



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Detecção automática de marcações em folha de resposta usando visão computacional

Autor: Lucas Costa Araújo
Orientador: Prof. Dr. Glauco Vitor Pedrosa

Brasília, DF
2020



Lucas Costa Araújo

Detecção automática de marcações em folha de resposta usando visão computacional

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Glauco Vitor Pedrosa

Brasília, DF

2020

Lucas Costa Araújo

Detecção automática de marcações em folha de resposta usando visão computacional/ Lucas Costa Araújo. – Brasília, DF, 2020-

43 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Glauco Vitor Pedrosa

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2020.

1. Visão-Computacional. 2. Aprendizado-de-Máquina. I. Prof. Dr. Glauco Vitor Pedrosa. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Detecção automática de marcações em folha de resposta usando visão computacional

CDU 02:141:005.6

Lucas Costa Araújo

Detecção automática de marcações em folha de resposta usando visão computacional

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 09 de dezembro 2020:

Prof. Dr. Glauco Vitor Pedrosa
Orientador

Prof. Dr. Bruno César Ribas
Convidado 1

**Prof. Dr. John Lenon Cardoso
Gardenghi**
Convidado 2

Brasília, DF
2020

Resumo

Folhas de respostas são utilizadas em atividades avaliativas objetivas, nas quais deve-se marcar uma resposta dentre um conjunto de possibilidades (estilo ENEM e vestibular). Este trabalho apresenta o desenvolvimento de um sistema baseado em visão computacional para identificar as marcações em folhas de respostas. Para isso, foram implementados e testados vários algoritmos/técnicas de processamento de imagens a fim de identificar a combinação que gerasse a melhor acurácia na segmentação automática dessas marcações. Essa implementação foi realizada em linguagem Python com o auxílio da biblioteca OpenCV. Para a realização de testes experimentais foi utilizada uma base de dados composta por 936 folhas de respostas preenchidas e escaneadas por máquinas diferentes. Como resultado, o método desenvolvido neste trabalho apresentou uma acurácia de 99,97%, demonstrando uma alta eficácia para a solução desenvolvida.

Palavras-chaves: Visão Computacional, Processamento de Imagens, OpenCV, Automação de Tarefas, Engenharia de Software.

Lista de ilustrações

Figura 1 – Esquema de um sistema de visão computacional	15
Figura 2 – Tipos de imagens digitais: (a) binária (b) nível de cinza (c) colorida . .	16
Figura 3 – Exemplos de elementos estruturantes binários.	17
Figura 4 – Imagem original (esquerda) e imagem dilatada (direita).	17
Figura 5 – Imagem original (esquerda) imagem erodida (direita).	18
Figura 6 – Imagem original (esquerda) e a mesma imagem após aplicada a técnica de abertura (direita).	18
Figura 7 – Imagem original (esquerda) e a mesma imagem após aplica a técnica de fechamento (esquerda).	19
Figura 8 – (a) Imagem original em nível de cinza (b) Binarização com threshold de 120 (c) Binarização usando o método de OTSU (d) Binarização usando Adaptive Thresholding.	20
Figura 9 – À esquerda a imagem original, as formas são pretas o fundo é branco. À direita a imagem com contornos identificados, representado pelas linhas coloridas. Pode parecer que o círculo branco (linha verde) é uma forma, porém é parte interna do retângulo preto. Há apenas formas pretas. . .	21
Figura 10 – Da esquerda para a direita. A primeira imagem é a original, capturada pela câmera e sem defeitos. A segunda, adicionando apenas ruído preto. A terceira, aplicando blur gaussiano com máscara de 5x5. A Quarta, aplicando filtro de mediana com máscara de 5x5.	22
Figura 11 – Da esquerda para a direita. A primeira imagem é a original, capturada pela câmera e sem defeitos. A segunda, adicionando ruído sal(ruído branco) e pimenta (ruído preto). A terceira, aplicando blur gaussiano com máscara de 15x15 em duas iterações. A quarta, aplicando filtro de mediana com máscara 15x15.	22
Figura 12 – Começando de cima para baixo. A primeira é a imagem original, percebe-se a dificuldade de reconhecer alguns detalhes como as fitas na estante e a escultura escura no topo. A segunda é a imagem equalizada com histograma global, percebe-se que a escultura escura no topo agora é reconhecível, mas a escultura ao centro da tela perdeu bastante de seus detalhes. A terceira é a imagem equalizada com CLAHE, que apresenta tanto a melhora na visualização de detalhes escuros como as fitas na estante, sem prejudicar a de detalhes claros como a escultura ao centro, demonstrando sua adaptabilidade à imagem de entrada.	24

Figura 13 – a) Folha de resposta não preenchida. b) Retângulo de referência, utilizado para identificar a ROI. c) Região de interesse (ROI), onde contém as informações para detecção das marcações. d) Região da questão, possuindo 5 células da grade que serão classificadas em preenchida ou não preenchida.	28
Figura 14 – Exemplo de um cartão de respostas parcialmente preenchido.	29
Figura 15 – Fluxo bpmn de processamento da folha de resposta	31
Figura 16 – Etapa de pré-processamento	32
Figura 17 – À esquerda o exemplo do resultado do pré-processamento realizado para verificação do retângulo de referência. À direita o exemplo do resultado do pré-processamento realizado para leitura individual de cada célula da grade.	32
Figura 18 – Etapa de identificação e recorte	33
Figura 19 – Exemplo de identificação da estrutura e definição dos pontos a serem analisados para a correção de perspectiva e recorte do retângulo delimitador das questões.	34
Figura 20 – Exemplo de borda fragmentada onde o algoritmo não foi capaz de identificar o retângulo de referência.	36
Figura 21 – Exemplo de leitura individual, acima a imagem original, abaixo a imagem pré-processada para leitura.	37
Figura 22 – À esquerda a imagem com deformação original, à direita a ROI recortada, corrigida a perspectiva, e realizada a leitura individual de cada célula da grade.	37
Figura 23 – À esquerda a imagem original recebida para leitura e à direita a leitura realizada com a questão 162 ampliada, percebe-se que possui uma leve marcação na questão 162 que não foi considerada preenchida pela classificação.	38
Figura 24 – À esquerda a imagem original recebida para leitura com bolinhas fracas e também vazadas no centro, e à direita a leitura realizada.	38
Figura 25 – À esquerda a imagem original recebida para leitura com exemplos de variação de força das marcações, e à direita a leitura realizada.	39

Lista de tabelas

Tabela 1 – Resultados finais da execução do algoritmo proposto	35
--	----

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	13
1.2	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Sistema de Visão Computacional	15
2.2	Tipos de Imagens Digitais	16
2.3	Morfologia Matemática	16
2.3.1	Dilatação e Erosão	17
2.3.2	Abertura e Fechamento	18
2.4	Segmentação de imagens	19
2.5	Análise estrutural ou <i>Shape Analysis</i>	20
2.6	Filtros	21
2.7	Equalização de histograma	22
3	MATERIAIS E MÉTODOS	25
3.1	Trabalhos Correlatos	25
3.2	Desenvolvimento do Sistema	27
3.2.1	Base de Dados	27
3.2.2	Validação do sistema	27
3.2.3	Ferramentas de Apoio	27
4	TÉCNICA PROPOSTA	31
4.1	Etapas do processamento	31
4.1.1	Etapa 1: Pré-processamento	31
4.1.2	Etapa 2: Identificação e recorte	32
4.1.3	Etapa 3: Classificação das marcações	33
4.2	Testes Experimentais	34
4.2.1	Resultados obtidos	35
5	CONCLUSÃO	41
	REFERÊNCIAS	43

1 Introdução

De acordo com o jornal Folha de São Paulo, o ENEM (Exame Nacional de Ensino Médio) de 2019 teve 5.1 milhões de inscrições realizadas. Para cada inscrito é impresso uma série de folhas, entre elas a folha de respostas. A folha de respostas ou gabarito, é uma folha especial previamente identificada que deve ser resistente o suficiente para passar pelo processo de transporte e chegar sem defeitos à máquina de leitura, também chamada de OMR (Optical Mark Recognition). Estas máquinas possuem configurações específicas para ler gabaritos e necessitam que a qualidade da folha atenda a gramatura correta e cor de impressão para realizar a leitura corretamente.

