






# ANÁLISE DE DIFERENTES ALGORITMOS DE DEEP-LEARNING PARA IDENTIFICAÇÃO DE DEFEITOS NA SUPERFÍCIE DE PAVIMENTOS

## IDENTIFICATION OF ASPHALT PAVEMENTS DISTRESSES BY DIFFERENT DEEP-LEARNING ALGORITHMS

Karen Amanda Barbosa da Silva<sup>1</sup>   
Gabriel Torresin de Oliveira Gardin<sup>2</sup>   
Vitor Hugo Salviatto<sup>3</sup>   
Thiago Vinícius Louro<sup>4</sup>   
Heliana Barbosa Fontenele<sup>5</sup> 

**Resumo:** A avaliação da condição do pavimento ocorre principalmente através de métodos trabalhosos e demorados. Automatizar as avaliações por meio da visão computacional pode garantir maior segurança, eficiência e produtividade aos avaliadores. O objetivo desta pesquisa é comparar diferentes arquiteturas de detecção de objetos na avaliação de defeitos em pavimentos flexíveis urbanos. Para isso, foram utilizadas as arquiteturas *You Only Look Once*, *Single Shot Detection*, *Faster Region-based Fully Convolutional Neural Networks* e *Faster Region-based Fully Convolutional Neural Networks* para detectar os defeitos do tipo buraco, trincas longitudinais e transversais, trincas por fadiga e remendos. A análise foi realizada comparando o número de defeitos detectados correta e incorretamente. A arquitetura *Single Shot Detection* foi a mais precisa na identificação da deterioração analisada, com 286 defeitos identificados corretamente. O modelo *You Only Look Once* foi o segundo mais eficiente, com 77 defeitos identificados corretamente. O algoritmo *Region Proposal Network*, presente nos modelos *Faster Region-based Fully Convolutional Neural Networks* e *Faster Region-based Fully Convolutional Neural Networks*, não foi capaz de aprender a generalizar regiões de interesse a partir das imagens, o que dificultou a identificação de deteriorações. Concluiu-se que a utilização de algoritmos de aprendizagem profunda para identificação de defeitos em pavimentos flexíveis é viável, com modelos arquitetônicos eficientes, capazes de identificar diferentes tipos de deterioração.

**Palavras-chave:** Sistema de Gerência de Pavimentos. Avaliação de condição. Detecção de objetos. imagem.

**Abstract:** The pavement condition's assessment occurs mainly through laborious and time-consuming methods. Automating evaluations through computer vision can ensure greater safety for evaluators, efficiency, and productivity. The objective of this research is to compare different object detection architectures in the evaluation of distresses in urban flexible pavements. For this, the architectures *You Only Look Once*, *Single Shot Detection*, *Faster Region-based Fully Convolutional Neural Networks* and *Faster Region-based Fully Convolutional Neural Networks*

<sup>1</sup> Graduado, Universidade Estadual do Paraná e Construção e karen\_amanda@outlook.com.

<sup>2</sup> Graduado, Universidade Estadual do Paraná e gabriel.torresin@uel.br.

<sup>3</sup> Mestrado, Universidade Estadual do Paraná e vitorhsalviatto@hotmail.com.

<sup>4</sup> Mestrado, Universidade Estadual do Paraná e tvinicius.louro@uel.br.

<sup>5</sup> Doutorado, Universidade Estadual de Londrina e heliana@uel.br.

were used to detect Pothole, longitudinal and transversal crack, alligator crack and patches. The analysis was performed by comparing the number of distresses detected correctly and incorrectly. The Single Shot Detection architecture was the most accurate in identifying the deterioration analyzed, with 286 distresses correctly identified. The You Only Look Once model was the second most efficient, with 77 distresses identified correctly. The Region Proposal Network algorithm, present in the Faster Region-based Fully Convolutional Neural Networks and Faster Region-based Fully Convolutional Neural Networks models, was unable to learn to generalize regions of interest from the images, which hindered the identification of deteriorations. It was concluded that the use of deep learning algorithms to identify distresses in flexible pavements is feasible, with efficient architectural models, capable of identifying different deterioration types.

**Keywords:** Pavement management system. Condition rating. Object Detection. Image.

# 1 INTRODUÇÃO

Investimentos em infraestrutura de transportes são um dos principais gastos públicos em muitos países do mundo. No Brasil, o sistema rodoviário é o principal meio para locomoção de cargas e de pessoas. Atualmente, o país conta com uma rede de 1.735.607 quilômetros de estradas e rodovias, o que representa 62,1% de toda matriz de transportes nacional (ZANCHETTA, 2017).

O Sistema de Gerência de Pavimentos (SGP) é o responsável por melhorar a manutenção da rede e o uso eficaz do dinheiro público. Para isso, é fundamental que toda a rede rodoviária seja constantemente monitorada. Porém, ainda são poucas as cidades que possuem um sistema de gerenciamento de pavimentos urbanos (SGPU), com constante monitoramento da qualidade do pavimento.

