Instance Segmentation via Deep Learning for Machinery Monitoring in Industry 5.0

Carlos Alves Lavor Neto, Elloá B. Guedes

[1]Grupo de Pesquisa em Sistemas Inteligentes
Universidade do Estado do Amazonas
Av. Darcy Vargas, 1200 – Manaus – Amazonas
{caln.eng22, ebgcosta}@uea.edu.br

Abstract. *Aiming to contribute to automated visual inspection in Industry 5.0, this work presents a comparative analysis of intelligent models for instance segmentation of defects in ball screw drives. The BSData benchmark was used to evaluate specialized end-to-end models (YOLO26-seg) and foundation models operating in a zero-shot regime (SAM2 and SAM3). Under a unified experimental protocol, the SAM2 Hiera-T architecture was identified as a candidate solution, achieving a mAP@0.5 of 0.799, a mIoU of 0.632, and an inference rate of 8.6 FPS. The selected model shows adaptive robustness and outperforms the instance segmentation baseline previously reported for selected benchmark.*

Resumo. *Visando contribuir para a inspeção visual automatizada na Indústria 5.0, este trabalho apresenta uma análise comparativa de modelos inteligentes para segmentação de instâncias de defeitos em fusos de esfera. O* benchmark *BSData foi utilizado para avaliar modelos especializados treinados de ponta a ponta (YOLO26-seg) e modelos de fundação em regime* zero-shot *(SAM2 e SAM3). Sob um protocolo experimental unificado, foi possível identificar a arquitetura SAM2 Hiera-T como solução candidata com mAP@*$0{,}5$ *de* $0{,}799$*, mIoU de* $0{,}632$ *e taxa de inferência de* $8{,}6$ FPS*. O modelo selecionado apresenta robustez adaptativa e supera o* baseline *de segmentação de instâncias previamente reportado para o BSData.*

1. Introdução

A Indústria 5.0 possui três princípios centrais: a centralidade humana, a sustentabilidade e a resiliência. No tocante ao primeiro princípio, a tecnologia é utilizada para complementar e aumentar as capacidades humanas, focando nos indivíduos no centro do processo de fabricação. Em termos de sustentabilidade, exige que a produção industrial respeite compromissos e limites econômicos, ambientais e sociais. Por fim, em termos de resiliência, refere-se à capacidade de um sistema industrial de manter o seu funcionamento ou recuperar-se rapidamente para um estado estável durante e após grandes interrupções [Leng et al. 2022].

A resiliência operacional na Indústria 5.0 fundamenta-se no monitoramento inteligente e contínuo do maquinário. Diante de anomalias, o sistema gera alertas para acionar rotinas de contenção, como a redução preventiva de velocidade ou o isolamento lógico do componente afetado, mantendo a operação parcial e segura. Essa abordagem mitiga falhas catastróficas (*shutdowns*) e protege a integridade do equipamento. Sob essa ótica,

os alertas servem de suporte ao especialista humano, preservando sua centralidade no processo ao direcionar sua perícia para manutenções e reparos especializados [Singh et al. 2023].

Com o objetivo de contribuir para o monitoramento inteligente de maquinário na perspectiva da Indústria 5.0, o presente trabalho aborda a segmentação de instâncias de defeitos empregando modelos de *Deep Learning* em uma tarefa de Visão Computacional. Nesse cenário, o objetivo consiste em gerar máscaras delimitadoras das áreas defeituosas dos fusos de esfera, componentes típicos em diversas máquinas industriais. Para tanto, utilizou-se um *benchmark* da literatura para avaliar e comparar modelos baseados em redes neurais convolucionais *single-stage* (YOLO26) e modelos de fundação em regime *zero-shot* (SAM2 e SAM3). A análise comparativa experimental permitiu identificar a arquitetura SAM2 Hiera-T como solução candidata, a qual superou numericamente o desempenho de contrapartidas da literatura para a mesma tarefa, consolidando um avanço no estado da arte para essa aplicação.

Para apresentar o que se propõe, o artigo está organizado como segue: os trabalhos relacionados encontram-se dispostos na Seção 2; a visão geral da solução e a metodologia para sua obtenção são descritos na Seção 3; os resultados experimentais são apresentados e discutidos na Seção 4; por fim, as conclusões e perspectivas futuras são dispostas na Seção 5.

2. Trabalhos Relacionados

Para compreender de forma abrangente as soluções existentes, identificar tendências e lacunas, realizou-se uma análise dos resultados de buscas realizadas no Google Acadêmico entre 2022 e 2026. Os critérios de inclusão contemplaram trabalhos baseados em *Deep Learning* para detecção de defeitos no *benchmark* BSData [Schlagenhauf et al. 2021], que contempla imagens coloridas de fusos de esfera (BSDs, do inglês, *Ball Screw Drives*), um componente típico do maquinário industrial. As imagens foram capturadas sob diferentes condições de iluminação, enquadramento e estado superficial, incluindo peças com resíduos provenientes do processo industrial, defeituosas e conformes. No primeiro caso, há uma máscara de segmentação anotada no *dataset*, a qual deve ser identificada automaticamente por métodos de Visão Computacional para fins de manutenção preventiva ou corretiva, a depender da sua extensão.

