

UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

GIOVANNA SOUZA TEODORO

**DETECÇÃO INTELIGENTE DE DEFEITOS PARA INDÚSTRIA 4.0: UM ESTUDO DE
CASO COM MODELOS SINGLE-SHOT YOLO**

Manaus

2024

GIOVANNA SOUZA TEODORO

**DETECÇÃO INTELIGENTE DE DEFEITOS PARA INDÚSTRIA 4.0: UM ESTUDO DE
CASO COM MODELOS SINGLE-SHOT YOLO**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Computação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Computação.

Orientador(a): Profa. Dra. Elloá Barreto Guedes da Costa

Manaus

2024



FOLHA DE APROVAÇÃO

Detecção Inteligente de Defeitos para Indústria 4.0: Um Estudo de Caso com Modelos Single-Shot YOLO

Giovanna de Souza Teodoro

Trabalho de Conclusão de Curso defendido e aprovado pela banca avaliadora constituída pelos professores:

Profª Drª Elloá Barreto Guedes da Costa
Presidente

Profº Dr. Rubelmar Maia de Azevedo Cruz Neto
Avaliador(a)

Profº Dr. Carlos Mauricio Serodio Figueiredo
Avaliador(a)

RESUMO

Este trabalho considera a detecção de defeitos em fusos de esfera como uma tarefa de Visão Computacional abordada com modelos de *Deep Learning* do tipo *single-shot* da Família YOLO (acrônimo para *You Only Look Once*). Para tanto, foram conduzidos experimentos computacionais a partir de uma base de dados realística cujos resultados, após análise com testes estatísticos, permitiram elencar o modelo YOLOv5 *Nano* como solução de referência, com mAP@0.5 médio de 93.27 e mostrando-se mais de 97.84% mais rápida e com 95.68% menos parâmetros que uma contrapartida da literatura. Essa solução é recomendada para dispositivos embarcados e colabora com o monitoramento de equipamentos na Indústria 4.0, sugerindo avanço no estado da arte em termos de eficiência na detecção de defeitos.

Palavras-chaves: Aprendizado Profundo; Visão Computacional; Indústria 4.0; Detecção de Objetos; Sistemas Tolerantes a Falhas.

ABSTRACT

This work aims at detecting ball screw driver defects as a Computer Vision task based on Deep Learning single-shot YOLO models. Computational experiments using a realistic dataset were performed and their results altogether with statistical tests enlisted YOLOv5 *Nano* as best suited for this purpose, with an average mAP@0.5 of 93.27 and with improvements of 97.84% on speed and on 95.68% less parameters than a related work counterpart. The proposed solution is recommended for embedded devices and favors equipment monitoring for Industry 4.0, also suggesting improvements in state of art in terms of efficiency of defects detection.

Keywords: Deep Learning; Computer Vision; Industry 4.0; Object Detection; Fault Tolerant Systems.

Detecção Inteligente de Defeitos para Indústria 4.0: Um Estudo de Caso com Modelos *Single-Shot* YOLO

Giovanna Souza Teodoro * Elloá B. Guedes *

* Grupo de Pesquisa em Sistemas Inteligentes
Universidade do Estado do Amazonas (UEA)
Av. Darcy Vargas, 1200 – Manaus – Amazonas
(e-mail: {gst.eng19,ebgcosta}@uea.edu.br)

Abstract: This work aims at detecting ball screw driver defects as a Computer Vision task based on Deep Learning single-shot YOLO models. Computational experiments using a realistic dataset were performed and their results altogether with statistical tests enlisted YOLOv5 *Nano* as best suited for this purpose, with an average mAP@0.5 of 93,27% and with improvements of 97,84% on speed and on 95,68% less parameters than a related work counterpart. The proposed solution is recommended for embedded devices and favors equipment monitoring for Industry 4.0, also suggesting improvements in state of art in terms of efficiency of defects detection.

Resumo: Este trabalho considera a detecção de defeitos em fusos de esfera como uma tarefa de Visão Computacional abordada com modelos de *Deep Learning* do tipo *single-shot* da Família YOLO (acrônimo para *You Only Look Once*). Para tanto, foram conduzidos experimentos computacionais a partir de uma base de dados realística cujos resultados, após análise com testes estatísticos, permitiram elencar o modelo YOLOv5 *Nano* como solução de referência, com mAP@0.5 médio de 93,27% e mostrando-se mais de 97,84% mais rápida e com 95,68% menos parâmetros que uma contrapartida da literatura. Essa solução é recomendada para dispositivos embarcados e colabora com o monitoramento de equipamentos na Indústria 4.0, sugerindo avanço no estado da arte em termos de eficiência na detecção de defeitos.

Keywords: Deep Learning; Computer Vision; Industry 4.0; Object Detection; Fault Tolerant Systems.

Palavras-chaves: Aprendizado Profundo; Visão Computacional; Indústria 4.0; Detecção de Objetos; Sistemas Tolerantes a Falhas.

1. INTRODUÇÃO

A Quarta Revolução Industrial, também conhecida como Indústria 4.0, tem como objetivo interconectar e permitir interação em tempo real com toda a cadeia logística das indústrias de manufatura, utilizando tecnologias inovadoras e disruptivas para projetar, construir e desenvolver um mundo virtual que conduza o mundo físico (Nayyar and Kumar, 2019). Para tanto, leva em consideração a combinação de Inteligência Artificial (IA), Impressão 3D, Internet das Coisas, Realidade Virtual e Aumentada, *Big Data*, Nanotecnologia, entre outras tecnologias emergentes, para mudar os processos industriais, tornando-os mais automatizados e menos dependentes de intervenção humana. Entender o potencial e os impactos dessas inovações tecnológicas será essencial para um diferencial competitivo daqueles que almejam uma posição de liderança e desenvolvimento socio-econômico nos próximos anos (Skilton and Hovsepian, 2017).

