Avaliação de técnicas de *Deep Learning* aplicadas à identificação de peças defeituosas em vagões de trem

Rafael L. Rocha*, Ana Carolina Q. Siravenha* , Ana C. S. Gomes†,
Gerson L. Serejo*, Alexandre F. B. Silva†, Luciano M. Rodrigues*, Júlio Braga‡,
Giovanni Dias‡, Schubert R. Carvalho* e Cleidson R. B. de Souza*§

*Instituto Tecnológico Vale (ITV), Belém, PA, Brasil
†Instituto SENAI de Inovação em Tecnologias Minerais ISI/SENAI, Belém, PA, Brasil
‡Vale S.A., São Luís, MA, Brasil
§Universidade Federal do Pará, Belém, PA, Brasil

***Resumo*—Inspecionar objetos é uma tarefa importante em várias áreas do conhecimento. Assim, a inspeção e é frequentemente usada na indústria a fim de garantir a qualidade do produto, permitindo a correção de problemas e o descarte de produtos danificados. A inspeção também é amplamente utilizada na manutenção ferroviária, onde diariamente centenas de vagões são inspecionados visualmente em um processo dependente de interpretação pessoal. Este artigo descreve uma abordagem de inspeção de componentes de vagão usando técnicas de *deep learning* que compreende as fases de detecção da peça de interesse e a identificação de sua condição. O componente inspecionado é o *pad*, peça que suporta os quadros laterais do truque. A detecção da peça é feita por um detector em cascata e a classificação entre três estados (não danificado, ausente e danificado) é feita por redes neurais convolucionais. Os resultados são muito encorajadores, principalmente quando observado o desempenho da rede AlexNet.**

***Abstract*—Inspecting objects is an important task in many areas and is often used in industry to ensure product quality, allowing problem correction and disposal of damaged products. Inspection is also widely used in railway maintenance, where every day, hundreds of wagons are inspected visually in a process dependent on personal interpretation. This article describes an inspection approach of wagon components using deep learning techniques that comprises the stages of the component detection and the identification of its condition. In this work, the analyzed component is the shear pad which is responsible for supporting the truck. Object detection is done by a cascade detector and the classification task among three possible states (undamaged, absent and damaged) is done by convolutional neural networks. Our results are very encouraging, especially when observing the performance of the AlexNet network.**

I. INTRODUÇÃO

Nos últimos anos, tecnologias de visão computacional tem sido amplamente utilizadas na indústria, em aplicações que envolvem inspeção e processos de controle de qualidade [1], [2]. O uso dessas técnicas também aumentou no âmbito ferroviário com o objetivo de inspecionar automaticamente os componentes de vagões, trilhos e rodeiros, como forma de tornar a inspeção mais eficiente, segura e objetiva. Tipicamente os sistemas de inspeção incluem componentes dos trilhos ferroviários, perfil de roda, sapata de freio e dispositivos de segurança [3]–[5].

A inspeção em ferrovias é particularmente importante devido aos descarrilamentos de trem que geralmente ocorrem quando há falhas nas rodas ou eixos, trilhos danificados ou objetos nas ferrovias [6]. O descarrilamento, como se pode imaginar, pode gerar causalidades e fatalidades, resultando também em danos aos ativos ferroviários [7]. Além disso, também geram implicações ambientais e financeiras como o custo de manutenção e o efeito sobre a logística ferroviária. Portanto, a inspeção dos componentes do vagão que podem causar descarrilamento é uma tarefa fundamental para a manutenção da logística ferroviária.

A Vale S.A. é segunda maior empresa de mineração no mundo, e o transporte ferroviário desempenha um papel fundamental nas suas operações. Apesar disso, seus vagões ainda são inspecionados de maneira visual por um técnico. Somente no Brasil, a empresa opera aproximadamente 2.000 quilômetros de trilhos. Uma de suas ferrovias opera o segundo maior trem do mundo, composto por quatro locomotivas e 330 vagões.

A inspeção visual tem várias desvantagens incluindo a lentidão e falta de objetividade, propensão ao erro devido à distração, estresse ou fadiga e divergência entre diferentes técnicos [8]. O problema se torna mais crítico quando alguns dos componentes a serem inspecionados estão localizados abaixo dos vagões, o que significa que é necessário mover esses itens para um local especialmente equipado para realizar a inspeção. Isso requer tempo adicional, equipamentos específicos e trabalho intenso para realizá-la em tempo aceitável. Além disso, a inspeção visual pode implicar em riscos de segurança para o funcionário [9].

As abordagens típicas dos sistemas automatizados de inspeção de componentes envolvem aquisição de imagens, pré-processamento, extração de características e classificação. En-

tre os métodos de classificação, aqueles que envolvem aprendizado de máquina, e em particular os métodos de aprendizado profundo (do inglês, *deep learning*), vem se estabelecendo, especialmente em problemas onde as imagens são complexas devido a condições ambientais, reflexão ou distorção da lente [10], [11]. A rede neural convolucional (CNN) é uma das abordagens de *deep learning* mais utilizadas atualmente, principalmente devido à sua eficiência na aprendizagem dos padrões extraídos diretamente dos dados bidimensionais, bem como a sua flexibilidade em termos de variação de translação ou distorções locais [12].