Com vistas a promover um *baseline* para a tarefa de segmentação de instâncias no BSData, Schlagenhauf et al. [2021] utilizaram uma arquitetura Mask R-CNN de dois estágios para localizar e criar máscaras para os defeitos, com mIoU igual a $0{,}316$. Segundo os autores, o desempenho do Mask R-CNN foi aquém do esperado na detecção de falhas muito pequenas, localizadas em áreas de baixo contraste ou com bordas demasiado irregulares, desafios recorrentes em cenários realistas, nos quais se espera maior robustez de modelos inteligentes.

Outros trabalhos reportados na literatura para o BSData reformularam o problema como uma tarefa de detecção de objetos, ainda no escopo da Visão Computacional. Li *et al.* [2022] investigaram mecanismos de atenção (*Attention Augmentation*) integrados à arquitetura CSPDarkNet-MHSA-FDS, com o objetivo de mitigar problemas como desfoque de movimento, variações de escala e interferências visuais comuns em ambientes industriais. Posteriormente, Teodoro e Guedes [2023] exploraram a Família YOLO (ver-

sões v5, v7 e v8), demonstrando que modelos *single-stage* operam com substancialmente menos parâmetros e maior taxa de FPS, mantendo altos índices de precisão e revocação para contextos de tempo real. Apesar dos bons resultados quantitativos, essas abordagens avaliam predominantemente a localização de uma caixa delimitadora externa para cada defeito, sem determinar, contudo, o seu formato preciso, o que é crucial no contexto da Indústria 5.0 quando se consideram estratégias inteligentes e automatizadas para reparo do maquinário.

A busca por modelos ainda mais leves e especializados em superfícies metálicas levou ao desenvolvimento da LiMS-Net [Zhu and Lin 2026]. Essa arquitetura utiliza blocos convolucionais síncronos e fusão de atributos de múltiplas escalas, superando modelos genéricos como o YOLOv10 ao tratar as particularidades das propriedades reflexivas dos metais. Um outro trabalho, cuja abordagem foi focada em segmentação semântica, utilizou redes do tipo *U-Net* com funções de perda personalizadas para medir a extensão do defeito [Landgraf et al. 2023]. Diferentemente da segmentação de instâncias, contudo, a segmentação semântica classifica todos os *pixels* da imagem, sem individualizar as ocorrências de defeitos individuais.

Ao examinar a literatura, conforme sintetizado na Tabela 1, tem-se que vários trabalhos abordam o *benchmark* BSData para identificação de defeitos em maquinários industriais com *Deep Learning*. Contudo, apenas o trabalho seminal de Schlagenhauf et al. [2021] considera a tarefa de segmentação de instâncias, enquanto todos os outros trabalhos correlatos simplificam o problema para uma detecção de defeitos ou o abordam como uma tarefa de segmentação semântica, o que traz impeditivos, em ambos os casos, para manutenção inteligente e automatizada ou para direcionamento de especialistas humanos na tarefa, conforme abordagem *human-in-the-loop*.

Tabela 1: Trabalhos relacionados que utilizaram o benchmark BSData.

Fonte	Modelo	Tarefa	mIoU	mAP@0.5	F1-Score
Schlagenhauf et al. 2021	Mask R-CNN	Segmentação de instâncias	0,316	–	–
Li et al. 2022	CSPDarkNet-MHSA-FDS	Detecção de objetos	–	0,820	–
Teodoro e Guedes 2023	YOLOv8-small	Detecção de objetos	0,745	0,883	0,877
Landgraf et al. 2023	U-Net + Focal Loss	Segmentação semântica	0,453	–	–
Zhu e Lin 2026	LiMS-Net	Detecção de objetos	–	0,879	0,865

A partir da investigação realizada, foi possível constatar que a literatura carece de avaliação comparativa de métodos do estado da arte de segmentação de instâncias para a tarefa proposta. Esta lacuna motivou o presente estudo.

3. Solução Proposta

Considerando o contexto da Indústria 5.0, em que se deseja monitorar o ambiente fabril de uma indústria com vistas a garantir que o maquinário que nele opera tenha maior resiliência e eficiência operacional, diminuindo os custos de manutenções corretivas e evitando falhas abruptas na linha de produção, tomou-se como referência o componente fuso de esfera, presente em diversas máquinas industriais. Tal componente, como ilustrado na Figura 2, passa a ser monitorado por meio de câmeras, as quais capturam imagens do fuso em intervalos fixos, e as enviam para um servidor computacional que executa um modelo inteligente de segmentação de instâncias, responsável por identificar defeitos do

tipo *pitting*. Em havendo sinalização positiva de defeito, o modelo elabora uma ou mais máscaras de segmentação delimitando a área afetada, informação esta que guiará especialistas humanos na manutenção preventiva ou corretiva da peça, colaborando assim para a abordagem *human-in-the-loop* preconizada pela Indústria 5.0, bem como para diminuir o ônus de inspeções visuais realizadas por operadores humanos em maquinários extensos.