As principais técnicas empregadas para avaliação da condição dos pavimentos ainda são técnicas manuais, na qual técnicos fazem o levantamento no campo ou analisam imagens dos pavimentos, a fim de detectar os defeitos, suas densidades e severidades. Tais métodos, além de serem muito laboriosos, são também suscetíveis a erros humanos e variam de acordo com a interpretação de cada avaliador (BIANCHINI; BANDINI; SMITH, 2010).

Assim, metodologias automatizadas e robustas para avaliar a qualidade da superfície do pavimento são cada vez mais necessárias e, portanto, alvo de estudo de grande interesse da comunidade acadêmica mundial, pois podem garantir: maior segurança para a equipe de coleta de dados, maior eficiência e produtividade de avaliações (KARGAH-OSTADI *et al.*, 2017).

Dentre as diversas metodologias, muitos autores fazem uso de técnicas de deep-learning e visão computacional para desenvolver modelos de inteligência artificial capazes de identificar e quantificar defeitos nos pavimentos.

Dentro do campo de deep-learning existem diversos algoritmos computacionais capazes de realizar a chamada detecção de objetos, que consiste em determinar a presença e a localização de objetos em imagens. Dentre os principais algoritmos pode-se destacar quatro arquiteturas de modelos: YOLO (*You Look Only Once*), SSD (*Single Shot Detector*), Faster-

RCNN (*Faster Region-based Fully Convolutional Neural Networks*) e Faster-RCNN (*Faster Region-based Fully Convolutional Neural Networks*). Tais algoritmos são corriqueiramente comparados de acordo com sua precisão e custo computacional (HUANG *et al.*, 2011).

Tais arquiteturas podem, dentro da gerência de pavimentos urbanos, ser utilizadas para o desenvolvimento de modelos de *deep-learning* capazes de reconhecer os mais diversos tipos de defeitos que ocorrem nos pavimentos. Porém, em muitos casos esses defeitos são pouco visíveis e/ou não possuem forma definida, o que torna a tarefa de identificação por técnicas de *deep-learning* um desafio.

Desta forma, a proposta deste trabalho foi analisar diferentes algoritmos de detecção de objetos a fim de avaliar quais algoritmos são os mais eficientes para detecção simultânea de diferentes tipos de defeitos e no auxílio às avaliações da condição de pavimentos flexíveis urbanos.

## **2. SISTEMA DE GERÊNCIA DE PAVIMENTOS**

Um SGP tem como objetivo possibilitar a melhoria das condições dos pavimentos e a redução dos custos de manutenção e reabilitação, de forma a proporcionar pavimentos seguros, confortáveis e econômicos e um melhor retorno possível dos recursos disponíveis (FERNANDES JÚNIOR, 2001).

A avaliação da condição do pavimento tem como função auxiliar os gestores de um SGP. A partir dos dados levantados é que são definidas as condições funcionais, estruturais e operacionais do pavimento DNIT (2011). Existem dois métodos principais de avaliação: o subjetivo e o objetivo. Por meio das avaliações subjetivas é obtido um valor para a condição do pavimento com base na opinião do usuário. Já as avaliações objetivas consistem na mensuração e classificação dos defeitos que ocorrem na superfície do pavimento a fim de auxiliar na identificação da melhor técnica de restauração ou reabilitação do pavimento.

Ambos os métodos dependem de uma inspeção manual, sendo um processo trabalhoso e demorado, devido a grande extensão da malha viária, o

que pode gerar inconsistências devido ao cansaço dos avaliadores (SULTANI; MOKHTARI; YUN, 2018).

De acordo com Kargah-Ostadi *et al.* (2017), o desenvolvimento e implementação bem-sucedidos de softwares de detecção automatizada de defeitos com base em visão computacional, gerariam as seguintes vantagens na obtenção dos dados e no SGP: maior segurança da equipe de coleta de dados; maior eficiência e produtividade das avaliações; maior objetividade na classificação de defeitos; maior exatidão e precisão na medição do defeito e melhores decisões em atividades de manutenção e reabilitação.

### **3. DETECÇÃO DE OBJETOS**

A visão computacional é um processo que estuda o desenvolvimento de métodos e técnicas que possibilitam a um sistema computacional interpretar imagens, ou seja, a forma como um computador enxerga o meio à sua volta, extraindo informações significativas a partir de imagens capturadas por câmeras de vídeo, sensores, scanners, entre outros dispositivos (MILANO; HONORATO, 2023).

Essa técnica de visão computacional pode ser aplicada em quatro áreas de conhecimento: reconhecimento, análise de movimento, reconstrução 3D e restauração de imagens. Neste estudo utilizou-se a área do reconhecimento, que consiste na aplicação de algoritmos para detecção, identificação e classificação de objetos em uma imagem.

Atualmente, as redes neurais convolucionais (CNN) são os algoritmos de reconhecimento mais utilizados, pois elas têm como base técnicas de *deep learning* que auxiliam no processamento de imagens, de modo que são muito eficientes na identificação e classificação de padrões (DE OLIVEIRA, 2019).