Com base no descrito acima, uma abordagem de inspeção automática é necessária para obter informações detalhadas sobre diferentes componentes do vagão de forma rápida e confiável. Esse trabalho consiste de um sistema que fará a integração de sensores e softwares de visão computacional para adquirir e analisar imagens de componentes dos vagões. Neste trabalho, o componente analisado é o *pad*, que é responsável por suportar os quadros laterais do truque ferroviário nos conjuntos de roda, sendo um dos componentes de vagão mais importantes de acordo com nossos parceiros industriais da Vale S.A..

A proposta apresentada é composta de duas fases: detecção e classificação. A fase de detecção refere-se a identificação, dentro de uma imagem, do *pad*. A imagem utilizada nessa fase é a representação do que o inspetor veria em uma situação real, ou seja, foi tomada a partir de uma distância segura do vagão em movimento. Nessas imagens, diversos componentes estão presentes (Fig. 1), porém não serão abordados durante este trabalho.

A fase de classificação é composta pelo treinamento e teste de quatro CNNs usando um banco de dados com recortes manuais das peças de interesse, e da validação dos modelos usados usando as imagens detectadas na primeira fase.

Este artigo está organizado da seguinte forma. A seção II apresenta a descrição do problema. A seção III descreve a metodologia de classificação utilizada para abordar o problema. A seção IV concentra-se nos resultados experimentais e a seção V discute os resultados e conclui a pesquisa.

II. O PROBLEMA

Um vagão ferroviário é apoiado sobre dois truques (ou *bogie*) (Fig. 1a) que é uma estrutura formada por travessa, laterais, suspensão e por dois rodeiros. Cada vagão possui tipicamente dois truques posicionados nas extremidades opostas do vagão para suportar sua caixa. Os truques são responsáveis pela transferência da carga imposta pelo veículo aos trilhos e pela movimentação dos vagões ao longo da linha (através dos rodeiros) [13], [14].

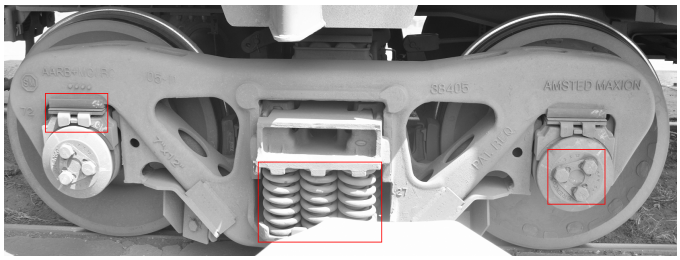
O *pad* (Fig. 1b) está posicionado entre cada um dos pedestais do quadro lateral e do adaptador do rolamento. É composto de metal e borracha, e funciona de forma semelhante a um amortecedor.

Os vagões são os ativos mais representativos da operação ferroviária, portanto, eles exigem recursos de manutenção. Em muitas empresas ferroviárias, a inspeção de vagões é realizada visualmente, onde o técnico deve avaliar 120 itens em 50 segundos, tais como: barra de compressão, triângulo, adaptador da caixa de rolamentos e placa de suporte do acoplamento. No entanto, esta técnica não é eficiente, devido à dificuldade de identificar visualmente uma grande quantidade de itens em tão pouco tempo. Além disso, o ambiente na área dos vagões oferece riscos para o inspetor.

Este trabalho consiste na construção de um sistema de inspeção de componentes de vagões através do desenvolvimento de algoritmos de processamento de imagem e visão computacional para identificar, através de fotografias digitais dos itens de interesse, os tipos de danos que o componente pode sofrer. Dado o grande e variado número de componentes dos vagões e o tipo de dano, o primeiro componente de interesse será o *pad*.

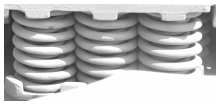
O foco será em torno de danos como uma quebra ou ruptura do componente, no qual o deslocamento da parte ou partes do componente em relação à sua posição esperada é observado. Os algoritmos desenvolvidos buscam detectar nas imagens deste componente ocorrências nas quais as peças não estão no formato e/ou posição esperados, indicando um possível dano.

No futuro, espera-se que outros componentes do vagão sejam inspecionados, como molas (Fig. 1c) e o conjunto de parafusos do rolamento (Fig. 1d).



(a) Truque ferroviário

(b) Pad



(c) Molas

(d) Parafusos do rolamento

Figura 1. Um truque ferroviário e alguns dos seus componentes. (A) Vista de um lado do truque e em detalhes, da esquerda para a direita, (b) o *pad*, (c) as molas e (d) um conjunto de parafusos do rolamento.

III. METODOLOGIA E MATERIAIS

O sistema em desenvolvimento consiste em câmeras digitais que se posicionam ao longo da da área do virador de vagões, que é um equipamento utilizado para a retirada do minério dos vagões. O processo de retirada do minério ocorre em um curto período de tempo o qual a composição de vagões permanece parada. As câmeras são usadas para capturar a imagem do truque (Fig. 1a) e são transmitidas para um equipamento de processamento para análise onde o algoritmo de visão computacional detecta o componente e realiza a classificação como ausente (Fig. 3a), não danificado (Fig. 3b), defeituoso (Fig. 3c) ou não *pad* (Fig. 3d).

