

Solução de Visão Computacional para Correção Automática de Folhas de Respostas de Múltipla Escolha

Everton Souza¹, Daniel Rosa¹, Andreza Falcão¹,
Cleon Xavier², Newarney Torrezão², Luiz Rodrigues³,
Seiji Isotani⁴, Ig Ibert Bittencourt⁵,
Rafael Ferreira Mello^{1,6}

¹ Universidade Federal Rural de Pernambuco (UFRPE)

² Instituto Federal Goiano (IF Goiano)

³ Universidade Tecnológica Federal do Paraná (UTFPR)

⁴ Universidade de São Paulo

⁵ Universidade Federal de Alagoas (UFAL)

⁶ Centro de Estudos e Sistemas Avançados do Recife (CESAR School)

rafael.mello@ufrpe.br

Resumo. A correção manual de provas de múltipla escolha consome tempo e atrasa o feedback aos estudantes. Scanners são uma solução, mas inacessíveis para muitas instituições. Este artigo propõe uma alternativa baseada em câmeras de smartphones, preenchendo uma lacuna sobre o custo-benefício de soluções móveis. Desenvolvemos uma ferramenta com o modelo You Only Look Once (YOLO) para reconhecer respostas em folhas de gabarito. Após testes e refinamentos, nossa solução atingiu 97% de acurácia com um tempo de inferência de 25ms, otimizando o processo de correção e acelerando o retorno avaliativo.

Abstract. Manual grading of multiple-choice tests is time-consuming and delays student feedback. While scanners are a common solution, they are inaccessible to many institutions. This paper proposes an alternative using smartphone cameras, addressing a gap regarding the cost-effectiveness of mobile solutions. We developed a tool using the You Only Look Once (YOLO) model to recognize answers on response sheets. After iterative testing and refinement, our solution achieved 97% accuracy with an inference time of 25ms, optimizing the grading process and accelerating evaluative feedback.

1. Introdução

A avaliação de questões de múltipla escolha, que são frequentemente realizadas com base em um gabarito, é uma tarefa comum em instituições de ensino [de Elias et al. 2021]. A correção manual tradicional é laboriosa, suscetível a erros humanos e pode levar a atrasos significativos no fornecimento de feedback aos estudantes [Centro de Inovação para a Educação Brasileira (CIEB)]. Como resultado, a avaliação

desse tipo de atividade não só sobrecarrega a equipe pedagógica (e.g., docentes e profissionais de pedagogia) com uma atividade manual e repetitiva, como também prejudica estudantes ao aumentar o tempo entre a avaliação e o recebimento de feedback [Centro de Inovação para a Educação Brasileira (CIEB)]. Uma alternativa para esse desafio é a automatização do processo de avaliação de questões de múltiplas escolhas, que pode aumentar sua eficiência, consistência e velocidade.

Na literatura de algoritmos de reconhecimento de marcação de gabarito, os sistemas mais comuns seguem as abordagens tradicionais de Reconhecimento de Marcação Óptica (do inglês *Optical Mark Recognition* - OMR) [de Elias et al. 2021]. Geralmente, as folhas de cartão-resposta de exercícios avaliativos são colocadas em um leitor do tipo scanner, que são lidas e passadas para um programa que faz a extração e processamento das questões marcadas para montar um gabarito totalmente digital. Essa versão digital contém a ordem das questões com suas respectivas marcações, que representam as opções escolhidas pelo estudante. A vantagem dessa abordagem é a facilidade de se obter as respostas dos estudantes de maneira rápida e fácil, uma vez que praticamente não há variação nas condições das imagens nos scanners modernos e também pela velocidade de leitura.

Por mais eficaz que essa abordagem possa ser, nem todas as instituições de ensino, a exemplo de uma grande parte das escolas públicas brasileiras, possuem recursos financeiros para adquirir scanners [Isotani et al. 2023]. Por conta disso, o processo de correção de folhas de avaliações ainda é exercido, em sua maior parte, de maneira manual, folha por folha. Esse processo pode exigir um tempo substancial para ser concluído por docentes ou profissionais de pedagogia, que também estão suscetíveis a cometer erros durante a correção das avaliações devido a fatores externos, como o cansaço gerado pela longa execução de uma atividade repetitiva como esta.

