

## **Sistema de visão computacional para inspeção automática de defeitos em bordados e estampas na indústria têxtil**

Computer vision system for automated defect inspection in embroidery and prints in the textile industry

Bruno Lucas Rodrigues da Silva <sup>1</sup>

Diogo David Macêdo Sena <sup>2</sup>

Jose Lucas da Silva Gomes <sup>3</sup>

Rafael Avelino dos Santos <sup>4</sup>

Stanley de Oliveira Souza <sup>5</sup>

Filipe de Carvalho Pinto Raulino <sup>6</sup>

Ivanilson França Vieira Júnior <sup>7</sup>

Diego da Silva Pereira <sup>8</sup>

### **Resumo**

A identificação de defeitos em bordados e estampas durante o processo de produção representa um desafio crítico para o controle de qualidade na indústria de customização têxtil, especialmente devido à variabilidade dos padrões e da dependência de inspeção visual humana, que é suscetível à subjetividade e à fadiga. Nesse sentido, a aplicação de técnicas de visão computacional com o uso de câmeras inteligentes configura-se como uma alternativa promissora para automatizar e padronizar esse processo. Este estudo visa o desenvolvimento e a validação de um sistema automatizado para a detecção de anomalias em tempo real, utilizando técnicas de visão computacional. Trata-se de um estudo de caráter aplicado, com

<sup>1</sup> Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: brunollucas99@gmail.com

<sup>2</sup> Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: diogodmsena@gmail.com

<sup>3</sup> Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: j1503lucas@gmail.com

<sup>4</sup> Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: rafaelavelino0908@gmail.com

<sup>5</sup> Discente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: ichosenstan@gmail.com

<sup>6</sup> Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: filipe.raulino@ifrn.edu.br

<sup>7</sup> Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: ivanilson.junior@ifrn.edu.br

<sup>8</sup> Docente do Curso de Formação Inicial e Continuada (Curso FIC) em Residência Tecnológica em Software Embarcado, na modalidade a distância. e-mail: diego.pereira@ifrn.edu.br

abordagem tecnológica, desenvolvido a partir da integração de técnicas de visão computacional e frameworks consolidados no mercado. A metodologia compreendeu a definição da arquitetura do sistema, a pesquisa bibliográfica, o treinamento de modelos de *Deep Learning*, a validação em um ambiente simulado, bem como a análise de repositórios técnicos e a validação de padrões de detecção. O sistema proposto demonstrou ser capaz de identificar falhas com alta precisão, contribuindo para a potencial redução de retrabalho, o aumento da padronização dos produtos e a otimização do controle de qualidade. Conclui-se que a visão computacional é uma estratégia eficaz para elevar a eficiência produtiva e a competitividade das empresas do setor.

**Palavras-chave:** Visão Computacional; Inspeção de Qualidade; Deep Learning; Redes Neurais Siamesas; YOLO; Indústria Têxtil.

## **Abstract**

The identification of defects in embroidery and textile printing during the production process represents a critical challenge for quality control in the textile customization industry, particularly due to pattern variability and the reliance on human visual inspection, which is susceptible to subjectivity and fatigue. In this context, the application of computer vision techniques using smart cameras emerges as a promising alternative for automating and standardizing this process. This study aims to develop and validate an automated system for real-time anomaly detection using computer vision techniques. It is an applied study with a technological approach, developed through the integration of computer vision techniques and well-established market frameworks. The methodology comprised the definition of the system architecture, bibliographic research, Deep Learning model training, validation in a simulated environment, as well as the analysis of technical repositories and the validation of detection standards. The proposed system proved capable of identifying defects with high accuracy, contributing to the potential reduction of rework, increased product standardization, and optimization of quality control processes. It is concluded that computer vision constitutes an effective strategy for enhancing production efficiency and the competitiveness of companies in the textile sector.