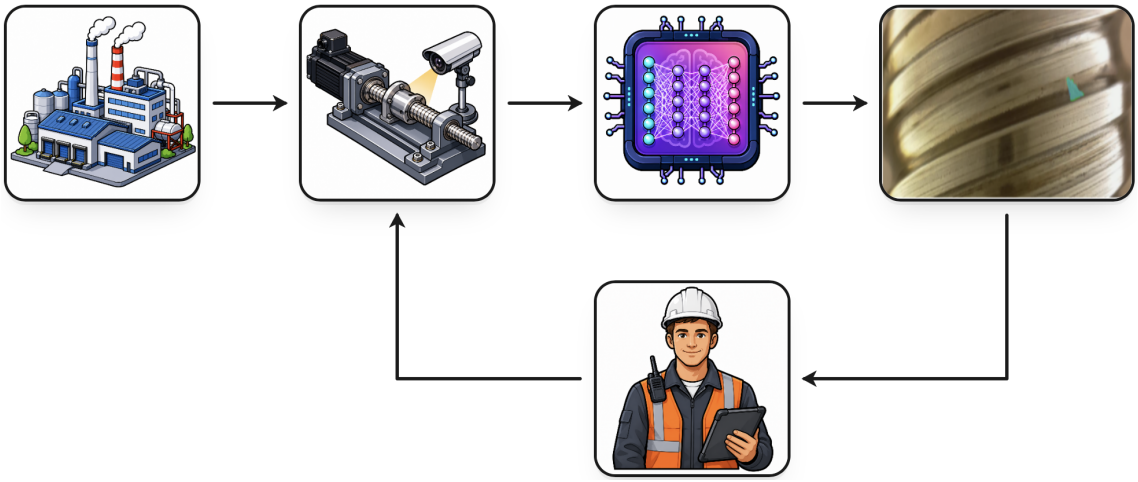


Figura 1: Visão geral da solução proposta

Para alcançar o que se espera, o escopo deste trabalho focou no desenvolvimento e avaliação comparativa dos modelos de Visão Computacional com *Deep Learning* para segmentação de instâncias. Para tanto, contrastou uma abordagem *single-stage* para detecção em tempo real com a YOLO26-seg *versus pipelines* de dois estágios em uma estratégia *zero-shot* com modelos SAM2 e SAM3. Assim, as subseções a seguir detalham a metodologia considerada, contemplando a descrição dos dados experimentais na Seção 3.1, os modelos, parâmetros e hiperparâmetros na Seção 3.2, as métricas de avaliação de desempenho na Seção 3.3 e a infraestrutura computacional na Seção 3.4.

3.1. Dados Experimentais

A base de dados BSData [Schlagenhauf et al. 2021] foi utilizada como fonte de experiência para este trabalho por ser um *benchmark* consolidado na literatura para segmentação de instâncias em maquinário industrial. Após a obtenção do *dataset*, foram identificadas 394 imagens com anotações poligonais válidas, totalizando 485 instâncias de defeitos. As demais imagens sem anotações foram preservadas como amostras de fundo (*background*) para o treinamento dos modelos, ilustrando exemplos conformes com vistas a diminuir a ocorrência de falsos positivos.

No que diz respeito às anotações, os defeitos estavam delimitados por polígonos de formato e dimensões variáveis, o que torna a tarefa de previsão da máscara de segmentação naturalmente desafiadora. Além disso, 187 instâncias possuíam área considerada pequena, com dimensões inferiores a $32 \times 32\,\text{px}$[1], evidenciando a presença de defeitos pequenos e potencialmente difíceis de segmentar. Além da complexidade individual de cada defeito, a distribuição quantitativa das máscaras variava entre as imagens. Embora

[1]Padrão definido pelo MS COCO em `http://cocodataset.org/`.

a maioria das amostras anotadas apresentasse apenas uma instância, havia imagens com múltiplas ocorrências simultâneas, chegando até a seis defeitos em uma única imagem. Para ilustrar essa variabilidade e a dificuldade descrita da tarefa, as Figuras 2 e 3 trazem dois exemplos de defeitos presentes no *dataset*. Na primeira imagem, observa-se o formato complexo da máscara, enquanto na segunda ressalta-se a presença de múltiplas instâncias de defeitos em um BSD sujo, situação típica em um ambiente industrial.

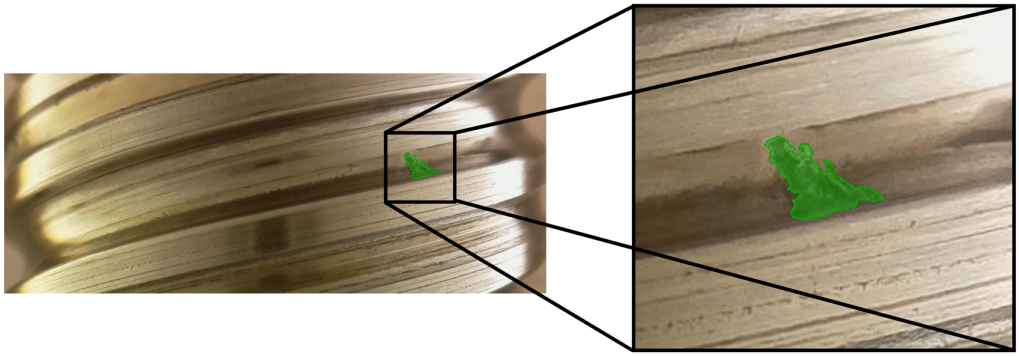


Figura 2: Exemplo de anotação *ground truth* em instância do *dataset* BSData (`01_200907130243_397500_315_crop_2.jpg`).