Existem várias arquiteturas de modelo de CNN. Neste trabalho foram analisadas as arquiteturas Faster-RCNN, Faster-RFCN, YOLO e SSD, cujas definições são dadas a seguir.

#### 4. ARQUITETURAS DE MODELOS

A arquitetura Faster-RCNN consiste em uma rede neural convolucional de região mais rápida, com modelos complexos para detecção e classificação de objetos com alto desempenho. Nesse modelo utiliza-se um algoritmo de geração de proposta de regiões (*Region Proposal Network* - RPN) que seleciona regiões com maior probabilidade de conterem objetos, as chamadas *Regions of Interest* (RoI). Essas regiões de interesse são então passadas para uma rede neural convolucional que determina as classes de objetos presentes e a posição das caixas delimitadoras (*Bounding boxes*) (MAJIDIFARD *et al.*, 2019)

O Faster-RFCN é uma arquitetura proposta por Dai *et al.* (2016). Trata-se de uma rede totalmente convolucional baseada em região para detecção precisa e eficiente de objetos. Esta arquitetura, derivada do Faster-RCNN, é mais eficiente, pois otimiza a forma com que as regiões de interesse, geradas pelo RPN, são formadas. Diferentemente do Faster-RCNN que recorta características da mesma camada onde as propostas de regiões (RoI) são previstas, o Faster-RFCN as retira da última camada de características, antes da previsão, o que minimiza a quantidade de análise computacional por região que deve ser executada. Com isso, esse modelo pode gerar resultados com precisão próximos ao do Faster-RCNN, mas com velocidade de execução mais rápida (HUANG *et al.*, 2011; MAEDA *et al.*, 2018).

O *Single Shot Detector* (SSD), como o próprio nome diz, é uma estrutura de detecção em apenas uma única etapa, do inglês “one shot”. Esta arquitetura não faz extração de regiões de interesse como as R-CNN e R-FCN, o que a torna um modelo mais rápido. O SSD utiliza um filtro convolucional para avaliar um conjunto de caixas delimitadoras e ver a probabilidade da caixa pertencer à classe de objeto. O modelo compara a similaridade das caixas originais com as previstas com base no método estatístico *Intersection our Union* (IoU) (DE OLIVEIRA, 2019).

Por fim, a arquitetura YOLO (*You Only Look Once*), assim como o SSD, é uma estrutura de detecção em uma única etapa, que não possui o algoritmo RPN. Por meio dessa arquitetura pode-se prever a região e a classe de objetos

com um único CNN. O YOLO tem como vantagem sua velocidade de processamento, muito mais rápida que as demais arquiteturas, pois utiliza técnicas de regressão para detectar objetos. Esse algoritmo funciona de maneira similar ao SDD, contudo, utiliza um método próprio de inferência nas caixas delimitadoras (MAEDA *et al.*, 2018).

## 5. ESTUDOS RELACIONADOS

Atualmente, diversos estudos têm sido realizados de modo a elaborar métodos para automatizar o levantamento de defeitos na superfície de pavimentos. Dentre esses estudos pode-se citar o trabalho de Koch e Brilakis (2011) que desenvolveram um método para detectar buracos em pavimentos por meio de uma câmera de alta velocidade instalada no local da câmera de ré de um carro. O modelo foi treinado e testado com base em um protótipo no *Matrix Laboratory* (MATLAB) por meio da análise de 120 imagens. De acordo com os autores, o modelo desenvolvido obteve 86% de acurácia e 82% de precisão.

Na pesquisa de Ukhwah, Yuniarno e Suprpto (2019) foram comparados três tipos de arquitetura YOLO, a saber: YOLOv3, YOLOv3 Tiny e YOLOv3 SPP, a fim de identificar qual modelo seria mais preciso para detectar buracos na superfície da estrada. Os resultados mostraram acurácia de 83,43%, 79,33% e 88,93% para YOLOv3, YOLOv3 Tiny e YOLOv3 SPP, respectivamente, concluindo que os três tipos analisados apresentaram resultados satisfatórios, sendo YOLOv3 SPP o mais preciso.

Outro exemplo de estudo nessa área são os trabalhos de Huang *et al.* (2011) e Redmon e Farhadi (2018) que analisaram a eficiência dos modelos SSD e YOLO, os autores concluíram que o modelo produzido pelo SSD é mais eficaz que o produzido pelo YOLO, uma vez que o principal diferencial do YOLO não é sua precisão, mas sim sua velocidade de execução que pode ser até 3 vezes mais rápido que o SSD.

Um banco de dados composto por 9.053 imagens de defeitos nos pavimentos em sete municípios do Japão foi desenvolvido por Maeda *et al.*, (2018). As imagens foram obtidas com um smartphone acoplado no painel de

um carro. Nesse trabalho a detecção dos defeitos foi testada por duas arquiteturas, a SSD *inception* e a SSD *mobileNet*.