Por outro lado, muitos profissionais de escolas públicas brasileiras possuem smartphones [Centro Regional para o Desenvolvimento da Sociedade da Informação 2022], e a maioria destes têm câmeras embutidas. Nessas condições, a captura de imagem das folhas de gabarito dos estudantes se torna possível, mesmo na ausência de scanners. Neste caso, soluções tradicionais de OMR ainda são referenciadas. No entanto, essas soluções tradicionais dependem frequentemente de condições de imagem ideais e gabaritos estritamente formatados, mostrando-se menos flexíveis a variações comuns em cenários reais de aplicação, como iluminação irregular, inclinação da folha ou diferentes formas de marcação [de Elias et al. 2021].

A captura da imagem pelo smartphone também permite a automatização do processo de correção com o uso de modelos de Visão Computacional e Aprendizado de Máquina. Abordagens baseadas em aprendizado profundo como o *You Only Look Once* (YOLO) têm ganhado destaque [Tinh and Minh 2024, Mahmud et al. 2024, TEKEREK and ÇETİNKAYA 2024]. Além disso, muitos trabalhos existentes que utilizam o YOLO, embora eficazes, podem não explorar as gerações mais recentes de modelos ou focar em um equilíbrio otimizado entre velocidade de inferência e precisão, uma lacuna que nossa pesquisa visa preencher com o uso do YOLOv11n.

Por isso, este artigo apresenta um novo pipeline de extração automática de respostas em folhas de gabarito para auxiliar professores na correção de avaliações de múltipla

escolha. Para isso, foi desenvolvido um sistema baseado em visão computacional projetado para ler e interpretar automaticamente respostas de folhas de múltipla escolha. O principal desafio é identificar com precisão a região de interesse (a grade de respostas) em uma imagem potencialmente distorcida ou mal iluminada de uma folha inteira e, em seguida, reconhecer corretamente cada resposta marcada. Sendo assim, nossas perguntas de pesquisa são:

Pergunta de pesquisa 1: É possível criar um pipeline que extraia de forma eficiente as alternativas corretas de um gabarito capturado com uma foto de celular?

Pergunta de pesquisa 2: Qual o melhor modelo para capturar alternativas corretas de um gabarito considerando desempenho e velocidade?

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta uma revisão da literatura sobre o tema. A Seção 3 detalha a metodologia empregada, incluindo a preparação dos conjuntos de dados e a arquitetura do pipeline. A Seção 4 apresenta os resultados quantitativos dos modelos, enquanto a Seção 5 discute esses resultados, responde às perguntas de pesquisa e aponta as limitações do estudo. Por fim, a Seção 6 conclui o trabalho e sugere direções futuras.

2. Revisão da Literatura

Dada a alta disponibilidade de celulares com câmeras embutidas [Centro Regional para o Desenvolvimento da Sociedade da Informação 2022], esses dispositivos apresentam uma alternativa interessante para a falta de scanners em escolas [Isotani et al. 2023]. Nesse contexto, para embasar nossa abordagem, foi realizada uma revisão bibliográfica para compreender os métodos existentes para correção automatizada de folhas de respostas de múltipla escolha em imagens capturadas sem a utilização de scanners.

Nossa revisão bibliográfica revelou uma tendência para o uso de modelos de detecção de objetos baseados em aprendizado profundo, sendo o YOLO uma escolha recorrente devido ao equilíbrio entre velocidade e precisão. Diversos estudos demonstraram a aplicação bem-sucedida de variantes do YOLO, como o YOLOv8, utilizado em conjunto com Tesseract OCR para avaliação de questões de múltipla escolha [Mahmud et al. 2024] e em um framework para pontuação automatizada de gabaritos em papel [Tinh and Minh 2024]. Outra abordagem explorou o YOLOv7 para a digitalização de questões de múltipla escolha impressas [TEKEREK and ÇETİNKAYA 2024]. Adicionalmente, uma busca em anais de eventos relevantes da área no Brasil, como o Simpósio Brasileiro de Informática na Educação (SBIE), não revelou trabalhos que abordem a extração de respostas de gabaritos com as mesmas técnicas, evidenciando uma lacuna na literatura nacional que este trabalho visa contribuir para preencher.