O banco de imagens usado neste trabalho foi obtido durante a operação da via ferroviária, por meio da parceria com a Vale S.A., assim devido a politica interna do nosso parceiro industrial a base de dados obtida é de propriedade intelectual privada.

A. Detector de objetos em cascata

As diversas propostas para inspeção de objetos visam a implementação de processos simples e eficientes, que consumam pouco tempo entre a captura da imagem e a tomada de decisão, de forma que sua utilização em ambiente industrial seja possível. Um importante passo na inspeção de peças é a correta definição sobre a localização do objeto de interesse na cena e a consequente rejeição de todos os outros componentes presentes [15].

Em Viola e Jones (2001) [16] é proposta uma estratégia que utiliza diversos modelos de classificação simples responsáveis por rejeitar as regiões (ou janelas) da imagem que não representam o objeto de interesse e identificar possíveis regiões onde esse objeto possa estar. Esse algoritmo é bastante robusto, com altas taxas de detecção correta, e é capaz de processar pelo menos 2 *frames* por segundo, o que torna-o aplicável à aplicações em tempo real.

O algoritmo de Viola-Jones utiliza o conceito de Imagem integral, uma representação intermediária da imagem original onde cada ponto contém a informação da imagem original sob a ótica de um extrator de características (Haar histograma de gradiente orientado, por exemplo). Usa também treinamento de classificadores usando boosting, que treina o sistema a partir de um banco de dados de imagens positivas (imagens contendo o objeto a ser detectado) e imagens negativas (todas as ocorrências que não correspondem ao objeto). O método boosting consiste em combinar diversos classificadores fracos (de média precisão) em um classificador de alta precisão. Por último, há uma combinação de classificadores fortes em cascata que processam eficientemente regiões da imagem em busca do padrão desejado.

Nesse tipo de modelo, a imagem é percorrida de forma janelada, e quando uma janela é reconhecida como contendo o objeto de interesse esta é passada para o próximo classificador para que ele decida sobre a positividade da mesma. Enquanto uma mesma janela continuar a ser classificada positivamente, ela será propagada para o próximo classificador até o fim da cascata. A resposta positiva à uma janela indica que o objeto de interesse foi encontrado naquela janela. Se uma janela, em qualquer etapa da cascata, for classificada negativamente, esta janela será descartada e não será propagada adiante, pois o objeto não foi encontrado.

O modelo de detecção em cascata é indicado para problemas onde o objeto de interesse possui razão de aspecto que não varia significantemente. Esse tipo de classificador é bastante sensível à variações de rotação e exige que, em caso de variações neste aspecto, seja treinado um classificador para cada rotação possível do objeto.

A aplicação em cascata visa excluir o quanto antes as janelas negativas, porém ao encontrar uma janela positiva, esta deverá passar por diversos crivos (classificadores seguintes) até ser efetivada como uma resposta positiva. Para isso, para cada estágio do modelo, a taxa de falso negativo deve ser baixa, evitando assim que uma janela erroneamente negativada seja definitivamente descartada, enquanto a taxa de falso positivo pode ser alta, pois se uma janela for erroneamente definida como positiva na primeira etapa da cascata, por exemplo, essa classificação pode ser corrigida nas etapas seguintes.

O número de estágios da cascata deve observar o fato de que quanto mais estágios, menor a taxa geral de falso positivo, porém também será menor a taxa geral de verdadeiro positivo do modelo.

1) Parâmetros do detector: O detector de *pads* implementado neste trabalho utilizou 597 amostras positivas e 673 amostras negativas em uma cascata de 4 estágios. Vale destacar que as amostras usadas durante essa fase do trabalho não são as mesmas usadas durante a classificação dada as particularidades de cada implementação, e não sofreram transformações para aumento de amostras. A taxa de falso positivo utilizada foi de $1e^{-5}$ e de verdadeiro positivo 0.995.

As características analisadas pelos classificadores foram baseadas no histograma de gradiente orientado, do inglês *Histogram of Oriented Gradient* (HOG) [17]. Esta técnica conta as ocorrências de orientação do gradiente em porções de uma imagem, partindo do entendimento que a forma e a aparência de um objeto podem ser descritas pela intensidade dos gradientes ou na orientação das bordas do objeto, sem que haja conhecimento prévio sobre a localização dessas bordas.

Nessa abordagem, a imagem é dividida em pequenas regiões conectadas chamadas células, e para os pixeis em cada célula, o histograma de gradiente orientado é extraído. O descritor HOG é a concatenação desses histogramas que podem ser normalizados com relação ao contraste, tornando os descritores mais invariantes às mudanças em iluminação e sombreamento. Os descritores HOG, por operarem em células, são invariantes à transformações geométricas e fotométricas (exceto a citada variação em orientação)

A Figura 2 apresenta um exemplo de características do tipo HOG. A partir de janelas de 8×8 pixeis são extraídos os histogramas orientados que serão utilizados para avaliar se na janela há ou não informação do objeto de interesse.