**Keywords:** Computer Vision; Quality Inspection; Deep Learning; Siamese Neural Networks; YOLO; Textile Industry.

## **1 INTRODUÇÃO**

O Brasil ocupa posição de destaque no cenário global da indústria têxtil e de confecção, apresentando, de acordo com estimativas de entidades representativas do setor, elevada capacidade produtiva, o que evidencia a relevância econômica dessa cadeia industrial no país (Fieg, 2018). Nesse contexto, numa avaliação temporal, o desenvolvimento e o aperfeiçoamento de técnicas e ferramentas têm impulsionado mudanças significativas na forma como as pessoas produzem, consomem e se relacionam, transformando continuamente os modos de vida e a organização da sociedade (Oliveira, 2021).

Atualmente, a indústria 4.0 tem redefinido a gestão da qualidade ao integrar tecnologias como inteligência artificial, internet das coisas e big data, promovendo processos produtivos mais eficientes, com maior capacidade de monitoramento e previsão (Pereira, 2023).

O papel da visão computacional envolve o desenvolvimento de algoritmos e técnicas que imitam a percepção visual do homem, o que permite que máquinas sejam capazes de analisar, reconhecer e interpretar dados visuais. Nesse sentido, sistemas de visão computacional podem, a partir da análise de vídeos e imagens, extrair informações valiosas e tomar decisões inteligentes (Khang *et al.*, 2025).

O estudo de caso deste trabalho é conduzido em uma fábrica de grande porte localizada no Nordeste do Brasil, especializada no ramo têxtil, responsável pela produção de milhões de peças de vestuário ao ano. A análise preliminar identificou que uma parcela significativa dos desperdícios está concentrada no setor de estampas e bordados, onde a inspeção manual apresenta elevada subjetividade e está sujeita à fadiga cognitiva dos operadores, por se tratar de um trabalho intenso e repetitivo. Diante desse cenário, o objetivo central desta pesquisa é desenvolver e implementar um sistema de visão computacional capaz de identificar anomalias em estampas com alta precisão, operando de forma integrada ao ambiente produtivo em tempo real.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Para compreender a solução proposta, é necessário introduzir dois conceitos fundamentais: Machine Learning (ML) e Deep Learning (DL), que constituem a base teórica dos sistemas modernos de visão computacional. O ML é um subcampo da Inteligência Artificial que desenvolve algoritmos capazes de aprender padrões diretamente a partir de dados, ajustando seus parâmetros conforme a experiência acumulada e melhorando o desempenho em tarefas específicas sem programação explícita. Como afirma Mitchell (1997), um sistema de aprendizado é aquele que melhora seu desempenho na tarefa T com base na experiência E, avaliada por uma métrica de desempenho P.

O Deep Learning, por sua vez, é um ramo do ML que utiliza redes neurais profundas formadas por múltiplas camadas, permitindo aprender representações cada vez mais abstratas dos dados. Essa abordagem elimina a necessidade de extrair manualmente características relevantes e viabiliza desempenho superior em tarefas como detecção de objetos, análise de imagens e reconhecimento de fala. Conforme LeCun, Bengio e Hinton (2015), o DL permite

que redes neurais aprendam representações complexas em grande escala, impulsionando os avanços recentes na área.

A visão computacional é um ramo da Inteligência Artificial que utiliza técnicas do aprendizado de máquina para permitir que sistemas automatizados interpretem e extraiam informações de imagens e vídeos. Diferentemente da programação tradicional, onde a máquina segue comandos definidos pelo usuário, os algoritmos de aprendizado de máquina aprendem padrões a partir de dados e, ao receber novas entradas, são capazes de realizar inferências baseadas em probabilidades de acordo com o que foi aprendido. Segundo Szeliski (2022), a visão computacional moderna depende fortemente de modelos estatísticos capazes de generalizar a partir de exemplos, substituindo abordagens rigidamente baseadas em regras por métodos de aprendizado.

