

CATEGORIA (3)

Abordagem baseada em regras profundas para a classificação de defeitos em superfícies de trilhos

INTRODUÇÃO.

Trilhos são elementos responsáveis pelo contato entre as rodas dos veículos ferroviários e a via permanente. Também são utilizados para transmitir as tensões causadas pelo peso destes veículos nos dormentes. Os trilhos são presos aos dormentes com auxílio de placas de apoio e fixações [1].

Segundo a ANTF (Associação Nacional dos Transportadores Ferroviários) o volume de mercadorias transportada pelas ferrovias tiveram um aumento de 125% no período de 1997 a 2018 [2]. Essa tendência ocasiona o aumento da velocidade e das cargas transportadas, alterando a dinâmica contato roda-trilho afetando diretamente os trilhos, aumentando assim os defeitos. Os defeitos na superfície do trilho ocorrem



devido a diferentes razões, por exemplo, como resultado de fadiga, devido às passagens repetitivas do material circulante sobre os componentes do trilho, como soldas, juntas e interruptores, ou devido aos impactos das rodas danificadas. Se os defeitos dos trilhos crescerem e forem tratados com atraso, eles podem levar a altos custos de manutenção [3],[4]. Os defeitos nessa parte do trilho têm um efeito significativo nas operações ferroviárias, pois diminuem a confiabilidade dos serviços e, em caso de falha, causam um descrédito para a empresa ferroviária. Em 2000, ocorreu um acidente com descarrilamento de trem causado por um erro na inspeção da ferrovia perto de Hatfield, no Reino Unido, matando 4 pessoas, como mostra a Figura 1 [21].



Fig. 1: Acidente ferroviário em Hatfield. Fonte: [21]

Como resultado, o *Network Rail* foi considerado culpado por violar a legislação de saúde e segurança durante o acidente. Foi multado em 3,5 milhões de libras e a empresa de manutenção *Balfour Beatty* em 10 milhões de libras. Ambos foram condenados a pagar 300.000 libras em custos [22]. Além disso, os passageiros



ferroviários do Reino Unido perderam cerca de 3,9 milhões de horas devido a atrasos em 2018, o que reduziu a confiabilidade do setor de transporte ferroviário, conforme exposto em [23]. Consequentemente, a classificação dos defeitos na superfície do trilho é crítica para gerenciar e monitorar as condições operacionais da ferrovia. Portanto, a classificação dos defeitos em superfícies de trilhos é um desenvolvimento crítico para ajudar empresas ferroviárias a verificarem suas condições de via, além de garantir maior confiabilidade dos serviços.

Com essa finalidade alguns trabalhos foram conduzidos nesse sentido, como em [5] onde foi proposto um método automático para detectar um tipo especifico de defeito de superfície usando medidas da aceleração da caixa de eixo dos trens. Além disso em [6] é discutido um método de inspeção para defeitos na superfície do trilho com base no sistema automatizado de visão de máquina, analisando dois tipos de defeitos através de imagens, incluindo fragmentação do trilho e rachaduras na superfície. Os autores em [7] tratam da detecção de defeitos na superfície do trilho de acordo com suas características, brilho e ruído irregulares e usa a morfologia matemática de elementos de escala múltipla e estrutura dupla como bases de detecção. Em [8], os autores apresentaram uma nova visão baseada na técnica de inspeção para detectar defeitos especiais de fadiga por contato de rolamento que ocorrem particularmente na superfície do trilho, implementando um sistema de detecção automática.



Entre todos os possíveis defeitos que podem ocorrer na superfície de trilhos, os quatro principais são *cracking, flaking, head-check e spalling*. Um aspecto comum de todos os trabalhos anteriores (ver referências [5] - [8]) é o foco na detecção da existência de um defeito, e não na classificação de diferentes tipos de defeitos, como os contemplados neste trabalho. Este trabalho foca na classificação de tipos de defeitos em superfície de trilho que são comumente estudados devido à sua gravidade e ocorrência: *cracking, flaking, head-check e spalling*. É importante enfatizar que a classificação desses tipos específicos de defeitos e a condição normal, extraindo atributos de imagens de superfície de trilhos, é nova.

As principais contribuições deste trabalho estão resumidas a seguir:

- Abordamos quatro tipos de defeitos que podem ocorrer na superfície do trilho: cracking, flaking, head-cehck e spalling. O estudo e a classificação destas falhas nunca foram abordadas anteriormente.
- Utilizamos uma rede neural convolucional profunda pré-treinada (do inglês deep convolutional neural network - DCNN) VGG19 para a extração dos atributos das imagens, com isso o classificador é capaz de aprender recursos abstratos e obter maior precisão.
- Apresentamos a análise de desempenho em termos da taxa de classificação e eficiência usando um conjunto de dados constituído por imagens adquiridas de um veículo de inspeção de via. Além disso é feita



uma análise comparativa entre o modelo proposto nesse trabalho (DRB) e modelos de classificação supervisionados já consagrados na literatura.

