

DETECÇÃO DE COMPONENTES INDUSTRIAIS EM IMAGENS UTILIZANDO REDES NEURAS CONVOLUCIONAIS BASEADAS EM REGIÃO

NELSON DUTRA JUNIOR¹; ANDERSON PRIEBE FERRUGEM²; FÁBIO KELLERMAR SCHRAMM³

¹Universidade Federal de Pelotas – ndjunior@inf.ufpel.edu.br

²Universidade Federal de Pelotas – ferrugem@inf.ufpel.edu.br

³Universidade Federal de Pelotas – fabioks@ufpel.edu.br

1. INTRODUÇÃO

Em resposta à crescente necessidade de agilidade e redução de custos, a utilização de kits industrializados surge como uma alternativa para a execução de instalações hidrossanitárias. As instalações hidráulicas são parte importante de uma obra e precisam ser conferidas com exatidão.

O objetivo geral é avaliar a aplicação de modelos de rede neural convolucional baseada em região para a detecção e classificação de componentes de kits hidrossanitários industrializados. Para isso, os objetivos específicos incluem a modelagem e construção de um *dataset* próprio para treinamento e validação, bem como o desenvolvimento de modelos capazes de identificar e classificar automaticamente os componentes presentes nas imagens.

As redes neurais convolucionais (RNC) são uma arquitetura de aprendizado profundo pensadas para aplicação em dados estruturados como matrizes, como imagens. As RNCs utilizam operações de convolução com o uso de filtros que varrem os dados de entrada, permitindo o reconhecimento de padrões como bordas e texturas. Uma particularidade das RNCs é a redução dimensional dos dados de entrada mantendo características importantes, as características e padrões extraídos são levados para camadas totalmente conectadas que são usadas para fazer as previsões finais (GU et al., 2018).

As redes neurais convolucionais baseadas em região (R-CNN), apresentam bons resultados para imagens com vários objetos, para isso elas identificam regiões de interesse e utilizam RNCs para categorizá-las como fundo ou primeiro plano (BHARATI; PRAMANIK, 2020).

Como apresentado por GIRSHICK et al. (2014) as R-CNNs são compostas por 3 componentes principais, o primeiro é responsável por gerar propostas de regiões, o segundo é uma RNC que para cada região proposta extrai um vetor de características de tamanho fixo, ao fim um conjunto de classificadores lineares que classifica cada região através de seu vetor de características e a regressão linear é utilizada para delimitar o objeto.

As variantes recentes de R-CNN, como Fast R-CNN e Faster R-CNN, adotam uma camada totalmente conectada, seguida por uma função *softmax* para realizar a classificação diretamente no modelo de rede neural (REN et al., 2016).

2. METODOLOGIA

Para sua melhor execução este trabalho foi dividido em etapas, seguindo um fluxo de trabalho, são elas: coleta de dados, pré-processamento dos dados, construção dos *datasets*, *data augmentation*, treinamento dos modelos e avaliação dos resultados.

Foram coletadas 63 fotos RGB em campo e obtidas 197 imagens de fabricantes contendo componentes das 4 classes abordadas neste trabalho em diferentes ângulos e exposições. As classes a serem trabalhadas são: joelho, misturador, registro e tê.

A resolução das imagens influencia diretamente o treinamento das Redes Neurais, impactando tanto o tempo de processamento quanto a acurácia do modelo (SAPONARA; ELHANASHI, 2022). Imagens muito grandes aumentam a complexidade e o uso de memória, enquanto imagens muito pequenas podem reduzir a qualidade dos resultados. Para padronizar os dados, as imagens coletadas foram pré-processadas: redimensionadas para 800×800 pixels com corte central quando necessário, evitando distorção, e convertidas de RGB para tons de cinza.

Foram construídos 2 *datasets* no formato COCO, o primeiro visando o treinamento dos modelos e contendo 50 exemplos para cada classe, o segundo pensado para avaliação dos modelos contendo em torno de 15 exemplos para cada classe.