Para o desenvolvimento do sistema, é utilizada a arquitetura de Redes Neurais Siamesas (Siamese Neural Networks - SNNs) que faz parte do campo de Deep Learning introduzidas por Bromley *et al.*, (1994), aplicadas em tarefas que envolvem a comparação por similaridade entre pares de entradas. O conceito fundamental de uma SNN é sua estrutura composta por duas sub-redes idênticas que operam em paralelo. Ambas possuem a mesma arquitetura (sejam redes neurais convolucionais ou recorrentes) e compartilham essencialmente os mesmos pesos e parâmetros (Harandi *et al.*, 2017).

As SNNs são uma aplicação proeminente do paradigma de Aprendizado de Métrica (Metric Learning). A rede é treinada para criar um espaço de características (o espaço de embeddings) onde a distância entre vetores é semanticamente significativa. Após a extração das características, uma camada de distância/similaridade (geralmente baseada em Distância Euclidiana, Distância Cosseno ou Distância Manhattan) é aplicada aos embeddings de saída ( $h_1$  e  $h_2$ ) para quantificar o quão semelhantes são as entradas originais (Chopra *et al.*, 2005).

O objetivo do Treinamento é maximizar a dissimilaridade, a rede deve ser treinada para aumentar a distância (distância grande) entre os embeddings de entradas que pertencem a classes diferentes, minimizar a dissimilaridade, a rede deve ser treinada para reduzir a distância (distância pequena) entre os embeddings de entradas que pertencem à mesma classe. A Função de Perda Contraste (Contrastive Loss) é o que impulsiona o aprendizado da métrica, ajustando os pesos compartilhados para atingir o objetivo de agrupar o semelhante e separar o diferente.

A principal vantagem da SNN reside na sua eficácia em tarefas de Aprendizado com Poucas Amostras (Few-Shot Learning) ou Aprendizado One-Shot (One-Shot Learning). Ao invés de aprender a classificar todas as categorias possíveis, a Rede Neural Siamesa aprende

uma métrica de similaridade robusta. Isso permite que ela se generalize para novas classes vistas com apenas uma ou poucas amostras, simplesmente comparando a nova amostra com um exemplar de referência da nova classe.

Complementarmente à análise de similaridade focada na qualidade, o sistema necessita de uma etapa prévia de localização espacial e quantificação dos itens produzidos. Para esta tarefa de detecção de objetos em tempo real, fundamenta-se o uso da arquitetura YOLO (You Only Look Once). Diferentemente de métodos tradicionais de dois estágios que primeiro propõem regiões de interesse para depois classificá-las, a arquitetura YOLO reestrutura a detecção de objetos como um problema único de regressão (Ultralytics, 2025).

A rede convolucional do YOLO divide a imagem de entrada em uma grade. Para cada célula da grade, a rede prevê simultaneamente as coordenadas das caixas delimitadoras (bounding boxes), o grau de certeza da presença do objeto (objectness score) e a probabilidade da classe à qual o objeto pertence. Essa abordagem unificada permite que a rede 'olhe apenas uma vez' para a imagem completa, extraíndo características globais e locais simultaneamente, o que garante a baixa latência necessária para acompanhar a velocidade dos carrosséis industriais (Ultralytics, 2025).

Para a funcionalidade específica de contagem, a detecção isolada frame a frame não é suficiente, exigindo a implementação de algoritmos de Rastreamento de Múltiplos Objetos (MOT - Multi-Object Tracking). Fundamentado em métodos como o ByteTrack ou BoT-SORT, o sistema realiza a associação de dados (data association), vinculando as detecções do YOLO em frames consecutivos baseando-se na sobreposição espacial (IoU - Intersection over Union) e nas características visuais. Isso atribui uma identidade única (ID) persistente a cada berço enquanto ele atravessa o campo de visão, permitindo a contagem precisa através de linhas virtuais de verificação, evitando contagens duplicadas causadas por paradas ou oscilações do carrossel (Ultralytics, 2025).