Nos ambientes fabris modernos, conjuntos de dados são continuamente gerados a partir de diversas fontes e contêm informações úteis que podem aumentar a produtividade, como um meio valioso para apoio à decisão em diversas

áreas, especialmente no monitoramento e manutenção de equipamentos (Borgi et al., 2017). Assim, um problema contemporâneo é a necessidade de sistemas que viabilizem inspeções inteligentes, especialmente do maquinário industrial, sinalizando necessidades de manutenção ou a ocorrência de defeitos, problemas estes que interrompem o processo fabril, causam perdas financeiras e podem até mesmo criar riscos ou danos de maior ordem (Zhang et al., 2017). No Brasil, em particular, a adoção de tecnologias relacionadas à Indústria 4.0 é incipiente e também urgente, sob a ameaça de perder reputação no mercado globalizado (Santos et al., 2018).

Muitos equipamentos industriais são compostos por peças metálicas do tipo fuso de esfera (BSD, do inglês *ball screw driver*), um atuador mecânico que transforma o movimento rotacional em movimento linear em que esferas metálicas ocas presentes no rolamento colaboram para a diminuição do atrito. No caso de Impressoras 3D, por exemplo, o BSD é usado para controlar o movimento do cabeçote extrusor ao longo do eixo z e seu bom funcionamento assegura o depósito do filamento com precisão e favorece a produção de peças com qualidade uniforme (Tasneem and Amer, 2019). Porém, BSDs são altamente

sujeitos a intempéries, as quais podem ser graves e causarem danos estruturais (desgaste, corrosão), ou simplesmente culminarem em contaminação da superfície com óleo ou outros materiais. No primeiro caso, uma intervenção corretiva faz-se necessária, pois a precisão e a estabilidade do maquinário como um todo podem ser afetadas (Schlagenhauf and Landwehr, 2021), o que é corroborado por observações *in situ* que constataram que 26% das falhas em tais ambientes ocorrem por danos em peças dessa natureza (Fleischer et al., 2009). Observa-se, portanto, a demanda pela concepção de sistemas para o monitoramento de BSDs, colaborando para o aumento da segurança e confiabilidade do maquinário industrial, evitando falhas abruptas e diminuindo perdas econômicas (Zhang et al., 2017).

Este trabalho apresenta um estudo de caso de uso de Redes Neurais Convolucionais Regionais Profundas (R-CNNs, do inglês *Region-based Convolutional Neural Networks*) da família YOLO (do inglês, *You Only Look Once*) para detecção de falhas estruturais em BSDs. Os resultados obtidos demonstram a viabilidade da aplicação de soluções de *Deep Learning* para Visão Computacional no monitoramento inteligente de equipamentos para Indústria 4.0.

Para apresentar os resultados obtidos, este trabalho está organizado como segue. Uma visão geral dos trabalhos relacionados encontra-se na Seção 2. As bases de dados utilizadas, modelos, configurações e estratégias de avaliação das soluções propostas encontram-se descritos na Seção 3. Os resultados obtidos são apresentados, contrastados e discutidos na Seção 4. Por fim, as considerações finais e sugestões de trabalhos futuros são apresentadas na Seção 5.

2. TRABALHOS RELACIONADOS

Tradicionalmente, a detecção de falhas em BSDs é primordialmente abordada por meio de técnicas da Engenharia Mecânica. Uma dessas estratégias consiste no monitoramento dos BSDs por meio de imagens térmicas, em que as regiões mais quentes denotam a existência de atrito, um indicativo de perda de energia cinética ocasionada por danos estruturais (Ramesh et al., 2000). Segundo a literatura, esse monitoramento deve ocorrer durante a realização de um ciclo operacional do dispositivo (Allen et al., 1970). Embora não haja menção direta, entende-se que havia apenas uma inspeção visual das imagens produzidas, o que demanda *expertise* e também introduz um viés de subjetividade. Essa estratégia, entretanto, pode ter sua viabilidade prática limitada em virtude da necessidade de mão de obra humana qualificada, do alto custo dos dispositivos de captura de informações térmicas e do tempo requerido para captura e processamento das imagens.

Uma alternativa para o monitoramento de BSDs consiste no uso de sensores de vibração, conforme reportado no trabalho de Lee et al. (2015). Nele, os autores ressaltam que muitas pesquisas de monitoramento de defeitos na Engenharia Mecânica têm sido conduzidas para elementos rotativos, tais como motores, turbinas, engrenagens, compressores e similares. Porém, esse monitoramento é essencialmente mais complexo em módulos linearmente alternativos, como os BSDs, pois os sinais são não-periódicos e as mudanças, quando ocorrem, são abruptas. Para contribuir

nesta perspectiva, no trabalho em questão foi feita a inserção de defeitos em BSDs e o posterior monitoramento do funcionamento desses componentes com sensores de vibração. O sinal coletado foi sujeito à Transformada Wavelet e então analisado com métodos tradicionais do Aprendizado de Máquina, resultando em um sistema autônomo capaz de apontar o local do defeito no BSD. Uma das limitações desse trabalho, entretanto, diz respeito à natureza das falhas, as quais foram artificialmente introduzidas e podem não corresponder aos defeitos que emergem em contextos realísticos. O uso de técnicas de *Machine Learning* na análise de sensores de vibração em BSDs foi reportada na literatura apenas com o intuito de diagnosticar a pré-carga desses componentes (Huang et al., 2018).

