

Universidade Tecnológica Federal do Paraná – UTFPR  
Departamento Acadêmico de Eletrônica – DAELN  
Engenharia Eletrônica  
Disciplina: ELTD2 – Processamento de Imagens  
Semestre: 2024.2  
Prof: Gustavo Borba

## RELATÓRIO Optical Mark Recognition (OMR)

Aluno:  
José Mário Nishihara de Albuquerque / 2191253

Fevereiro 2025

---

### 1 Objetivo

O objetivo deste trabalho é implementar um sistema de OMR para ser aplicado na correção de provas. O sistema desenvolvido deve ser capaz de:

- Adquirir as imagens das folhas de resposta (FR);
- Identificar as marcas oculares (MO) e a resposta associada a cada questão;
- Comparar as respostas com as de um gabarito;
- Atribuir uma nota.

### 2 Fundamentação Teórica

O Reconhecimento Óptico de Marcas (OMR, do inglês Optical Mark Recognition) é uma tecnologia amplamente utilizada para automatizar a correção de provas e o processamento de formulários, através da leitura de marcas feitas por um usuário em uma folha de resposta. As marcas, geralmente feitas com um lápis ou caneta, são interpretadas por sistemas de captura óptica que convertem as imagens em dados processáveis.

De acordo com Elias et al. (2021), o OMR utiliza sensores ópticos para identificar as marcas em uma folha, convertendo-as em valores digitais que podem ser analisados. Esse processo envolve várias etapas, incluindo a aquisição de imagem, a localização das marcas, a interpretação das respostas e a comparação com um gabarito. Os principais desafios encontrados nesse processo envolvem a variação na qualidade da imagem, a interferência de ruídos e a necessidade de alta precisão na detecção das marcas [1].

A figura 1 ilustra um exemplo de um formulário de OMR.

Além disso, o processo de detecção das marcas envolve técnicas de processamento de imagem, como binarização, segmentação e eliminação de ruídos. Essas técnicas ajudam a isolar as marcas das outras partes do formulário, garantindo que o sistema possa identificar corretamente as respostas.

Para processar imagens em tons de cinza e convertê-las em imagens binárias, uma técnica comum é a binarização adaptativa. Essa abordagem permite que o limiar de binarização seja ajustado de forma dinâmica, com base nas características locais da imagem, em vez de utilizar um único valor de limiar para toda a imagem.

A binarização adaptativa é útil em cenários onde a imagem possui variações de iluminação ou regiões sombreadas, o que dificultaria o uso de um limiar global. No método

Nome: \_\_\_\_\_

Número da prova:		①	②	③	④						
Número	A	B	C	D	E	Número	A	B	C	D	E
1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	26	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	27	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	28	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	29	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	30	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	31	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	32	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	33	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	34	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	35	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	36	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	37	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	38	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	39	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	40	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	41	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	42	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	43	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	44	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	45	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	46	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	47	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	48	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
24	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	49	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
25	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	50	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 1: Exemplo de formulário de OMR, com as áreas de marcação indicadas para o preenchimento das respostas.

adaptativo, o limiar de cada região da imagem é ajustado considerando o valor médio local e uma medida de sensibilidade, que controla a flexibilidade do limiar.

Ao utilizar essa técnica, o processo de segmentação da imagem se torna mais robusto, permitindo isolar de forma mais precisa os objetos de interesse, como marcas em formulários, por exemplo. A sensibilidade do algoritmo pode ser ajustada para tornar a binarização mais ou menos rígida, dependendo da necessidade de preservar ou remover detalhes sutis nas regiões de transição entre áreas claras e escuras.

Para o reconhecimento de marcas ópticas em formulários, é necessário identificar as regiões que contêm essas marcas de maneira precisa. Para isso, pode-se utilizar técnicas de análise de regiões em uma imagem binária, as quais permitem detectar áreas de interesse e extrair suas características geométricas.

Uma abordagem comum é a análise das regiões conectadas, que são grupos de pixels adjacentes com valores semelhantes, geralmente representados por valores de 1 em uma imagem binária. A partir dessas regiões, é possível calcular várias propriedades que ajudam na segmentação e identificação das marcas.