B. Algoritmos de aprendizagem

1) Banco de Dados: As imagens capturadas para compor o banco de dados usados durante o processo de aprendizagem contemplavam todo o truque (como exemplificado pela Fig. 1a), de diferentes pontos de vista. Desta forma, os cortes feitos para isolar o *pad* tinham dimensões diferentes. Como forma de padronizar esta medida e considerando o tamanho médio dos recortes, a resolução final das imagens foi definida como $(128 \times 256 \times 1)$ (linhas$\times$colunas$\times$ canais). Devido a resolução da câmera, as imagens foram capturadas em escala de cinza, o que explica a terceira dimensão na resolução final.

Inicialmente, o banco de dados possuía um total de 334 de imagens, divididas em 3 de classes distintas ou etiquetas, que são: classe 1 (*pad* ausente - Fig. 3a), classe 2 (*pad* não

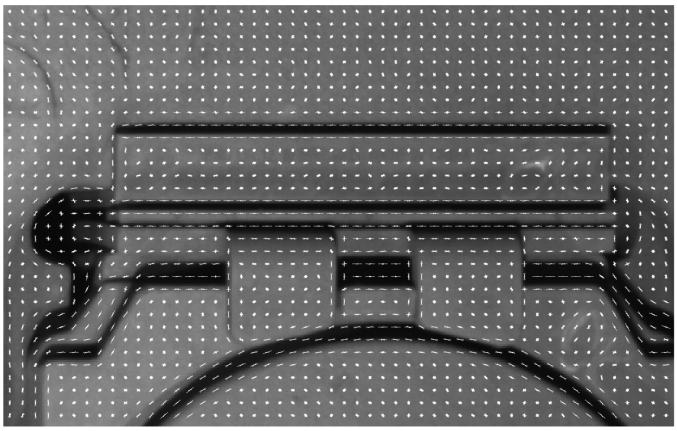


Figura 2. Características HOG de um *pad* não defeituoso. Para gerar os histogramas orientados foram utilizadas janelas deslizantes de 8×8 pixeis com sobreposição de 1 pixel.

danificado - Fig. 3b) e classe 3 (*pad* danificado - Fig. 3c), com 53, 241 e 40 imagens, respectivamente. As imagens foram coletadas em um ambiente real de operação.

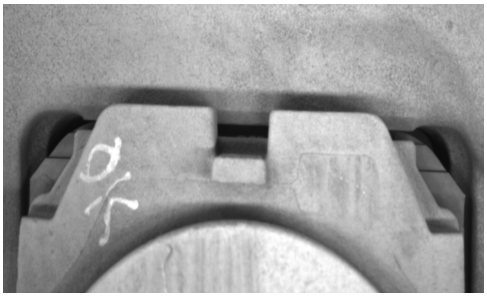
Para aumentar o número de imagens, utilizou-se o método de aumento de dados (*data augmentation*), considerado como método mais simples e mais comum para reduzir o *overfitting* [18]–[21]. Os pixeis das imagens foram contaminados com ruídos do tipo sal e pimenta, gaussiano e de Poison [22]. Além disso, algumas imagens foram rotacionadas ou tiveram a sua resolução em pixeis reduzida. As rotações feitas nas imagens não excederam ângulos maiores que $10°$, que visam simular o desalinhamento eventual nas câmeras instaladas.

Após o aumento de dados, o conjunto passou a contar com 3674 imagens, sendo 583 amostras da classe 1, 2651 da classe 2 e 440 da classe 3. As figuras 3a-3c exemplificam três das quatro classes presentes no conjunto de dados. Como pode ser observado, há também variação de luminosidade em todas as imagens, o que é extremamente importante em situações reais.

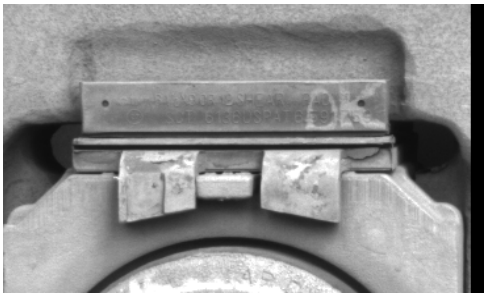
Para complementar o banco de dados usado durante o treinamento e teste dos modelos de aprendizagem, foram adicionadas 2180 amostras que representam as regiões das imagens que não contém a peça de interesse, como ilustra a Fig. 3d. Essas imagens contemplam os falsos positivos que podem ser detectados durante a primeira fase desta proposta e foram nomeados como classe 4.

2) Support Vector Machine: O *Support Vector Machine* (SVM) é um algoritmo de aprendizagem de máquina supervisionado definido por um hiperplano separador [23]. Conceitualmente, os vetores de entrada são mapeados de forma não linear para um espaço de característica de grande dimensão, no qual uma superfície de decisão linear é construída (os hiperplanos).

O ponto chave do SVM é a definição do *kernel* que define os hiperplanos e que deve refletir a natureza do conjunto de dados. Os *kernels* mais utilizados são: linear (o mais simples), quadrático, gaussiano e cúbico [24]. A ausência de mínimos locais, a escassez da solução e o controle de capacidade obtido pela otimização da margem são vantagens importantes a serem

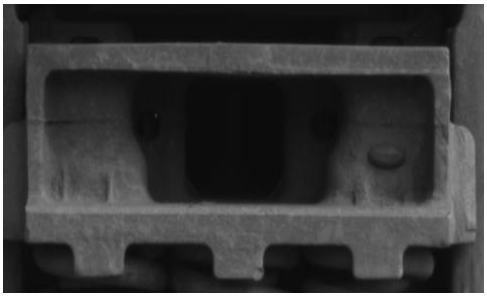


(a) Pad ausente.