Os relatos recentes da literatura já ressaltam, por exemplo, o uso de *Deep Learning* com CNNs na Engenharia Térmica como um advento com grande potencial de efetividade na detecção de defeitos (Patel et al., 2021). No caso dos BSDs, a utilização de tais técnicas foram consideradas no trabalho pioneiro de Schlagenhauf and Landwehr (2021), no qual foi proposta uma base de dados pública, gratuita e rotulada para este problema, a chamada BSDData (Schlagenhauf et al., 2021). Nesse trabalho, o problema da detecção de defeitos em BSDs foi abordado como uma tarefa de segmentação usando uma *Mask R-CNN* de dois estágios, a qual possui cerca de 44 Mi parâmetros pré-treinados na base de dados MS COCO e que opera a 5 FPS¹. Os autores reportaram mIoU (do inglês, *Mean Intersection Over Union*) igual a 0,316 e denotaram uma especial dificuldade do modelo proposto para com defeitos arbitrariamente pequenos (Schlagenhauf and Landwehr, 2021). Ressalta-se que não foi possível identificar qual estratégia de validação cruzada foi utilizada pelos autores nos experimentos.

Levando em consideração as contribuições recentes na literatura de arquiteturas de R-CNNs *single-shot* da Família YOLO (Diwan et al., 2023), o objetivo deste trabalho consistiu em explorar o problema em questão como uma tarefa de detecção de objetos, visando investigar se haveria benefícios ao usar modelos com substancialmente menos parâmetros, com alta acurácia e velocidade de detecção, o que poderia viabilizar uma solução para contextos realísticos da Indústria 4.0. Nessa perspectiva, a seção a seguir detalha os procedimentos metodológicos adotados para condução dos experimentos.

3. METODOLOGIA

O problema considerado no escopo deste trabalho foi abordado como uma tarefa de detecção mediante Aprendizado Supervisionado com R-CNNs da Família YOLO. A infraestrutura computacional utilizada consistiu em um servidor com Intel(R) Core(TM) i7-8700 CPU @ 3,20 GHz, 32 GB de memória principal, 2,4 TB de memória secundária e 2 GPUs NVIDIA GeForce GTX 1080 Ti com 11 GB de VRAM para aceleração em *hardware* do treinamento. Os dados experimentais, modelos e a avaliação de desempenho da tarefa considerada são descritos detalhadamente nas subseções a seguir.

¹ Acrônimo para *Frames per Second*

3.1 Dados Experimentais

A base de dados BSData, proposta por Schlagenhauf et al. (2021), contém 1107 imagens coloridas de BSDs, as quais possuem dimensões variadas, e nela constam 394 rótulos de segmentação delimitando a região afetada quando a peça correspondente encontrava-se danificada. Os exemplos variam no tocante à cor do BSD e às condições de iluminação, estando algumas peças contaminadas por solo ou por resíduos do processo industrial (e.g., crosta de óleo). Alguns exemplos oriundos desta base de dados encontram-se ilustrados na Figura 1.

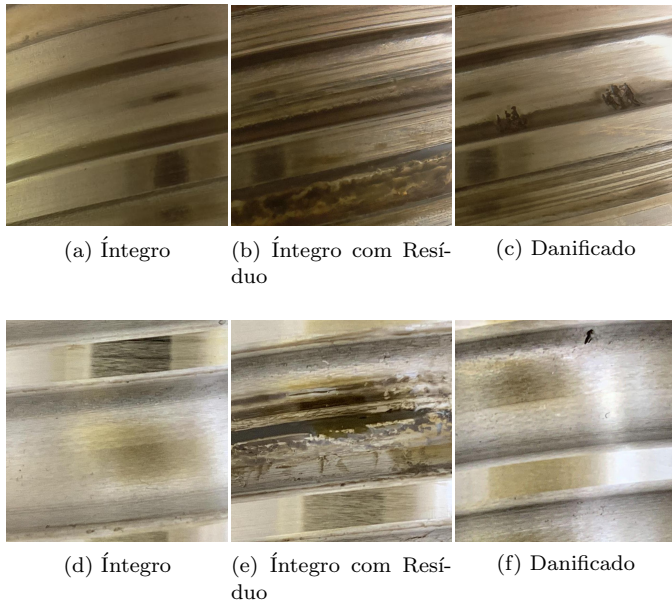


Figura 1. Exemplos de imagens oriundas do BSDATA. Fonte: (Schlagenhauf et al., 2021; Schlagenhauf and Landwehr, 2021).

No caso particular das peças danificadas, tal como ilustrado na Figura 2a, os rótulos delimitadores da área danificada possuíam formato convexo arbitrário, conforme ilustrado na Figura 2b, o que ensejava a abordagem como uma tarefa de segmentação. Para adaptar ao que se propunha, derivou-se, a partir das coordenadas mais externas, caixas delimitadoras em formato retangular, as quais foram utilizadas como atributo alvo na tarefa de detecção, vide Figura 2c.