Uma das propriedades mais importantes para esse fim é a *Bounding Box*, ou caixa delimitadora. A *Bounding Box* é uma caixa retangular mínima que envolve completamente a região conectada, sendo definida pelas coordenadas do canto superior esquerdo e pelas dimensões (largura e altura) do retângulo que circunscreve a região. Essa propriedade é

útil para isolar a marca de interesse e realizar uma análise mais detalhada da sua posição e tamanho.

Com a *Bounding Box*, é possível, por exemplo, localizar a posição das marcas no formulário e determinar se elas correspondem às respostas corretas. Além disso, a utilização dessa propriedade permite filtrar áreas não relevantes, como ruídos ou outras partes da imagem que não contêm marcas, otimizando o processo de reconhecimento e aumentando a precisão do sistema.

### 3 Implementação

A implementação do sistema de reconhecimento de marcas ópticas (OMR) foi realizada em MATLAB, utilizando diversas funções de processamento de imagem para detectar e analisar as respostas dos formulários. A seguir, descrevemos as principais etapas do código desenvolvido.

O código começa com a leitura da imagem do formulário, que é redimensionada para um tamanho específico:

```
img = imread("Prova.jpeg");  
img = imresize(img, [NaN, 1414]);
```

Em seguida, a imagem é convertida para escala de cinza e binarizada utilizando um método adaptativo, o qual é ajustado com uma sensibilidade de 0.7:

```
img_gray = im2gray(img);  
img_bin = imbinarize(img_gray, 'adaptive', 'Sensitivity', 0.7);  
img_bin = ~img_bin;
```

A figura 2 mostra o resultado da binarização da imagem.

A função `regionprops` é utilizada para extrair as propriedades das regiões conectadas na imagem binária, especificamente a `BoundingBox`, que define as caixas delimitadoras de cada marca de resposta encontrada:

```
stats = regionprops(img_bin, 'BoundingBox');
```

Essas propriedades são então analisadas para filtrar as regiões que correspondem a possíveis marcas de resposta, com base em critérios de largura e altura:

```
largura_minima = 30;  
altura_minima = 30;  
largura_maxima = 50;  
altura_maxima = 50;
```

Após filtrar as regiões com base nas dimensões, o código continua com a identificação da maior caixa delimitadora (Bounding Box) da folha e o código verifica se cada alternativa está dentro dos limites da maior caixa encontrada. O trecho a seguir realiza esse processo:

```
bboxDentro = false(length(alternativas), 1);  
  
for i = 1:length(alternativas)  
    bboxAlternativa = alternativas(i).BoundingBox;
```

Nome: _____											
Número da prova:		①	②	③	④						
Número	A	B	C	D	E	Número	A	B	C	D	E
1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	26	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	27	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	28	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	29	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	30	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	31	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	32	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	33	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	34	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	35	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	36	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	37	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	38	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	39	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	40	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	41	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	42	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	43	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	44	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	45	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	46	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	47	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	48	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
24	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	49	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
25	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	50	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 2: Resultado da binarização da imagem

```

if bboxAlternativa(1) >= maiorBBox.BoundingBox(1) && ...
    bboxAlternativa(2) >= maiorBBox.BoundingBox(2) && ...
    (bboxAlternativa(1) + bboxAlternativa(3)) <= (maiorBBox.BoundingBox(1) + maiorBBox.BoundingBox(3)) && ...
    (bboxAlternativa(2) + bboxAlternativa(4)) <= (maiorBBox.BoundingBox(2) + maiorBBox.BoundingBox(4)) && ...
    bboxDentro(i) = true;
end
end
end

```

Esse trecho de código tem como objetivo garantir que apenas as alternativas localizadas dentro da maior caixa delimitadora (geralmente a região onde as alternativas de resposta estão) sejam consideradas. O algoritmo compara as coordenadas e as dimensões de cada alternativa com as da maior caixa delimitadora. Se uma alternativa estiver dentro da área da maior caixa, ela é marcada como “dentro” (com  $\text{bboxDentro}(i) = \text{true}$ ). Isso resulta em uma lista de alternativas que estão dentro da área da maior caixa, que são as regiões de interesse para a detecção das respostas.