Cursinhos, escolas de ensino médio ou preparatórias, utilizam o sistema de gabarito para facilitar e diminuir o gasto com correção. Porém o uso de máquinas OMR não é acessível para todos, então soluções alternativas são bem-vindas. Uma alternativa é utilizar o processamento digital de imagem para extrair a marcação das respostas, porém essa não é uma atividade trivial e a complexidade aumenta de acordo com os requisitos necessários de acurácia e remoção de ruídos que são gerados no transporte, forma de escaneamento, qualidade de impressão, forma de preenchimento, entre outros.

No caso de cursinhos, o layout ou template da folha de respostas é desenvolvido para que se assemelhe ao gabarito utilizado no ENEM. A folha de respostas do ENEM possui 90 questões, cada uma com cinco opções de marcação totalizando 450 possíveis lugares de marcação. Alguns colégios utilizam essas avaliações como parte efetiva da nota dos alunos e necessitam que as leituras sejam confiáveis. Ao mesmo tempo, utilizam o mínimo de recursos necessários para imprimir e escanear as folhas de respostas, o que aumenta a vulnerabilidade das imagens de possuírem ruídos e dificulta garantir a confiabilidade do processamento das folhas.

1.1 Objetivos

Este trabalho possui o objetivo geral de detectar, de maneira automática, marcações em folhas de respostas de avaliações objetivas usando a visão computacional.

Para atingir o objetivo geral, alguns objetivos específicos foram definidos:

- Definir uma base de imagens para realizar treinamentos/testes;
- Investigar diferentes técnicas de processamento de imagem para detecção de marcações nas folhas de respostas;

- Realizar testes comparativos para analisar o desempenho dos algoritmos investigados.

1.2 Organização do Trabalho

No [Capítulo 2](#) - Fundamentação Teórica - é apresentada uma revisão técnica sobre alguns fundamentos de processamento de imagens digitais, abordando questões básicas como a definição do que é imagem digital, e algoritmos de pré e pós processamento, como transformações morfológicas, limiarização, segmentação, etc.

No [Capítulo 3](#) - Materiais e Métodos - é apresentado o método utilizado para desenvolvimento do trabalho. Também são analisados e discutidos alguns artigos que aplicaram algoritmos em leitura de marcação, quais etapas foram utilizadas no processamento, quantidade de base de teste, qualidade das imagens testadas.

No [Capítulo 4](#) - Técnica Proposta - relata-se o desenvolvimento dos algoritmos, execução dos algoritmos na base de dados, e coleta dos dados de leitura e métricas.

No [Capítulo 5](#) - Conclusão - são apresentados os resultados provenientes da experimentação, dificuldades e casos específicos encontrados na base que ocasionaram falha na execução do algoritmo e que são oportunidades para o desenvolvimento de trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo são apresentados conceitos básicos sobre a visão computacional, bem como algumas técnicas de processamento digital de imagens que foram utilizadas para o desenvolvimento desse trabalho.

2.1 Sistema de Visão Computacional

A visão computacional é uma área de estudo que tenta replicar nas máquinas a capacidade da visão humana. A visão não se refere apenas ao ato de captar imagens, mas também na capacidade de extrair várias informações da imagem analisada, por exemplo, cor, forma e textura, bem como relacionar as imagens com outras vistas anteriormente.

Um sistema de visão computacional é constituído de várias fases, tal como esquematizado pela Figura 1: aquisição, processamento de imagens, segmentação, extração de características/análise de imagens e reconhecimento de padrões. Não há uma descrição única para todas as fases de um sistema de visão computacional. Alguns autores chamam “processamento de imagens” todas as fases. Além disso, algumas fases podem ser suprimidas dependendo do problema, por exemplo, uma imagem pode seguir direto para a fase de extração de características sem passar pela fase de segmentação.

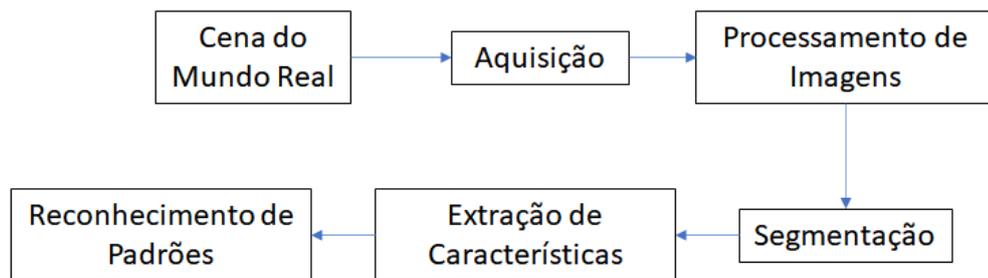


Figura 1 – Esquema de um sistema de visão computacional

O fase de processamento de imagens digitais se refere ao processamento de imagens digitais por um computador e a diferença entre visão computacional e processamento de imagens pode se tornar difícil de se perceber em alguns momentos. Alguns consideram que a área de processamento de imagens é definida apenas quando uma rotina de computador tem como sua entrada e saída, uma imagem digital. Essa definição é superficial e não representa totalmente esse campo de pesquisa, pois apenas por realizar uma tarefa básica como processar uma imagem para encontrar sua intensidade média, tem como saída um número (GONZALEZ, 2008).

2.2 Tipos de Imagens Digitais

Uma imagem digital é uma matriz de pixels, em que cada pixel é caracterizado por uma localização (em dois eixos para imagens 2D) e uma intensidade. Dependendo da intensidade, uma imagem pode ser de três tipos, tal como ilustrada pela Figura 2, que são:

- Imagem Binária
- Imagem em Nível de Cinza
- Imagem Colorida



Figura 2 – Tipos de imagens digitais: (a) binária (b) nível de cinza (c) colorida

Uma imagem binária é aquela em que os pixels assumem apenas duas intensidades (geralmente 0 e 1 ou 0 e 255), uma intensidade representando a cor branca e a outra intensidade representando a cor preta (que seria o fundo da imagem). Imagens em nível de cinza (grayscale) possuem pixels com valores que representam diferentes tonalidades de cinzas. No formato mais comumente usado, essas intensidades assumem 256 valores inteiros entre 0 (cor preta) e 255 (cor branca).

Se a imagem for colorida a intensidade por ser dada por mais de um valor, dependendo do modelo de cor utilizado. O modelo de cor mais comum é o RGB (red-green-blue). Neste modelo, a intensidade é representada por 3 valores: 1 valor representando o canal de cor vermelho (*red*), 1 valor representando o canal de cor verde (*green*) e 1 valor representando o canal de cor azul (*blue*) (GONZALEZ, 2008).

2.3 Morfologia Matemática

Na matemática a morfologia se refere ao estudo da geometria das formas de objetos. Na área de processamento de imagens, o processamento morfológico contribui para várias etapas de pré e pós processamento e de análise de imagem, como realce, filtragem, segmentação, detecção de bordas, esqueletização, etc. (GONZALEZ, 2008).

A morfologia matemática se baseia na teoria dos conjuntos (FILHO; NETO, 1999). Sendo assim, várias operações válidas sobre os conjuntos podem ser aplicadas em imagens,

tais como: união, intersecção, diferença, etc. Uma operação morfológica pode ser entendida como uma transformação na imagem, que é realizada utilizando uma segunda imagem menor e cuja estrutura e geometria sejam conhecidas. Essa segunda imagem é conhecida como elemento estruturante.

As operações morfológicas podem ser aplicadas em imagens binárias ou em níveis de cinza. Nesse trabalho, o foco será somente em operações realizadas em imagens binárias. Para isso, deve ser definido um elemento estruturante do tipo binário, tal como ilustrado na Figura 3. Um elemento estruturante binário é um conjunto de coordenadas de pixels que correspondem ao objeto (cor preta) sendo um desses pontos definidos como o centro do elemento (marcado com um ponto). O elemento estruturante é deslizado por todos os pixels da imagem e, para cada um deles, a operação morfológica é realizada.

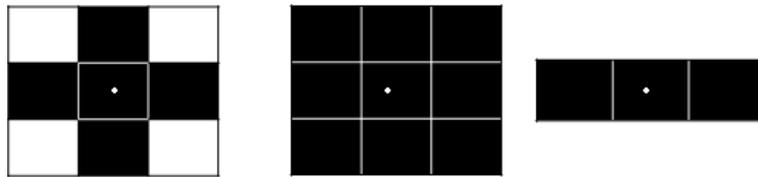


Figura 3 – Exemplos de elementos estruturantes binários.