### **3 METODOLOGIA**

Este trabalho caracteriza-se como uma pesquisa aplicada com abordagem tecnológica, orientada ao desenvolvimento e validação de um sistema de visão computacional para uso em ambiente industrial. O desenvolvimento foi conduzido de forma iterativa, com ciclos de implementação e validação, inspirado em práticas ágeis aplicadas a esse tipo de sistema (Szeliński, 2022), com o objetivo de facilitar ajustes em parâmetros do sistema e reduzir riscos de integração no ambiente de produção.

A solução foi projetada como um pipeline modular composto por dois subsistemas integrados: (i) um módulo de contagem automática e (ii) um módulo de inspeção de falhas por comparação de peças, alinhando rastreabilidade produtiva e controle de qualidade em tempo real. Essa arquitetura segue princípios de pipelines industriais de visão computacional que enfatizam separação de responsabilidades e rastreabilidade (Bradski; Kaehler, 2008). Em termos de fluxo, o sistema realiza a aquisição de vídeo e imagens, aplica técnicas de detecção para localizar o item, executa rastreamento para manter a identidade do objeto ao longo do tempo e efetua a contagem por evento de cruzamento. Quando aplicável, a região de interesse é recortada para a etapa de comparação e emissão de veredito de qualidade.

Os dados de entrada foram obtidos por câmeras posicionadas de modo a capturar continuamente o campo de visão do carrossel/estação, operando tipicamente a 30 FPS em resolução 1280x720 ou 1920x1080, conforme requisitos de operação e limites computacionais. Para aumentar a reprodutibilidade e mitigar variações indesejadas de perspectiva e iluminação, buscou-se padronizar o posicionamento e as condições de captura por estação. Tal padronização é crucial, visto que essas variáveis influenciam diretamente a robustez do rastreamento e a estabilidade da comparação visual.

No módulo de contagem, a etapa central consiste na detecção de objetos em tempo real utilizando YOLO v8, uma das arquiteturas de detecção de objetos com melhor desempenho em cenários de alta velocidade e baixa latência (Jocher *et al.*, 2023). Como a detecção por frame não garante unicidade temporal, empregou-se rastreamento multi-objeto (MOT), estratégia amplamente utilizada em cenários industriais (Bewley *et al.*, 2016), mantendo um identificador persistente (ID) para cada item enquanto ele atravessa o campo de visão. A contagem é então realizada por meio de uma linha virtual configurável, na qual o sistema incrementa o contador quando o objeto cruza essa referência respeitando regras de direção e persistência, reduzindo ocorrências de contagem duplicada em situações como paradas, oscilações e re-identificações temporárias. Por fim, os eventos são registrados com timestamp e podem ser acompanhados por saídas auxiliares (como logs e visualizações de depuração), o que favorece a auditoria e a análise do processo.

No módulo de inspeção de falhas, a estratégia adotada baseia-se em comparação por similaridade entre a peça em produção e uma referência considerada adequada (denominada de “bandeira”), permitindo detectar desvios visuais associados a defeitos do processo. Para isso, após localizar a região de interesse, o sistema extrai um vetor de características e calcula uma medida de distância ou similaridade em relação ao conjunto de referências, definindo o veredito por meio de um limiar configurável (threshold). Quando necessário, para aumentar a

sensibilidade a falhas localizadas, a comparação pode ser conduzida em sub-regiões (*patches*) da área inspecionada, de modo que o maior desvio observado em uma região específica contribua para a decisão final de aprovação ou reprovação.

O sistema foi projetado para operar em ambiente Linux, empregando um *pipeline* desenvolvido em Python. Este *pipeline* utiliza bibliotecas especializadas (como OpenCV) para a aquisição e o processamento de imagens e vídeos, além de *frameworks* de inferência para os modelos de detecção e comparação. O foco principal do desenvolvimento foi garantir baixa latência e operação ininterrupta no ambiente de produção (*chão de fábrica*). A estratégia de utilizar a computação de borda (*edge computing*) foi adotada para minimizar a dependência de conectividade externa, assegurando um tempo de resposta rápido e compatível com a dinâmica do processo industrial.