Além das técnicas de aprendizado profundo, soluções baseadas em processamento de imagem tradicional, como o uso de OpenCV para OMR, também foram exploradas. Um exemplo é a abordagem detalhada no projeto de [Hassan 2021], que emprega a biblioteca OpenCV para implementar um pipeline de OMR. Este pipeline inclui etapas como o pré-processamento da imagem (conversão para tons de cinza, aplicação de desfoque gaussiano e detecção de bordas com Canny), a identificação de contornos para isolar a folha de respostas, a aplicação de uma transformação de perspectiva para corrigir distorções e obter uma visualização retificada, e, finalmente, a análise das regiões das

bolhas de marcação através de limiarização e contagem de pixels não nulos para identificar as opções assinaladas. Embora eficazes em condições controladas e com gabaritos bem definidos, essas abordagens tradicionais frequentemente apresentam sensibilidade a variações de iluminação, desalinhamento da folha, diferentes tipos de caneta ou preenchimento incompleto das bolhas, exigindo calibração cuidadosa [de Elias et al. 2021]. Nesse contexto, técnicas baseadas em processamento de imagem tradicional são, em geral, menos robustas e flexíveis que as soluções baseadas em aprendizado profundo frente à diversidade de cenários reais.

Apesar de haver abordagens utilizando modelos treinados do YOLO que demonstram alta precisão, como as que empregam YOLOv8 [Mahmud et al. 2024, Tinh and Minh 2024] ou YOLOv7 [TEKERREK and ÇETİNKAYA 2024], muitas vezes essas soluções podem utilizar variantes mais pesadas dos modelos ou focar primariamente na acurácia, resultando em tempos de inferência que podem não ser ideais para aplicações que demandam alta velocidade ou que rodam em hardware com recursos limitados. Frequentemente, o equilíbrio entre a velocidade de processamento e a precisão da detecção não é o foco principal, ou as versões utilizadas antecedem otimizações presentes em arquiteturas mais recentes e leves.

Dessa forma, este trabalho realiza uma avaliação de modelos mais novos e eficientes como o YOLOv11n, que é um modelo mais eficiente para as principais tarefas de detecção, visando um equilíbrio otimizado entre precisão, velocidade de inferência e a capacidade de lidar com variações nas imagens. Além disto, este artigo propõe um pipeline completo para detecção das questões a partir de uma foto da página como um todo e não apenas da área do gabarito.

3. Método

Nesta seção, descrevemos a metodologia utilizada para desenvolver e avaliar nossa solução de visão computacional. Detalhamos a criação dos conjuntos de dados, as métricas de avaliação, o processo de treinamento dos modelos e a arquitetura final do pipeline.

3.1. Conjuntos de Dados (Datasets)

Para o desenvolvimento da nossa solução, foram preparados dois conjuntos de imagens para serem usados como insumo no treinamento dos modelos de detecção de objetos. As imagens desses conjuntos de dados foram coletadas em pesquisas anteriores e disponibilizadas para execução deste estudo. Cada conjunto denota um conteúdo diferente a ser extraído: o primeiro representa a imagem da folha completa capturada pelo professor, enquanto o segundo apresenta o recorte da região de interesse do gabarito. As imagens foram registradas conforme o seguinte.

- **Folhas Completas:** Um conjunto de 4.441 imagens representando folhas de respostas inteiras. Estas foram usadas para treinar o modelo de localização da região do gabarito. A Figura 1 apresenta um exemplo. A única classe ('gabarito') foi dividida em 3.105 instâncias para treino (70%), 888 para teste (20%) e 444 para validação (10%).
- **Folhas de Respostas Recortadas:** Um conjunto de 1.587 imagens mostrando apenas a região do gabarito. Este dataset foi utilizado para treinar o modelo focado na detecção das alternativas marcadas (A, B, C, D ou inválido). A Figura 2

apresenta um exemplo. A distribuição de classes neste conjunto é desbalanceada: 'A' (8.006), 'B' (11.541), 'C' (8.169), 'D' (6.448) e 'inválido' (338). A divisão seguiu a proporção de 70% treino, 20% teste e 10% validação.