Neste estudo foram verificadas oito classes, onde cada classe correspondia a um grupo de defeitos, compostos por sete tipos de trincas, buracos e faixas de sinalização. Os autores não realizaram a diferenciação entre os tipos de trincas, podendo ter a mesma trinca em mais de uma classe. Maeda *et al.* (2018) concluíram que o modelo SSD *mobileNet* foi o mais preciso, com um coeficiente de precisão maior que 75%. Os autores disponibilizaram publicamente a base de dados, os resultados obtidos e o aplicativo para smartphone, a fim de desenvolver um banco de dados uniforme e métodos de referência para outra pesquisa.

Em seus estudos Destri Junior *et al.* (2019) criaram um modelo para identificar e estimar o percentual de trincas em pavimentos por meio de vídeos e com base na textura e na detecção de objetos com o auxílio de CNN. Esse modelo foi avaliado com um banco de dados composto por 138.909 imagens. Os autores concluíram que o método é aceitável, pois, obteve-se uma acurácia de 90% e uma precisão 75%. Apesar disso, os autores ainda ressaltam que houve muita dificuldade em se detectar trincas em ambientes sombreados, assim como na pesquisa de (ZHANG *et al.*, 2016).

Outro estudo desenvolvido nessa área foi o de Majidifard *et al.* (2019) que compararam duas arquiteturas de modelo, o YOLO.v2 e o Faster-RCNN por meio de 7.237 imagens obtidas no *Google Street View*. Os autores concluíram que ambas as arquiteturas atingiram resultados aceitáveis. Contudo, o YOLO.v2 foi o mais preciso, apresentando um coeficiente de precisão de 84%, enquanto o Faster-RCNN obteve 65% de precisão. Entretanto, o modelo proposto não quantificava a densidade dos defeitos no pavimento.

Em um novo estudo, Majidifard, Adu-Gyamfi e Buttlar (2020) desenvolveram um novo modelo para avaliar, por meio de imagens, a densidade dos defeitos observados no pavimento. As arquiteturas YOLO e U-Net foram utilizadas para desenvolver quatro modelos de previsão de condição de pavimentos. Os modelos foram treinados com as imagens de Majidifard *et al.* (2019). Após o desenvolvimento, os resultados dos modelos foram comparados

com os resultados do *Pavement Surface Evaluation and Rating* (PASER), método que utiliza a inspeção visual para avaliar a condição dos pavimentos. Os autores observaram que os modelos possuíam altos valores de correlação. Além disso, os modelos propostos mostraram-se robustos, flexíveis, econômicos e capazes de prever a condição de pavimentos.

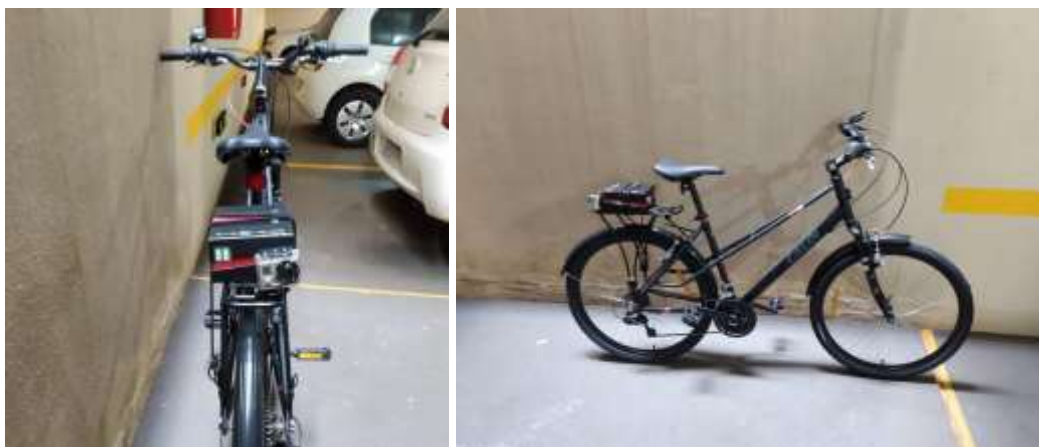
## 6. MATERIAIS E MÉTODOS

Os procedimentos adotados neste estudo foram divididos em 3 etapas: *i)* obtenção do banco de dados, *ii)* treinamento dos modelos e *iii)* teste dos modelos.

### 6.1. Obtenção de banco de dados

As imagens de pavimentos utilizadas neste trabalho foram obtidas de três formas: *i)* nas vias de um campus universitário com uma câmera GoPro acoplada no protótipo desenvolvido por Louro *et al.* (2021) (Figura 1), estando a câmera posicionada de forma que sua inclinação permitisse visualizar a largura total das faixas de tráfego, como apresentado na Figura 2; *ii)* do banco de dados desenvolvido por Maeda *et al.* (2018) e *iii)* em algumas vias do município com um smartphone, para complementar o banco de imagens, principalmente com imagens com os defeitos menos ocorrentes nas vias do campus.