Nossas principais conclusões são as seguintes:

- A utilização da DCCN VGG19 para a extração de atributos das imagens é eficaz para a aplicação atual.
- A taxa de classificação gerada pelo classificador semi-supervisionado
 DRB gerou taxas de acerto acima de 93% superando os modelos supervisionados de classificação.
- Através dos resultados obtidos o classificador semi-supervisionado DRB se mostrou uma excelente alternativa para a classificação de falhas superficiais em trilhos. Podendo auxiliar no processo de inspeção da via reduzindo os tempos de inspeção, garantindo maior confiabilidade e disponibilidade da via.

O restante do artigo está organizado da seguinte forma: a Seção 2 trata da formulação do problema. A Seção 3 discute os resultados de simulações em computador. A Seção 4 apresenta as principais conclusões sobre as propostas.

FORMULAÇÃO DO PROBLEMA

Os trilhos são elementos que sofrem grande desgaste em serviço e apresentam o maior custo de manutenção da via, o desgaste desse elemento pode causar graves acidentes, prejudicando a produtividade, segurança e confiabilidade da ferrovia [9].



Afim de melhorar o processo de inspeção da via, as operadoras ferrovias podem empregar soluções mais completas como os veículos de inspeção de via (do inglês *rail inspection vehicle* - RIV). Trata-se de veículos adaptados com inúmeros sensores e equipamentos para a realização da inspeção completa de diversos parâmetros da superestrutura da ferrovia, porém possuem custo elevado e sua operação ocupa a via.



Fig. 2: Veículo de inspeção de via. Fonte: MRS Logística

Esses veículos podem capturar imagens da superfície dos trilhos, através de câmeras embarcadas, que serão analisadas por especialistas, podendo reportar alguma anormalidade nos trilhos. Com base nisso o presente trabalho consiste no desenvolvimento de um modelo computacional autônomo para diagnóstico e classificação de falhas em superfícies de trilhos através de imagens adquiridas por um veículo de inspeção de via, classificando-as em 5 classes dentre elas bom, *cracking*, *flaking*, *head-check e spalling*. Para isso é utilizado um classificador semi-supervisionado baseado em regras profundas (do inglês *deep rule based* – DRB). O algoritmo pode



auxiliar no processo de inspeção da via de forma *online* ou *offline* afim de reduzir os tempos, garantindo maior disponibilidade e confiabilidade das mesmas. O classificador DRB foi recentemente introduzido [10-12]. O classificador em questão é uma abordagem geral que serve como uma forte alternativa as redes neurais profundas (do Inglês *deep neural net* – DNNs) atuais [10–12]. O DRB é não paramétrico; não iterativo; altamente paralelizável e eficiente em termos de computação; alcança taxas de classificação muito alta superando outros métodos [12]. O classificador gera de forma autônoma um conjunto totalmente compreensível de regras condicionais do tipo "SE ... ENTÃO ..." após um processo de treinamento do tipo "única etapa" [10–12]. O "núcleo" do classificador DRB é formado pelas regras condicionais "SE ... ENTÃO ...", que consistem em um número de protótipos. Elas podem ser representadas como uma série de regras difusas mais simples com um único protótipo conectado por um operador lógico "OU" [12].

Uma descrição breve da arquitetura, processo de treinamento e a validação do classificador baseado em regras profundas (DRB) é apresenta em [10-12].



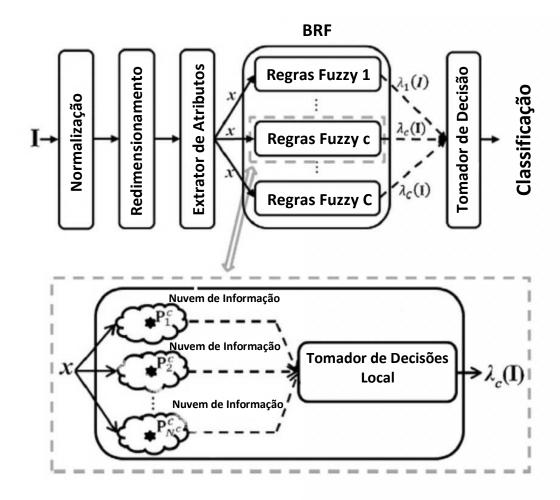


Fig. 3: Estrutura do DRB. Fonte: [24]

Arquitetura

A arquitetura geral do classificador DRB foi apresentada em [10–12]. No presente artigo, usamos o classificador DRB com arquitetura, conforme ilustrado na Figura 3. Ele consiste nos seguintes componentes:

1. Camada de normalização



Essa camada normaliza os valores de *pixel* das imagens em uma faixa com a qual o extrator de atributos trabalha. Devido ao extrator de atributos específicos usado neste trabalho, o intervalo necessário dos valores de *pixel* é [0, 255] [16].