Com o objetivo de aumentar a disponibilidade de dados para treinamento foram aplicadas técnicas de *data augmentation*, sendo aplicadas 4 transformações por imagem aumentando o número de 50 exemplos por classe para 200 exemplos por classe, totalizando 800 exemplos para treinamento. As técnicas aplicadas foram: o espelhamento vertical com probabilidade de 50%, o espelhamento horizontal também com probabilidade de 50%, rotações fixas em 90° , 180° e 270° e rotação aleatória entre $0,5^\circ$ e 30° . O processo de *data augmentation* foi realizado pelo *framework* Detectron2 na etapa de carregamento de dados.

O Detectron2 foi utilizado como plataforma para treinar, validar e comparar os modelos de detecção, aproveitando sua flexibilidade. O Detectron2 é um *framework* de código aberto para implementação de modelos de detecção de objetos e segmentação de imagens, oferecendo arquiteturas modernas, como a Faster R-CNN. Ele fornece ferramentas integradas para treinamento, avaliação e inferência, permitindo a personalização de pipelines e facilitando a experimentação com diferentes modelos (WU et al., 2019).

Foi realizado o treinamento dos modelos pré treinados no cronograma 3x disponíveis no Detectron2 com o *dataset* construído variando os parâmetros configuráveis como: número de iterações, número de imagens por lote e a taxa de aprendizado.

Os resultados obtidos, foram avaliados para cada modelo treinado e para cada variação de parâmetros adotada, através das métricas de *average precision* (AP), acurácia, precisão e F1-Score. As métricas foram obtidas por meio da matriz de confusão gerada com os resultados de cada modelo treinado aplicado ao *dataset* de avaliação construído.

3. RESULTADOS E DISCUSSÃO

Como ponto de partida para os experimentos foram utilizados os parâmetros predefinidos na documentação do Detectron2, onde o número de iterações é definido como 300, o tamanho de lote em 2 imagens e a taxa de aprendizado em 0,00025.

No primeiro experimento, foram testados todas as variações de modelos pré treinados no cronograma 3x disponíveis para a arquitetura Faster R-CNN, mantendo os parâmetros de treinamento em seus valores padrões, a fim de

identificar a resposta dos modelos ao *dataset* criado. Na Tabela 1, são apresentados os resultados obtidos bem como os parâmetros utilizados. O modelo X101-FPN foi o que apresentou os melhores resultados para AP, acurácia e precisão.

Tabela 1. Avaliação dos modelos da arquitetura Faster R-CNN

Modelo	Iterações	Lote	Taxa de aprendizado	AP	Acurácia	Precisão	F1-Score
R50-C4	300	2	0,00025	43,82	0,80	0,82	0,81
R50-DC5	300	2	0,00025	45,56	0,75	0,62	0,68
R50-FPN	300	2	0,00025	47,25	0,87	0,84	0,86
R101-DC5	300	2	0,00025	51,78	0,76	0,88	0,82
R101-FPN	300	2	0,00025	33,83	0,57	0,74	0,59
X101-FPN	300	2	0,00025	58,18	0,87	0,88	0,87

No segundo experimento, o objetivo foi avaliar o impacto do tamanho do lote, ou seja, o número de imagens utilizadas em cada iteração do treinamento, nas métricas utilizadas. O parâmetro variado foi o tamanho do lote, testado com os valores: 1, 2, 4 e 8. Verificou-se que o aumento do número de imagens por iteração traz um impacto positivo nos resultados, mas aumenta significativamente o tempo de treinamento do modelo.

O número de iterações é um dos parâmetros mais importantes para o treinamento do modelo e, em um *dataset* prático, 300 iterações podem ser insuficientes. Assim, no terceiro experimento, o número de iterações foi aumentado para 600, 1200, 2400 e 4800. Observou-se uma evolução considerável nos resultados à medida que o número de iterações aumentou.

O último parâmetro a ser testado foi a taxa de aprendizado, que representa o quanto os pesos da rede podem ser atualizados em uma iteração. Foram testadas as seguintes taxas de aprendizado: 0,00025, 0,00050, 0,00100 e 0,01000. Verificou-se que apenas o aumento da taxa de aprendizado para 0,00050 foi eficaz, enquanto valores superiores resultaram em piora dos resultados.