**Figura 1** – Bicicleta com o protótipo e a câmera instalados



**Fonte:** Próprios autores

**Figura 2** - Visualização da via pela câmera GoPro



**Fonte:** Próprios autores

Na Figura 3 está representado uma das imagens selecionadas do banco de dados de Maeda *et al.* (2018).

Neste estudo foram verificados os cinco principais defeitos que ocorrem em pavimentos flexíveis urbanos: trincas por fadiga, trincas longitudinais, trincas transversais, buracos e remendos.

**Figura 3** - Imagem capturada por um smartphone dentro de um carro



**Fonte:** Maeda et al. (2018)

Assim, neste trabalho foram utilizadas um total de 2.019 imagens de pavimentos com a presença de defeitos na superfície, de modo que as 1.800 imagens obtidas nos itens *i* e *ii* foram empregadas no treinamento dos modelos e as 219 imagens obtidas no item *iii* foram usadas na fase de teste.

## 6.2. Treinamento dos Modelos

Primeiramente, para o treinamento dos modelos foi necessário identificar os defeitos em cada uma das imagens do banco de dados e, em seguida, realizar a anotação dos defeitos. A anotação é feita por meio da identificação do local do defeito na imagem e o seu tipo. Para isso, utilizou-se o site *makesense.ai*, que fornece um ambiente gráfico de fácil uso para anotação de imagens. Todas as anotações seguiram o formato Pascal VOC, um formato de anotação de imagens que gera um arquivo XML para cada imagem anotada. Estes arquivos contêm a posição e os tipos dos defeitos presentes nas imagens.

Para o treinamento dos modelos foi utilizado a biblioteca *TensorFlow*, uma biblioteca de código aberto desenvolvido pelo Google para aprendizado de máquina (*Machine Learning*). Também se desenvolveu um pequeno código em Python que automatiza algumas das etapas necessárias para utilização das Interfaces de Programação de aplicativos (*Application Programming Interface - APIs*) de detecção de objetos do *TensorFlow*. Todos os modelos analisados foram treinados com as mesmas imagens.

A fim de diminuir o tempo necessário para o treinamento dos modelos, foi utilizado o *Google Colab*, uma ferramenta de computação na nuvem fornecida gratuitamente pelo Google. Os servidores do *Google Colab* são equipados com boas placas de vídeos, e por isso são uma ótima alternativa para treinar os modelos de *Deep-Learning*.

## 6.3. Teste dos modelos

Os modelos de detecção de objetos analisados foram YOLO, SSD, Faster-RCNN e Faster-RFCN. Os quatro modelos foram testados com as 219 imagens de pavimentos do município, todas contendo ao menos um dos quatro defeitos analisados. Para isso foi desenvolvido um código em *Python* que utiliza os modelos treinados para desenhar retângulos em volta dos defeitos identificados nas imagens de teste e então cria uma imagem com os defeitos destacados pelos retângulos. Esta etapa, por não ser computacionalmente

custosa, não foi executada nos servidores do Google Colab, mas sim em um computador normal.

Como resultado, foram obtidas 219 imagens para cada um dos modelos testados. As eficiências dos modelos foram comparadas por meio da análise do número de defeitos corretamente identificados e no número de defeitos identificados de forma errônea (seja por identificar um defeito inexistente ou por identificar um defeito com o tipo errado). Os defeitos em duplicata não foram considerados como detecções errôneas, apenas foram contabilizados uma vez.

Em seguida, de modo a verificar o funcionamento do algoritmo RPN presente nos modelos Faster-RCNN e Faster-RFCN, foram observados os gráficos de evolução do treinamento. Assim, os gráficos obtidos durante o treinamento das arquiteturas Faster-RCNN e Faster-RFCN dizem respeito à eficácia do algoritmo RPN. Esses gráficos são denominados como “Perda de objetividade RPN” (*RPN Objectness Loss*) e “Perda de localização RPN” (*RPN Localization Loss*). O primeiro diz respeito ao erro do algoritmo RPN em detectar possíveis objetos nas imagens e o segundo consiste no erro do algoritmo RPN em propor a posição dos possíveis objetos.

## 7. APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Para avaliação da eficiência das arquiteturas foi analisado o número de defeitos identificados de forma correta e incorreta por cada modelo nas 219 imagens utilizadas na fase de teste. Os resultados obtidos são apresentados na Tabela 1.