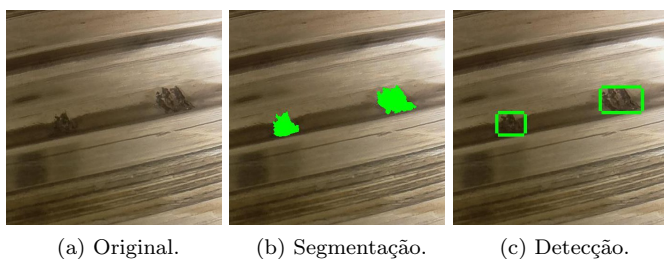


Figura 2. Etapa de adaptação dos rótulos dos defeitos. Fonte: (Schlagenhauf et al., 2021; Schlagenhauf and Landwehr, 2021).

Após a adaptação dos rótulos, partiu-se para uma análise exploratória com o intuito de examinar a frequência de tais

caixas por imagem e o tamanho das mesmas. Conforme resultados dispostos na Figura 3 e Tabela 1, respectivamente, foi possível perceber que o número de defeitos por imagem era variado e que as caixas delimitadoras possuíam dimensões arbitrárias não uniformes.

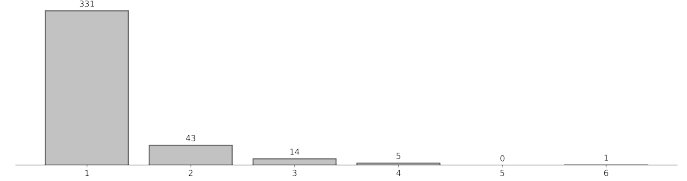


Figura 3. Quantidade de defeitos (eixo horizontal) por imagem (eixo vertical).

3.2 Arquiteturas da Família YOLO e Parametrização

As arquiteturas de R-CNNs da Família YOLO foram propostas originalmente em 2015 com o objetivo de unificar as tarefas necessárias para detecção (localizar e classificar os objetos) em uma única passagem (*single-shot*) da entrada pelo modelo. Ademais, reformularam o problema de detecção como uma tarefa de regressão, em que a rede prevê as coordenadas das caixas delimitadoras, seus coeficientes de confiança e também uma distribuição de probabilidades das classes à elas associados (Redmon et al., 2016). Em virtude dessas características, os modelos da Família YOLO são considerados rápidos e acurados, de baixo custo computacional e atualmente constituem o estado da arte para este tipo de problema, com aplicações em diversos contextos (Michelucci, 2019; Diwan et al., 2023).

Ao longo do tempo houve várias melhorias na proposição seminal da YOLO que ensejaram versões sucessoras. Inicialmente, por exemplo, a proposição de regiões de interesse era feita por caixas ancoradas, enquanto nas versões mais recentes tal proposição é feita com uma CNN. O *backbone* original foi otimizado, tornando-se mais leve e rápido para extração de características, bem como mais eficiente para detecção de objetos pequenos. Citam-se ainda paralelização do treinamento, classificação em contextos de alta resolução e *batch normalization*. Além disso, as versões mais recentes propõem um melhor custo-benefício entre número de parâmetros e eficiência na detecção, o que facilita a adoção em dispositivos embarcados, por exemplo (Abdullah et al., 2022).

No escopo deste trabalho, versões recentes de arquiteturas da Família YOLO foram consideradas e são apresentadas a seguir. Informações acerca do número de parâmetros e do tamanho encontram-se detalhadas na Tabela 2.

- (1) **YOLOv5**. Decorrente da continuidade de melhorias propostas na YOLOv4 (Bochkovskiy et al., 2020), foi desenvolvida nativamente com o *framework* Pytorch especialmente com o intuito de simplificar a implementação e o *deployment*. A principal inovação foi a introdução do *Mosaic Augmentation*, uma técnica de regularização por meio do aumento artificial de dados que combina quatro imagens em quatro blocos de proporção aleatória. Conforme argumentam seus proponentes, a principal vantagem dessa inovação é a melhoria de desempenho na detecção de objetos pequenos (Jocher et al., 2022);

Tabela 1. Descrição estatística das caixas delimitadoras da base de dados (em px).

Comprimento			Largura		
Média e Desvio Padrão	Máximo	Mínimo	Média e Desvio Padrão	Máximo	Mínimo
55,32 ± 27,07	107	6	67,33 ± 45,56	197	3

- (2) **YOLOv7**. Dando continuidade ao aprimoramento dos modelos da Família YOLO, introduziu um processo denominado reparametrização, com o intuito de diminuir o tamanho do modelo para *deployment*, o que é especialmente útil para sistemas embarcados. Também faz uso de *Feature Pyramid Networks* em sua arquitetura, as quais empilham camadas convolucionais e produzem previsões em diferentes escalas, colaborando para um melhor desempenho. Diferentemente das versões anteriores, propôs uma arquitetura padrão e uma estratégia de dimensionamento em escala para outras quantidades de parâmetros, sejam elas maiores ou menores (Wang et al., 2022);
- (3) **YOLOv8**. Diferencia-se das suas anteriores por não usar a abordagem de caixas ancoradas para localização dos objetos em favor da localização do centro dos objetos, diminuindo o ônus computacional para execução do algoritmo de *Non-Maximum Supression* requerido anteriormente. Os filtros de algumas camadas convolucionais foram diminuídos de 6×6 para 3×3 , o que implica na redução do número de parâmetros. Não utiliza a técnica de *Mosaic Augmentation* em todas as épocas, pois os autores observaram empiricamente sua maior efetividade quando interrompida 10 épocas antes do final do treinamento (Jocher et al., 2023).