As marcas de resposta são armazenadas em uma estrutura chamada `alternativas`, e as caixas delimitadoras são ordenadas com base na posição vertical ( $y$ ):

```
alternativas = alternativas(bboxDentro,:);
alternativasOrdenadas = reshape([alternativas.BoundingBox], 4, []).';
sortedByY = sortrows(alternativasOrdenadas, 2);
```

Após ordenar as alternativas com base na posição vertical (*y*), o código segue para dividir as alternativas em três grupos. O primeiro grupo, `group1`, representa o tipo da prova e é extraído das primeiras quatro alternativas ordenadas:

```
group1 = sortedByY(1:4, :);
alternativasOrdenadas = setdiff(alternativasOrdenadas, group1, 'rows');
alternativasOrdenadas = sortrows(alternativasOrdenadas, 2);
```

Já os grupos `group2` e `group3` representam as questões do lado esquerdo e direito da prova, respectivamente. O código faz a ordenação das alternativas com base na posição horizontal (*x*) para dividir as alternativas em duas partes:

```
numGrupos = 25;
tamanhoGrupo = size(alternativasOrdenadas, 1) / numGrupos;
grupos = cell(numGrupos, 1);
for i = 1:numGrupos
    inicio = (i-1) * tamanhoGrupo + 1;
    fim = i * tamanhoGrupo;
    grupos{i} = alternativasOrdenadas(inicio:fim, :);
end
group2 = [];
group3 = [];
for i = 1:numGrupos
    sortedByX = sortrows(grupos{i}, 1);
    group2 = [group2; sortedByX(1:5, :)];
    group3 = [group3; sortedByX(end-4:end, :)];
end
alternativasOrdenadas = [group2; group3];
```

O grupo `group2` contém as alternativas localizadas no lado esquerdo da prova, enquanto o grupo `group3` contém as alternativas do lado direito. Ambos os grupos são então ordenados novamente, agora considerando tanto a posição vertical (*y*) quanto a horizontal (*x*) das alternativas. Ao final, as alternativas ordenadas são recombinadas, com o `group2` vindo antes do `group3`, para garantir que as alternativas da prova estejam dispostas corretamente, facilitando a análise posterior das respostas. A figura 3 mostra o resultado final da ordenação.

Após ordenar as alternativas, as questões são agrupadas e armazenadas em uma estrutura de células, chamada `questoes`, onde cada célula contém as alternativas de uma questão específica:

```
questoes = cell(50, 1);
for i = 1:50
    startIdx = (i - 1) * 5 + 1;
    endIdx = i * 5;
    questoes{i} = alternativas(startIdx:endIdx);
end
```

Nome: \_\_\_\_\_

Número da prova:						①	②	③	④		
Número	A	B	C	D	E	Número	A	B	C	D	E
1	A 1	A 2	A 3	A 4	A 5	26	A 126	A 127	A 128	A 129	A 130
2	A 6	A 7	A 8	A 9	A 10	27	A 131	A 132	A 133	A 134	A 135
3	A 11	A 12	A 13	A 14	A 15	28	A 136	A 137	A 138	A 139	A 140
4	A 16	A 17	A 18	A 19	A 20	29	A 141	A 142	A 143	A 144	A 145
5	A 21	A 22	A 23	A 24	A 25	30	A 146	A 147	A 148	A 149	A 150
6	A 26	A 27	A 28	A 29	A 30	31	A 151	A 152	A 153	A 154	A 155
7	A 31	A 32	A 33	A 34	A 35	32	A 156	A 157	A 158	A 159	A 160
8	A 36	A 37	A 38	A 39	A 40	33	A 161	A 162	A 163	A 164	A 165
9	A 41	A 42	A 43	A 44	A 45	34	A 166	A 167	A 168	A 169	A 170
10	A 46	A 47	A 48	A 49	A 50	35	A 171	A 172	A 173	A 174	A 175
11	A 51	A 52	A 53	A 54	A 55	36	A 176	A 177	A 178	A 179	A 180
12	A 56	A 57	A 58	A 59	A 60	37	A 181	A 182	A 183	A 184	A 185
13	A 61	A 62	A 63	A 64	A 65	38	A 186	A 187	A 188	A 189	A 190
14	A 66	A 67	A 68	A 69	A 70	39	A 191	A 192	A 193	A 194	A 195
15	A 71	A 72	A 73	A 74	A 75	40	A 196	A 197	A 198	A 199	A 200
16	A 76	A 77	A 78	A 79	A 80	41	A 201	A 202	A 203	A 204	A 205
17	A 81	A 82	A 83	A 84	A 85	42	A 206	A 207	A 208	A 209	A 210
18	A 86	A 87	A 88	A 89	A 90	43	A 211	A 212	A 213	A 214	A 215
19	A 91	A 92	A 93	A 94	A 95	44	A 216	A 217	A 218	A 219	A 220
20	A 96	A 97	A 98	A 99	A 100	45	A 221	A 222	A 223	A 224	A 225
21	A 101	A 102	A 103	A 104	A 105	46	A 226	A 227	A 228	A 229	A 230
22	A 106	A 107	A 108	A 109	A 110	47	A 231	A 232	A 233	A 234	A 235
23	A 111	A 112	A 113	A 114	A 115	48	A 236	A 237	A 238	A 239	A 240
24	A 116	A 117	A 118	A 119	A 120	49	A 241	A 242	A 243	A 244	A 245
25	A 121	A 122	A 123	A 124	A 125	50	A 246	A 247	A 248	A 249	A 250