**Tabela 1:** Detecções corretas e incorretas dos defeitos

Arquitetura	SSD	YOLO	FASTER- RCNN	FASTER- RFCN
Detecções corretas	286	77	19	0
Detecções incorretas	1	0	3	0

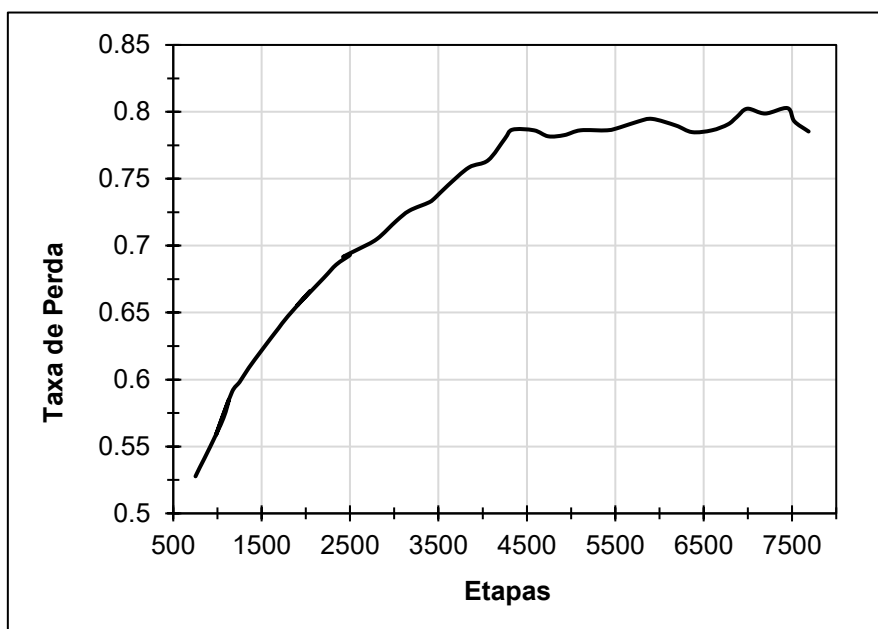
**Fonte:** Próprios autores

O modelo produzido pela arquitetura SSD foi capaz de identificar corretamente 286 defeitos presentes nas imagens de teste, enquanto o modelo com base na arquitetura YOLO, que apresentou o segundo maior número de detecções corretas, contabilizou apenas 77, resultado 3,7 vezes menor. Como reportado nos trabalhos de Huang *et al.* (2011) e de Redmon e Farhadi (2018) esperava-se que o modelo produzido pelo SSD fosse mais eficaz que o produzido pelo YOLO.

Já os modelos de arquiteturas Faster-RCNN e Faster-RFCN identificaram 19 e 0 defeitos, respectivamente. Este resultado não foi condizente com o esperado, uma vez que os dois modelos, no trabalho de Huang *et al.* (2011) foram mais precisos. Para entender o motivo por trás deste resultado inesperado analisou-se o funcionamento interno de cada um dos detectores.

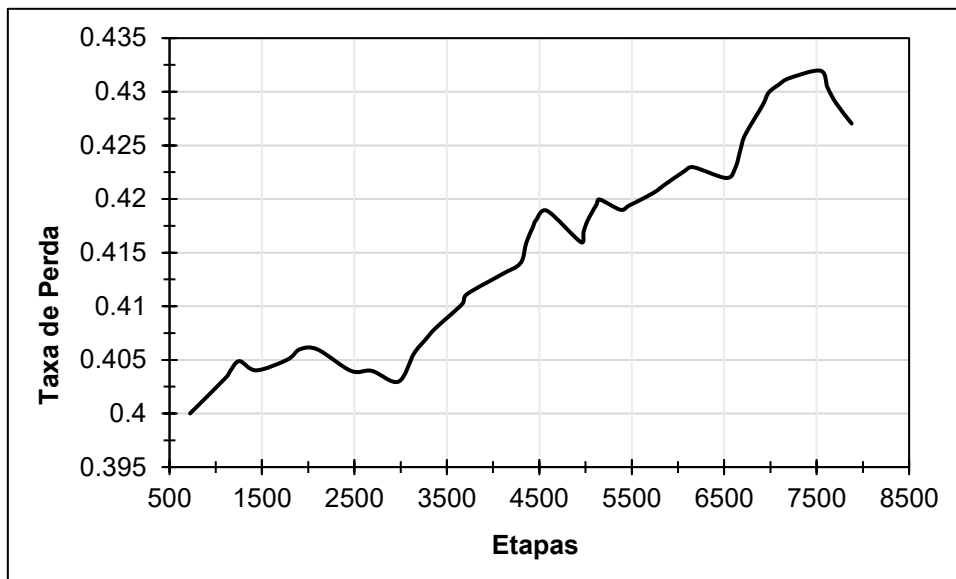
De modo que o algoritmo RPN era a principal diferença entre as arquiteturas, foi verificado o funcionamento do algoritmo. Nas Figuras 4 e 5 são apresentados os gráficos com as curvas de aprendizagem para a arquitetura Faster-RCNN.