2. Camada de redimensionamento

Essa camada serve para redimensionar o tamanho original das imagens para várias finalidades, ou seja, reduzir a complexidade computacional, aumentar a generalização e etc. No classificador DRB usado neste artigo, todas as imagens são redimensionadas para o tamanho de 227 × 227 *pixels* [16], devido ao extrator de atributos específico usado.

3. Extrator de atributos

O extrator de atributos extrai vetores de atributos globais das imagens que serão usadas para treinamento ou teste. Atualmente, as redes neurais convolucionais profundas (do inglês deep convolutional neural network - DCNN) tornaram-se a abordagem mais avançada no reconhecimento de imagens [14]. No entanto, o treinamento de uma DCNN para um problema específico é computacionalmente caro e requer muitas amostras de treinamento rotuladas [12]. No entanto, para executar a classificação usando uma DCNN pré-treinada em um novo problema no qual o DCNN não é direcionado, é necessário um grande esforço e experiência humana para o ajuste fino. Uma abordagem alternativa e amplamente usada é aplicar a DCNN pré-treinada para extrair vetores globais de atributos das imagens de treinamento e usar esses



vetores de atributos para treinar um classificador genético [12,17]. Ao usar a DCNN prétreinada, o classificador pode aprender mais atributos abstratos e discriminativos de alto nível e obter maior precisão. Seguindo esse conceito, neste trabalho, a DCNN prétreinada vgg-verydeep-19 (VGG19) [19] é empregada para a extração de atributos devido à sua estrutura mais simples e melhor desempenho. As ativações dimensionais 1×4096 da primeira camada totalmente conectada são utilizadas como o vetor de atributos da imagem [17]. Isso ajuda a evitar processos manuais e automatiza todo o processo.

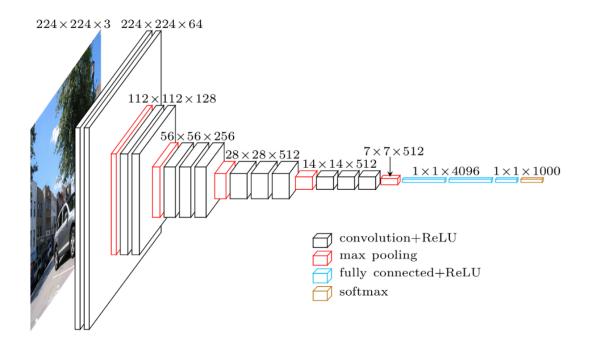


Fig. 4: DCNN vgg-verydeep-19. Fonte [25]

4. Base de regras fuzzy (BRF)

A camada BRF é um conjunto massivamente paralelo de regras fuzzy de ordem 0 do tipo AnYa [18], a qual é o "mecanismo de aprendizado" do classificador DRB. A camada



BRF consiste em subsistemas baseados em regras fuzzy (BRF) de ordem 0 do tipo AnYa C treinados em paralelo, correspondendo às classes C do conjunto de imagens (uma por classe). Os subsistemas BRF do tipo C AnYa são totalmente independentes um do outro e cada um deles pode ser alterado sem influenciar os outros. Cada subsistema BRF contém um conjunto de regras fuzzy massivamente paralelos da ordem 0 do tipo AnYa formulados em torno dos protótipos (que são os picos locais da densidade de dados [13]) identificados durante o estágio de treinamento a partir das imagens da classe correspondente com base em seus vetores de características. As regras fuzzy do tipo AnYa de ordem 0 têm o seguinte formato:

$$SE(I \sim P_i^c) ENTÃO(CLASSEC)$$
 (1)

Na Eq. 1 "~" denota similaridade; c=1,2,...C; N^c é o número de protótipos da classe c; P^c_i são os protótipos identificados, $i=1,2,...,N_c$.

Como todos eles têm a mesma consequência (mesmo rótulo de classe, c), eles podem ser combinados em uma regra com várias condições vinculadas ao operador lógico "OU" [12]. Essa regra difusa combinada tem o seguinte formulário por classe:

$$SE (I \sim P_1^c) OU (I \sim P_2^c) OU \dots OU (I \sim P_{N^c}^c) ENTÃO (CLASSE C)$$
 (2)

Durante o estágio de validação, para cada imagem de teste, cada regra fuzzy gera uma pontuação de confiança / grau de satisfação usando o princípio "vencedor leva tudo". Então, todas as pontuações de confidencialidade geradas pela base de regras difusa



serão passadas para a próxima camada (o tomador de decisão) para uma decisão geral do rótulo da classe.