No que se refere ao desenvolvimento dos modelos de aprendizado profundo, especificamente para a arquitetura de Redes Neurais Siamesas, utilizou-se a biblioteca **Keras**, operando sobre o *framework* TensorFlow, que oferece flexibilidade na definição de arquiteturas e experimentação rápida (Chollet, 2018). A escolha dessa ferramenta deve-se à sua interface de alto nível, que facilita a prototipagem rápida e a experimentação de diferentes arquiteturas de *backbone* para a extração de vetores de características (*embeddings*).

Visando a otimização do sistema para o ambiente de produção e o cumprimento dos requisitos de baixa latência em hardware de borda, adotou-se o formato ONNX (Open Neural Network Exchange), que facilita portabilidade entre frameworks e acelera inferência (BAI et al., 2019). Para maximizar a eficiência computacional, aplicou-se o processo de quantização dinâmica, convertendo os parâmetros dos modelos de precisão de ponto flutuante de 32 bits (FP32) para inteiros de 8 bits (INT8), técnica documentada como eficaz na redução de latência e tamanho de modelos, sem perda significativa de acurácia (Jacob et al., 2018). Essa estratégia resultou na redução do tamanho final dos arquivos e no aumento da velocidade de inferência (FPS), viabilizando a execução em tempo real nos dispositivos embarcados sem degradação significativa na acurácia da detecção de falhas.

## 4 RESULTADOS E DISCUSSÕES OU ANÁLISE DOS DADOS

Os resultados apresentados nesta seção referem-se à validação em ambiente simulado e controlado, com o objetivo de verificar a viabilidade técnica do sistema antes de sua aplicação completa no ambiente produtivo. Assim, os achados devem ser interpretados como evidências preliminares de desempenho, uma vez que o pipeline ainda não foi integralmente

implantado em produção e seguirá ciclos de ajustes, novos treinamentos e melhorias incrementais.

No escopo deste trabalho, a avaliação foi conduzida considerando a arquitetura modular proposta, composta por (i) um módulo de contagem automática baseado em detecção e rastreamento e (ii) um módulo de inspeção de falhas por comparação, conforme a organização do pipeline descrita no modelo do artigo. Dessa forma, os resultados foram analisados sob duas perspectivas: desempenho do sistema de contagem em vídeo e resultados preliminares do mecanismo de inspeção por similaridade.

O módulo de contagem foi avaliado utilizando o fluxo implementado no *notebook*. Nesse processo, o sistema aplica rastreamento com persistência e contabiliza um evento quando um objeto cruza uma linha vertical, posicionada no centro do *frame*. Para mitigar a dupla contagem (ocorrências comuns quando o objeto se mantém próximo à linha ou devido a pequenas oscilações de movimento), a lógica empregada utiliza duas memórias: um conjunto (*set*) de IDs já contabilizados e um registro de posições anteriores por ID, permitindo identificar o cruzamento na direção esperada.

Nos testes registrados, o vídeo foi processado em resolução  $1920 \times 1080$  e 30 FPS, com incremento do contador a cada cruzamento confirmado pelo ID rastreado. O log de execução do notebook evidencia eventos sucessivos de contagem (por exemplo, “Peça X contada! Total Y”), culminando no total de 13 itens contabilizados ao final do vídeo de teste. Esse resultado indica que a estratégia de contagem por cruzamento de linha, associada ao rastreamento, é funcional para a etapa de rastreabilidade produtiva, sobretudo por evitar que o mesmo ID seja contabilizado mais de uma vez durante sua passagem.

Em termos de desempenho computacional, o relatório de validação do detector no notebook registra tempos médios por imagem na ordem de milissegundos (pré processamento de 0,2 ms, inferência de 3,5 ms e pós-processamento de 1,3 ms por imagem, no ambiente de execução indicado), o que sustenta a viabilidade de operação em tempo real para o módulo de contagem. Além disso, as métricas reportadas na validação do modelo de detecção apresentam valores elevados (BoxP igual a 1, Recall de 0,974 e mAP 50/mAP 50-95 de 0,987), reforçando a capacidade do detector em localizar o objeto de interesse no conjunto avaliado.