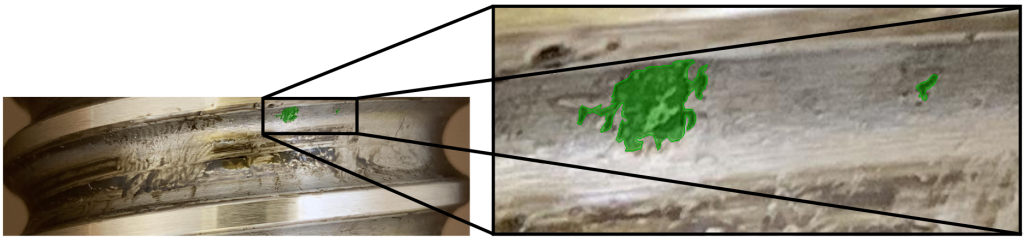


Figura 3: Exemplo de múltiplas anotações *ground truth* em instância do *dataset* BSData (`KGT2-V-295-A08_crop_0.jpg`).

3.2. Modelos Utilizados

Para lidar com o problema de segmentação de instância proposto, consideraram-se dois paradigmas distintos dentre modelos de *Deep Learning*: o uso de modelos especializados, treinados diretamente a partir de exemplos do conjunto-alvo; e o uso de modelos pré-treinados de larga escala, avaliados na tarefa em regime *zero-shot*. Para o primeiro paradigma, considerou-se o uso do modelo YOLO26-seg, enquanto no segundo paradigma considerou-se os modelos SAM2 e SAM3, os quais são descritos a seguir, com suas respectivas parametrizações.

Baseado em redes neurais convolucionais (CNNs, do inglês *Convolutional Neural Networks*), o modelo YOLO26 possui um *backbone* extrator de características baseado na CSP-DarkNet, *neck* responsável pela fusão bidirecional de características de diferentes dimensões e *head* otimizado para efetuar previsões de máscaras de segmentação eliminando a etapa de Supressão Não-Máxima [Ultralytics 2026], o que colabora para um desempenho em ambientes reais de implantação, tais como em dispositivos de Computação de Borda para monitoramento industrial. Embora tipicamente reconhecido para detecção de

objetos, no escopo desse trabalho será utilizada a variante YOLO26-seg para segmentação de instâncias, cuja previsão consiste em máscaras que contornam a forma e a região do defeito no BSD, sendo relevantes para a análise visual.

Da variante YOLO26-seg foram consideradas cinco versões com tamanhos distintos, a citar: *nano*, *small*, *medium*, *large* e *xlarge*, permitindo avaliar o *trade-off* entre desempenho preditivo e custo computacional. O treinamento desses modelos considerou entradas com tamanho 640×640 px, 600 épocas, *early stopping* com paciência igual a 100, tamanhos de *batch* calculados de forma proporcional ao número de parâmetros do modelo, em que as versões *nano*, *small* e *medium* usaram *batch size* igual a 16, a versão *large* usou igual a 8 e a *xlarge* igual a 6. O otimizador adotado foi o AdamW com taxa de aprendizado inicial 10^{-2}.

O *Segment Anything Model 2* (SAM2) é um modelo de fundação para segmentação visual *promptable* em imagens e vídeos com o objetivo de generalizar o paradigma de segmentação interativa com dimensão temporal [Ravi et al. 2024]. Arquiteturalmente, o modelo é composto por um codificador de imagens hierárquico (*image encoder*) baseado na Hiera, um *Vision Transformer* hierárquico que dispensa componentes específicos de visão presentes em arquiteturas anteriores [Ryali et al. 2023], pré-treinado de forma auto-supervisionada com *Masked Autoencoders* (MAE) e responsável por extrair características multiescala; um codificador de *prompts* (*prompt encoder*) que processa entradas esparsas, como pontos e caixas delimitadoras, e também entradas densas, como máscaras; e um decodificador de máscaras (*mask decoder*) leve, fundamentado em atenção bidirecional, que produz a máscara final juntamente com uma estimativa de *Intersection over Union* (IoU) usada para resolver ambiguidades de segmentação. Neste trabalho, o SAM2 foi empregado como refinador de máscaras em regime *zero-shot*, sem retreinamento específico para o domínio-alvo, recebendo como *prompt* as caixas delimitadoras previstas na etapa anterior do *pipeline*. Do SAM2 foram consideradas quatro variantes que diferem unicamente na capacidade do codificador Hiera, a citar: *tiny*, *small*, *base_plus* e *large*, distintas em capacidade representacional e custo computacional, o que permite avaliar o *trade-off* entre qualidade de segmentação e desempenho de inferência.