(b) Pad sem danos.

(c) Pad danificado.



(d) Falso-positivo.

Figura 3. Amostras de quatro classes presentes no conjunto de dados usado. 3674 amostras foram obtidas por aumento de dados a partir do conjunto de dados original e mais 2180 amostras negativas foram adicionadas ao banco.

citadas. Porém, treinamento de um SVM e os tempos de teste podem representar uma limitação significativa, dependendo do problema [25]. Além disso, a complexidade algorítmica e os requisitos computacionais em tarefas de grande escala podem prejudicar seu uso.

O SVM é essencialmente uma técnica de classificação binária (de duas classes), mas algumas modificações permitem sua aplicação em problemas de várias classes. Alguns métodos

podem ser usados para habilitar esta adaptação, incluindo as técnicas de um-vs-um (OvO) e um-vs-todos (OvA) [26]. A diferença entre as técnicas baseia-se principalmente na quantidade de modelos produzidos para a tarefa de classificação. O OvA divide um conjunto de dados N classes em modelos N de duas classes, enquanto a OvO constrói uma máquina para cada par de classes ($N(N-1)/2$ modelos). A técnica OvA pode ter seu desempenho prejudicado devido a conjuntos de dados de treinamento desequilibrados e OvO requer mais esforços computacionais do que OvA devido à quantidade de modelos analisados [27].

Nos testes apresentados na seção IV, o SVM é usado em substituição da última camada classificadora em ambas abordagens de *deep learning* descritas na sequência.

3) Deep Learning: O padrão de aprendizagem de imagens é uma tarefa complexa que pode ser suportada por métodos de extração de características. No entanto, a escolha das características é outra tarefa complexa, que requer conhecimento total sobre o domínio do problema. As redes de aprendizado profundo vem sendo bastante utilizadas para processar dados complexos e, em particular, as redes neurais convolucionais (CNN) vem sendo empregadas eficientemente no aprendizado de dados bidimensionais [28]. Sua topologia de grade não requer etapas comuns de pré-processamento e extração de características, e é flexível em relação à translação ou distorções locais [12].

A representação hierárquica de uma CNN, leva a uma solução que mantém um certo nível de invariância à escala, rotações e translação, por meio de interações esparsas e pelo compartilhamento de parâmetros [29], [30]. Uma camada é tipicamente definida por vários filtros convoluídos com a entrada, seguido por uma função de ativação e uma função de *pooling*. O resultados das operações aplicadas em uma camada será a entrada da próxima camada.

A função de ativação utilizada pode variar de acordo com as definições do projeto, mas amplamente utilizada Unidade Linear Retificada (*Rectified Linear Unit* - ReLU) obtém representações dos dados que muito se assemelham à codificação de informações biológicas. A transformação ReLU é definida como $ReLU(z) = max(0, z)$ [31]. A camada de *pooling* ou agrupamento resume a saída de um grupo de neurônios vizinhos de um mesmo *kernel*, reduzindo a resolução da imagem sem alterar fundamentalmente a sua aparência [32].

Um possível quarto componente de uma camada é o método de regularização (*dropout*). Este componente visa reduzir o erro de generalização ao combinar vários modelos de forma exponencial em um método computacionalmente de baixo custo [33].

4) AlexNet: Uma das CNNs pré-treinadas mais utilizadas é a AlexNet [32]. É uma grande rede capaz de classificar $1,3$ milhões de imagens de alta resolução do conjunto de treinamento ImageNet em 1000 classes diferentes. A rede contém 8 camadas, sendo 5 camadas convolucionais, algumas das quais são seguidas por camadas de *pooling*, duas camadas totalmente conectadas e uma *softmax* final de 1000 saídas. Possui 60 milhões de parâmetros e 500 mil neurônios otimizados para reduzir o *overfitting*.

Para adequar o conjunto de dados presente para a arquitetura AlexNet, as imagens originais (antes do redimensionamento mencionado na Secção III-B1) foram redimensionadas para a dimensão necessária (227 linhas e 227 colunas). A camada de entrada e a primeira camada convolucional foram modificadas para admitir uma imagem em escala de cinza. Além disso, a camada *softmax* foi redimensionada para retornar três classes, ao invés de 1000.

Também foi testada a influência das SVMs na estrutura da AlexNet. Para fazer isso, a camada *softmax* foi substituída por 30 configurações diferentes de máquinas e a melhor configuração é apresentada para comparação na seção IV.

5) Nossa Proposta de CNN: Dada a natureza das imagens e a resolução da imagem descritas na Secção III-B1, a presente proposta visa utilizar uma arquitetura simples e eficiente para aprender os padrões dos *pads*. Para fazer isso, é apresentada uma arquitetura composta por 2 camadas convolutivas, 1 camada totalmente conectada e uma camada *softmax* de 3 saídas.