A seguir serão detalhadas algumas técnicas comumente utilizadas no processamento morfológico de imagens digitais. Vale ressaltar que as técnicas que serão apresentadas são aplicadas somente à imagens binárias, ou seja, imagens que possuem apenas duas intensidades de cores: preto e branco.

2.3.1 Dilatação e Erosão

A dilatação e a erosão são duas técnicas populares do processamento morfológico de imagens digitais. A dilatação consiste em “expandir” os pixels do objeto da imagem. Essa expansão ocorre se o elemento estruturante obtiver ao menos uma de suas posições diferente de zero. A Figura 4 mostra um exemplo de imagem digital após aplicada a dilatação.



Figura 4 – Imagem original (esquerda) e imagem dilatada (direita).

Por outro lado, a erosão é o inverso da dilatação. Na erosão ocorre uma “diminuição” do objeto presente na imagem. Essa diminuição ocorre por somente ser contabilizado o pixel transformado, caso o elemento estruturante inteiro estiver inserido totalmente no objeto. A Figura 5 mostra um exemplo de imagem digital após aplicada a erosão.



Figura 5 – Imagem original (esquerda) imagem erodida (direita).

2.3.2 Abertura e Fechamento

Abertura e fechamento são duas abordagens que utilizam as técnicas de erosão e dilatação mostradas anteriormente. A combinação dessas duas técnicas intercaladas pode ser utilizada para “fechar” ou “abrir” objetos em uma imagem digital. A abertura e o fechamento possuem o intuito de preservar regiões maiores da imagem e/ou remover pequenas regiões, por exemplo, ruídos.

A abertura aplica primeiro uma erosão que irá diminuir o objeto da imagem e assim, permitir a remoção de pequenas regiões brancas da imagem que podem ser ruídos. Logo depois, aplica-se a dilatação para recuperar o objeto ao seu tamanho original. A Figura 6 mostra um exemplo de imagem que sofreu a aplicação da técnica de abertura.



Figura 6 – Imagem original (esquerda) e a mesma imagem após aplicada a técnica de abertura (direita).

Por outro lado, o fechamento aplica primeiro uma dilatação, o que irá expandir o objeto e assim irá permitir diminuir possíveis “buracos” que possam existir dentro do objeto na imagem. Em seguida é aplicada a erosão para retornar o objeto ao seu tamanho

original. A Figura 7 mostra um exemplo de imagem que sofreu a aplicação da técnica de abertura.



Figura 7 – Imagem original (esquerda) e a mesma imagem após aplica a técnica de fechamento (esquerda).

2.4 Segmentação de imagens

Segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões. Ela é tipicamente usada para localizar objetos e formas (linhas, curvas, etc) em imagens. O resultado da segmentação de imagens é um conjunto de regiões/objetos ou um conjunto de contornos extraídos da imagem. Como resultado, cada um dos pixels em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. Regiões adjacentes devem possuir diferenças significativas com respeito a mesma característica(s).

Thresholding (ou binarização), é uma das técnicas de segmentação mais populares na área de processamento de imagens (GONZALEZ; WOODS, 2018). Ela tem como objetivo a separação dos pixels de uma imagem em dois grupos: um grupo seria o fundo e o outro grupo seria o objeto alvo. Para realizar essa segmentação se aplica uma verificação na intensidade do valor de cada pixel da imagem: caso o pixel esteja acima de um nível (limiar) determinado, então ele é definido como um pixel pertencente ao conjunto do objeto alvo, caso contrário ele é considerado como fundo.

Quando o algoritmo é aplicado em toda a imagem, a técnica é denominada de *global thresholding*, quando o valor de threshold varia de acordo com uma determinada região da imagem é chamado de *variable thresholding* ou *adaptive thresholding*. Essa variação ocorre considerando os vizinhos do pixel que está sendo analisado (GONZALEZ; WOODS, 2018).

O sucesso da segmentação por *thresholding* consiste em escolher um limiar (valor) adequado. Para que essa escolha não dependa do usuário, foi criado o método de Otsu. Esse é um método matemático que almeja encontrar um limiar ótimo que divida uma imagem em duas regiões de modo a minimizar a variância em cada região. Para isso ele utiliza

a análise do histograma da imagem. A Figura 8 mostra a diferença entre a segmentação utilizando o algoritmo de binarização simples, utilizando o método de OTSU e a técnica de Adaptive Thresholding (GONZALEZ; WOODS, 2018).



Figura 8 – (a) Imagem original em nível de cinza (b) Binarização com threshold de 120 (c) Binarização usando o método de OTSU (d) Binarização usando Adaptive Thresholding.

2.5 Análise estrutural ou *Shape Analysis*

Análise estrutural é uma área que recebe bastante atenção de pesquisadores. Uma imagem mesmo com o pré-processamento ainda possui muitos dados que são complexos para extrair informações. A análise estrutural verifica padrões para extrair dados. Algoritmos de análise estrutural reduzem as formas encontradas na imagem em descritores, que representam a forma analisada, tornando mais claro a classificação das formas, obtendo informações de interesse.

Algoritmos de esqueletização normalmente descrevem o interior da forma. Em (OGIELA, 2002) é utilizado esqueletização para reconhecer a linha central das artérias a partir de imagens coronográficas. Com a linha central definida, é possível verificar a largura da artéria e identificar pontos de estreitamento que representam perigo ao paciente.

Algoritmos de borda descrevem o exterior da forma. Em (KRISHNAN, 2012) é apresentado como exemplo, a utilização de dois algoritmos Moore, e Chain code, para identificação da borda de um coração humano. Algoritmos de borda podem ser utilizados em conjunto com esqueletização, sendo o primeiro, interessante para segmentação, o segundo para descrição interna.

Bordas ou contornos podem ser definidos como um conjunto de pixels contínuos de mesma intensidade que acompanham a borda do objeto. Em (SUZUKI, 1985) é apresentado uma proposta como evolução do reconhecimento de bordas por seguimento de bordas ou *Border following*, da qual estabelece uma hierarquia de composições e categoriza vários objetos, em uma imagem binária. O *Border following* é o algoritmo que estabelece uma corrente de coordenadas, individualmente categorizadas, que identificam um objeto. Como se pode ver na Figura 9, a identificação ocorre tanto para borda externa como

borda interna da forma. Este algoritmo é muito interessante ao trabalho atual e pode ser encontrado na biblioteca OpenCV, como *findContour*.

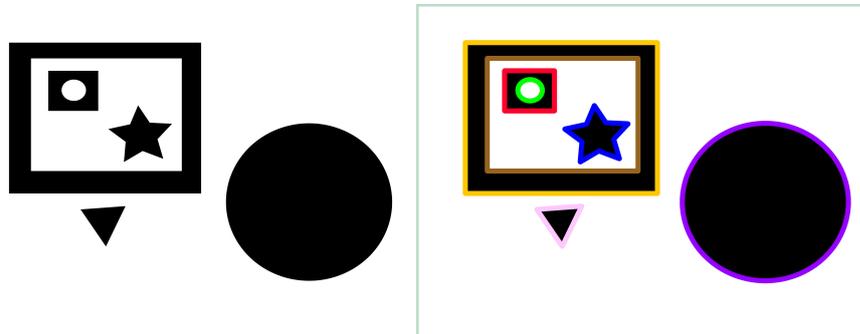


Figura 9 – À esquerda a imagem original, as formas são pretas o fundo é branco. À direita a imagem com contornos identificados, representado pelas linhas coloridas. Pode parecer que o círculo branco (linha verde) é uma forma, porém é parte interna do retângulo preto. Há apenas formas pretas.

2.6 Filtros

Filtragem espacial pode ser definida sendo uma convolução, que se utiliza de uma máscara (ou kernel, semelhante ao elemento estruturante das operações morfológicas), para realizar operações no espaço da imagem (QUEIROZ, 2001). A máscara pode ser configurada mudando seu tamanho e distribuição de pesos a fim de alcançar as operações desejadas.