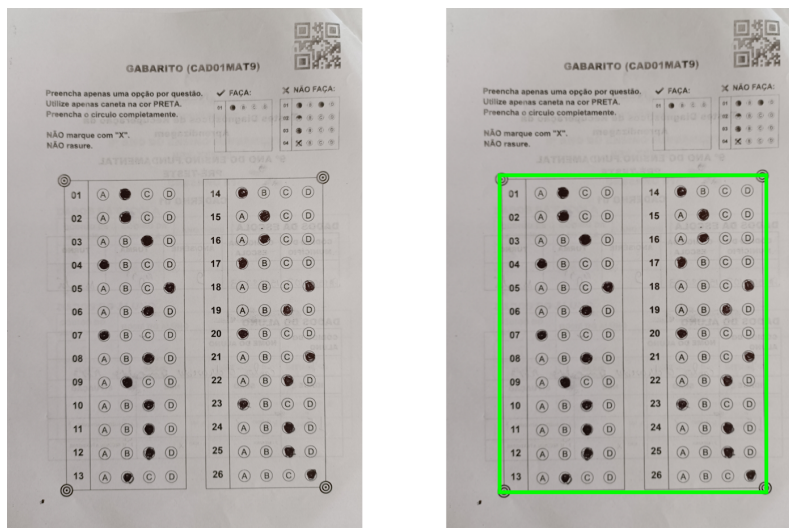


Figura 1. Exemplo de imagem do conjunto de dados de folhas completas. À esquerda está a imagem capturada pelo celular e à direita está a sua respectiva anotação.

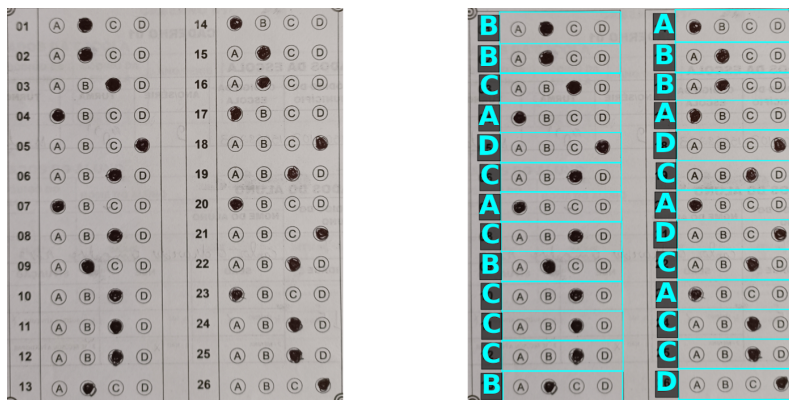


Figura 2. Exemplo de imagem do conjunto de dados de folhas de respostas recortadas. À esquerda está a imagem recortada do gabarito e à direita estão as suas respectivas anotações.

O processo de anotação para ambos os datasets combinou um esforço manual inicial, utilizando a ferramenta LabelImg, com uma fase subsequente de autoanotação. No esforço manual inicial, as bases de dados foram divididas igualmente entre três pesquisadores. Dada a objetividade desta tarefa, consideramos um anotador por imagem suficiente para alcançar a confiabilidade necessária. Na fase de autoanotação, os modelos treinados inicialmente (descritos na Seção 3.3) foram utilizados para gerar anotações para as imagens restantes, as quais foram então revisadas e corrigidas manualmente.

Os arquivos de anotação de cada base seguiram o padrão do formato YOLO. Para cada imagem no conjunto de dados (e.g., `imagem001.jpg`), um arquivo de texto cor-

respondente (e.g., `imagem001.txt`) é criado. Cada linha neste arquivo de texto representa uma caixa delimitadora (bounding box) de um objeto detectado na imagem. O formato de cada linha é: `<class_id> <x_center_norm> <y_center_norm> <width_norm> <height_norm>`, onde:

- `<class_id>` é um inteiro que representa a classe do objeto (e.g., 0 para a classe ‘A’, 1 para ‘B’, 2 para ‘C’, 3 para ‘D’ e 4 para ‘inválido’, no caso do modelo de detecção de respostas; ou 0 para ‘gabarito’ no modelo de detecção de gabarito).
- `<x_center_norm>` é a coordenada x do centro da caixa delimitadora, normalizada pela largura da imagem (valor entre 0 e 1).
- `<y_center_norm>` é a coordenada y do centro da caixa delimitadora, normalizada pela altura da imagem (valor entre 0 e 1).
- `<width_norm>` é a largura da caixa delimitadora, normalizada pela largura da imagem (valor entre 0 e 1).
- `<height_norm>` é a altura da caixa delimitadora, normalizada pela altura da imagem (valor entre 0 e 1).

Por exemplo, uma linha em um arquivo de anotação para o modelo de detecção de respostas poderia ser: ‘0 0.25 0.5 0.1 0.05’, indicando um objeto da classe 0 (digamos, ‘A’) centrado em 25% da largura e 50% da altura da imagem, com uma largura de 10% da largura da imagem e altura de 5% da altura da imagem.