Os dados de entrada são as imagens em escala de cinza com $128 \times 256 \times 1$. A primeira camada convolucional possui 32 filtros, com *kernel* de 3×3, sem sobreposição e passo (ou *stride*) de 1 e ReLU como função de ativação. A segunda camada convolucional possui 64 filtros, do mesmo tamanho de *kernel* que a primeira, com *stride* de 3, sem sobreposição e seguido pela função ReLU. Os mapas são agrupados e reduzidos pela metade com *dropout* de 25%. A camada totalmente conectada possui 128 unidades, seguido da ativação do ReLU e 50% de *dropout*. A camada final é uma função *softmax* que responde a distribuição de probabilidade de três classes com função de perda de entropia cruzada.

Em comparação, enquanto a AlexNet soma 60 milhões de parâmetros, a abordagem atual tem menos de 13 milhões. Presume-se a redução do consumo de tempo durante as fases de treinamento e teste, devido à redução do número de parâmetros da proposta utilizada. Além disso, a arquitetura atual usa as imagens tão próximas quanto as originais em termos de dimensões, o que implica em menos perda de informação (ou criação de artefatos) devido ao redimensionamento da imagem.

Uma modificação da CNN utilizada também é proposta. Da mesma forma que para AlexNet, a camada *softmax* é substituída por classificadores SVM. Um conjunto de 30 configurações diferentes é testado e o melhor resultado é apresentado para discussão.

IV. Resultados

Nossos resultados comparam quatro abordagens diferentes, que são (1) AlexNet, e (2) AlexNet + SVM, (3) CNN e (4) CNN + SVM. AlexNet é usado como a primeira abordagem. Enquanto isso, a 2ª abordagem é uma adaptação da AlexNet com SVM como classificador na última camada. A 3ª abordagem mostra o desempenho da rede convolucional projetada para este trabalho (Seção III-B5), e a 4ª abordagem mostra a

Tabela I
COMPARAÇÃO DOS MÉTODOS DE CLASSIFICAÇÃO. AS PERFORMANCES DOS MÉTODOS SÃO AVALIADAS PELAS MEDIDAS DE ACURÁCIA DAS FASES DE TREINAMENTO, TESTE E VALIDAÇÃO. AS IMAGENS DE VALIDAÇÃO USADAS SÃO AQUELAS EXTRAÍDAS PELO DETECTOR EM CASCATA.

Método	Acurácia (%)		
	Treino	Teste	Validação
AlexNet	100	94.10	88.91
AlexNet + SVM	100	90.57	73.46
CNN	100	93.75	46.45
CNN + SVM	100	97.36	40.97

substituição da camada Softmax da CNN proposta por uma SVM.

A Tabela I mostra o resultado das fases de treinamento, teste e validação do modelo proposto. Tanto a AlexNet quanto a CNN proposta foram treinadas durante 60 épocas, numa média de 10 iterações por época. A taxa inicial de aprendizagem e o tamanho do lote (*batch*) foram iguais a $1e^{-4}$ e 32, respectivamente. A taxa de aprendizagem foi atualizada a cada 10 épocas por uma taxa de $1e^{-1}$ e 90% da informação do passo anterior, o chamado *momentum*, foi mantido para a iteração atual.

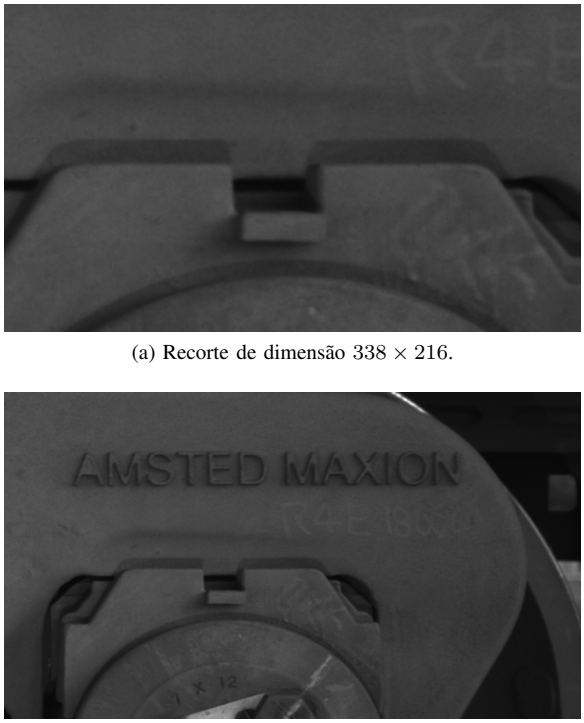
Os testes que combinam uma rede convolucional com SVM contemplam várias combinações de diferentes hiperparâmetros e modelos de otimizador com as duas estratégias de SVM multi-classe (OVA e OVO). A melhor combinação de todos os parâmetros, ou seja, a abordagem que atingiu o mínimo valor da função objetivo é apresentada na Tabela I.

Para as fases de treinamento e teste foi usado o banco de amostras extraídos conforme descrito na Seção III-B1, enquanto que para a fase de validação, apenas as imagens extraídas pelo detector foram usadas. Observa-se que os valores de acurácia de treinamento são os mesmos para todas as abordagens (100%) e que quando observadas as acurácias de teste, apesar de menores, todas apresentam uma elevada taxa de precisão, tendo alcançado a média de 93.95% de acerto.