A abordagem de inspeção por similaridade mostrou-se adequada para cenários em que se deseja comparar padrões visuais entre uma referência e a amostra inspecionada, permitindo identificar desvios localizados ao longo da peça, inclusive por meio de análise por sub-regiões (*patches*). Em avaliações preliminares no contexto considerado, observou-se uma



assertividade em torno de 90% na detecção de anomalias visuais, indicando potencial para aplicação em rotinas de controle de qualidade automatizado. Ainda assim, nesta fase do projeto, esses resultados devem ser interpretados como parciais, pois a robustez do limiar de decisão e a estabilidade do recorte da região de interesse são sensíveis a variações de captura, como iluminação, perspectiva e padronização de enquadramento, comuns no chão de fábrica.

Diante disso, a evolução do sistema prevê novos ciclos de treinamento, refinamento de parâmetros e validações em condições progressivamente mais próximas do cenário real de operação, buscando aumentar a capacidade de generalização frente às variações do processo industrial.

Com o objetivo de validar o ganho de desempenho do detector mantendo o mesmo conjunto de dados, foi conduzido um novo treinamento utilizando o YOLO26 (variante nano), a partir do dataset “Bercocarrosseis-3” (versão 3) exportado no formato YOLO26 via Roboflow, em GPU Tesla T4. O treinamento foi executado com 100 épocas, resolução 640 e batch 16, mantendo a mesma configuração base aplicada nos testes anteriores de detecção.

Na validação do YOLO26, observou-se desempenho elevado no conjunto avaliado (Precision  $\approx 0,999$  e Recall  $\approx 0,974$ ), com mAP50  $\approx 0,986$  e mAP50-95  $\approx 0,986$ . Em termos de custo computacional, o modelo apresentou inferência média de 4,3 ms por imagem, com pós-processamento de 0,3 ms por imagem, indicando uma execução eficiente no estágio de pós-processamento. Além disso, o modelo treinado apresentou redução de complexidade (aprox. 2,38 milhões de parâmetros e 5,2 GFLOPs na versão “fused”), o que pode favorecer cenários com restrições de recursos e implementação em hardware embarcado.

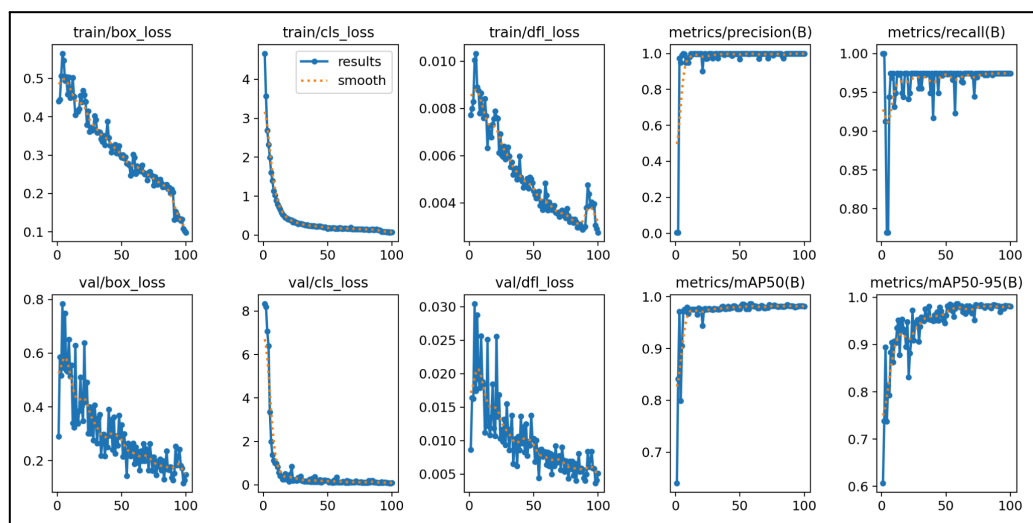


Figura 1 - Métricas da etapa de treinamento

A Figura 1 apresenta a evolução das perdas de treinamento e validação (box\_loss, cls\_loss e dfl\_loss) ao longo das épocas, evidenciando queda consistente das curvas no