Uma das utilizações de filtros é a remoção de ruído que normalmente é utilizada na etapa de pré-processamento. Uma imagem com ruído pode prejudicar uma análise ou transformação morfológica influenciando a identificação e operação das formas. A fim de remover o ruído, uma técnica amplamente utilizada é o Blur Gaussiano ou Filtro Gaussiano. Essa técnica utiliza um kernel de tamanho predefinido, que passa pela imagem aplicando a distribuição de Gauss para definir os valores dos pixels adjacentes, gerando assim, a suavização dos pixels.

O resultado da aplicação de dois filtros pode ser visto nas Figuras 10 e 11. Na primeira a utilização do filtro da mediana é muito mais eficaz, restaura quase que completamente a imagem original, enquanto o blur gaussiano apenas suavizou o ruído. O impacto negativo do filtro da mediana se apresenta mais na segunda imagem, onde apesar de recuperar boa parte do fundo, a imagem sofreu deformações e aglutinou ruídos pretos. O filtro gaussiano preservou mais características, e não gerou concentrações de preto, sendo uma opção mais confiável para limiarização.

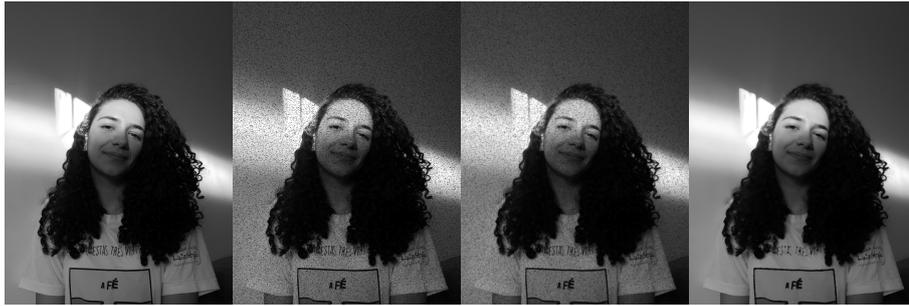


Figura 10 – Da esquerda para a direita. A primeira imagem é a original, capturada pela câmera e sem defeitos. A segunda, adicionando apenas ruído preto. A terceira, aplicando blur gaussiano com máscara de 5x5. A Quarta, aplicando filtro de mediana com máscara de 5x5.



Figura 11 – Da esquerda para a direita. A primeira imagem é a original, capturada pela câmera e sem defeitos. A segunda, adicionando ruído sal(ruído branco) e pimenta (ruído preto). A terceira, aplicando blur gaussiano com máscara de 15x15 em duas iterações. A quarta, aplicando filtro de mediana com máscara 15x15.

2.7 Equalização de histograma

Histogramas representam a distribuição de valores, no caso de imagens em nível de cinza, representam os valores de cinza, para imagens em RGB, é necessário três histogramas para representar a imagem, um histograma para cada cor.

([ACHARYA; RAY, 2005](#)) define a equalização de histograma como uma técnica para ajustar a escala de cinza da imagem, mapeando o histograma da imagem de entrada, para um histograma uniforme, gerando a imagem de saída. O histograma é analisado baseado na imagem como um todo. De acordo com ([EASTON, 2010](#)) a equalização de histogramas pode mudar tanto o caráter espacial, a distribuição de valores, quanto a quantização, o valor geral da imagem. Concentrações de valores são redistribuídas pela imagem.

Equalização de histograma é utilizado para pré-processamento e também como melhoria e destaque de detalhes na imagem, utilizado em uma grande variedade de áreas. Na área médica como apresentado em ([RUBINI, 2019](#)), onde a equalização de histograma é utilizada para melhorar os detalhes de imagens de ressonância magnética a fim de facilitar

a visualização pelo médico. Também em (RUBINI, 2019) é discutido a eficácia no uso de equalizações adaptativas como AHE e CLAHE e ao final é concluído que CLAHE é mais eficiente na equalização.

Adaptative Histogram Equalization ou AHE, é uma técnica da qual se diferencia da equalização global, por não mais considerar a imagem toda como uma só. AHE secciona a imagem em histogramas, onde são analisados e computados para produzir ajustes de contraste locais, proporcionando melhor valorização dos detalhes. Porém esses ajustes também podem gerar amplificação de ruídos.

Contrast Limited Adaptive Histogram Equalization ou CLAHE, é uma técnica aprimorada a partir do AHE, que limita a amplificação de contraste e previne a amplificação de ruídos. A Figura 12 exemplifica a efetividade do CLAHE para valorizar os detalhes sem amplificar ruídos.



Figura 12 – Começando de cima para baixo. A primeira é a imagem original, percebe-se a dificuldade de reconhecer alguns detalhes como as fitas na estante e a escultura escura no topo. A segunda é a imagem equalizada com histograma global, percebe-se que a escultura escura no topo agora é reconhecível, mas a escultura ao centro da tela perdeu bastante de seus detalhes. A terceira é a imagem equalizada com CLAHE, que apresenta tanto a melhora na visualização de detalhes escuros como as fitas na estante, sem prejudicar a de detalhes claros como a escultura ao centro, demonstrando sua adaptabilidade à imagem de entrada.

3 Materiais e Métodos

A fim de desenvolver este trabalho definiu-se a seguinte metodologia, dividida em quatro etapas:

1. Estudo e definição dos conhecimentos necessários sobre a área de processamento digital de imagens e aprendizado de máquina para o desenvolvimento do trabalho;
2. Levantamento de trabalhos correlatos na área de detecção de marcações em folhas de resposta.
3. Definição dos materiais e métodos para o desenvolvimento do sistema
4. Validação do sistema desenvolvido

Os conceitos relacionados à etapa 1 já foram apresentados no capítulo anterior desse trabalho. O detalhamento das demais etapas será descrito a seguir.

3.1 Trabalhos Correlatos

Para uma análise de cada um dos trabalhos correlatos levou-se em consideração os seguintes tópicos:

- Identificação das técnicas de processamento de imagens utilizadas
- Quantidade de cartões processados para obtenção da leitura
- Qualidade dos cartões submetidos para leitura
- Técnica de aquisição da imagem, se foi escaneada ou fotografada por uma câmera
- Quais foram os resultados obtidos.

No trabalho de (GUPTA, 2016) foi utilizado um modelo previamente definido para realizar a leitura dos cartões-respostas. Técnicas de transformação como correção de alinhamento e escala, rotacionando e redimensionando a imagem baseada nos pixels dos quatro cantos extremos da imagem. Não foi relatada a quantidade de cartões lidos para verificar o grau de assertividade do projeto e também a variação de qualidade das imagens. As imagens utilizadas foram escaneadas por uma máquina scanner comum.

Em (BELAG, 2018), os autores utilizaram também um modelo (template) previamente definido para realizar a leitura das questões. Primeiramente são detectados

pontos de referência, que são encontrados a partir de linhas horizontais empilhadas ao lado esquerdo da folha, indicando a linha para cada referência de leitura. Em seguida, transforma-se a imagem para escala de cinza, aplica-se um threshold de 128 (não é relatado qual técnica de threshold é utilizada), detecta-se a primeira e última barra horizontal, e corrige-se o alinhamento da folha, baseando-se no centro da barra horizontal de referência. Depois segmenta-se a folha de resposta de acordo com o estipulado pelo template, e verifica cada resposta utilizando threshold e projeção, não é relatado especificamente como é realizado a projeção, mas tende-se a pensar que é uma distância padrão das barras de referência. Não é dito sobre variação de qualidade dos cartões processados, mas é utilizado um scanner para aquisição de imagem colorida. Foram processadas 250 imagens, mas analisadas uma amostragem de 100 imagens da qual foi observada uma assertividade média de 100%.

O trabalho de (RACHCHH; GOPI, 2019) utiliza um template previamente configurado para leitura da folha. Na confecção do cartão de respostas utiliza-se uma barra vertical larga totalmente escura ao meio do cartão. No artigo é relatado que essa barra é necessária para aumentar a precisão do algoritmo de correção da matriz de homografia da folha. O algoritmo utilizado é o PCA, no trabalho se verifica a precisão de uma correlação por entropia e outro por PCA que se apresentou ideal. Primeiramente ocorre a aquisição da imagem por uma câmera de 1MP a 13MP, envia-se ao computador para processamento, transforma em escala de cinza, aplica-se o threshold com método de OTSU, executa a correção com o PCA, detecta e verifica cada linha e coluna. O experimento é realizado com 100 imagens, tendo 100% de acurácia. A quantidade de questões, demonstrado do template é de apenas 40 questões.