Para a etapa de validação, foram utilizadas 335 imagens tomadas a partir de diversos ângulos e distâncias câmera-alvo, em condições reais de iluminação. Em uma operação ideal, o detector deveria ser capaz de capturar 448 *pads*, uma vez que parte das imagens contemplam uma face completa do truque que possui 2 peças-alvo como na Fig. 1a.

O detector em cascata implementado utilizou características HOG, com taxas de falsos e verdadeiros positivos definidos como $1e^{-5}$ e 0.995, respectivamente, em 4 estágios, conforme Seção III-A1. Nessas configurações, o detector selecionou um total de 642 amostras. Não houve processamento posterior para evitar que uma mesma região (contendo o alvo ou não) fosse detectada duas vezes (ou mais) em diferentes recortes, como exemplifica a Fig 4. Esse fato, somado aos erros de detecção explicam o total de amostras 43% maior que o ideal.

Das 53 amostras de *pad* ausente, o detector selecionou 54 amostras, dos 241 *pads* sem danos, 233 foram corretamente detectados, e dos 41 *pads* defeituosos apresentados, 23 foram capturados pelo seletor. O total de amostras erroneamente detectadas foi de 332.



(a) Recorte de dimensão 338×216.

(b) Recorte de dimensão 740×474.

Figura 4. Exemplo de dois recortes feitos a partir da mesma região de uma imagem. O primeiro recorte é o que mais se aproxima dos recortes usados para treinamento e teste, diferentemente do segundo que, apesar de conter a região alvo, difere consideravelmente das amostras do banco de dados. Por conveniência as imagens foram redimensionadas para apresentação, os valores de dimensão apresentados são os valores reais dos recortes.

O desempenho dos modelos de classificação diante das imagens de validação foi pior, principalmente quando observamos o resultado da CNN proposta aliada à SVM, cuja acurácia não alcançou 41%. A abordagem mais robusta foi a do modelo AlexNet, cuja performance atingiu 88.91%. Por classe, os valores de acurácia desse modelo foram: classe 1 = 98.28%, classe 2 = 97.74%, classe 3 = 61.36% e classe 4 = 98.25%. Os valores de *recall* foram: classe 1 = 91.94%, classe 2 = 93.84%, classe 3 = 90% e classe 4 = 100%. Esses resultados refletem a dificuldade em identificar *pads* defeituosos, principalmente quando observa-se a quantidade de amostras desse tipo presente no banco de dados. Por outro lado, a abundância de amostras negativas permitiu que houvesse menos erros com relação à essa classe em todos os modelos testados.

V. DISCUSSÃO E CONCLUSÃO

Neste artigo, foi proposta a utilização da técnica de aprendizagem profunda aplicada à inspeção automatizada de componentes de vagões. Essa atividade é crucial para garantir a segurança das operações e evitar acidentes e descarrilamentos. O descarrilamento é um problema sério nas ferrovias, devido às perdas ambientais e financeiras geradas, bem como perdas pessoais por meio de mortes.

De acordo com o nosso parceiro industrial, um desses componentes é o *pad*. O *pad* suporta os quadros laterais do truque nos rodeiros e está posicionado entre cada um dos pedestais do quadro lateral e o adaptador do rolamento. Devido ao grande número de vagões, e também às limitações da inspeção visual humana [8], é desejável implementar técnicas de aprendizado automático para identificar *pads*, sinalizando sobre possíveis defeitos no componente.

Neste trabalho, é abordada parte de um sistema automático de inspeção por imagens, que compreende a detecção e classificação das peças. Nessa fase, a partir das imagens capturadas por uma câmera, um algoritmo é responsável pela identificação da peça em análise dentro da cena (fase de detecção) e o resultado dessa fase é classificado por um modelo de decisão, principalmente para a identificação de peças defeituosas. O resultado dessa classificação deve compor o relatório de inspeção a ser avaliado pelo gestor da ferrovia.

Para a primeira fase, foi implementado um detector em cascata baseado nas características de histograma orientado, HOG, cujos resultados foram apresentados à modelos de aprendizado profundo. O detector identificou 642 amostras a partir de 335 imagens, sendo que 310 amostras detectadas referem-se às peças de interesse.

Os modelos de classificação testados são baseados em redes convolucionais: a pré-treinada AlexNet e uma arquitetura de CNN proposta para comparação. A principal diferença entre as abordagens refere-se à quantidade de parâmetros utilizados para extrair e aprender características dos objetos. Os modelos apresentaram desempenho semelhante durante as fases de treino e teste. Porém, os resultados de validação mostram dois fatores importantes a serem explorados futuramente: a eficiência do detector e a capacidade de generalização dos modelos de classificação.

O detector, apesar ter um bom desempenho em identificar os *pads*, apresentou uma alta taxa de falsas detecções. Além de aumentar o tempo de processamento de um futuro sistema em tempo real, isso aumenta a suscetibilidade do sistema à erros. A quantidade de amostras no banco de dados ainda está aquém do que deveria. Ainda que tenha sido complementado por amostras modificadas (*data augmentation*), ainda é preciso que novas imagens sejam adicionadas ao banco, principalmente amostras contendo *pads* danificados.