O dataset de folhas completas foi anotado de maneira a destacar somente a região do gabarito completo. Conforme ilustrado na Figura 1, a região do gabarito é tipicamente delimitada por marcadores visuais (como quatro bolinhas) nas extremidades da área retangular das questões. Por sua vez, o dataset das folhas de respostas recortadas foi anotado para destacar cada uma das N questões presentes na imagem.

3.2. Métricas de Avaliação

Para avaliar o desempenho dos modelos de detecção de objetos, foram utilizadas as seguintes métricas padrão:

- **Precisão (Precision):** Mede a proporção de detecções corretas entre todas as detecções realizadas pelo modelo. É calculada como $\frac{VP}{VP+FP}$.
- **Recall (Revocação):** Mede a proporção de objetos reais que foram corretamente detectados pelo modelo. É calculado como $\frac{VP}{VP+FN}$.
- **mAP50 (mean Average Precision @ IoU=0.50):** É uma métrica comum para avaliar a precisão geral de um detector de objetos. O sufixo “50” indica que a métrica é calculada considerando uma detecção como correta se a Interseção sobre União (IoU) entre a caixa delimitadora prevista e a real for de pelo menos 0.50 [Padilla et al. 2020].

Estas métricas fornecem uma visão abrangente da capacidade do modelo em localizar e classificar corretamente os objetos de interesse.

3.3. Definição de Classes e Treinamento Inicial

Inicialmente, 600 imagens foram selecionadas manualmente de cada conjunto de dados para anotação no formato YOLO, servindo como base para o primeiro ciclo de treinamento.

Para o **Modelo de Detecção de Respostas**, as anotações nas imagens de folhas recortadas utilizaram cinco classes: 'A', 'B', 'C', 'D' e 'inválido'. A classe 'inválido' foi designada para questões onde nenhuma opção foi marcada ou quando mais de uma opção foi marcada. Com base em discussões técnicas posteriores, esta definição foi refinada para explicitamente considerar marcações duplas como 'inválidas', levando a uma reanotação manual de parte do dataset e ao retreinamento específico desse aspecto, que resultou em um mAP de 94.3% para a detecção correta dessas instâncias. O treinamento inicial do modelo YOLOv11n com as 600 imagens anotadas (antes do refinamento) produziu métricas de mAP50 de 86.7%, precisão de 86.1% e *recall* de 94.9%.

Para o **Modelo de Detecção do Gabarito (Template)**, o objetivo foi treinar um detector para localizar a área retangular que engloba todas as questões e alternativas na folha completa. Assim, nas 600 imagens de folhas completas, as anotações foram feitas de modo a delimitar essa região de interesse (bounding box), utilizando o rótulo 'gabarito'. O modelo inicial do YOLOv11n treinado neste subconjunto alcançou precisão de 99,9%, recall de 100% e mAP50 de 99,5%.

3.4. Seleção da Arquitetura e Hiperparâmetros do Modelo YOLO

Com os conjuntos de dados expandidos e verificados, procedeu-se à seleção da arquitetura final do modelo YOLO. Foram treinados e avaliados oito modelos distintos, variando a versão do YOLO (v8 e v11), o tamanho dos pesos (nano 'n' e small 's') e a resolução da imagem de entrada (320px e 640px). Os resultados dos treinamentos estão expostos na seção de resultados.

3.5. Arquitetura do Pipeline

A solução de ponta a ponta processa imagens em um pipeline sequencial. Após a detecção inicial das respostas marcadas pelo modelo YOLO, as saídas brutas consistem em uma lista de caixas delimitadoras com suas respectivas classes e coordenadas. Estas saídas, contudo, não fornecem intrinsecamente a ordem sequencial correta das questões, nem lidam com possíveis falhas de detecção. Portanto, uma etapa subsequente de pós-processamento é crucial. Esta etapa visa organizar as detecções, assegurar que cada resposta seja corretamente associada à sua questão, mitigar erros e estruturar os dados para a saída final. O pipeline completo, ilustrado na Figura 3, pode ser detalhado conforme o seguinte:

1. **Entrada:** O sistema recebe uma imagem completa de uma folha de respostas.
2. **Detecção do Gabarito:** O primeiro modelo (detector de 'gabarito') identifica e recorta a região da grade de respostas da imagem completa.
3. **Detecção de Respostas:** A grade recortada é passada para o segundo modelo (detector de ABCD ou 'inválido'), que detecta e classifica a resposta marcada para cada questão.
4. **Pós-processamento:** Esta etapa organiza as detecções, assegura que cada resposta seja corretamente associada à sua questão e mitiga possíveis erros. O processo ordena as respostas por colunas e linhas, insere marcadores para questões não detectadas (lacunas) e remove detecções redundantes com Supressão Não Máxima (NMS).
5. **Saída:** O pipeline retorna um objeto JSON estruturado mapeando cada número de questão para a resposta detectada.