5. Tomador de decisão

O tomador de decisão forma a decisão geral atribuindo rótulos às imagens de validação com base no grau de semelhança com os protótipos obtidos pela camada BRF.

Processo de treinamento:

Durante o processo de treinamento, os protótipos estão sendo identificados como os picos da densidade de dados no espaço de atributos [10–12]. Com base nesses protótipos, são geradas as regras nebulosas correspondentes. Devido à alta dimensionalidade dos vetores de características, é usado dissimilaridade de cosseno como medida de distância, que é dada abaixo [19]:

$$d(x,y) = \sqrt{2 - 2\cos\theta_{x,y}} = \sqrt{2 - 2\frac{\sum_{i=1}^{M} x_i y_i}{\|x\| \|y\|}} = \left\| \frac{x}{\|x\|} - \frac{y}{\|y\|} \right\|$$
(3)

Na Eq. 3 $\theta_{x,y}$ é o ângulo entre o vetor de características x e y;

$$cos\theta_{x,y} = \frac{(x,y)}{\|x\| \|y\|} = \frac{\sum_{l=1}^{M} x_i y_i}{\|x\| \|y\|}$$

$$||x|| = \sqrt{(x,y)} = \sqrt{\sum_{l=1}^{M} x_i^2}$$



A norma de x é denotada, M é a dimensionalidade do vetor, para esse trabalho M=4096.

Foi demonstrado em [20] que a dissimilaridade de cosseno entre os vetores originais das características globais é equivalente à distância euclidiana entre os vetores normalizados por suas normas ($x \leftarrow \frac{x}{\|x\|}$), conforme descrito na Eq. 3. Isso é importante porque facilita a eficiência computacional, permitindo o cálculo recursivo. Portanto, no restante deste trabalho, sem declaração especial, todos os vetores de recursos extraídos das imagens são normatizados por suas normas como pré-processamento padrão.

Como o processo de treinamento pode ser realizado em paralelo para cada regra difusa, consideramos as regras difusas paralelas c ($c=1,2,\ldots,C$) massivamente paralelas e resumimos seu processo de treinamento da seguinte forma.

I) Estágio 0: Inicialização

A regra c difusa é inicializada pela primeira imagem (denotada por I_1^c) da classe correspondente com o vetor de característica denotado por x_1^c ($x_1^c = x_{1,1}^c, x_{1,2}^c, \dots, x_{1,M}^c$) Em seguida, os meta-parâmetros do sistema são inicializados da seguinte maneira:

$$k \leftarrow 1; \ \mathbb{N}^c \leftarrow 1; \ P_1^c \leftarrow I_1^c; \ \mu^c \leftarrow x_1^c; \ P_1^c \leftarrow x_1^c; \ S_1^c \leftarrow 1; \ r_1^c \leftarrow r_o \quad (4)$$

Em que k é a instância de tempo atual; N^c é o número de protótipos; P_1^c é o primeiro protótipo de imagem da regra difusa; μ^c é a média global de todos os vetores de características das imagens observadas da classe; p_1^c é a média



dos vetores de características das imagens associadas ao P_1^c , que também é o centro da nuvem de dados inicializada por P_1^c ; S_1^c é o suporte da primeira nuvem de dados; r_1^c é o raio da área de influência da nuvem de dados; r_0 é um pequeno valor usado para estabilizar o status inicial dos protótipos recentemente identificados.

Nesta problemática, $r_o = \sqrt{2-2\cos(\pi/6)} \approx 0.5176$; a lógica é que dois vetores para os quais o ângulo entre eles é menor que 30° ou $\pi/6$ podem ser considerados como apontando em direções próximas ou similares. O limiar é definido com base no fato de que para dois vetores com um ângulo entre eles menor que 30° pode ser considerado semelhante. No entanto, devemos enfatizar que o r_o não é um parâmetro específico do problema ou do usuário e pode ser definido sem o conhecimento prévio, mas apenas pelo senso comum. Em geral, quanto mais alto o r_o , menos protótipos serão identificados durante o processo de treinamento, mais eficiente será o classificador DRB, mas menos detalhes o classificador DRB aprenderá com o treinamento e vice-versa.