O *Segment Anything Model 3* (SAM3) é a versão mais recente dos modelos SAM para ampliar a segmentação *promptable* ao paradigma de *Promptable Concept Segmentation* (PCS), no qual o modelo detecta, segmenta e rastreia todas as instâncias de um conceito visual em imagens e vídeos a partir de *prompts* conceituais, como compostos nominais curtos, exemplares visuais ou a combinação de ambos [Carion et al. 2025]. Diferentemente do SAM2, voltado à segmentação visual a partir de *prompts* geométricos como pontos, caixas e máscaras, o SAM3 adota uma arquitetura unificada composta por um detector em nível de imagem e um rastreador baseado em memória que compartilham um único *backbone*. O detector segue o paradigma DETR (*Detection Transformer*), de natureza baseada em *queries*, sendo condicionado por texto, geometria e exemplares visuais, ao passo que o rastreador herda a arquitetura codificador-decodificador do SAM2. Uma contribuição central do modelo é o desacoplamento entre reconhecimento e localização por meio de uma *presence head*, responsável por estimar a probabilidade de o conceito consultado estar presente na imagem, o que eleva a acurácia de detecção, especialmente diante de conceitos negativos desafiadores. No escopo deste trabalho, embora o SAM3 possua suporte a *prompts* textuais e exemplares visuais, sua utilização foi padroni-

zada com a mesma estratégia aplicada ao SAM2, isto é, a segmentação a partir de caixas delimitadoras, padronização que permite comparar os modelos *promptable* sob condições equivalentes de entrada.

Nos modelos SAM2 e SAM3, o YOLO26 *nano* foi utilizado para etapa de detecção, fornecendo as caixas delimitadoras usadas como *prompt*, de modo que o *pipeline* completo em dois estágios consiste em, primeiro, o YOLO26 *nano* predizer as caixas candidatas e, em seguida, o SAM2 ou o SAM3 gerar a máscara de segmentação correspondente a cada caixa. A escolha da variante *nano* como geradora de *prompts* foi motivada por sua relação favorável entre custo computacional e desempenho observado nos experimentos preliminares, uma vez que, por possuir menor número de parâmetros e menor custo de inferência, essa variante reduz o tempo da etapa de detecção no *pipeline* em dois estágios.

3.3. Avaliação de Desempenho

As imagens anotadas do conjunto BSData foram divididas segundo um particionamento do tipo *holdout*, no qual 70% das amostras foram utilizadas para treinamento, 10% para validação e 20% para teste. Para avaliar a robustez dos modelos e reduzir a dependência dos resultados em relação a uma única divisão dos dados, foram efetuadas três repetições do experimento proposto. As repetições diferenciam-se quanto aos exemplos presentes em cada partição, cuja seleção foi feita de maneira aleatória segundo uma distribuição uniforme. Essa estratégia também permitiu avaliar o comportamento dos modelos diante de diferentes composições do conjunto de teste, incluindo a presença de objetos pequenos, com dimensões inferiores a 32×32 px, que representam um desafio particular para a segmentação de defeitos no BSData.

O desempenho dos modelos foi aferido na partição de teste utilizando métricas compatíveis com a natureza da tarefa de segmentação de instâncias, em que foram coletadas: precisão, revocação, F_1-*Score*, mAP (*Mean Average Precision*) considerando o limiar de IoU $\geq 0{,}5$, mIoU (*mean IoU*) e o FPS (*Frames per Second*).

3.4. Infraestrutura Computacional e Parametrização

Todos os experimentos foram executados em um servidor com sistema operacional Ubuntu 24.04.4 LTS, equipado com processador Intel Core i7-8700 de 3,20 GHz, com 6 núcleos físicos e 12 *threads*, 54 GiB de memória RAM e duas GPUs NVIDIA GeForce GTX 1080 Ti, cada uma com 11 GiB de memória dedicada. O ambiente computacional utilizou driver NVIDIA 560.35.03, CUDA 12.6, Python 3.12.3, PyTorch 2.6.0 com suporte a CUDA 12.6 e Ultralytics 8.4.30.

No treinamento dos modelos YOLO26-seg foi utilizado treinamento distribuído nas duas GPUs disponíveis. O otimizador foi configurado automaticamente pela biblioteca Ultralytics, utilizando AdamW com taxa inicial de aprendizado ($lr_0 = 0{,}01$). O treinamento foi limitado a 600 épocas, com parada antecipada configurada com paciência de 100 épocas. O uso de AMP (*Automatic Mixed Precision*) foi desabilitado para priorizar a estabilidade do treinamento. Embora essa técnica combine precisão mista de 32 e 16 bits (FP32 e FP16) para reduzir o consumo de memória e acelerar execuções em GPU, foram observadas instabilidades e erros de falta de memória durante as execuções com modelos YOLO26-seg de maior porte. Como as GPUs utilizadas são baseadas na

arquitetura Pascal e não possuem *Tensor Cores*, os ganhos de desempenho esperados seriam limitados. Por isso, optou-se por manter a precisão padrão e ajustar o *batch size* das variantes maiores conforme a memória disponível.

Os experimentos com YOLO26-seg foram repetidos com três sementes aleatórias, totalizando quinze treinamentos, considerando as cinco variantes avaliadas: *nano*, *small*, *medium*, *large* e *xlarge*. Para os modelos SAM2 e SAM3, não houve atualização de pesos; ambos foram avaliados diretamente a partir de *checkpoints-base*, em regime de inferência. Também foram efetuadas três repetições experimentais para diminuir o viés da flutuação estocástica. As parametrizações dos modelos encontram-se dispostas na Tabela 2.

Tabela 2: Principais configurações de treinamento e inferência dos modelos avaliados.