Tabela 2. Descrição dos parâmetros e tamanho dos modelos.

Arquitetura	Modelo	Parâmetros (Mi)	Tamanho (MB)
YOLOv5	<i>Nano</i>	1,9	3,8
	<i>Small</i>	7,2	14,3
	<i>Medium</i>	21,2	42,1
YOLOv7	<i>Tiny</i>	6,2	12,3
	Padrão	36,9	74,8
	<i>X</i>	71,3	142,1
YOLOv8	<i>Nano</i>	3,2	6,2
	<i>Small</i>	11,2	22,5
	<i>Medium</i>	25,9	52,0

Com o propósito de padronizar e garantir o treinamento equitativo de todos os modelos, alguns hiperparâmetros foram estabelecidos, a citar: *batch size* igual a 16, redimensionamento das imagens para resolução de 640×640 e um total de 300 épocas, sem regularização por *early stopping*. Os pesos das redes foram inicializados aleatoriamente, sem transferência de aprendizado de outras tarefas. Os demais hiperparâmetros foram utilizados conforme os valores padrão de cada versão da YOLO.

3.3 Avaliação de Desempenho

As imagens do conjunto de dados BSDData foram divididas segundo uma validação cruzada do tipo *holdout* na qual

70 % das imagens foram utilizadas para treinamento, 10 % para a validação e 20 % para teste. Para avaliar a robustez dos modelos, evitar *overfitting* e diminuir os vieses introduzidos pela flutuação estocástica, foram efetuadas 3 repetições do experimento proposto. Ressalta-se que os experimentos diferenciavam-se no tocante aos exemplos em cada partição, cuja seleção foi feita de maneira aleatória segundo uma distribuição uniforme. As repetições experimentais foram oportunas para avaliar também a detecção de objetos pequenos (com dimensões inferiores a 32×32 px), um desafio particular neste tipo de tarefa (Lin et al., 2014). A Tabela 3 mostra o quantitativo de objetos em cada partição, o número de objetos pequenos e sua proporção na respectiva partição.

O desempenho dos modelos foi aferido no tocante à correta classificação e ao grau de sobreposição das caixas delimitadoras em comparação com os rótulos disponíveis na partição de testes, o qual foi sintetizado conforme métricas de tarefas de detecção: precisão, revocação, *F₁-Score*, AP (do inglês, *Average Precision*) e mAP (do inglês, *Mean Average Precision*), em que nesta última considerou-se o limiar do IoU (do inglês, *Intersection Over Union*) como sendo maior ou igual a 0,5. Ademais, aferiu-se também o mIoU para viabilizar análises comparativas com trabalhos da literatura. Uma explicação detalhada do cálculo de tais métricas no contexto da detecção de objetos em Visão Computacional encontra-se disponível no *survey* de Padilla et al. (2020).

4. RESULTADOS E DISCUSSÃO

Os experimentos computacionais foram executados respeitando os passos da metodologia proposta e a síntese dos resultados obtidos encontra-se na Tabela 4, em que as métricas são denotadas em termos da média e do desvio padrão das três repetições realizadas. A Figura 4 detalha os valores observados da métrica de mAP@0.5 por modelo e por arquitetura, em virtude desta ser considerada uma referência para sintetizar a qualidade de detectores de objetos.

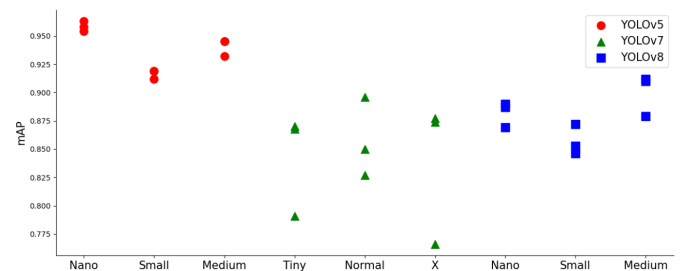


Figura 4. Comparativo dos modelos YOLO ante a métrica mAP com limiar de 0,5.

Tabela 3. Detalhamento das repetições experimentais.

Experimento	Treinamento		Validação		Teste	
	Total	Objetos Pequenos	Total	Objetos Pequenos	Total	Objetos Pequenos
Experimento 1	338	78 (23,07 %)	48	10 (20,83 %)	99	22 (22,22 %)
Experimento 2	329	82 (24,92 %)	49	9 (18,36 %)	107	19 (17,75 %)
Experimento 3	337	76 (22,55 %)	47	14 (29,78 %)	101	20 (19,80 %)

Tabela 4. Resultados da avaliação de desempenho dos modelos no contexto proposto.

Arquitetura	Modelo	Precisão	Revocação	F_1 -Score	mAP@0,5	mIoU
YOLOv5	<i>Nano</i>	95,27 ± 0,66	91,27 ± 2,09	93,22 ± 1,40	93,27 ± 1,72	0,724 ± 0,005
	<i>Small</i>	97,13 ± 0,52	91,50 ± 2,04	94,23 ± 1,32	94,07 ± 1,62	0,730 ± 0,009
	<i>Medium</i>	97,47 ± 0,86	93,83 ± 3,74	95,59 ± 2,28	94,23 ± 1,81	0,738 ± 0,003
YOLOv7	<i>Tiny</i>	92,83 ± 8,3	75,67 ± 4,27	82,94 ± 1,43	79,47 ± 2,50	0,679 ± 0,013
	Padrão	98,37 ± 2,31	79,17 ± 1,67	87,69 ± 0,54	88,10 ± 1,10	0,693 ± 0,013
	<i>X</i>	92,77 ± 1,80	79,17 ± 1,67	85,40 ± 0,92	86,40 ± 1,02	0,700 ± 0,014
YOLOv8	<i>Nano</i>	96,13 ± 1,28	79,57 ± 2,35	87,03 ± 0,89	86,70 ± 1,07	0,731 ± 0,035
	<i>Small</i>	96,03 ± 1,43	80,93 ± 4,23	87,75 ± 2,11	88,27 ± 2,74	0,745 ± 0,036
	<i>Medium</i>	95,90 ± 1,31	81,80 ± 1,39	88,29 ± 1,18	88,97 ± 1,56	0,740 ± 0,025