treinamento e comportamento convergente na validação, ainda que com oscilações esperadas devido ao tamanho do conjunto de validação. Em paralelo, as métricas de desempenho (precision e recall) se estabilizam em patamar elevado após as primeiras épocas, indicando que o modelo rapidamente aprende a localizar e classificar o objeto de interesse, com baixa taxa de falsos positivos e boa cobertura das ocorrências reais.

Observa-se também que as curvas de mAP 50 e mAP 50-95 aumentam progressivamente e se mantêm próximas do teto durante a maior parte do treinamento, sugerindo boa capacidade de generalização para o conjunto avaliado e reforçando a adequação do detector para a etapa de identificação do berço no fluxo proposto. Em conjunto, esses resultados indicam convergência do treinamento e estabilidade das métricas, com redução de perdas e manutenção de desempenho elevado nas métricas de detecção.

Além do treinamento, o YOLO26 foi aplicado em um teste de contagem em vídeo com rastreamento persistente e lógica de cruzamento de linha, registrando eventos em CSV e contabilizando 16 peças no vídeo utilizado nesse experimento. Esse resultado reforça a aplicabilidade do detector como componente do módulo de contagem, principalmente quando integrado à lógica de rastreamento e registro de eventos, favorecendo a auditoria e a validação do processo.

## **5 CONCLUSÃO/CONSIDERAÇÕES FINAIS**

Este trabalho apresentou o desenvolvimento de um sistema de visão computacional destinado à automação de etapas críticas do processo produtivo no setor têxtil, contemplando dois módulos principais: (i) detecção e rastreamento para contagem automática de peças e (ii) inspeção visual baseada em similaridade para identificação de anomalias em bordados e estampas. Os experimentos conduzidos em ambiente controlado demonstraram que a abordagem proposta é tecnicamente viável, apresentando desempenho elevado na detecção de objetos, estabilidade no rastreamento e resultados consistentes na identificação de desvios visuais.

A realização de um segundo ciclo de treinamento utilizando o modelo YOLO26, com o mesmo conjunto de dados previamente aplicado ao YOLOv8, possibilitou a obtenção de métricas igualmente elevadas, aliadas à redução da complexidade computacional do modelo. Esses resultados reforçam o potencial de adoção da solução em sistemas embarcados e plataformas de computação de borda, cenário comum em linhas de produção que demandam baixa latência e operação contínua.

Entretanto, por se tratar de uma validação restrita a condições simuladas, os resultados devem ser interpretados como preliminares. Fatores ambientais típicos do ambiente fabril - como variações de iluminação, oscilações mecânicas, mudança de perspectiva e inconsistências no posicionamento das peças - tendem a impactar significativamente a estabilidade dos módulos de rastreamento e de inspeção por comparação visual. Assim, ainda não é possível generalizar integralmente o desempenho observado para o contexto produtivo real.

Como desdobramento natural desta pesquisa, as próximas etapas incluem:

- Implantar o sistema em ambiente industrial real, com o objetivo de avaliar seu desempenho diante das variáveis operacionais presentes no chão de fábrica;
- Coletar métricas reais de operação, tais como taxa de acertos na inspeção, impacto na redução de retrabalho, efetividade na padronização visual e estabilidade do sistema em longos ciclos produtivos;
- Refazer treinamentos e ajustar parâmetros com base em dados capturados diretamente na produção, ampliando o conjunto de amostras e aumentando a capacidade de generalização do modelo;
- Investigar abordagens complementares, como segmentação semântica, análise por regiões de interesse dinâmicas e técnicas avançadas de data augmentation específicas para o domínio têxtil;
- Avaliar a possibilidade de integração do sistema com plataformas de supervisão industrial, possibilitando rastreabilidade completa dos eventos detectados e tomada de decisão automatizada.