Configuração	YOLO26-seg	SAM2	SAM3
Resolução de treinamento/inferência	640×640	1024×1024	1024×1024
Prompt	—	bbox do YOLO nano	bbox do YOLO nano
Épocas máximas	600	—	—
Paciência	100	—	—
Otimizador	AdamW ($lr_0 = 0{,}01$)	—	—
AMP	Desabilitado	—	—

4. Resultados e Discussão

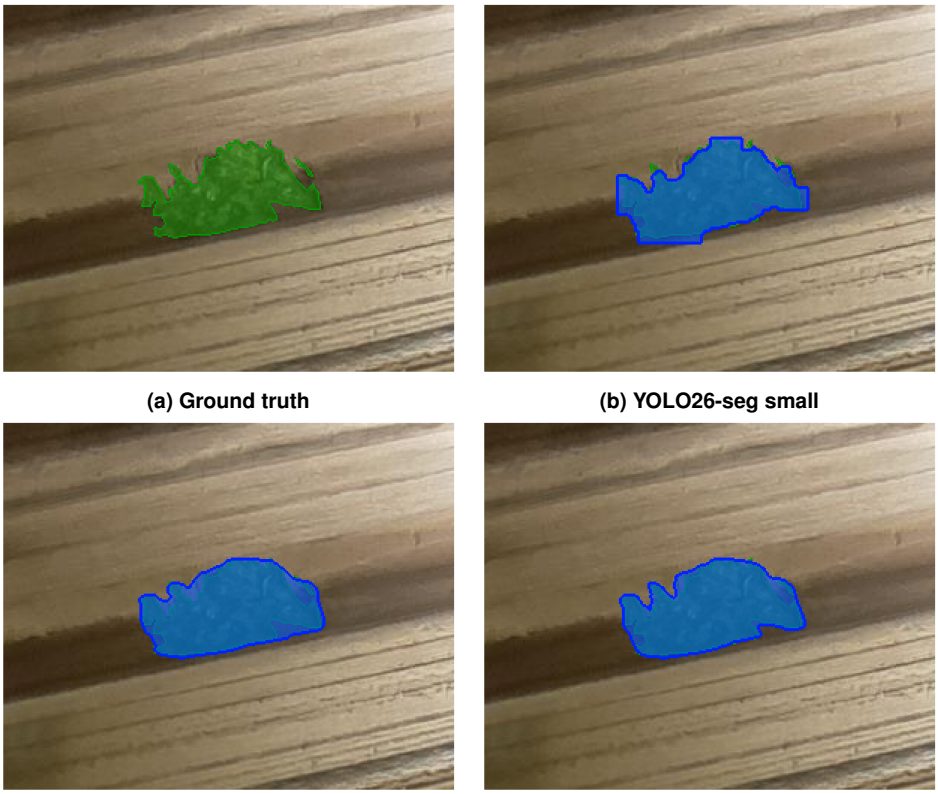
Os experimentos foram conduzidos conforme descrito na metodologia e os resultados obtidos encontram-se dispostos na Tabela 3. A discussão dos resultados levará em conta inicialmente duas métricas cruciais no contexto da segmentação de instâncias, o mIoU e o mAP@$0{,}5$ cuja semântica é recapitulada como segue: o mIoU denota a média do IoU entre as máscaras de segmentação previstas pelo modelo e as anotações *ground truth*, quantificando a precisão do alinhamento espacial dos *pixels* de cada instância, enquanto o mAP@$0{,}5$, por sua vez, expressa o balanço global entre precisão e revocação na identificação das instâncias, considerando um acerto sempre que a correspondência espacial atinge ou supera o patamar mínimo de $50\,\%$ [Padilla et al. 2020].

Os maiores valores de mIoU foram observados nos modelos SAM2 e SAM3 em comparação com todas as variantes da família YOLO, independentemente da escala de parâmetros destas últimas. Esse comportamento ressalta que a alta capacidade de generalização espacial dos modelos de fundação, decorrente do pré-treinamento em larga escala, mesmo operando em regime *zero-shot*, supera a acurácia geométrica obtida pelo aprendizado supervisionado restrito ao domínio industrial mapeado. No tocante ao mAP@$0{,}5$, apenas o SAM2 Hiera-L se equiparou aos modelos YOLO26, em que a superioridade numérica foi observada no modelo SAM2 Hiera-T, sendo igual a $0{,}799$. Este modelo é $3{,}09\,\%$ superior ao melhor desempenho observado em uma YOLO26-seg, versão *small*, e $0{,}75\,\%$ superior ao SAM2 Hiera-S, segundo melhor modelo dentre o SAM2 e o SAM3. Cabe destacar ainda que a solução elencada, o SAM2 Hiera-T, além de apresentar o melhor mAP@$0{,}5$, é a variante mais compacta do SAM2, com $82{,}66\,\%$ menos parâmetros que o SAM2 Hiera-L.

Tabela 3: Síntese dos resultados experimentais (média de 3 experimentos por modelo).