O *survey* apresentado em (PATEL; PRAJAPATI, 2015) levanta um conjunto de técnicas de processamento de cartões de resposta. O trabalho apresenta vantagens e desvantagens da utilização de uma máquina especializada na leitura de cartões de resposta, a principal vantagem é pela precisão da leitura, as principais desvantagens são que para acompanhar a precisão é necessário que a marcação precisa estar bem marcada e não haver sujeiras, e que é relativamente lenta lendo 10 mil folhas por hora. Da técnica proposta utiliza de threshold adaptativo para considerar variação de luz pela imagem, extração de bordas para determinar a região de interesse, mapeamento por grade da região de interesse para se determinar a localização das questões, e análise da marcação pelo valor médio de nível de cinza da região específica marcada. Não analisa diferentes tipos de cartões, nem demonstra experimentação com quantidade e acurácia. Utilizou template de 30 questões para exemplo de uso do algoritmo proposto.

3.2 Desenvolvimento do Sistema

A seguir são descritas as ferramentas utilizadas para o desenvolvimento de testes experimentais e para a validação do sistema desenvolvido.

3.2.1 Base de Dados

A base de dados utilizada neste trabalho é composta por 936 folhas de resposta preenchidas, escaneadas por máquinas diferentes, cada folha de resposta possui 90 questões que serão alvo da classificação de marcação, cada questão possuindo 5 opções de marcação (A, B, C, D e E). Na base de dados há folhas de respostas que começam na questão 1 e também outras que começam na 91, mas todas possuem 90 questões.

A Figura 14 mostra um modelo de folha de resposta utilizado. Este modelo conta com um retângulo de borda preta, cujo propósito é utilizá-lo como referência para a segmentação da área das questões. Após a segmentação é utilizada a técnica de mapeamento por grade, onde cada célula representa um possível região para leitura da marcação. A marcação é realizada verificando o nível médio de cinza, tal como em (PATEL; PRAJAPATI, 2015).

3.2.2 Validação do sistema

A validação consiste em verificar o desempenho do sistema na detecção correta das marcações nas folhas de resposta. Esse desempenho será medido pelas métricas de acurácia para leitura individual de questões, para acurácia de leitura perfeita individual de folhas de resposta, e pelo número de folhas que não foi possível de encontrar o retângulo de referência, dados como tempo total não serão abordados pelo foco estar na acurácia e não em velocidade do algoritmo. O retângulo de referência é a principal estrutura utilizada no trabalho para definir a região do objeto de interesse (ROI), e corrigir perspectiva. É o retângulo de borda preta que circunda a área das questões. O retângulo e a ROI podem ser identificados na Figura 13.

3.2.3 Ferramentas de Apoio

Para o desenvolvimento do sistema foram utilizadas as seguintes ferramentas de apoio:

- OpenCV é uma ferramenta open source desenvolvida pela Intel que é direcionada para realizar operações em imagens, e conta com uma vasta coleção de bibliotecas para diversas áreas de processamento de imagens digitais, tendo suporte a todas as técnicas propostas neste trabalho.

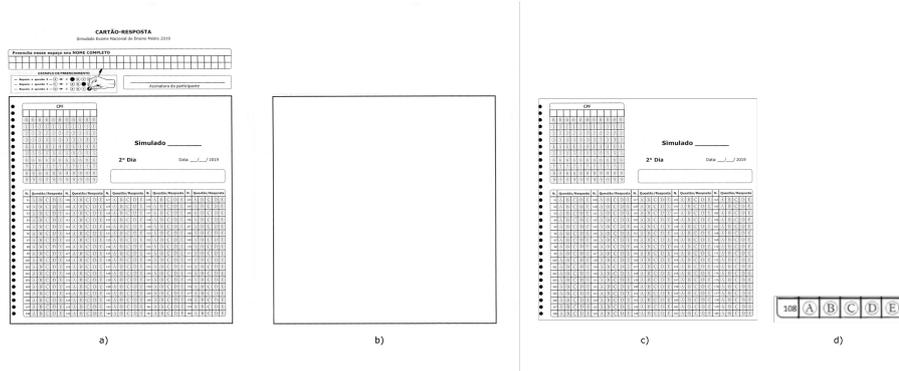


Figura 13 – a) Folha de resposta não preenchida. b) Retângulo de referência, utilizado para identificar a ROI. c) Região de interesse (ROI), onde contém as informações para detecção das marcações. d) Região da questão, possuindo 5 células da grade que serão classificadas em preenchida ou não preenchida.

- Python é uma linguagem ascendente no mercado, fácil de usar, que possui também uma vasta coleção de bibliotecas. Entre elas o OpenCV para python. Foi selecionada por ser uma opção que oferece poucas barreiras para desenvolvimento, sendo ótima para realizar estudos, e tendo boa produtividade.
- Numpy é uma biblioteca de python direcionada a realizar operações matriciais, de forma otimizada. É ideal para operação em imagens mantendo uma complexidade baixa de código.
- Jupyter é uma plataforma que oferece um interface amigável e um ótimo gerenciamento de código python que é executado.
- Docker é um gerenciador de ambiente, utilizado para manter a versão das tecnologias utilizadas definidas corretamente, sem interferência de versões externas ao ambiente.
- PiP é um gerenciador de pacotes, utilizado para instalar os pacotes python necessários para execução do trabalho.

CARTÃO-RESPOSTA

Simulado Exame Nacional do Ensino Médio 2019

Preencha nesse espaço seu NOME COMPLETO

EXEMPLO DE PREENCHIMENTO

— Resposta a questão X — A → X: A B C D

— Resposta a questão Y — C → Y: A B C D

— Resposta a questão Z — D → Z: A B C D

Assinatura do participante

N.	Questão/Resposta	N.	Questão/Resposta	N.	Questão/Resposta	N.	Questão/Resposta	N.	Questão/Resposta
1	A B <input checked="" type="radio"/> D E	19	<input checked="" type="radio"/> B C D E	37	A B C D E	55	A B C D E	73	A B C D E
2	A B <input checked="" type="radio"/> D E	20	<input checked="" type="radio"/> B C D E	38	A B C D E	56	A B C D E	74	A B C D E
3	<input checked="" type="radio"/> B C D E	21	A B <input checked="" type="radio"/> D E	39	A B C D E	57	A B C D E	75	A B C D E
4	A B C D <input checked="" type="radio"/> E	22	A B C <input checked="" type="radio"/> E	40	A B C D E	58	A B C D E	76	A B C D E
5	A B <input checked="" type="radio"/> D E	23	A B C <input checked="" type="radio"/> E	41	A B C D E	59	A B C D E	77	A B C D E
6	<input checked="" type="radio"/> B C D E	24	A B <input checked="" type="radio"/> D E	42	A B C D E	60	A B C D E	78	A B C D E
7	A <input checked="" type="radio"/> C D E	25	A B C D <input checked="" type="radio"/> E	43	A B C D E	61	A B C D E	79	A B C D E
8	A B <input checked="" type="radio"/> D E	26	A B <input checked="" type="radio"/> D E	44	A B C D E	62	A B C D E	80	A B C D E
9	A B <input checked="" type="radio"/> D E	27	A B C D E	45	A B C D E	63	A B C D E	81	A B C D E
10	A B C <input checked="" type="radio"/> E	28	A B C D E	46	A B C D E	64	A B C D E	82	A B C D E
11	<input checked="" type="radio"/> B C D E	29	A B C D E	47	A B C D E	65	A B C D E	83	A B C D E
12	A B <input checked="" type="radio"/> D E	30	A B C D E	48	A B C D E	66	A B C D E	84	A B C D E
13	A <input checked="" type="radio"/> C D E	31	A B C D E	49	A B C D E	67	A B C D E	85	A B C D E
14	A B C <input checked="" type="radio"/> E	32	A B C D E	50	A B C D E	68	A B C D E	86	A B C D E
15	<input checked="" type="radio"/> B C D E	33	A B C D E	51	A B C D E	69	A B C D E	87	A B C D E
16	<input checked="" type="radio"/> B C D E	34	A B C D E	52	A B C D E	70	A B C D E	88	A B C D E
17	A B <input checked="" type="radio"/> D E	35	A B C D E	53	A B C D E	71	A B C D E	89	A B C D E
18	A B C D <input checked="" type="radio"/> E	36	A B C D E	54	A B C D E	72	A B C D E	90	A B C D E

Figura 14 – Exemplo de um cartão de respostas parcialmente preenchido.

