

# Processamento de Cartão-Resposta

1<sup>st</sup> Rogério Succuglia Andrade Junior  
Universidade Estadual do Norte do Paraná - CLM  
Bandeirantes, Brasil  
rogeriosuccuglia@gmail.com

1<sup>st</sup> Gustavo Denobi  
Universidade Estadual do Norte do Paraná - CLM  
Bandeirantes, Brasil  
gustavodenobi2022@gmail.com

**Abstract**—Utilizando Python e sua biblioteca OpenCV, foi desenvolvida uma solução que possibilita a identificação automática de questões de múltipla escolha e as respostas assinaladas em cartões-resposta, previamente digitalizados e anonimizados, do vestibular da Universidade Estadual do Norte do Paraná.

**Keywords**— *image processing*

## I. INTRODUÇÃO

A correção automatizada de cartões-resposta é uma demanda crescente em processos avaliativos de larga escala, como vestibulares, concursos e exames institucionais. Tradicionalmente, a leitura dessas respostas é realizada por equipamentos dedicados, que podem apresentar custos elevados e pouca flexibilidade para adaptações. Com o avanço das técnicas de processamento digital de imagens e o aumento da capacidade computacional, torna-se viável o desenvolvimento de soluções baseadas em software para a detecção e análise automática de marcações em cartões de múltipla escolha.

Este trabalho apresenta o desenvolvimento de um sistema automatizado para reconhecimento de respostas em gabaritos de provas, utilizando técnicas de visão computacional e processamento de imagens. O sistema proposto é capaz de identificar, segmentar e interpretar as marcações realizadas pelos candidatos, gerando arquivos digitais com os resultados e possibilitando a comparação automática com gabaritos oficiais. A solução visa oferecer uma alternativa de baixo custo, flexível e de fácil adaptação a diferentes modelos de cartões, contribuindo para modernização e eficiência de processos avaliativos.

## II. METODOLOGIA

### A. Aquisição das Imagens

As imagens dos cartões-resposta foram obtidas por meio da digitalização dos gabaritos preenchidos pelos candidatos durante o vestibular da Universidade Estadual do Norte do Paraná. Para garantir a anonimização dos participantes, os arquivos foram renomeados de forma padronizada e sem identificação pessoal.

### B. Pré-processamento das Imagens

O pré-processamento visa preparar as imagens para a etapa de reconhecimento. Inicialmente, as imagens são convertidas para escala de cinza, facilitando a aplicação de técnicas de processamento digital. Em seguida, são aplicados filtros para redução de ruído e realce das regiões de interesse (ROI). O sistema detecta automaticamente as colunas presentes no cartões, recortando-as e armazenando-as em diretórios organizados.

Para que seja possível realizar esta etapa, é necessário padronizar a folha de respostas (gabarito) preenchida pelos alunos, sendo composta por duas partes: o cabeçalho, o qual

contém as informações do aluno, sua assinatura e dados da prova; e a área destinada às respostas, sendo esta a ROI do presente trabalho.

A área destinada às respostas se encontra dividida em três (3) colunas, cada qual contendo vinte (20) questões de múltipla escolha, com cinco (5) possíveis alternativas, totalizando sessenta (60) questões a serem reconhecidas e analisadas.

Foram analisados quatrocentos e noventa e oito (498) cartões-resposta, o que torna possível prever que, à princípio, serão analisadas vinte e nove mil oitocentos e oitenta (29.880) questões no total.

### C. Ferramentas Utilizadas

Para realizar o presente trabalho, foi utilizada a linguagem de programação Python, devido à quantidade de bibliotecas disponíveis que possuem o intuito de processar imagens e analisar dados.

Para realizar a leitura, recorte e reconhecimento nas imagens, foi utilizada a biblioteca OpenCV. Aliada à ela, foi utilizada para possibilitar a detecção das respostas marcadas a biblioteca NumPy, por meio do processamento da intensidade dos pixels de cada parte da ROI utilizada.