**Figura 4 - Perda de objetividade RPN para Faster-RCNN**



**Fonte:** Próprios autores

**Figura 5 - Perda de localização RPN para Faster-RCNN**

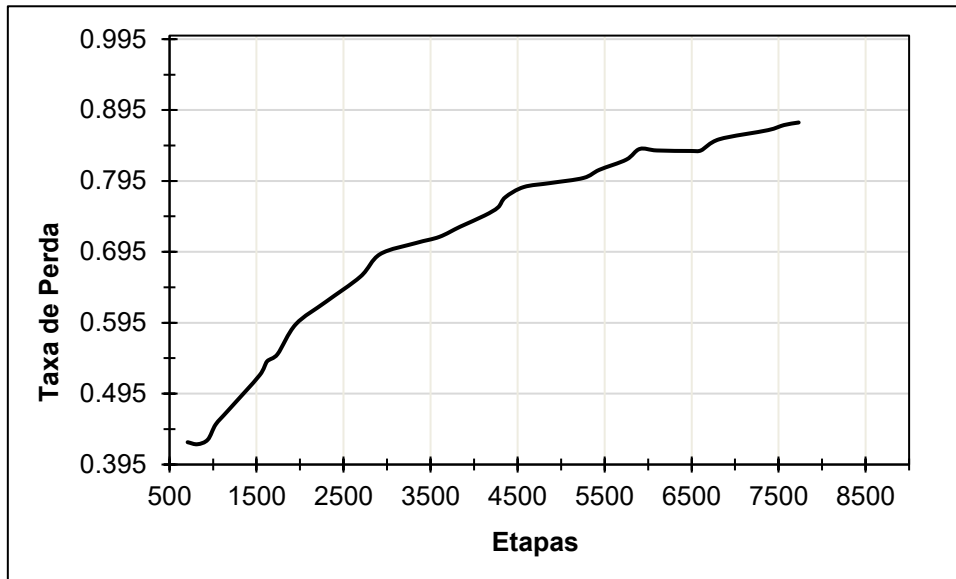


**Fonte:** Próprios autores

Ao analisar os gráficos das imagens 4 e 5 pode-se observar que, conforme o treinamento do modelo Faster-RCNN avançou, o erro do algoritmo RPN aumentou, ou seja, o algoritmo não foi capaz de aprender a generalizar regiões de interesse a partir das imagens de treinamento. Se o algoritmo RPN é incapaz de gerar boas regiões de interesse, espera-se que a etapa de detecção também seja fortemente impactada, como de fato ocorreu.

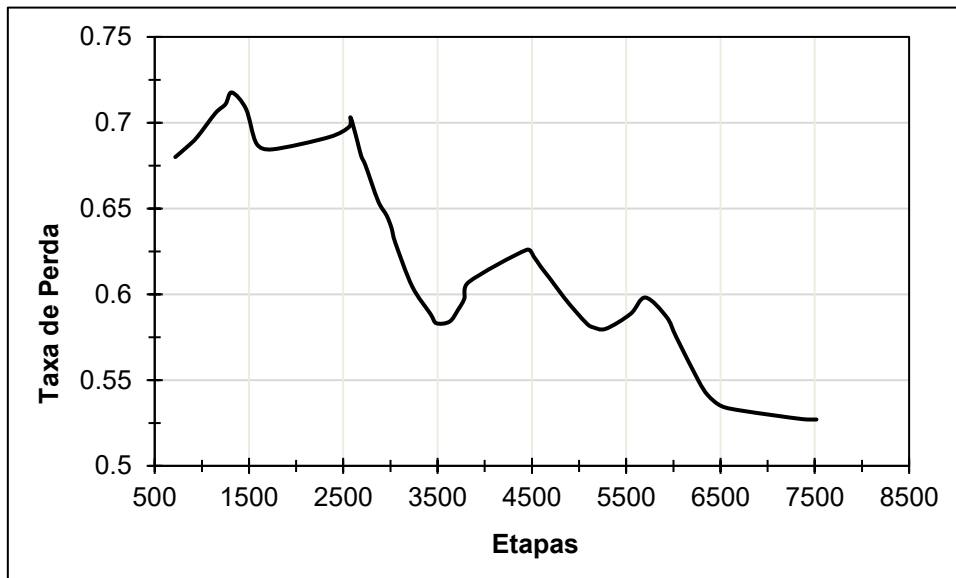
Nas Figuras 6 e 7 são expostos os gráficos referentes à arquitetura Faster-RFCN. Para os gráficos obtidos para a arquitetura Faster-RFCN, os resultados parecem um pouco mais promissoras, uma vez que o erro do RPN de perda de localização diminuiu à medida que o treinamento avançou. Em contrapartida, o erro do RPN de perda de objetividade para esta arquitetura foi ainda maior do que para o Faster-RCNN, de modo a alcançar, no final do treinamento, um valor muito próximo de 1.

**Figure 6 - Perda de objetividade RPN para Faster-RFCN**



Fonte: Próprios autores

**Figura 7 - Perda de localização RPN para Faster-RFCN**



Fonte: Próprios autores

## 8. CONCLUSÕES

Com base no exposto, pode-se concluir que a arquitetura SSD foi a mais eficiente na identificação dos quatro defeitos analisados, obtendo-se 286 defeitos corretamente identificados, um número 3,7 vezes maior que o modelo YOLO, o segundo mais preciso, com 77 defeitos identificados corretamente.