Com a primeira nuvem de dados, a regra difusa é gerada com o protótipo P_1^c :

$$SE(I \sim P_1^c) ENTÃO(CLASSEC)$$
 (5)

II) Estágio 1: Atualização do sistema:



Atualização do parâmetro, k ($k \leftarrow k+1$) da recém-chegada imagem (I_k^c) que pertence à classe c, a média global μ^c atualizou primeiro com seu vetor de características, denotado por x_k^c , da seguinte maneira:

$$\mu^c \leftarrow \frac{k-1}{k}\mu^c + \frac{1}{k}x_k^c \tag{6}$$

E calcula-se as densidades de dados de p_i^c ($i=1,2,...,N^c$), N^c é o número de protótipos identificados, e o vetor de características I_k^c, x_k^c de acordo com o framework Empirical Data Analytics [13]:

$$D(z) = \frac{1}{1 + d^2(z, \mu^c)/(\sigma^c)^2}; \quad z = p_1^c, p_{2,\dots,p_N^c}^c, x_k^c$$
 (7)

Em que $\sigma^c=\sqrt{X^c-\|\mu^c\|^2}=\sqrt{1-\|\mu^c\|^2}$; X^c é o vetor de característica médio das imagens observadas, que é sempre igual a 1 devido à operação de normalização do vetor. Nesta etapa, atualizamos a estrutura do sistema e os meta-parâmetros para refletir a imagem recém-chegada. Primeiro, o seguinte princípio é verificado para verificar se I_k^c é um novo protótipo [15]:

$$SE\left(D(x_{k}^{c}) > \max\left(D(p_{j}^{c})\right)\right) OU\left(D(x_{k}^{c})\right) < \min\left(D(p_{j}^{c})\right)\right) ENTÃO\left(I_{k}^{c} \neq protótipo\right)$$
(8)

Quando a Condição 1 é atendida, I_k^c é definido como um novo protótipo e inicializa uma nova nuvem de dados:

Condição 1:



$$N^c \leftarrow N^c + 1; P_{N^c}^c \leftarrow I_k^c; P_{N^c}^c \leftarrow x_k^c; S_{N^c}^c \leftarrow 1; r_{N^c}^c \leftarrow r_o$$
 (9)

Se a Condição 1 não for atendida, encontramos o protótipo mais próximo de I_k^c indicado por p_n^c pela seguinte equação:

$$P_n^c = \arg\min\left(d(p_j^c, x_k^c)\right) \tag{10}$$

Antes de associarmos I_k^c a nuvem de dados de P_n^c a Condição 2 é verificada para ver se I_k^c está na área de influência de P_n^c .

Condição 2:

$$SE(d(P_n^c, x_k^c) > r_n^c) ENTÃO(I_k^c \'e um novo prot\'otipo)$$
 (11)

Se a condição 2 for atendida, significa que I_k^c estará fora da área de influência da nuvem de dados mais próxima e I_k^c se tornará um novo protótipo de uma nova nuvem de dados com meta-parâmetros inicializados pela Eq. (9).

Caso contrário, I_k^c é atribuído à nuvem de dados formada em torno do protótipo p_n^c e os meta-parâmetros dessa nuvem de dados são atualizados da seguinte forma:

$$p_n^c \leftarrow \frac{S_n^c}{S_n^c + 1} p_n^c + \frac{1}{S_n^c + 1} x_k^c; \quad S_n^c \leftarrow S_n^c + 1$$

$$r_n^c = \sqrt{\frac{1}{2} (r_n^c) + \frac{1}{2} (1 - ||p_n^c||^2)}$$
(12)

e o protótipo p_n^c permanece o mesmo.



III) Estágio 2: Atualização regras fuzzy

Após a conclusão do estágio 1, o sistema DRB atualizará a regra difusa. Em seguida, a próxima imagem é lida e o sistema volta ao Estágio 1 para iniciar um novo ciclo de processamento. O processo de treinamento é resumido no pseudocódigo a seguir.

```
While o novo vetor de atributos x_k^c de k^{th} da imagem I_k^c de classe c^{th} for válido:
              If (k = 1) Then:
              1. Inicializa o sistema usando a Eq. (4);
              2. Gera as regras fuzzy do tipo AnYa (Eq. (5));
      ii.
              1. Atualiza u^c usando a Eq. (6);
                 Calcula D(x_k^c) e D(p_i^c)(j=1,2,...,N^c) usando a Eq. (7);
              3. If (Condição 1 é conhecida) Then:

    Inicializa uma nuvem de dados usando a Eq. (9);

                 Else:
                  - Encontre P_N^c usando Eq. (10)
                  - If (Condição 2 é conhecida) Then:

    * Inicializa uma nova nuvem de dados usando Eq. (9);

                     * Atualiza a nuvem de dados existente usando a Eq. (12);
                  - End If:
              5. End If:
              6. Atualiza regras fuzzy tipo AnYa;
      iii.
              End If:
End While
```

Fig. 5: Processo de treinamento do DRB. Fonte: MatLab

Processo de validação

Durante o processo de validação, para cada imagem de teste, as regras fuzzy identificadas em C do classificador DRB geram pontuações C de confiança correspondentes às classes C no conjunto de treinamento [12].

$$\gamma^{c}(I) = \max(\exp(-d^{2}(x, p_{i}^{c})))$$
(14)



Como resultado, para cada imagem de teste, é gerado um vetor de pontuação dimensional $1 \times C$ de confiança ou graus de proximidade dos protótipos mais próximos (um por classe): $\gamma(I) = [\gamma^1(I), ..., \gamma^C(I)]$.