4 Técnica Proposta

Este capítulo apresenta o funcionamento do sistema desenvolvido. Serão apresentados o fluxo de processamento adotado pelo sistema, bem como os resultados de testes realizados para avaliar a eficiência da técnica proposta.

4.1 Etapas do processamento

A Figura 15 mostra o fluxo geral de processamento do método proposto para a identificação das marcações nas folhas de resposta. Porém, para demonstrar melhor o funcionamento do método proposto, o fluxo geral foi dividido em três etapas que serão detalhadas a seguir.

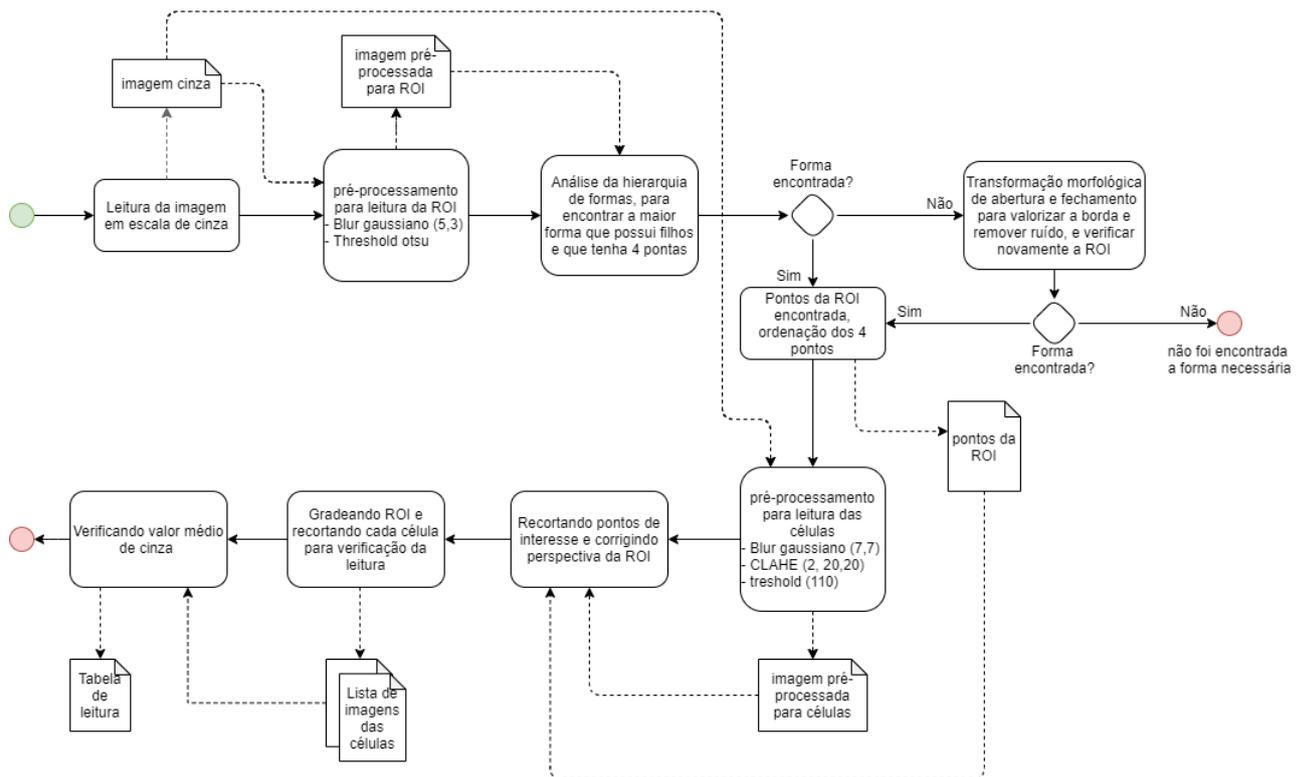


Figura 15 – Fluxo bpmn de processamento da folha de resposta

4.1.1 Etapa 1: Pré-processamento

A etapa de pré-processamento compreende as atividades mostradas na Figura 16. Nessa primeira etapa duas imagens diferentes são geradas a partir da imagem original: uma imagem para auxiliar na identificação do retângulo de referência onde estão as questões e

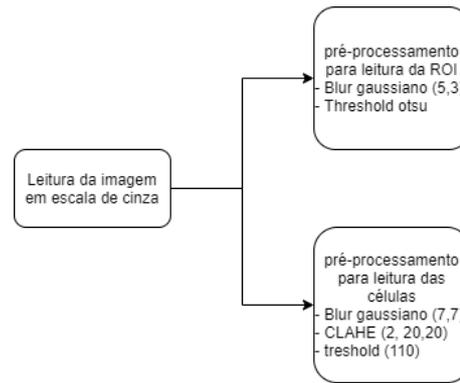


Figura 16 – Etapa de pré-processamento

outra imagem para auxiliar na leitura individual das marcações. A Figura 17 mostra um exemplo do resultado gerado nesta etapa.

O pré-processamento valoriza o retângulo de referência e alarga as marcações. Além disso, o pré-processamento ajuda a suavizar e eliminar o máximo de ruídos, valorizando apenas as concentrações de preto, o que é bastante útil para encontrar o retângulo de referência pois fecha possíveis fragmentos no contorno do retângulo.

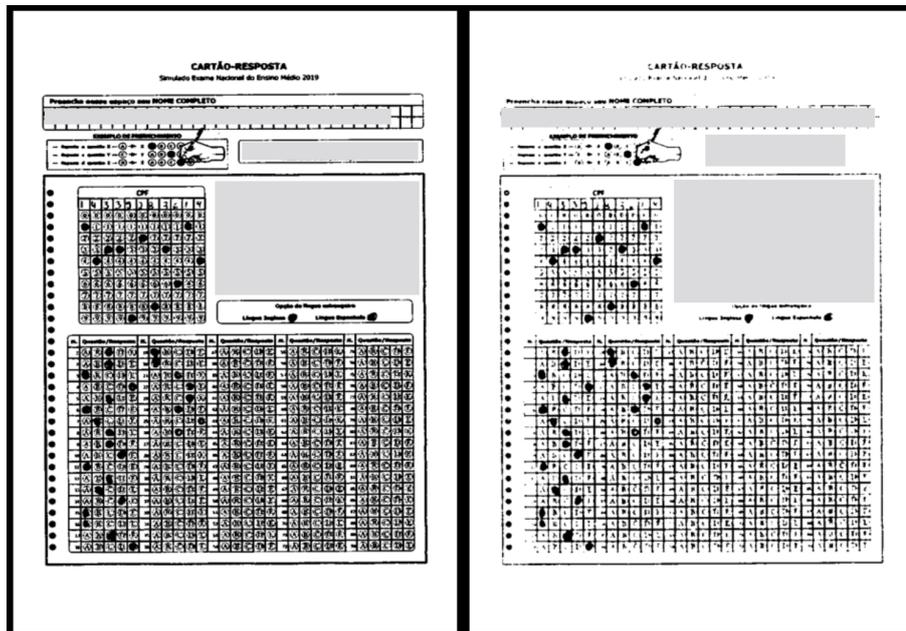


Figura 17 – À esquerda o exemplo do resultado do pré-processamento realizado para verificação do retângulo de referência. À direita o exemplo do resultado do pré-processamento realizado para leitura individual de cada célula da grade.

4.1.2 Etapa 2: Identificação e recorte

A Figura 18 mostra as atividades da Etapa 2 do processamento, que tem o objetivo de identificar e segmentar as áreas para verificação da marcação.

Há dois momentos de recorte, o primeiro ocorre após a análise da estrutura das formas na imagem pelo método de border following, ao identificar os quatro pontos que definem o retângulo de referência. O segundo recorte é realizado para individualizar cada célula a ser lida, após a definição da grade de leitura.