Por fim, para trabalhar com os dados obtidos nas etapas anteriores, foi utilizada a biblioteca Pandas, com o intuito de gerar um arquivo CSV contendo as colunas “questão”, “resposta” e “candidato”, seguindo a mesma estrutura do arquivo CSV que contém as respostas esperadas do reconhecimento.

### D. Etapas do sistema

O sistema desenvolvido para realizar o reconhecimento e a verificação automática das respostas dos cartões se divide em quatro (4) etapas, sendo elas a detecção da ROI e das colunas, detecção das questões e marcações, registro das respostas reconhecidas e comparação com as respostas esperadas.

a) *Detecção da ROI e das colunas*: a primeira etapa do sistema desenvolvido consiste na detecção da região de interesse (ROI) e das colunas presentes nos cartões-resposta. Para isso, utiliza-se a biblioteca OpenCV, onde cada imagem é inicialmente convertida para escala de cinza a fim de facilitar o processamento e reduzir a influência de variações de cor. Em seguida, são aplicados filtros de desfoque, como o Gaussian Blur, para suavizar a imagem minimizando ruídos, o que torna as bordas das colunas mais evidentes. O algoritmo de detecção de bordas de Canny é utilizado em seguida para realçar as estruturas retangulares presentes no cartão. Após detectar as bordas, uma operação de dilatação morfológica é realizada para a unificação de segmentos próximos, com o intuito de facilitar a identificação de contornos. Ao serem encontrados, ocorre uma filtragem para selecionar os

correspondentes às colunas de respostas, sendo posteriormente ordenadas da esquerda para a direita, garantindo a sequência das questões.

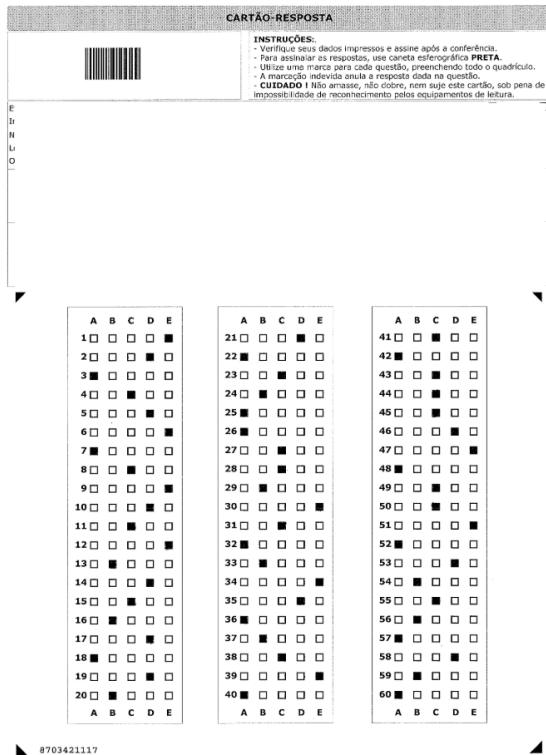


Figura 1. Cartão-resposta processado

A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
1					21					41				
2					22					42				
3					23					43				
4					24					44				
5					25					45				
6					26					46				
7					27					47				
8					28					48				
9					29					49				
10					30					50				
11					31					51				
12					32					52				
13					33					53				
14					34					54				
15					35					55				
16					36					56				
17					37					57				
18					38					58				
19					39					59				
20					40					60				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E