Conclui-se que o sistema apresenta potencial significativo para contribuir com a modernização do controle de qualidade na indústria têxtil. Sua consolidação, entretanto, depende da validação em ambiente produtivo e da coleta de métricas reais que permitam quantificar a efetividade do sistema em condições operacionais. As etapas futuras serão determinantes para confirmar a robustez da solução e sua capacidade de gerar valor em escala industrial.

## REFERÊNCIAS

ABADI, M. et al. **TensorFlow**: Large-scale machine learning on heterogeneous systems. 2015. Software available from: <https://www.tensorflow.org>. Acesso em: 15 jan. 2026.

BAI, J. et al. **ONNX**: Open Neural Network Exchange. GitHub repository, 2019. Disponível em: <https://onnx.ai>. Acesso em: 15 jan. 2026.

BARBOSA DE OLIVEIRA, G. DESEMPREGO TECNOLÓGICO NA INDÚSTRIA: UMA ANÁLISE SOBRE O SETOR TÊXTIL NO BRASIL. **Revista Economia e Políticas Públicas**, v. 9, n. 1, p. 1–35, 27 out. 2021.

BEWLEY, A. et al. **Simple Online and Realtime Tracking**. In: IEEE International Conference on Image Processing (ICIP), 2016. Anais [...]. IEEE, 2016.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library**. Sebastopol: O'Reilly Media, 2008.

BROMLEY, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. **Signature verification using a “siamese” time delay neural network**. In Advances in neural information processing systems, 737–744, 1994.

CHOLLET, F. et al. **Keras**. 2015. Disponível em: <https://keras.io>. Acesso em: 15 jan. 2026.

CHOLLET, F. **Deep Learning with Python**. Shelter Island: Manning Publications, 2018.

CHOPRA, S., Hadsell, R., and LeCun, Y. **Learning a Similarity Metric Discriminatively, with Application to Face Verification**. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, 539–546. doi:10.1109/CVPR.2005.202. 2005.

FIEG. **Relatório sobre o posicionamento da Indústria Brasileira**. 2018. Goiana.

HARANDI, M., Roy Kumar, S., and Nock, R. **Siamese Networks: A Thing or Two to Know**. Technical report. 2017.

JACOB, B. et al. **Quantization and training of neural networks for efficient integer-arithmetic-only inference**. In: Conference on Computer Vision and Pattern Recognition (CVPR), 2018. Anais [...]. IEEE, 2018.

JOCHER, G.; CHAURASIA, A.; QIU, J. **YOLO by Ultralytics**. 2023. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 16 jan. 2026.

KHANG, Alex *et al.* **Machine Vision and Industrial Robotics in Manufacturing: approaches, technologies, and applications**. Boca Raton: Crc Press, 2025.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. *Nature*, v. 521, p. 436-444, 2015.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997.

PEREIRA, Filipe et al. **Intelligent Computer Vision System for Analysis and Characterization of Yarn Quality**. *Electronics*, v. 12, n. 5, p. 1134, 2023.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. 2. ed. Cham: Springer, 2022.

ULTRALYTICS. Ultralytics YOLOv8: state-of-the-art real-time object detection and vision models. PyPI, 2024. Disponível em: <https://pypi.org/project/ultralytics/8.2.64/>. Acesso em: 08 dez. 2025.

ULTRALYTICS. Explore Ultralytics YOLOv8 – models and documentation. 2025. Disponível em: <https://docs.ultralytics.com/models/yolov8/> . Acesso em: 08 dez. 2025.

ULTRALYTICS. Explore Ultralytics YOLO26 – models and documentation. 2026. Disponível em: <https://docs.ultralytics.com/pt/models/yolo26/> . Acesso em: 23 jan. 2026.

ULTRALYTICS. Ultralytics YOLO – official GitHub repository. 2022. Disponível em: <https://github.com/ultralytics/ultralytics/> . Acesso em: 08 dez. 2025.