	Treinamento		Teste					
Modelo	**Parâm. (M)**	**Tempo (h)**	**mIoU**	**mAP@0,5**	**Precisão**	**Revocação**	**F1-Score**	**FPS**
YOLO26-seg nano	3,4	1,9	0,569	0,749	0,892	0,692	0,778	77,6
YOLO26-seg small	11,8	3,3	0,586	0,775	0,862	0,758	0,803	70,4
YOLO26-seg medium	23,5	3,1	0,582	0,768	0,911	0,707	0,795	42,7
YOLO26-seg large	27,9	9,6	0,582	0,774	0,899	0,746	0,815	36,0
YOLO26-seg xlarge	62,0	21,2	0,576	0,771	0,928	0,738	0,820	23,0
SAM2 Hiera-T	38,9	–	0,632	0,799	0,915	0,731	0,811	8,6
SAM2 Hiera-S	46,0	–	0,629	0,793	0,915	0,731	0,811	7,7
SAM2 Hiera-B+	80,8	–	0,627	0,789	0,892	0,742	0,808	4,8
SAM2 Hiera-L	224,4	–	0,614	0,771	0,897	0,713	0,793	2,4
SAM3	860,1	–	0,634	0,790	0,914	0,735	0,813	–

Visando analisar qualitativamente a capacidade de segmentação das arquiteturas avaliadas, foram selecionadas instâncias do conjunto de testes para inspeção visual. A primeira amostra, denotada na Figura 4, compreende duas máscaras de *ground truth*: uma diminuta e outra com formato complexo. O modelo YOLO26-seg *small* previu uma máscara de contorno serrilhado com área superestimada. No caso do SAM2 Hiera-T, a predição aproximou-se de um envoltório convexo, apresentando *pixels* excedentes, enquanto o SAM3 gerou uma máscara inferior à área real do defeito. Nota-se que nenhuma das abordagens foi capaz de isolar e detectar os dois defeitos simultaneamente.

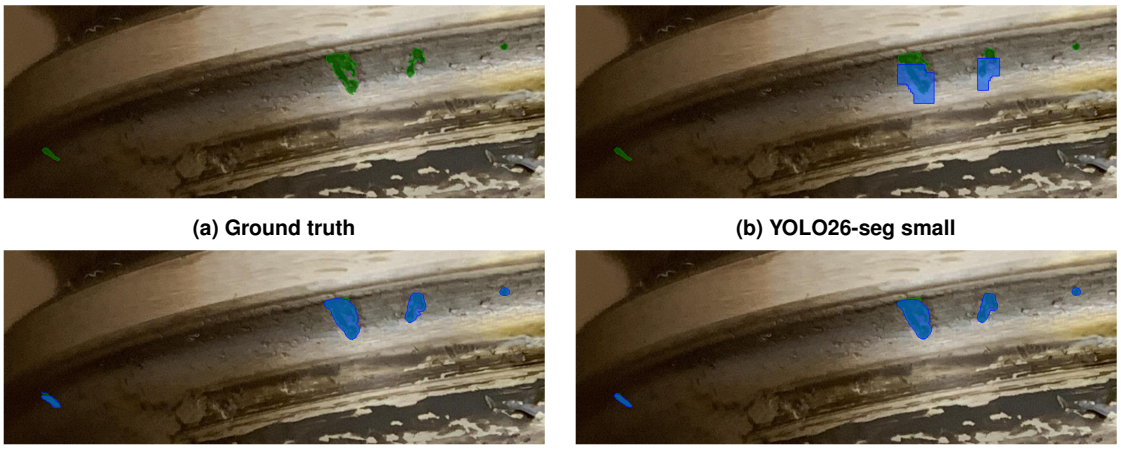


(a) Ground truth **(b) YOLO26-seg small**

(c) SAM2 Hiera-T **(d) SAM3**

Figura 4: Comparação visual das predições dos melhores modelos avaliados em uma imagem do conjunto de teste (`01_200907090403_462500_045_crop_2.jpg`).

A segunda amostra para inspeção visual é denotada na Figura 5, em que há quatro defeitos em um contexto mais desafiador porque o BSD encontra-se sujo. O modelo YOLO26-seg *small* detectou apenas dois defeitos e a máscara produzida possuía baixa intersecção com a solução desejada. Ambos os modelos SAM2 Hiera-T e SAM3 detectaram os quatro defeitos, diferindo apenas na sobreposição da máscara proposta e da solução desejada, em que o primeiro mostrou-se quantitativamente superior.



(a) Ground truth (b) YOLO26-seg small

(c) SAM2 Hiera-T (d) SAM3

Figura 5: Comparação visual das predições dos melhores modelos avaliados em uma imagem do conjunto de teste (`KGT2-V-302-A08_crop_0.jpg`).

No que tange à mensuração do FPS, cumpre destacar que, para a arquitetura SAM2, computou-se o tempo de execução latente de todo o *pipeline* de inferência. Quanto ao SAM3, essa métrica não pôde ser aferida devido a inconsistências de compatibilidade entre os *frameworks* utilizados e o *hardware* disponível. A partir dos dados observados, constata-se que todas as variantes da família YOLO26, à exceção da versão *xlarge*, atendem ao critério de processamento em tempo real (valor igual ou superior a 30 FPS). Em contrapartida, os modelos SAM2, devido à magnitude de parâmetros, impõem um ônus computacional proibitivo para demandas em tempo real e restrições de Computação de Borda, dois critérios importantes para sistemas de monitoramento industrial inteligente.