O rótulo desta imagem de teste é decidido usando o princípio "vencedor leva tudo" [12]:

$$Rotulo = \arg\max(\gamma^{c}(I)) \tag{15}$$

RESULTADOS

A partir da teoria abordado no tópico anterior, foram realizados testes para avaliar o modelo de classificação proposto nesse trabalho. O banco de dados que será usado nesta aplicação é fornecido pela empresa MRS Logística S.A. (https://www.mrs.com.br/). Tal banco de dados é composto por imagens que são capturadas por uma câmera acoplada em um veículo de inspeção de via permanente, contendo defeitos de superfície em trilhos. As imagens da superfície do trilho são tiradas na ordem de três imagens por metro. As imagens adquiridas se enquadram nas cinco classes a seguir: *cracking, flaking, head-check, spalling* e condição normal da superfície do trilho. Mostramos exemplos dessas imagens na Figura:



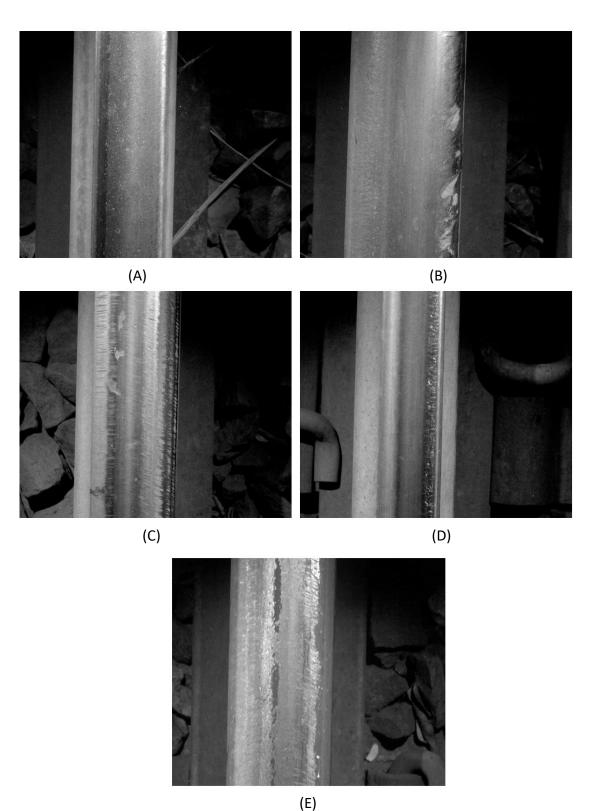


Fig. 6: Amostras de 5 classes A) condição normal B) cracking C) flaking D) head-check E) spalling. Fonte: MRS Logística



Afim de alcançar o melhor desempenho do classificador a base de dados original foi aumentada utilizando o processo de *autoencoder*, causando distorções nas imagens e garantindo o aumento das amostras. Como resultado obteve-se uma base de dados com 500 imagens distribuídas igualmente em 5 classes.

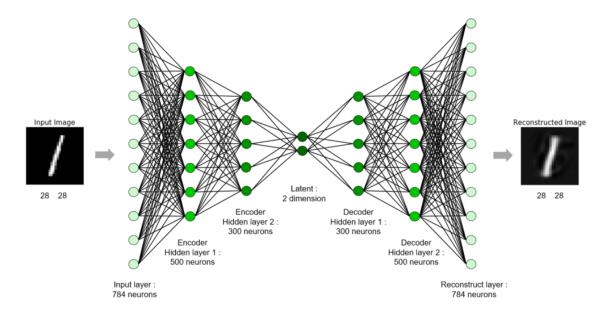


Fig. 7: Processo de autoencoder. Fonte: [26]

Para avaliar o desempenho do classificador semi-supervisionado DRB os resultados foram comparados com os seguintes classificadores supervisionados já consagrados na literatura KNN, Linear SVM, Decision Tree, Random Forest, Rede Neural Artificial, AdaBoost, Naive Bayes e QDA.