Por fim, no último experimento foi treinado o melhor modelo para a arquitetura Faster R-CNN, o X101-FPN, encontrado através do experimento 1. Para os treinamentos foram utilizados os parâmetros que apresentaram os melhores resultados nos experimentos 2, 3 e 4. O modelo foi treinado em 4800 e 9600 iterações, para a taxa de aprendizado foram utilizados os valores de 0,00025 e 0,00050, e para o tamanho do lote foram testadas 2 e 4 imagens.

Na Tabela 2 são apresentados os resultados desse experimento, o melhor resultado foi atingido com 4800 iterações e não houve melhora ao treinar o modelo em 9600 iterações. Os resultados da combinação de 2 imagens por lote e taxa de aprendizado de 0,00025, e a combinação de 4 imagens por lote e taxa de aprendizado de 0,00050 também apresentaram resultados muito próximos.

Tabela 2. Resultados da combinação de múltiplos parâmetros de treinamento

Modelo	Iterações	Lote	Taxa de aprendizado	AP	Acurácia	Precisão	F1-Score
X101-FPN	4800	2	0,00025	85,16	0,88	0,89	0,88
X101-FPN	4800	2	0,00050	84,89	0,85	0,85	0,85
X101-FPN	4800	4	0,00050	85,43	0,87	0,87	0,87
X101-FPN	9600	2	0,00025	84,59	0,87	0,87	0,87

X101-FPN	9600	4	0,00050	86,29	0,85	0,85	0,85
----------	------	---	---------	-------	------	------	------

Utilizando a AP como fator de decisão consideramos o modelo Faster R-CNN X101-FPN treinado em 9600 iterações, 4 imagens por lote e taxa de aprendizado de 0,00050. Dos 15 exemplos de avaliação para cada classe, o modelo acertou todas as predições para as classes misturador e registro, errando 3 para a classe joelho e 5 para a classe tê.

4. CONCLUSÕES

O objetivo da construção dos *datasets* foi alcançado pela rotulação e segmentação de fotografias dos componentes, bem como de imagens extraídas dos catálogos dos fabricantes, totalizando 260 imagens utilizadas nos *datasets* de treinamento e validação.

Este trabalho demonstrou que, por meio da arquitetura Faster R-CNN, foi possível desenvolver um modelo de rede neural convolucional baseada em região capaz de localizar e detectar os componentes dos kits hidrossanitários industrializados.

A aplicação da visão computacional e métodos de detecção de objetos baseadas em imagem para a detecção, localização e classificação de tubos e conexões hidráulicas no âmbito da construção civil, é um tema hoje não abordado na literatura, desta forma, trazendo um viés inovador ao projeto.

5. REFERÊNCIAS BIBLIOGRÁFICAS

BHARATI, P.; PRAMANIK, A. Deep Learning Techniques - R-CNN to Mask R-CNN: A Survey. In: **COMPUTATIONAL INTELLIGENCE IN PATTERN RECOGNITION**, 2020, Singapore. **Anais...** Springer Singapore, 2020. p.657–668

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2014. **Proceedings...** IEEE, 2014. p.580–587.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J. et al. Recent advances in convolutional neural networks. **Pattern recognition**, New York, NY, United States, v.77, p.354–377, 2018.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. 2016. Disponível em: <<https://arxiv.org/abs/1506.01497>>. Acesso em: 20 set. 2024.

SAPONARA, S.; ELHANASHI, A. Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance. In: **APPLICATIONS IN ELECTRONICS PERVADING INDUSTRY, ENVIRONMENT AND SOCIETY**, 2022, Cham. **Anais...** Springer International Publishing, 2022. p.10–17.

WU, Y.; KIRILLOV, A.; MASSA, F.; LO, W.-Y.; GIRSHICK, R. **Detectron2**. 2019. Disponível em: <<https://github.com/facebookresearch/detectron2>>. Acesso em: 12 ago. 2024.