Ao final do processo têm-se pequenas imagens preparadas para a leitura. Pode ocorrer da imagem não possuir o retângulo de referência ou o retângulo não foi identificado corretamente. Para minimizar o impacto de ruídos que dificultam identificar a forma, é realizada uma nova tentativa aplicando o método de abertura e fechamento, a fim de fortalecer o retângulo de referência e aumentar a efetividade do método de border following para identificar a forma que corresponde ao retângulo de referência.

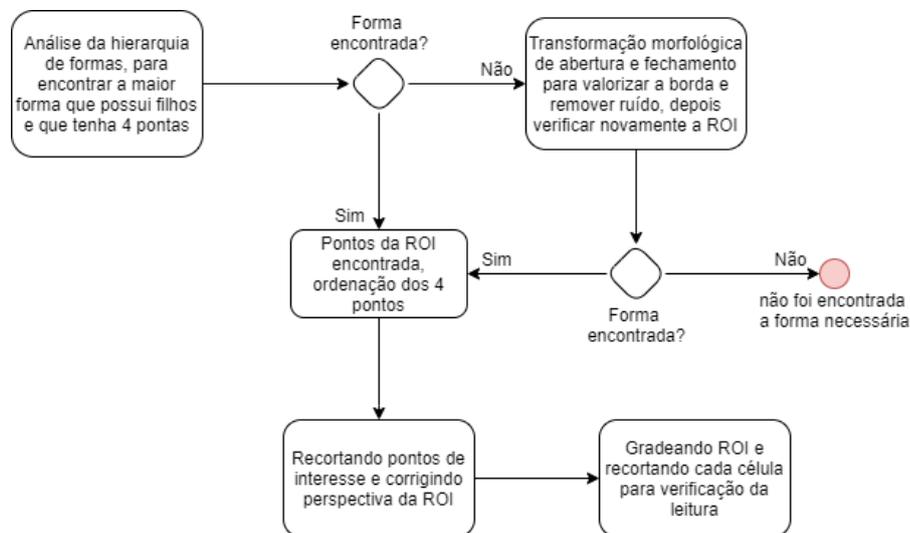


Figura 18 – Etapa de identificação e recorte

4.1.3 Etapa 3: Classificação das marcações

A terceira etapa consiste em verificar cada célula recortada na etapa anterior e realizar a análise do nível médio de cinza de cada célula: a partir de determinado nível a célula é classificada como marcada. Após todas as células serem classificadas, é realizado um processo lógico para analisar a posição das marcações e, então, assim obter qual a resposta esperada de cada marcação em cada questão. Na Figura 23 nota-se a grade gerada e suas células internas, que caso verificada, para fins demonstrativos foi marcada de azul.

No método atual foi utilizado um arquivo de referência que especifica a correspondência entre posição da célula na grade e a questão a ser analisada. A partir desse arquivo é possível traduzir a grade para uma lista de respostas marcadas.

Para início desse trabalho foi realizado previamente uma verificação manual de todas as marcações, em todas as folhas de resposta, gerando um arquivo contendo todas

tanto referências de folhas de respostas boas para leitura, que possuíssem pouco ruído, quanto referências de qualidade baixa. A Figura 14 é um exemplo das imagens que estão na base de calibração, ela apesar de possuir poucas questões respondidas, possui bastante ruído.

Durante a experimentação na base de calibração, foram identificados os melhores parâmetros dos algoritmos de pré-processamento para que os ruídos fossem eliminados, mas não prejudicassem a leitura em outras folhas. Foi definido o parâmetro de nível de cinza para ser utilizado como afinação da classificação. Foi executada uma afinação manual desse parâmetro começando a partir valor de 20 indo até 60, variando em 2. O valor otimizado obtido foi de 48%, ou seja, caso a célula esteja acima de 48% preenchida de preto, ela será classificada como marcada.

4.2.1 Resultados obtidos

Na Tabela 1 são apresentadas as métricas finais que foram obtidas pelo método proposto.

Descrição da métrica	Valor obtido
Acurácia de verificação de marcação	99.973%
Acurácia de leitura ideal de folha	98.174%
Número de falhas ao encontrar o retângulo de referência	5
Número total de questões lidas	83790
Número total de falhas de leitura individual da questão	23

Tabela 1 – Resultados finais da execução do algoritmo proposto

Analisando as folhas que não foram captadas pela etapa de extração do retângulo de referência, há um limite do tamanho da fragmentação para que a borda possa ser reconstituída pelas operações morfológicas realizadas, como exemplo a Figura 20 possuiu duas fragmentações, as quais impediram a identificação da forma fechada do retângulo de referência.

Analisando as marcações que falharam a leitura, houve pequenos casos em que o ruído foi aumentado causando assim uma leitura incorreta da célula, como pode ser visto na Figura 21.

Um pré-processamento utilizando valores de threshold acima de 110 a fim de eliminar mais a presença do template, se demonstrou ser menos eficaz por reduzir a chance da leitura de marcações mais suaves ou menores. Percebe-se que em alguns casos aparentemente houve dupla marcação ou o início de uma marcação, que levanta a questão se a marcação deveria ser considerada ou não.

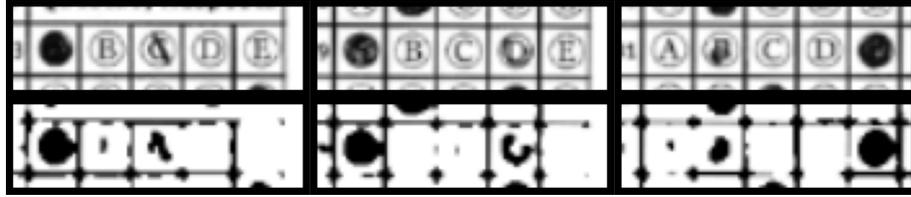


Figura 21 – Exemplo de leitura individual, acima a imagem original, abaixo a imagem pré-processada para leitura.

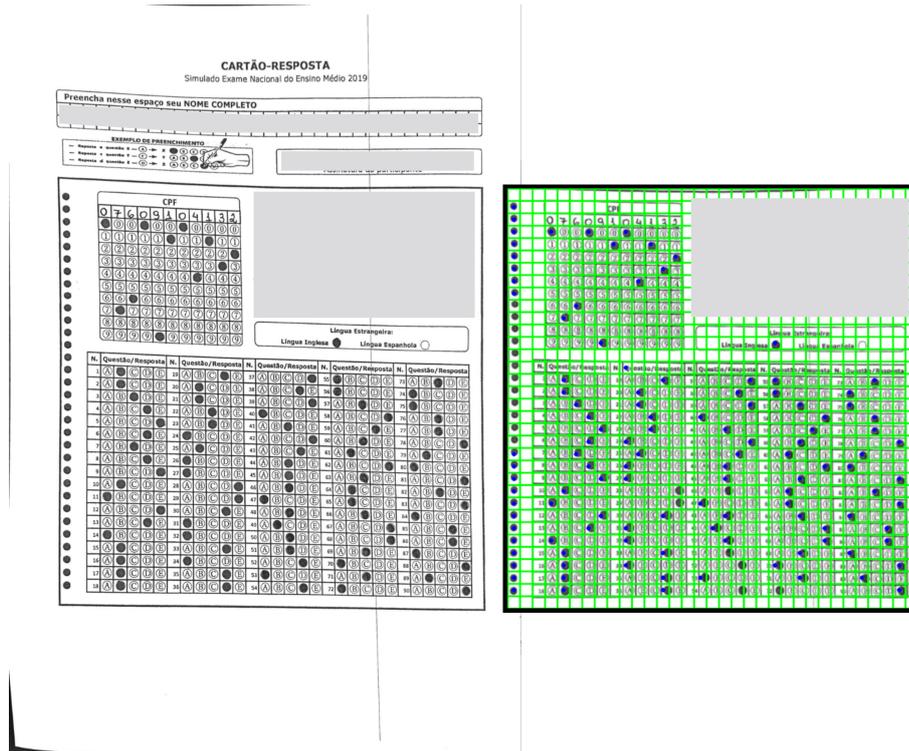


Figura 22 – À esquerda a imagem com deformação original, à direita a ROI recortada, corrigida a perspectiva, e realizada a leitura individual de cada célula da grade.