Os resultados indicam que, o longo tempo que a arquitetura AlexNet requer para otimizar seus pesos e parâmetros reflete positivamente na classificação das amostras detectadas. A arquitetura proposta, apesar de ser consideravelmente menor necessita de reformulação para que obtenha um resultado melhor em um tempo menor de treinamento e principalmente de teste (em relação à rede pré-treinada). As implementações que utilizam SVM podem ter sofrido com as alterações que compuseram o aumento do banco de dados, uma vez que esse método é sensível à variações em rotação.

Para trabalhos futuros, a proposta é testar outras técnicas de classificação para comparar os resultados atuais, investigar a possibilidade de *overfitting* das redes, melhorar o tempo de treinamento e de teste e ampliar o modelo de inspeção para outros componentes do vagão. O tempo de teste melhorado é de extrema importância devido a uma futura aplicação real para a inspeção automatizada de componentes de vagões na ferrovia.

AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq pelo apoio financeiro (processos 440880/2013-0, 310468/2014-0, 443111/2015-4 e 420801/2016-2.).

REFERÊNCIAS

[1] D.-B. Perng, H.-W. Liu, and C.-C. Chang, "Automated smd led inspection using machine vision," *The International Journal of Advanced Manufacturing Technology*, vol. 57, no. 9, pp. 1065–1077, Dec 2011. [Online]. Available: http://dx.doi.org/10.1007/s00170-011-3338-y

[2] U. S. Khan, J. Iqbal, and M. A. Khan, "Automatic inspection system using machine vision," in *Proceedings of the 34th Applied Imagery and Pattern Recognition Workshop*, ser. AIPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 210–217. [Online]. Available: http://dx.doi.org/10.1109/AIPR.2005.20

[3] C. Wöhler, *3D Computer Vision. Efficient Methods and Applications*. Springer-Verlag London, 2013, p. 382. [Online]. Available: http://www.springer.com/gp/book/9781447141495

[4] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 153–164, Jan 2017.

[5] E. Resendiz, J. M. Hart, and N. Ahuja, "Automated visual inspection of railroad tracks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 751–760, June 2013.

[6] M. Macucci, S. Di Pascoli, P. Marconcini, and B. Tellini, "Derailment detection and data collection in freight trains, based on a wireless sensor network," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 1977–1987, 2016.

[7] J. Zhao, A. Chan, and A. Stirling, "Risk analysis of derailment induced by rail breaks-a probabilistic approach," in *Reliability and Maintainability Symposium, 2006. RAMS'06. Annual*. IEEE, 2006, pp. 486–491.

[8] B. Park, Y. Chen, M. Nguyen, and H. Hwang, "Characterizing multispectral images of tumorous, bruised, skin-torn, and wholesome poultry carcasses," *Transactions of the ASAE*, vol. 39, no. 5, pp. 1933–1941, 1996.

[9] J. Hart, E. Resendiz, B. Freid, S. Sawadisavi, C. Barkan, and N. Ahuja, "Machine vision using multi-spectral imaging for undercarriage inspection of railroad equipment," in *Proceedings of the 8th World Congress on Railway Research, Seoul, Korea*, 2008.

[10] S. Ravikumar, K. I. Ramachandran, and V. Sugumaran, "Machine learning approach for automated visual inspection of machine components," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3260–3266, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.09.012

[11] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, Jul 2016. [Online]. Available: https://doi.org/10.1007/s40684-016-0039-x

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

[13] S. IWnicki, *Handbook of Railway Vehicle Dynamics*. CRC Press, 2006, p. 548.

[14] A. Sisdelli, "Estudo de desgastes de rodas e suas consequências no material rodante e na via permanente," 2006.

[15] J. Gama and P. Brazdil, "Cascade generalization," *Machine Learning*, vol. 41, no. 3, pp. 315–343, Dec 2000. [Online]. Available: https://doi.org/10.1023/A:1007652114878

[16] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features." in *CVPR (1)*. IEEE Computer Society, 2001, pp. 511–518. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2001-1.html#ViolaJ01

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[18] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[19] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[20] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High-performance neural networks for visual object classification," *arXiv preprint arXiv:1102.0183*, 2011.

[21] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis." in *ICDAR*, vol. 3, 2003, pp. 958–962.

[22] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[24] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[25] E. Osuna and F. Girosi, "Reducing the run-time complexity of support vector machines," in *International Conference on Pattern Recognition*, 1998.

[26] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Trans. Neur. Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1109/72.991427

[27] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *AIPR Workshop: Advances in Computer Assisted Recognition*. SPIE, 1998, pp. 2211–232.

[28] Y. LeCun, "Generalization and network design strategies," *Connectionism in perspective*, pp. 143–155, 1989.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[30] S. Min, B. Lee, and S. Yoon, *Deep learning in bioinformatics*. Oxford University Press, 2016, p. 42. [Online]. Available: https://arxiv.org/pdf/1603.06430.pdf

[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, G. J. Gordon and D. B. Dunson, Eds., vol. 15. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011, pp. 315–323. [Online]. Available: http://www.jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[33] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.