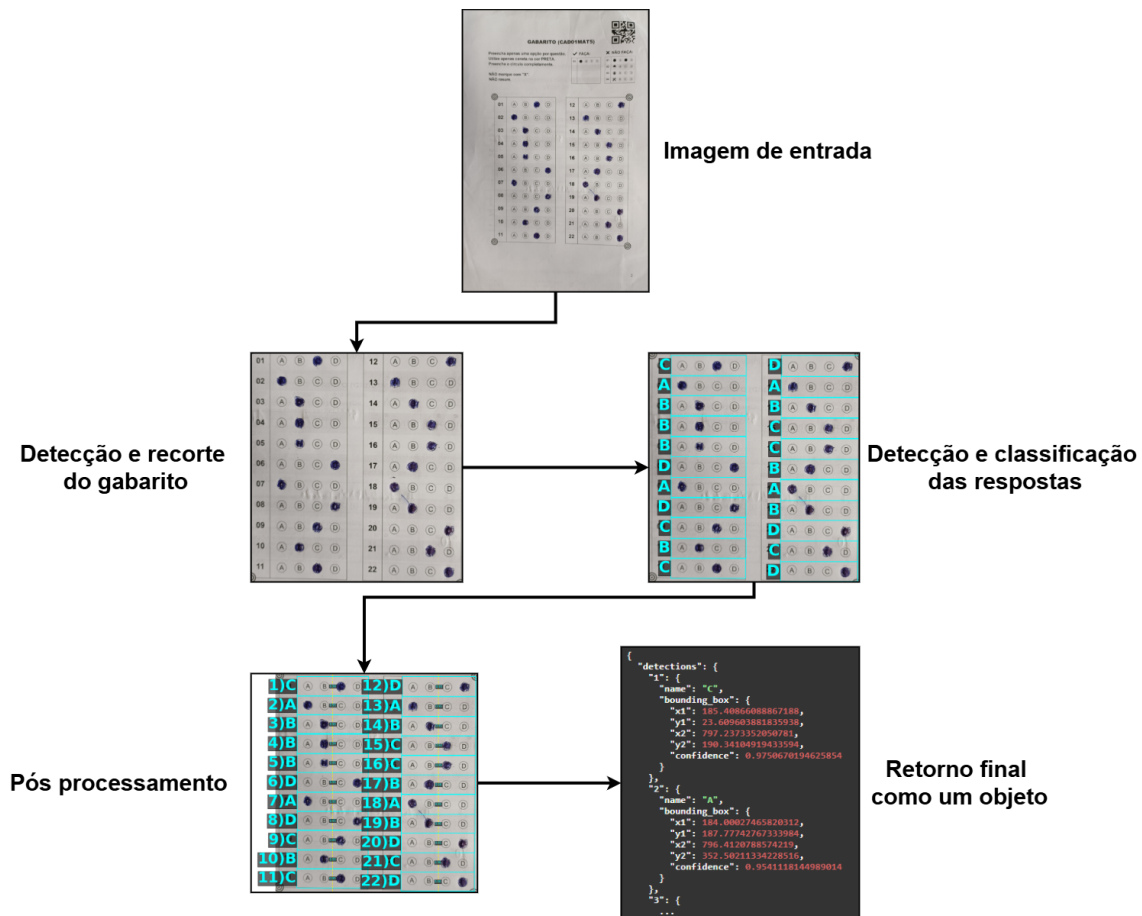


Figura 3. Exemplo visual do funcionamento do pipeline

4. Resultados

A Tabela 1 mostra os valores de mAP50, precisão e *recall* que cada modelo atingiu no subset de teste, variando os pesos e o tamanho de entrada de imagem. Além disso, o tempo de inferência em milissegundos também foi registrado na última coluna.