Considerando os diferentes modelos ($k = 9$) e replicações experimentais ($n = 3$), verificou-se a necessidade de realizar testes estatísticos para responder alguns questionamentos sobre os resultados obtidos. Assim, tomando o mAP@0.5, a primeira investigação deu-se com o propósito de avaliar a hipótese nula (H_0) de que a mediana das n populações de todos os k grupos eram iguais, i.e., os modelos seriam equivalentes para a tarefa de detecção proposta. Na impossibilidade de assegurar que os dados seguem a distribuição normal, partiu-se para utilização do Teste Não-Paramétrico H de Kruskal-Wallis com nível de confiança de 95 % ($\alpha = 0,05$) (Walpole et al., 2016). O p -valor obtido foi igual a 0,00541 e, como $p < \alpha$, rejeita-se H_0 em favor da hipótese alternativa, com 95 % de confiança, de que os modelos não são equivalentes para o problema.

O resultado do teste de hipóteses efetuado suscitou outros questionamentos que também foram investigados. Desejava-se então identificar, caso houvesse, qual arquitetura ou modelo teria desempenho superior aos demais. Ao observar a Tabela 4, percebeu-se que a média do mAP@0.5 de todos os modelos da arquitetura YOLOv5 foi superior a 93 % e que nenhuma das outras arquiteturas obteve média superior a 90 %. Em decorrência, lançou-se mão do teste estatístico *post hoc* de Dunn (1964) para efetuar múltiplas comparações dos modelos YOLOv5, em que a hipótese nula (H_0) assertou sobre a equivalência entre dois dados modelos (digam-se, Modelo A e Modelo B) no tocante ao desempenho. Manteve-se o nível de confiança de 95 % ($\alpha = 0,05$) e os resultados obtidos encontram-se detalhados na Tabela 5.

A partir da análise estatística realizada, os diferentes modelos YOLOv5 se mostraram equivalentes, em termos do mAP@0.5, perante o cenário experimental conduzido. Uma inspeção visual que corrobora essa similitude é ilustrada na Figura 5, em que um exemplo do conjunto de testes possui 3 defeitos, denotados por caixas delimitado-

ras verdes, e as detecções realizadas estão denotadas em azul.

Em face da equivalência, partiu-se para examinar os recursos computacionais utilizados no cenário experimental conduzido, conforme reportado na Tabela 6. Foi possível observar que os níveis de confiança no maior F_1 -Score foram bastante similares, mas que os aspectos relativos ao ônus computacional foram demasiado díspares. Por exemplo, o número de GFLOPs requerido para treinar a YOLOv5 *Medium* foi 168,46 % superior ao utilizado pela YOLOv5 *Nano*, mas esta última mostrou-se 52,17 % mais rápida que a primeira ao detectar objetos. Assim, elencou-se então o modelo YOLOv5 *Nano* como solução de referência proposta no escopo deste trabalho em virtude da menor quantidade de parâmetros, o que implica em menor ônus computacional para treinamento e para posterior inferência em contextos práticos. Uma vantagem dessa solução é que a mesma pode ser facilmente implantada em dispositivos embarcados disponíveis atualmente, podendo vir a atuar como um nó sensor inteligente para monitorar fusos de esfera numa Rede de Sensores sem Fio (Loureiro et al., 2003) em uma planta industrial, nos termos do que se propõe na Indústria 4.0.

Em comparação com um trabalho relacionado disponível na literatura sobre a mesma base de dados (Schlagenhauf and Landwehr, 2021), tem-se que tanto o pior quanto o melhor mIoU observados experimentalmente, de 0,660 na YOLOv7 *Tiny* e de 0,775 na YOLOv8 *Small*, respectivamente, são 70,49 % e 84,14 % superiores ao valor por eles reportado, indicando uma alta probabilidade de melhorias no estado da arte no problema da detecção de defeitos em fusos de esfera a partir de imagens realísticas. Ademais, a YOLOv5 *Nano*, solução de referência neste trabalho, é capaz de detectar defeitos em tempo real a uma taxa 97,84 % superior ao trabalho relacionado, favorecendo o uso em contextos práticos. Esses resultados são corroborados por outras observações da literatura que afirmam que modelos *single-shot* YOLO são mais eficazes e eficientes

Tabela 5. Resultados da comparação dos modelos YOLOv5 com o Teste de Dunn.