Diante das premissas de resiliência de maquinário preconizadas pela Indústria 5.0, o ônus computacional da arquitetura SAM2 Hiera-T é contraposto por sua robustez adaptativa, a qual pode ser verificada pela segmentação mesmo em cenários de oclusão por sujeira no BSD, conforme Figura 5. Adicionalmente, o acoplamento de sensores de imagem (câmeras) a infraestruturas de computação centralizada de alto desempenho (servidores) subsidia o monitoramento em tempo real. Em ocorrências de fadiga superficial por *pitting* no BSD, esse arranjo viabiliza estratégias de contenção baseadas na estratégia detecção-isolamento, mitigando o desperdício. Por fim, a delimitação precisa das fronteiras das máscaras pelo modelo mitiga a carga cognitiva do operador, instrumentalizando o especialista humano para intervenções precisas na linha de produção, de acordo com o modelo de governança *human-in-the-loop*.

5. Considerações Finais

Visando contribuir para a inspeção visual automatizada de componentes de maquinário no contexto da Indústria 5.0, este trabalho retomou a tarefa de segmentação de instâncias de defeitos do tipo *pitting* em fusos de esfera a partir do *benchmark* BSData [Schlagenhauf et al. 2021], uma vez que a literatura recente havia majoritariamente reformulado o problema como detecção de objetos ou segmentação semântica. Para tanto, conduziu-se uma avaliação comparativa entre dois paradigmas, contrastando modelos especializados treinados de ponta a ponta, representados pela família YOLO26-seg em abordagem *single-stage*, e modelos de fundação aplicados em regime *zero-shot*, representados pelos *pipelines* de duas etapas com SAM2 e SAM3.

Os resultados revelaram melhor capacidade de produção das máscaras de segmentação de defeitos pelos modelos de fundação SAM2 e SAM3 frente aos modelos YOLO26-seg. Em particular, elencou-se o SAM2 Hiera-T como solução de referência, com mAP@$0{,}5$ de $0{,}799$ e mIoU de $0{,}632$ com detecções a $8{,}6$ FPS. Observou-se que o modelo possui robustez adaptativa e que, embora aquém do desejado para detecções em tempo real, mostra-se adequado para o modelo de governança *human-in-the-loop* e para uso mediante estratégia detecção-isolamento. Ressalta-se ainda que a solução identificada supera o *baseline* no *benchmark* BSData, que reporta mIoU igual a $0{,}316$ [Schlagenhauf et al. 2021], ainda que sob protocolo experimental distinto, evidência sugestiva de um avanço no estado da arte para a tarefa considerada.

Em investigações futuras, projeta-se o *fine-tuning* dos modelos SAM no domínio industrial avaliado e a exploração de novas modalidades de *prompt* no SAM3. Adicionalmente, pretende-se investigar métodos de poda e quantização para reduzir o número de parâmetros e a latência dos modelos de fundação, com vistas a viabilizar sua utilização em dispositivos de Computação de Borda para fins de detecção em tempo real.

Transparência da IA e Considerações Éticas

Os ícones apresentados na Figura 1 foram gerados com auxílio do modelo DALL·E. A revisão gramatical e ortográfica foi realizada com auxílio dos modelos GPT-5.5 Thinking e Gemini 3.5 Flash. Além disso, todas as revisões automáticas foram complementadas por revisão humana, a fim de garantir a precisão e a qualidade do texto.

Agradecimentos

Os autores agradecem ao Grupo de Pesquisa em Sistemas Inteligentes da Universidade do Estado do Amazonas pelo suporte à infraestrutura computacional utilizada neste trabalho. EBG agradece o apoio financeiro da CAPES por meio do Projeto COFECUB-PJ3126170P.

Referências

Carion, N., Gustafson, L., Hu, Y.-T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K. V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.-H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H. K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., and Feichtenhofer, C. (2025). SAM 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719.

Landgraf, J., Kompenhans, M., Christ, T., Roland, T., and Heinig, A. (2023). Computer vision for industrial defect detection. In *Materials Research Proceedings*, volume 25 of *Sheet Metal 2023*, pages 371–378. Materials Research Forum LLC.

Leng, J., Sha, W., Wang, B., Zheng, P., Zhuang, C., Liu, Q., Wuest, T., Mourtzis, D., and Wang, L. (2022). Industry 5.0: Prospect and retrospect. *Journal of Manufacturing Systems*, 65:279–295.

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)*, page 237–242. IEEE.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. (2024). SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y., and Feichtenhofer, C. (2023). Hiera: A hierarchical vision transformer without the bells-and-whistles. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 29441–29454.

Schlagenhauf, T., Landwehr, M., and Fleischer, J. (2021). Industrial machine tool component surface defect dataset. *Data in Brief*, 39:107643.

Singh, T., Singh, D., Singh, C. D., and Singh, K. (2023). *Industry 5.0*, chapter 2, pages 21–45. John Wiley & Sons, Ltd.

Teodoro, G. S. and Guedes, E. B. (2023). Detecção inteligente de defeitos para indústria 4.0: Um estudo de caso com modelos Single-Shot YOLO. In *Proceedings do Simpósio Brasileiro de Sistemas Elétricos*, Manaus - AM, Brasil. SBA Sociedade Brasileira de Automática.

Ultralytics (2026). Instance segmentation with ultralytics yolo26. Acesso em: 27 maio 2026.

Zhu, Y. and Lin, Y.-C. (2026). Lims-net: Lightweight metal surface defect detection network. *Displays*, 91:103227.