamente pequeno e, mesmo com *data augmentation*, pode não representar adequadamente todas as situações encontradas na realidade. Automatizar elementos da rotulação pode contribuir com um treinamento usando maiores conjuntos de dados, que abordem imagens mais desafiadoras à rede. Ainda assim, os resultados são interessantes e encorajam a aplicação da proposta em ambiente industrial.

Para a aplicação direta na indústria, é interessante que as tarefas de inspeção sejam incorporadas a um único sistema capaz de gerar relatórios e configurar alarmes de acordo com os resultados encontrados. Câmeras devidamente posicionadas e combinadas a sistemas de inspeção mais completos podem oferecer informação suficiente para grande parte da manutenção ferroviária.

AGRADECIMENTOS

Os autores agradecem à empresa Vale S.A. pelo apoio através do projeto "Medição através de Processamento de Imagens em Wayside", processo FEST/UFES 23068.054526/2021-63.

REFERÊNCIAS

Chollet, F. (2018). *Deep learning with python*. Manning, Shelter Island.

Deng, J., Dong, W., Socher, R., Li, L., Kai, L., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 27, 248–255. IEEE, Miami.

Franca, A. and Vassallo, R. (2021). A method of classifying railway sleepers and surface defects in real environment. *IEEE Sensors Journal*, 21(10), 11301–11309.

Hao, S., Zhou, Y., and Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406, 302–321.

Haykin, S. (2009). *Neural networks and learning machines*. Pearson Education, Upper Saddle River.

He, K., Zhang, X., Ren, S., and Shaoqing, J. (2016). Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 29, 770–778. IEEE, Las Vegas.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 32, 448–456. PMLR, Lille.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7, 128837–128868.

Kingma, D. and Ba, J. (2015). Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 3. Conference Track Proceedings, San Diego.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Internationl Conference on Neural Information Processing Systems*, 25, 1097–1105. Curran Associates Inc., Lake Tahoe.

Labelbox (2021). Image segmentation made fast and intuitive. URL `https://labelbox.com/product/image-segmentation`. Acesso em: 14 ago. 2021.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 28, 3431–3440. IEEE, Boston.

Luchi, L. and Adami, A. (2020). Deep learning aplicado a inspeção visual da presença de um componente de conjunto de eixo. *Scientia Cum Industria*, 8(2), 135–144.

Machado, F. (2020). *Medição automática de calado utilizando deep learning*. Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória.

Marim, Y. (2019). Detecção de objetos: estudo e aplicação da arquitetura r-cnn.

Pacheco, C. and Pereira, N. (2018). Deep learning conceitos e utilização nas diversas áreas do conhecimento. *Revista Ada Lovelace*, (2), 34–49.

Rectlabel (2021). Rectlabel. URL `https://rectlabel.com/`. Acesso em: 22 mar. 2021.

Rocha, R. (2020). *Redes neurais convolucionais aplicadas à inspeção de componentes do vagão ferroviário*. Master's thesis, Programa de Pós-Graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia, Universidade Federal do Pará, Tucuruí.

Rocha, R., Siravenha, A., Gomes, A., Serejo, G., Silva, A., Rodrigues, L., Braga, J., Dias, G., Carvalho, S., and Souza, C. (2017). Avaliação de técnicas de deep learning aplicadas à identificação de peças defeituosas em vagões de trem. In *Conference on Graphics, Patterns and Images*, 30. Sociedade Brasileira de Computação, Porto Alegre.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 18, 234–241. Springer, Munich.

Sadad, T., Rehman, A., Munir, A., Saba, T., Tariq, U., Ayesha, N., and Abbasi, R. (2021). Brain tumor detection and multi-classification using advanced deep learning techniques. *Microscopy Research and Technique*, 84(6), 1296–1308.

Scalabel (2021). Scalabel. URL `https://www.scalabel.ai/`. Acesso em: 22 mar. 2021.

Shorten, C. and Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(60).

Tomar, N. (2021). Semantic-segmentation-architecture. URL `https://github.com/nikhilroxtomar/Semantic-Segmentation-Architecture/blob/main/TensorFlow/resnet50_unet.py`. Acesso em: 09 ago. 2021.

Vale (2020). Logística. URL `http://www.vale.com/brasil/PT/business/logistics/Paginas/default.aspx`. Acesso em: 7 nov. 2020.

Zhang, R., Du, L., Xiao, Q., and Liu, J. (2020). Comparison of backbones for semantic segmentation network. *Journal of Physics: Conference Series*, 1544(5), 012196.

Zhou, X. and Yang, G. (2019). Normalization in training u-net for 2-d biomedical semantic segmentation. *Robotics and Automation Letters*, 4(2), 1792–1799.