Modelo A	Modelo B	p-valor	Observação	Conclusão
YOLOv5 <i>Nano</i>	YOLOv5 <i>Small</i>	0,9074	$p > \alpha$	Não há evidências para rejeitar H_0
YOLOv5 <i>Nano</i>	YOLOv5 <i>Medium</i>	0,8356	$p > \alpha$	Não há evidências para rejeitar H_0
YOLOv5 <i>Small</i>	YOLOv5 <i>Medium</i>	0,9982	$p > \alpha$	Não há evidências para rejeitar H_0

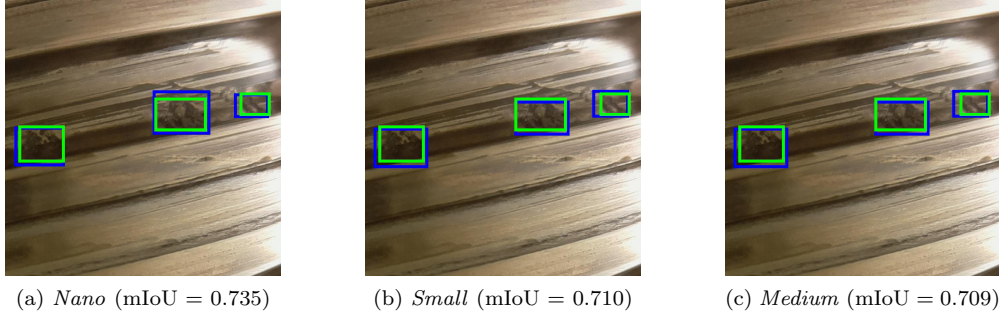


Figura 5. Detalhe de exemplo de detecção realizado pelos modelos YOLOv5.

Tabela 6. Recursos computacionais utilizados e métricas coletadas.

Modelo	GFLOPs	Melhor F_1 -Score	Limiar de Confiança	FPS
YOLOv5 <i>Nano</i>	4,1	0,95	0,47	232
YOLOv5 <i>Small</i>	15,8	0,96	0,47	222
YOLOv5 <i>Medium</i>	47,9	0,97	0,46	136

em contextos práticos de detecção de objetos com *Deep Learning* que *Mask R-CNNs* (Sumit et al., 2020).

5. CONSIDERAÇÕES FINAIS

Visando contribuir para o desenvolvimento de soluções para a Indústria 4.0, este trabalho considerou o problema de detecção automática e inteligente de defeitos em fusos de esfera a partir de uma base de dados realística disponível na literatura. Para tanto, considerou diferentes versões de R-CNNs da Família YOLO em um cenário experimental em que também havia múltiplos objetos pequenos, uma dificuldade particular no âmbito da detecção de objetos. Três modelos de cada arquitetura (YOLOv5, YOLOv7 e YOLOv8) foram aferidos perante três repetições, totalizando 27 treinamentos e testes executados. Os resultados obtidos mostraram-se satisfatórios para todos os modelos, nos quais o F_1 -Score médio não foi inferior a 82%. Escolhendo a métrica mAP@0.5 para denotar a qualidade de detecção, testes estatísticos revelaram que o desempenho dos modelos não era equânime, com as Arquiteturas YOLOv5 se sobressaindo sobre as demais versões. Ao considerar também o número de parâmetros e recursos computacionais para treinamento e inferência, a YOLOv5 *Nano* foi classificada como solução de referência no escopo deste trabalho.

A YOLOv5 *Nano* apresentou um baixo desvio padrão nas métricas coletadas durante os experimentos, o que reforçou evidências de sua robustez, registrou a maior taxa de detecção dentre os modelos avaliados e possuía o menor número de parâmetros. Essas vantagens observadas nos experimentos pode ser usufruída em cenários práticos, inclusive quando implantada em dispositivos embarcados, aproximando sua utilização em contextos práticos da In-

dústria 4.0. Em comparação com um trabalho relacionado da literatura sobre a mesma base de dados, mostrou-se 97,84% mais rápida na detecção, com mIoU 56,35% superior e com 95,68% menos parâmetros. Tais resultados são sugestivos de um avanço no estado da arte perante a tarefa considerada.

Um outro aspecto a ressaltar do trabalho em questão foi a experimentação com modelos recém-propostos na literatura, em especial a YOLOv7 (Wang et al., 2022) e a YOLOv8 (Jocher et al., 2023). Embora os autores argumentem acerca do aumento no desempenho das mesmas frente ao desafio MS COCO (Lin et al., 2014), tais vantagens não se reproduziram em um cenário mais específico, em que tais modelos tiveram desempenho aquém aos seus antecessores. Entretanto, outras investigações empíricas de tais modelos em cenários diversos são necessárias para conclusões mais categóricas.

Em trabalhos futuros almeja-se expandir os experimentos realizados considerando outras versões e modelos da Família YOLO, outras abordagens *single-shot* para detecção de objetos, transferência de aprendizado, ajuste fino de parâmetros e hiperparâmetros e também validação externa com outras bases de dados para o mesmo problema disponíveis na literatura.

AGRADECIMENTOS

As autoras agradecem o apoio financeiro da Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por meio do programa PAIC/UEA 2022-2023 e o apoio material do Laboratório de Sistemas Inteligentes (LSI) da Universidade do Estado do Amazonas (UEA).