É possível observar na Tabela 1 que os modelos de ambas as versões do YOLO obtiveram resultados similares, com variações de desempenho de até no máximo de 1.4%, 2.0% e 1.2% nas métricas de mAP50, precisão e *recall*, respectivamente. O modelo com o maior mAP50 foi o do YOLOv8 de tamanho *small*, com valor de 97.7%. Este modelo também alcançou o maior nível de precisão, atingindo uma taxa de 98.3%. Os modelos do YOLOv11 *small* de 640px, e o YOLOv8 *nano* de 320px obtiveram o mesmo desempenho de 97.4% em *recall*.

Particularmente, analisando os valores da coluna de mAP50, a média dos resultados dos modelos da arquitetura YOLOv11 foi de 97.45%, enquanto a média obtida com os modelos do YOLOv8 totalizou 97.07%. Se olharmos para a velocidade de predição, a média dos tempos de inferência dos modelos do YOLOv11 foi ligeiramente menor que a dos YOLOv8. Dentre todos os modelos treinados, somente o YOLOv11 *nano* de 320px obteve uma velocidade inferior a 30ms, atingindo 25ms de tempo de predição.

Tabela 1. Comparação de Desempenho entre Configurações do Modelo YOLO para Detecção de Respostas.

Modelo	Pesos	Img. Size	mAP50	Precision	Recall	Inferência (ms)
YOLOv11	nano	320px	0,973	0,969	0,954	25
YOLOv11	nano	640px	0,975	0,979	0,973	44
YOLOv11	small	320px	0,975	0,981	0,965	38
YOLOv11	small	640px	0,975	0,971	0,974	95
Média			0,974	0,975	0,966	50,5
YOLOv8	nano	320px	0,972	0,964	0,974	32
YOLOv8	nano	640px	0,963	0,981	0,962	41
YOLOv8	small	320px	0,977	0,983	0,971	35
YOLOv8	small	640px	0,971	0,963	0,966	98
Média			0,970	0,972	0,968	51,5

5. Discussão

Nesta seção, discutimos os resultados, respondemos às perguntas de pesquisa e abordamos as limitações do estudo.

Respondendo à Pergunta de Pesquisa 1, sobre a viabilidade de um pipeline eficiente, nossos resultados demonstram que é possível criar um sistema robusto. O pipeline proposto, combinando dois modelos YOLOv11n e uma etapa de pós-processamento, conseguiu extrair e ordenar as respostas com alta acurácia (mAP50 de 97,3%) e velocidade (25ms por inferência). A etapa de pós-processamento foi crucial para lidar com falhas de detecção e estruturar a saída, garantindo a usabilidade dos dados extraídos.

Respondendo à Pergunta de Pesquisa 2, sobre o melhor modelo em termos de desempenho e velocidade, a Tabela 1 mostra que o YOLOv8s com imagem de 320px obteve o maior mAP50 (97,7%). No entanto, o YOLOv11n com imagem de 320px oferece o melhor custo-benefício, com um mAP50 muito competitivo (97,3%) e o menor tempo de inferência (25ms), sendo 10ms mais rápido que seu concorrente mais próximo. A leveza do modelo *nano* o torna ideal para aplicações em larga escala ou em hardware com recursos limitados, justificando sua escolha para a solução final.

Uma análise qualitativa dos erros revela que falsos negativos (questões não detectadas) ocorrem principalmente com marcações muito fracas ou preenchimento parcial da bolha. Falsos positivos, embora raros, podem surgir de manchas ou rasuras próximas às áreas de marcação. A classe 'inválido' apresentou os maiores desafios, especialmente em casos de dupla marcação onde uma das respostas estava significativamente mais fraca que a outra, levando o modelo a classificar erroneamente como uma marcação única. O pós-processamento mitiga parte desses erros ao identificar lacunas, mas a precisão na detecção de casos ambíguos permanece um desafio.

É importante reconhecer as **limitações** deste estudo. Primeiramente, o conjunto de dados, embora variado em iluminação, apresenta pouca variação angular nas fotos das folhas, o que pode ter simplificado a tarefa de detecção. Em cenários reais, folhas mais amassadas ou fotografadas de ângulos extremos podem representar um desafio maior.

Além disso, o estudo se concentra na extração das respostas, não na correção automática, que dependeria de um sistema externo com o gabarito oficial. A robustez do modelo contra diferentes tipos de caneta e cores de preenchimento também não foi sistematicamente avaliada.