Figura 2. Colunas obtidas da esquerda para a direita

*b) Detecção das Questões e Marcações:* na segunda etapa, são utilizadas as bibliotecas OpenCV e NumPy para detectar as respostas assinaladas pelos candidatos. Para realizar isso, são calculadas as coordenadas exatas das áreas onde cada alternativa de resposta está localizada dentro de cada coluna, considerando parâmetros como o número total de questões, o número de alternativas por questão, as margens do cartão e o espaçamento entre as áreas de marcação. Com as regiões de cada alternativa definida, é aplicada uma

limiarização adaptativa para transformar a imagem em preto e branco, invertendo as cores para que as marcações fiquem destacadas. Em cada área correspondente a uma alternativa, é feita a contagem de pixels escuros utilizando o NumPy, o que indica a presença de uma resposta assinalada. A alternativa com maior quantidade de pixels escuros é entendida com escolhida, contando que a diferença em relação as outras seja considerável. Caso não exista esse destaque, o sistema entende que a questão foi deixada em branco e prossegue para a questão seguinte.

A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
1					21					41				
2					22					42				
3					23					43				
4					24					44				
5					25					45				
6					26					46				
7					27					47				
8					28					48				
9					29					49				
10					30					50				
11					31					51				
12					32					52				
13					33					53				
14					34					54				
15					35					55				
16					36					56				
17					37					57				
18					38					58				
19					39					59				
20					40					60				
A	B	C	D	E	A	B	C	D	E	A	B	C	D	E

Figura 3. Respostas reconhecidas e marcadas

*c) Registro das Respostas Reconhecidas:* a terceira etapa é responsável pelo registro das respostas obtidas anteriormente. Para cada gabarito processado, o sistema associa o número da questão, a alternativa identificada como marcada e o identificador do candidato. Esses dados são organizados em uma estrutura tabular utilizando a biblioteca Pandas, facilitando a manipulação e exportação dos dados obtidos. O registro é feito de forma padronizada em um arquivo CSV, garantindo que cada linha contenha as informações necessárias para a conferência das respostas, seguindo o mesmo padrão do arquivo CSV que contém as respostas esperadas.

*d) Comparação com as Respostas Esperadas:* a quarta e última etapa consiste em comparar as repostas reconhecidas com as respostas esperadas, a fim de averiguar a taxa de acerto do algoritmo de reconhecimento. Para realizar determinada tarefa, são utilizadas as bibliotecas Pandas a biblioteca padrão do Python, especificamente o módulo CSV, organizando-os de modo a permitir a comparação direta entre as repostas reconhecidas de cada candidato e seus respectivos gabaritos digitalizados. Após a comparação, é calculada a taxa de acerto em porcentagem, tanto de cada candidato quanto à média de todos eles; sendo esses resultados escritos em um arquivo de texto.

### III. RESULTADOS

O sistema descrito na seção II, que evidencia as etapas desenvolvidas a fim de realizar o reconhecimento e

comparação das respostas contidas nos gabaritos do vestibular da Universidade Estadual do Norte do Paraná, se mostrou altamente eficaz em relação ao objetivo proposto para o mesmo. Ao realizar o processamento do conjunto de imagens fornecidos, foi obtida a taxa de acerto de 100%, tanto de cada candidato individualmente, quanto à média de todos eles, indicando que o reconhecimento das repostas foi eficaz.

Contudo, vale ressaltar novamente que o objetivo do sistema desenvolvido não é averiguar se as respostas assinaladas por cada candidato estão corretas ou não, mas sim averiguar se o reconhecimento da área de interesse (ROI), das três colunas com vinte questões e das repostas assinaladas em cada uma delas foi efetuado corretamente, ou seja, se no final as respostas reconhecidas correspondem às respostas esperadas.

## REFERENCES

- [1] Python Software Foundation, “Python 3.12 documentation,” <https://docs.python.org/3.12/>, accessed May 12, 2025.
- [2] OpenCV Team, “OpenCV documentation,” <https://docs.opencv.org/4.x/index.html>, accessed May 12, 2025.
- [3] NumPy Developers, “NumPy Documentation,” <https://numpy.org/doc/>, accessed May 12, 2025.
- [4] Pandas, NumFOCUS, Inc, “Pandas Documentation,” <https://pandas.pydata.org/docs/>, accessed May 12, 2025.