REFERÊNCIAS

- Abdullah, S., AL-Nuaimi, B., and Abed, H. (2022). A Survey of Deep Learning-Based Object Detection: Application and Open Issues. *International Journal of Nonlinear Analysis and Applications*, 13(2), 1495–1504.
- Allen, J., Postlethwaite, S., and Ford, D. (1970). Practical Application of Thermal Error Correction – 4 Case Studies. *WIT Transactions on Engineering Sciences*, 16.
- Bochkovskiy, A., Wang, C.Y., and Liao, H.Y.M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Disponível em <https://arxiv.org/abs/2004.10934>. Acesso em 17 de agosto de 2023.
- Borgi, T., Hidri, A., Neef, B., and Naceur, M.S. (2017). Data Analytics for Predictive Maintenance of Industrial Robots. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. IEEE, Tunísia.
- Diwan, T., Anirudh, G., and Tembhurne, J.V. (2023). Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications. *Multimed. Tools Appl.*, 82(6), 9243–9275.
- Dunn, O.J. (1964). Multiple Comparisons using Rank Sums. *Technometrics*, 6(3), 241–252.
- Fleischer, J., Broos, A., Schopp, M., Wieser, J., and Henrich, H. (2009). Lifecycle-Oriented Component Selection for Machine Tools Based on Multibody Simulation and Component Life Prediction. *CIRP J. Manuf. Sci. Technol.*, 1(3), 179–184.
- Huang, Y.C., Kao, C.H., and Chen, S.J. (2018). Diagnosis of the Hollow Ball Screw Preload Classification using Machine Learning. *Appl. Sci. (Basel)*, 8(7), 1072.
- Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLOv8 by Ultralytics. Disponível em <https://github.com/ultralytics/ultralytics>. Acesso em 17 de agosto de 2023.
- Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomamma, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, J., Fang, J., Yu, L., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, and Rai, P. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Disponível em <http://dx.doi.org/10.5281/ZENODO.7347926>. Acesso em 17 de agosto de 2023.
- Lee, W.G., Lee, J.W., Hong, M.S., Nam, S.H., Jeon, Y., and Lee, M.G. (2015). Failure Diagnosis System for a Ball-Screw by using Vibration Signals. *Shock Vib.*, 2015, 1–9.
- Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C.L. (2014). Microsoft COCO: Common Objects in Context. In *Proceedings of the 13th European Conference on Computer Vision (ECVV 2014)*, 740–755. Springer, Suíça.
- Loureiro, A.A., Nogueira, J.M.S., Ruiz, L.B., Mini, R.A.d.F., Nakamura, E.F., and Figueiredo, C.M.S. (2003). Redes de Sensores Sem Fio. In *Simpósio Brasileiro de Redes de Computadores (SBRC)*, 179–226. Editora da UFRN, Natal, Rio Grande do Norte.
- Michelucci, U. (2019). *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection*. Apres, Suíça.
- Nayyar, A. and Kumar, A. (2019). *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*. Advances in Science, Technology & Innovation. Springer, Suíça, 1 edition.
- Padilla, R., Netto, S.L., and da Silva, E.A.B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237–242. Niterói, Brasil. doi:10.1109/IWSSIP48289.2020.9145130.
- Patel, A.R., Ramaiya, K.K., Bhatia, C.V., Shah, H.N., and Bhavsar, S.N. (2021). Artificial Intelligence: Prospect in Mechanical Engineering Field – A Review. In *Lecture Notes on Data Engineering and Communications Technologies*, 267–282. Springer Singapore, Singapore.
- Ramesh, R., Mannan, M.A., and Poo, A.N. (2000). Error Compensation in Machine Tools – A Review: Part II: Thermal Errors. *Int. J. Mach. Tools Manuf.*, 40(9), 1257–1284.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, Estados Unidos.
- Santos, M., Manhães, A.M., and Lima, A.R. (2018). Indústria 4.0: Desafios e Oportunidades para o Brasil. In *Anais do X Simpósio de Engenharia de Produção de Sergipe*, 317–329. Departamento de Engenharia de Produção – Universidade Federal de Sergipe, Sergipe. Disponível em https://ri.ufs.br/bitstream/riufs/10423/2/Industria_4_0.pdf. Acesso em 17 de agosto de 2023.
- Schlagenhauf, T. and Landwehr, M. (2021). Industrial Machine Tool Component Surface Defect Dataset. *Data Brief*, 39(107643), 107643.
- Schlagenhauf, T., Landwehr, M., and Fleischer, J. (2021). Industrial Machine Tool Element Surface Defect Dataset. Disponível em <https://doi.org/10.5445/IR/1000129520>. Acesso em 17 de agosto de 2023.
- Skilton, M. and Hovsepian, F. (2017). *The 4th Industrial Revolution – Responding to the Impact of Artificial Intelligence on Business*. Springer International Publishing, Suíça, 1 edition.
- Sumit, S.S., Watada, J., Roy, A., and Rambli, D. (2020). In Object Detection Deep Learning Methods, YOLO Shows Supremum to Mask R-CNN. *Journal of Physics: Conference Series*, 1529(4), 042086.
- Tasneem, M.H.B. and Amer, G.T. (2019). Design, Fabrication and Testing of a 3D Printer. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2334–2344. Pilsen, Czech Republic.
- Walpole, R.E., Myers, R.H., Myers, S.L., and Ye, K.E. (2016). *Probability & Statistics for Engineers & Scientists*. Pearson, Estados Unidos, 9 edition.
- Wang, C.Y., Bochkovskiy, A., and Liao, H.Y.M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. Disponível em <https://arxiv.org/abs/2207.02696>. Acesso em 17 de agosto de 2023.
- Zhang, L., Gao, H., Wen, J., Li, S., and Liu, Q. (2017). A Deep Learning-Based Recognition Method for Degradation Monitoring of Ball Screw with Multi-Sensor Data Fusion. *Microelectronics Reliability*, 75, 215–222.