6. Conclusão

A tarefa da correção de gabaritos em folhas de respostas pode se tornar uma atividade custosa para professores, tendo em vista o tempo de organização e comparação das marcações. O uso de scanners para leitura automática consegue acelerar o processo, porém tais equipamentos nem sempre estão disponíveis. Para auxiliar nesse cenário, a abordagem proposta neste artigo para extração de respostas a partir de fotos de celular se mostrou eficiente, capaz de devolver os gabaritos dos estudantes de forma rápida e com alto grau de confiabilidade.

Em resumo, a solução consiste na integração de dois modelos de detecção de objetos que, ao receber uma folha como entrada, primeiro recorta a região do gabarito e, em seguida, detecta as respostas marcadas. Após uma etapa de pós-processamento que organiza e valida as detecções, o algoritmo retorna o gabarito do estudante completamente anotado e pronto para análise. Como resultado, nossa proposta atingiu alta taxa de acerto na extração das marcações, com o modelo escolhido alcançando um mAP50 de 97,3% e um tempo de inferência de apenas 25ms.

A partir disso, pode-se afirmar que a abordagem desenvolvida consegue extrair e ordenar as respostas assinaladas com alta confiabilidade e velocidade, tornando-se uma alternativa eficaz para auxiliar os professores na correção de suas atividades. Sua qualidade é atestada pelo uso de modelos de detecção de objetos de última geração e por um método robusto de ordenação e processamento das informações extraídas, garantindo a estabilidade e a coesão dos dados. Desse modo, concluímos que a solução pode ser muito útil na composição de sistemas de correção automática, dando celeridade aos processos pedagógicos e dinamizando os feedbacks escolares.

Como **trabalhos futuros**, pretendemos expandir o pipeline para um sistema de correção de ponta a ponta, integrando-o com uma base de dados de gabaritos oficiais para fornecer a nota final automaticamente. Outra direção é aumentar a robustez do modelo, treinando-o com um conjunto de dados mais desafiador, incluindo maiores variações de ângulo, folhas amassadas e diferentes tipos de marcações. Finalmente, planejamos otimizar o modelo para implantação em dispositivos móveis de baixo custo, garantindo acessibilidade e usabilidade em um espectro mais amplo de contextos educacionais.

Agradecimentos

Reconhecemos o uso de IA Generativa para auxiliar na elaboração e revisão deste artigo, que foi devidamente revisado, e assumimos total responsabilidade por seu conteúdo.

Considerações Éticas

Este estudo se baseia em um conjunto de dados coletados/publicados anteriormente. Portanto, não é elegível para avaliação por um Comitê de Ética em Pesquisa.

Disponibilidade de artefatos

Os artefatos deste estudo serão disponibilizados pelos autores correspondentes.

Referências

- Centro de Inovação para a Educação Brasileira (CIEB). Como a inteligência artificial irá impactar o trabalho dos professores.
- Centro Regional para o Desenvolvimento da Sociedade da Informação (2022). TIC Educação 2022 - Professores. Pesquisa, Comitê Gestor da Internet no Brasil (CGI.br) / Centro Regional para o Desenvolvimento da Sociedade da Informação (Cetic.br), São Paulo. Acesso em: 09 jun. 2025.
- de Elias, E. M., Tasinaffo, P. M., and Hirata Jr., R. (2021). Optical Mark Recognition: Advances, Difficulties, and Limitations. *SN Computer Science*, 2:367.
- Hassan, M. (2021). Optical mark recognition using opencv. GitHub repository. Acesso em: 26 ago. 2025.
- Isotani, S., Bittencourt, I. I., Challco, G. C., Dermeval, D., and Mello, R. F. (2023). Aied unplugged: Leapfrogging the digital divide to reach the underserved. In *International Conference on Artificial Intelligence in Education*, pages 772–779. Springer.
- Mahmud, S., Biswas, K., Alam, A., Rudro, R. A. M., Anannya, N. J., Mouri, I. J., and Nur, K. (2024). Automatic multiple choice question evaluation using tesseract ocr and yolov8. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 246–252. IEEE.
- 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)*, pages 237–242.
- TEKEREK, A. and ÇETİNKAYA, Y. (2024). Digitization of printed multiple choice questions using object detection methods: A yolov7-based approach. *Artificial intelligence Studies*, 7(2).
- Tinh, P. D. and Minh, T. Q. (2024). Automated paper-based multiple choice scoring framework using fast object detection algorithm. *International Journal of Advanced Computer Science & Applications*, 15(1).