Na Figura 23, especificamente na questão 162 localizada ao final do quarto quadrante de questões, pode-se perceber uma leve marcação realizada, porém não foi captada pelo algoritmo, nesse caso pode-se considerar como algo positivo e que a marcação leve na opção “E” é um ruído. Na Figura 24 há várias bolinhas marcadas com o meio vazado, que pode ser um desafio para outros algoritmos, o proposto realizou a leitura da forma correta, pois a forma da bolinha não é um fator considerado, e sim o preenchimento verificado da célula como um todo. Na Figura 25 é um ótimo exemplo, que durante a verificação de uma mesma folha, a presença das marcações varia bastante entre as primeiras com as últimas, dependendo da forma de pré-processamento as marcações mais suaves podem ser eliminadas. O algoritmo proposto também se apresentou satisfatório nesse quesito.

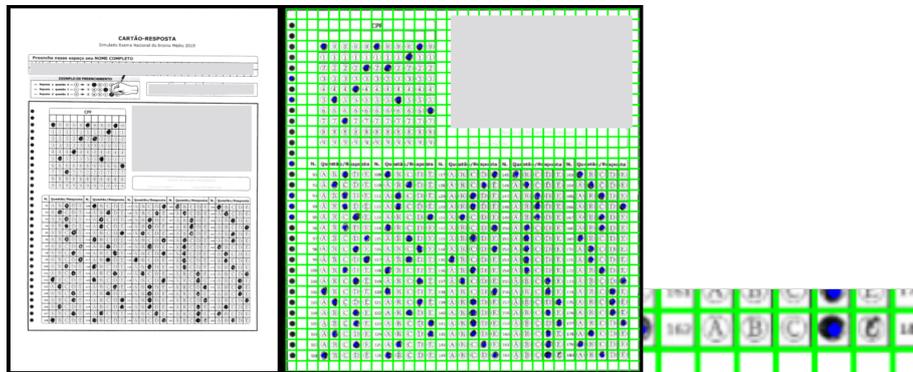


Figura 23 – À esquerda a imagem original recebida para leitura e à direita a leitura realizada com a questão 162 ampliada, percebe-se que possui uma leve marcação na questão 162 que não foi considerada preenchida pela classificação.

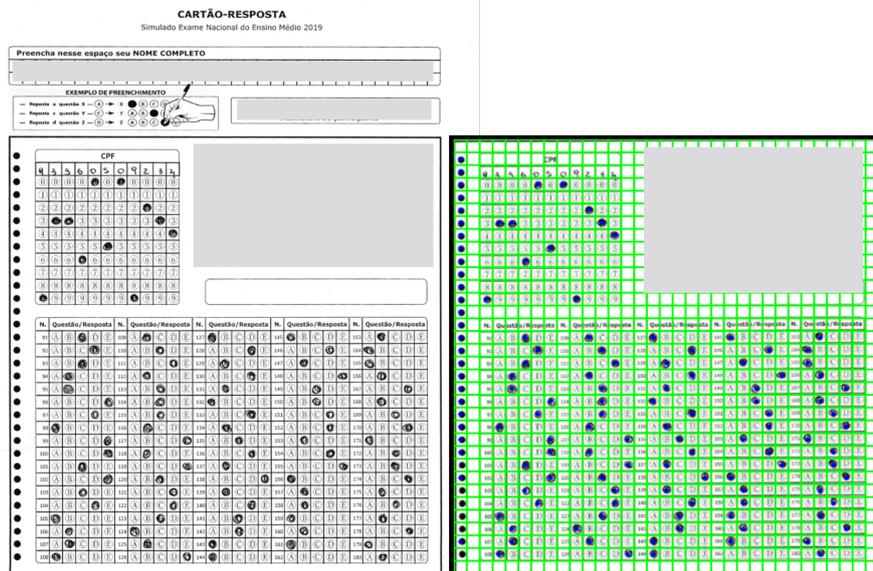


Figura 24 – À esquerda a imagem original recebida para leitura com bolinhas fracas e também vazadas no centro, e à direita a leitura realizada.

5 Conclusão

Com os resultados obtidos, foi possível validar a possibilidade de utilização da visão computacional para a automatização de tarefas manuais, ou como solução alternativa à máquinas onerosas especializadas em leitura de cartões de resposta. Neste trabalho foi proposto e desenvolvido um sistema simples baseado em um fluxo de atividades, usando técnicas de processamento de imagens, a fim de detectar marcações em folhas de respostas de atividades avaliativas objetivas.

O método proposto conseguiu uma excelente acurácia na detecção de marcações (98%), porém foram encontradas algumas dificuldades e fraquezas do método proposto, que podem ser alvos em trabalhos futuros, tais como:

- A deformação do retângulo de referência é a principal fraqueza, pois impacta outras etapas e pode gerar falsas leituras. Ao contrário da fragmentação onde é acusado o erro durante o processamento da folha e assim pode ser tratado, a deformação pode passar despercebida pelo operador do sistema que utiliza o algoritmo proposto, demonstrando a fragilidade do sistema.
- A fragmentação da borda também é uma fraqueza, a proposta de utilizar operações de fechamento e abertura morfológica são interessantes apenas até um certo tamanho da fragmentação. Análises estruturais da imagem, tentando conectar os pontos e recriar o retângulo de referência, ou algoritmos alternativos ao *border following* que não mais simplesmente acompanham a borda, mas analisam as quinas do retângulo de referência, podem ser objeto de estudo de trabalhos futuros.

A utilização do CLAHE juntamente com o blur gaussiano, como parte do pré-processamento para leitura, se apresentou muito eficaz para melhorar a diferença entre a marcação realizada pela pessoa e o template das questões, possibilitando remove-lo na maioria dos casos quase que completamente, utilizando um threshold global simples de 110. A equalização sendo parte do pré-processamento reduziu a diferença entre as imagens proporcionando maior eficácia para uso de um único valor para o threshold.

A separação de pré-processamento para identificação do retângulo de referência e outra para leitura das questões também demonstrou ser uma abordagem eficaz, otimizando os resultados para cada objetivo específico.

Referências

- ACHARYA, T.; RAY, A. K. *Image Processing Principles and Applications*. John Wiley Sons, 2005. Citado na página 22.
- BELAG, I. A. An image processing based optical mark recognition with the help of scanner. Kastamonu University, Science Institute, Kastamonu - TURKEY, 2018. Citado na página 25.
- EASTON, J. R. L. *Fundamentals of Digital Image Processing*. [S.l.], 2010. Citado na página 22.
- FILHO, O. M.; NETO, H. V. *Processamento de imagens digitais*. Rio de Janeiro, 1999. Citado na página 16.
- GONZALEZ, R. C. *Digital Image Processing*. University of Tennessee, 2008. Citado 2 vezes nas páginas 15 e 16.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. University of Tennessee, 2018. Citado 2 vezes nas páginas 19 e 20.
- GUPTA, A. Image based low cost method to the omr process for surveys and research. Krishna Engineering College, Ghaziabad, India, 2016. Citado na página 25.
- KRISHNAN, N. G. S. Algorithm on tracing the boundary of medical images using abstract cellular complex. Department of Applied Mathematics and Computational Sciences PSG College of Technology Coimbatore, India, 2012. Citado na página 20.
- OGIELA, R. T. M. R. Syntactic reasoning and pattern recognition for analysis of coronary artery images. Institute of Automatics 30 Mickiewicza Avenue, University of Mining and Metallurgy, PL-30-059, Krakow, Poland, 2002. Citado na página 20.
- PATEL, N. V.; PRAJAPATI, G. I. Various techniques for assessment of omr sheets through ordinary 2d scanner: A survey. SVMIT Bharuch, India, 2015. Citado 2 vezes nas páginas 26 e 27.
- QUEIROZ, H. M. G. José Eustáquio Rangel de. Introdução ao processamento digital de imagens. UFCG, Departamento de Sistemas e Computação, 2001. Citado na página 21.
- RACHCHH, K.; GOPI, E. S. Inclusion of vertical bar in the omr sheet for image-based robust and fast omr evaluation technique using mobile phone camera. National Institute of Technology Trichy, Tamil Nadu, India, 2019. Citado na página 26.
- RUBINI, N. P. C. Contrast enhancement of mri images using ahe and clahe techniques. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019. Citado 2 vezes nas páginas 22 e 23.
- SUZUKI, S. Topological structural analysis of digitized binary images by border following. Shizuoka University, Hamamatsu 432, Japan, 1985. Citado na página 20.