Em um primeiro momento a base de dados foi dividida em 80% para base de treinamento e 20% para base de teste. Nota-se que a fase de teste foi executada 200 vezes de forma aleatória e a média aritmética foi consolidada na tabela abaixo:



Classificadores	DRB	KNN	Linear SVM	Decision Tree	Random Forest	Redes Neurais	AdaBoost	Naive Bayes	QDA
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
Condição Normal	100.0	100.0	100.0	83.4	91.3	97.5	20.3	85.7	62.2
Cracking	100.0	100.0	99.5	78.8	92.6	95.5	47.4	91.3	54.1
Flaking	96.7	95.4	97.2	77.0	77.9	88.0	17.6	80.9	53.4
Head Check	96.0	94.4	95.8	2.5	73.4	75.9	84.9	14.9	76.5
Spalling	100.0	100.0	100.0	82.3	93.2	95.7	78.8	85.6	55.5
Eficiência	98.5	98.0	98.5	64.8	85.7	90.5	49.8	71.7	60.3

Tabela 1: Eficiência DRB

Logo após com intuito de confrontar o modelo, foi inserido um ruído branco Gaussiano (AGWN) na base de dados original seguindo a premissa de 18,93 dB de relação sinal-ruído (SNR). A aplicação do ruído tem for finalidade simular condições adversas de operação, na qual as imagens podem sofrer interferência de agentes externos como intempéries. Novamente a fase de testes foi executada 200 vezes e a média aritmética dos resultados foram consolidadas na tabela abaixo:

Classificadores	DRB	KNN	Linear SVM	Decision Tree	Random Forest	Redes Neurais	AdaBoost	Naive Bayes	QDA
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
Condição Normal	97.8	97.5	97.1	77.1	81.9	93.7	26.4	77.0	57.1
Cracking	95.1	93.9	93.9	78.8	86.2	93.0	51.7	89.1	54.2
Flaking	88.5	88.1	89.7	70.1	65.0	85.7	27.7	60.3	44.4
Head Check	87.1	80.0	85.6	60.4	64.6	81.0	25.7	61.7	44.0
Spalling	99.5	99.5	97.4	78.4	86.4	96.0	79.5	82.5	55.2
Eficiência	93.6	91.8	92.7	73.0	76.8	89.9	42.2	74.1	51.0

Tabela 2: Eficiência DRB com adição de ruído branco Guassiano (AGWM)

Como foi detalhado em tópicos anteriores o processo de treinamento do classificador semi-supervisionado *DRB* ocorre em uma única etapa, na qual imagens previamente rotuladas passam pela etapa de treinamento e são criados os protótipos (Imagens comparativas). Quanto maior for o universo de imagens de treinamento, maior



será a quantidade de protótipos criados na fase de treino e consequente mais eficiente se torna o classificador. Com a finalidade de avaliar a eficiência do classificador em função da quantidade de imagens por classe, foram traçados os gráficos:



Fig. 8: Eficiência do classificador sem adição de ruído. Fonte: Excel



Fig. 9: Eficiência do classificador com adição de ruído. Fonte: Excel

Analisando os gráficos anteriores, pode-se concluir que quanto maior for a base de treinamento do classificador mais eficiente será o modelo.



CONCLUSÕES

Nesse trabalho foi discutido o uso de técnicas de processamento de imagem e inteligência computacional, introduzindo o classificador semi-supervisionado DRB com objetivo de analisar sua eficácia na classificação de defeitos comuns em superfícies de trilhos. Para o pré-processamento das imagens foi utilizado a DCNN VGG19 pré-treinada que se mostrou bastante eficiente na extração de atributos.

Se comparado com os classificadores supervisionados, o classificador semisupervisonado DRB apresentou uma maior eficiência nos testes. Além disso o modelo apresentado nesse trabalho possui outras vantagens não contempladas nos métodos supervisionados, como processo de aprendizado de fácil interpretação por especialista; possibilidade de ser treinado online ou offline; capacidade de classificar imagens fora da amostra; capacidade de aprender novas classes se tornando um processo autoevolutivo. Com isso o classificador DRB é capaz de aprender outros defeitos de superfícies de trilhos como defeitos de solda e patinação de veículos ferroviários.

No contexto atual, a detecção e classificação de todos os tipos de defeitos em superfícies de trilhos é frequentemente realizado, gerando grades custos e tempo. Então vale ressaltar que a utilização de sistemas inteligentes pode auxiliar nos processos de tomada de decisão trazendo mais agilidade e eficiência ao processo. Nesse quesito o classificador DRB apresenta ser uma alternativa bem atraente para o diagnóstico e classificação de falhas em trilhos ferroviários de forma rápida e eficiente, reduzindo os custos e tempos gastos com inspeção.