As arquiteturas Faster-RCNN e Faster-RFCN obtiveram um rendimento muito abaixo do esperado, identificando poucos defeitos em comparação as outras arquiteturas ou até mesmo nenhum defeito, como foi o caso do Faster-RFCN. Nesses modelos, o algoritmo RPN não foi capaz de gerar boas regiões de interesse com base nas imagens de treinamento, o que acabou interferindo na detecção dos defeitos.

Por fim, pôde-se concluir que obtenção das imagens com a câmera acoplada a uma bicicleta proporcionou imagens de boa qualidade para o satisfatório reconhecimento dos defeitos e que a utilização de algoritmos de *deep learning* para identificação de defeitos em pavimentos flexíveis é viável, havendo modelos de arquitetura eficientes, capazes de identificar diferentes tipos. Ainda, com o avanço das tecnologias de visão computacional, pode-se futuramente desenvolver algoritmos capazes não só de identificar, mas de quantificar os defeitos e gerar índices de condição de pavimentos flexíveis com base em análises computacionais.

## **AGRADECIMENTOS**

## **REFERÊNCIAS**

BIANCHINI, A.; BANDINI, P.; SMITH, D. W. Interrater Reliability of Manual Pavement Distress Evaluations. **Journal of Transportation Engineering**, v. 136, n. 2, p. 165–172, 2010.

DAI, J. et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks. **Computer Science - Computer Vision and Pattern Recognition**, 2016.

DE OLIVEIRA, D. T. **Um sistema inteligente que prevê a produtividade do algodão em imagens de lavouras comerciais**. Dissertação (mestrado)—Jaboticabal: Universidade Estadual Paulista (Unesp), 2019.

DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES (DNIT). **Manual de gerência de pavimentos**. Rio de Janeiro: IPR, 2011.

DESTRI JUNIOR, J. et al. **Detecção e quantificação automatizadas de trincas em pavimentos de rodovias**. 33º Congresso de Pesquisa e Ensino em Transporte da ANPET. **Anais...**Balneário Camboriú: ANPET, nov. 2019.

FERNANDES JÚNIOR, J. L. **Sistemas de gerência de pavimentos urbanos para cidades de médio porte**. [s.l.] Universidade de São Paulo, 2001.

HUANG, J. et al. **Speed/accuracy trade-offs for modern convolutional object detectors**. Proceedings of the IEEE conference on computer vision and pattern recognition. **Anais...**2011. Acesso em: 22 maio. 2023

KARGAH-OSTADI, N. et al. Evaluation Framework for Automated Pavement Distress Identification and Quantification Applications. **Transportation Research Record: Journal of the Transportation Research**, v. 2639, p. 46–54, 2017.

KOCH, C.; BRILAKIS, I. Pothole detection in asphalt pavement images. **Advanced Engineering Informatics**, v. 25, p. 507–515, 2011.

LOURO, T. V. et al. **Is There Space For Cyclists? Understanding the Impact of Cycle Lanes on Passing Distance**. Transportation Research Board 100th Annual Meeting. **Anais...**Washington DC: Transportation Research Board, jan. 2021. Acesso em: 22 maio. 2023

MAEDA, H. et al. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. **Computer-Aided Civil and Infrastructure Engineering**, v. 33, p. 1127–1141, 2018.

MAJIDIFARD, H. et al. Pavement Image Datasets: A New Benchmark Dataset to Classify and Densify Pavement Distresses. **Transportation Research Record**, v. 2674, p. 328–339, 2019.

MAJIDIFARD, H.; ADU-GYAMFI, Y.; BUTTLAR, W. G. Deep machine learning approach to develop a new asphalt pavement condition index. **Construction and Building Materials**, v. 247, p. 118513, 30 jun. 2020.

MILANO, D.; HONORATO, L. B. **Visão Computacional**. Disponível em: <<https://docplayer.com.br/3058305-Visao-computacional-danilo-de-milano-luciano-barrozo-honorato-unicamp-universidade-estadual-de-campinas-ft-faculdade-de-tecnologia.html>>. Acesso em: 22 maio. 2023.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. **ArXiv**, v. abs/1804.02767, 2018.

SULTANI, W.; MOKHTARI, S.; YUN, H.-B. Automatic Pavement Object Detection Using Superpixel Segmentation Combined With Conditional Random Field. **IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS**, v. 19, n. 7, p. 2076–2085, 2018.

UKHWAH, E. N.; YUNIARNO, E. M.; SUPRAPTO, Y. K. **Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network**. 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA). **Anais...**2019.

ZANCHETTA, F. **Sistema de gerência de pavimentos urbanos: avaliação de campo, modelo de desempenho e análise econômica**. Tese (Doutorado)—São Carlos: Universidade de São Paulo (USP) - Escola de Engenharia de São Carlos (EESC), 2017.

ZHANG, L. et al. **Road crack detection using deep convolutional neural network**. 2016 IEEE International Conference on Image Processing (ICIP). **Anais...**2016.