Figura 3: Alternativas ordenadas na imagem da prova.

Uma vez que as alternativas estão organizadas e agrupadas, a detecção da resposta escolhida por cada aluno é realizada pela contagem de pixels na região de cada alternativa. A alternativa com o maior número de pixels é considerada como a resposta correta para cada questão, desde que esse número ultrapasse um limite mínimo, definido pelo `threshold_num_pixel`:

```
threshold_num_pixel = 700;
for k = 1:50
    max_pixels = 0;
    resposta_atual = 0;

    for j = 1:5
        bbox = questoes{k}(j).BoundingBox;
        region = imcrop(img_bin, bbox);
        num_pixels = sum(region(:) == 1);

        if num_pixels > max_pixels
            if num_pixels > threshold_num_pixel
                max_pixels = num_pixels;
                resposta_atual = j;
            end
        end
    end
end
```

```
        else
            resposta_atual = 0;
        end
    end
end
respostas(k) = resposta_atual;
end
```

Ao final, o vetor **respostas** contém a solução do formulário, ou seja, as alternativas escolhidas pelo aluno para cada questão.

#### 4 Resultados e Conclusões

O código desenvolvido é capaz de operar somente com o gabarito desenvolvido para ele; qualquer mudança nesse gabarito requer uma alteração no código para continuar funcionando. A utilização da função **regionprops** para reconhecer as marcas na folha se mostrou muito eficiente. A função obtém os valores das coordenadas precisamente, o que facilita a ordenação das questões.

Apesar de sua eficiência, o código apresenta limitações. Ele não consegue lidar adequadamente com imagens que estejam muito tortas, pois o processo de ordenação e posterior classificação depende da correta localização das coordenadas de cada questão na imagem. Caso a orientação da imagem esteja incorreta, o algoritmo pode falhar na organização das alternativas.

A binarização da imagem utilizando o método de Otsu adaptativo com uma sensibilidade relativamente alta se provou eficaz, sendo capaz de binarizar imagens corretamente, mesmo em condições de iluminação desfavoráveis. Essa abordagem foi essencial para o processamento de imagens de provas que, por vezes, apresentaram variações significativas na iluminação.

Embora o sistema tenha se mostrado funcional dentro do contexto do gabarito específico, algumas melhorias podem ser consideradas para aumentar a robustez e a flexibilidade da implementação. Entre as sugestões de melhorias estão:

- Desenvolvimento de um método para corrigir a orientação da imagem, evitando erros causados pela rotação incorreta da folha.
- Implementação de um maior número de questões no gabarito, ampliando a capacidade de análise do código.
- Melhoria da capacidade do código para operar com diferentes formatos de gabarito, tornando-o mais versátil e aplicável a outros tipos de provas.

#### Referências

- [1] Erik Miguel de Elias, Paulo Marcelo Tasinaffo, and R Hirata Jr. Optical mark recognition: Advances, difficulties, and limitations. *SN Computer Science*, 2(5):367, 2021.