Como trabalho futuro está a criação de um sistema capaz de ser integrado ao veículo de inspeção de via RIV utilizado pela empresa MRS Logística S.A, o sistema dará suporte as equipes de manutenção identificando e classificando possíveis falhas na superfície de trilhos ferroviários. Além disso pretendemos melhorar o processo, através de pesquisas de técnicas mais eficientes para o pré-processamento e classificação de imagens.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PEDRONI, P. G. (2008). ANÁLISE DE FALHAS DAS FRATURAS DE TRILHOS: O CASO DA FERROVIA. INSTITUTO MILITAR DE ENGENHARIA, Rio de Janeiro.
- [2] ANTF ASSOCIAÇÃO NACIONAL DOS TRANSPORTADORES FERROVIÁRIOS.

 Informações Gerais. Disponível em: http://www.antf.org.br/informacoes-gerais/>.

 Acesso em: 03 dez 2017.
- [3] M. Molodova, Z. Li, A. Nunez, and R. Dollevoet, "Automatic detection of squats in railway infrastructure," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1980–1990, 2014.
- [4] M. Oregui, M. Molodova, A. N´u˜nez, R. Dollevoet, and Z. Li, "Experimental investigation into the condition of insulated rail joints by impact excitation," Experimental Mechanics, vol. 55, no. 9, pp. 1597–1612, 2015.
- [5] Molodova, M., Li, Z., Nunez, A., Dollevoet, R.: Automatic detection of squats in railway infrastructure. IEEE Transactions on Intelligent Transportation Systems 15(5) (Oct. 2014) 1980–1990.



- [6] Liu, Z., Wang, W., Zhang, X., Jia, W.: Inspection of rail surface defects based on image processing. in Proc. International Asia Conference on Informatics in Control, Automation and Robotics 1 (Mar. 2010) 472–475.
- [7] Hu, G., Xiong, L., Tang, J.: Heavy rail surface defects detection based on the morphology of multi-scale and dual-structure elements. in Proc. Chinese Automation Congress (Nov. 2015) 2126–2129.
- [8] Jie, L., Siwei, L., Qingyong, L., Hanqing, Z., Shengwei, R.: Real-time rail head surface defect detection: A geometrical approach. in Proc. IEEE International Symposium on Industrial Electronics (Jul. 2009) 769–774.
- [9] Paiva, Cassio Eduardo Lima de. Super e Infraestrutura de Ferrovias, 1. ed. Rio de Janeiro: Elsevier. 2016. 313 p.
- [10] P.P. Angelov, X. Gu, MICE: Multi-layer multi-model images classifierensemble, IEEE International Conference on Cybernetics (2017) 436–443.
- [11] P. Angelov, X. Gu, A cascade of deep learning fuzzy rule-based image classifier and SVM, International Conference on Systems Man and Cybernetics (2017)1–8.
- [12] X. Gu, P. Angelov, C. Zhang, P. Atkinson, A massively parallel deep rule-basedensemble classifier for remote sensing scenes, IEEE Geosci. Remote Sens. Lett.15 (3) (2018) 345–349.
- [13] P.P. Angelov, X. Gu, J. Principe, A generalized methodology for data analysis, IEEE Trans. Cybern. (2017).



- [14] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
- [15] P. Angelov, Autonomous Learning Systems: From Data Streams to Knowledgein Real Time, John Wiley & Sons, Ltd., 2012.
- [16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scaleimage recognition, International Conference on Learning Representations (2015) 1–14.
- [17] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, AID: A benchmarkdataset for performance evaluation of aerial scene classification, IEEE Trans.Geosci. Remote Sens. 55 (7) (2017) 3965–3981.
- [18] P. Angelov, R. Yager, A new type of simplified fuzzy rule-based system, Int. J.Gen. Syst. 41 (2) (2011) 163–185.
- [19] Mateen, Muhammad, et al. "Fundus image classification using VGG-19 architecture with PCA and SVD." *Symmetry* 11.1 (2019): 1.
- [20] X. Gu, P.P. Angelov, D. Kangin, J.C. Principe, A new type of distance metric andits use for clustering, Evol. Syst. 8 (3) (2017) 167–178.
- [21] H. Bottom, "No.29 hatfield train crash", New Civil Engineer, 2012.
- [22] "Four dead in hatfield rail crash", BBC News, 2000.
- [23] "Rail passengers lost 3.9m hours in 2018, which? report says" BBC News, 2019.
- [24] Gu, Xiaowei, and Plamen P. Angelov. "Semi-supervised deep rule-based approach for image classification." Applied Soft Computing 68 (2018): 53-68.



- [25] "CNN Como fazer uma Transferência de Estilo facilmente", médium, 2019.
- [26] "auto-encoder-in-biology", mc.ai, 2019.