

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PAULO VICTOR ZUFFO OYAMA

**APLICAÇÃO DE PROCESSAMENTO DE IMAGENS PARA MEDIÇÃO DO NÍVEL
DO LEITE NA COLETA A GRANEL**

MEDIANEIRA

2022

PAULO VICTOR ZUFFO OYAMA

**APLICAÇÃO DE PROCESSAMENTO DE IMAGENS PARA MEDIÇÃO DO NÍVEL
DO LEITE NA COLETA A GRANEL**

Image processing application to measure the milk level in bulk collection

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Diogo Marujo.

Coorientador(a): Pedro Luiz de Paula Filho.

MEDIANEIRA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

PAULO VICTOR ZUFFO OYAMA

**APLICAÇÃO DE PROCESSAMENTO DE IMAGENS PARA MEDIÇÃO DO NÍVEL
DO LEITE NA COLETA A GRANEL**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel em Engenharia Elétrica da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 15/junho/2022

Diogo Marujo
Doutorado
Universidade Tecnológica Federal do Paraná

Pedro Luiz de Paula Filho
Doutorado
Universidade Tecnológica Federal do Paraná

Alex Lemes Guedes
Mestrado
Universidade Tecnológica Federal do Paraná

Leandro Antonio Pasa
Doutorado
Universidade Tecnológica Federal do Paraná

MEDIANEIRA

2022

RESUMO

O presente trabalho apresenta a implementação de um protótipo que realize a leitura da régua na medição do nível do leite na coleta a granel através de processamento de imagens, oferecendo uma alternativa para a leitura manual, devido ao fato de a etapa da medição do nível do leite na coleta a granel ser subjetiva e estar sujeita a erros humanos, que podem causar prejuízos para o produtor e para os laticínios. Um estúdio de fotos portátil foi desenvolvido, com o qual foram tiradas algumas fotos. O algoritmo foi testado com essas fotos e teve êxito em determinar o nível do leite no tanque em todas as fotos coletadas, obtendo erro médio absoluto de 0,086 cm e erro médio absoluto percentual de 0,255%, demonstrando a viabilidade da técnica da proposta, obtendo erros baixos, tempo de processamento pequeno e atingindo os objetivos.

Palavras-chave: processamento de imagens; visão por computador; python; inteligência artificial.

ABSTRACT

The present work exhibits the implementation of a prototype that performs the reading of the ruler in the measurement of the level of milk in the bulk collection through image processing, offering an alternative to the manual reading, due to the fact that the step of measuring the level of milk in bulk collection is subjective and subject to human errors, which can cause damage to the producer and the dairy. A portable photo studio was developed, with which some pictures were taken. The algorithm was tested with these photos and was successful in determining the level of milk in the tank in all the photos collected, obtaining mean absolute error of 0.086 cm and mean absolute percentage error of 0.255%, demonstrating the feasibility of the proposed technique, obtaining low errors, small processing time and reaching the objectives.

Keywords: image processing; computer vision; python; artificial intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 - Medição do nível do leite.....	14
Figura 2 - Representação dos pixels de uma imagem	17
Figura 3 - Espaço RGB	18
Figura 4 - Espaços de cores RGB e CMYK.....	19
Figura 5 - Representação do espaço de cores HSV	19
Figura 6 - Imagem em escala de cinza e seu histograma	20
Figura 7 - Imagem Binarizada	21
Figura 8 - Limiarização adaptativa pelo método de Niblack	22
Figura 9 - Limiarização inRange.....	22
Figura 10 - Transformações locais	23
Figura 11 - Filtro de média	24
Figura 12 - Filtro de média gaussiana	24
Figura 13 - Filtro de mediana	25
Figura 14 - Detector de bordas Canny	26
Figura 15 - Transformada de Hough (cartesianas).....	27
Figura 16 - Transformada de Hough (polares)	27
Figura 17 - Elementos Estruturantes	28
Figura 18 - Exemplo de erosão 1	29
Figura 19 - Exemplo de erosão 2	30
Figura 20 - Dilatação	31
Figura 21 - Imagem e Espectro de magnitude	32
Figura 22 - Detecção da escala da régua	33
Figura 23 - Imagem e espectro de magnitude.....	33
Figura 24 - Filtragem passa-baixa.....	34
Figura 25 - Filtragem passa alta.....	34
Figura 26 - Arquitetura do Tesseract OCR.....	35
Figura 27 - Aplicação do Tesseract.....	36
Figura 28 - Estúdio de fotos	37
Figura 29 - Foto tirada com estúdio de fotos	38
Figura 30 - Funcionamento do algoritmo.....	39
Figura 31 - Identificação das bordas da régua	40
Figura 32 - Identificação da linha do leite	41
Figura 33 - Reconhecimento de caracteres	42
Figura 34 - Cálculo do intervalo de escala	43
Figura 35 - Segmentação das linhas da escala.....	44
Figura 36 - Imagens obtidas durante a coleta	47
Figura 37 - Região de interesse durante execução.....	48
Figura 38 - Segmentação das linhas da graduação da régua	48

Figura 39 - Identificação de caracteres com TesseractOCR.....	49
Figura 40 - Resultado final obtido.....	50

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Trabalhos correlatos	15
1.2	Justificativa.....	16
1.3	Objetivos	16
1.3.1	Objetivos específicos.....	17
1.4	Fundamentação teórica	17
1.4.1	Espaços de cores.....	17
<u>1.4.1.1</u>	<u>RGB e RGBα.....</u>	<u>18</u>
<u>1.4.1.2</u>	<u>CMYK.....</u>	<u>18</u>
<u>1.4.1.3</u>	<u>HSV e HSL.....</u>	<u>19</u>
1.4.2	Limiarização de imagens.....	20
<u>1.4.2.1</u>	<u>Histograma de uma imagem</u>	<u>20</u>
<u>1.4.2.2</u>	<u>Limiarização</u>	<u>20</u>
<u>1.4.2.3</u>	<u>Limiarização adaptativa.....</u>	<u>21</u>
<u>1.4.2.4</u>	<u>Inrange</u>	<u>22</u>
1.4.3	Transformações locais	23
<u>1.4.3.1</u>	<u>Filtro de média.....</u>	<u>24</u>
<u>1.4.3.2</u>	<u>Filtro de mediana.....</u>	<u>25</u>
<u>1.4.3.3</u>	<u>Detector de bordas canny</u>	<u>25</u>
<i>1.4.3.3.1</i>	<i>Transformada de hough</i>	<i>26</i>
<i>1.4.3.3.2</i>	<i>Find contours.....</i>	<i>28</i>
1.4.4	Morfologia Matemática	28
<u>1.4.4.1</u>	<u>Erosão</u>	<u>29</u>
<u>1.4.4.2</u>	<u>Dilatação</u>	<u>30</u>
1.4.5	Transformada de Fourier.....	31
<u>1.4.5.1</u>	<u>Detecção da escala da régua.....</u>	<u>32</u>
<u>1.4.5.2</u>	<u>Filtragem passa-baixa</u>	<u>33</u>
<u>1.4.5.3</u>	<u>Filtragem passa-alta</u>	<u>34</u>
1.4.6	Tesseract OCR.....	35
2	MATERIAL E MÉTODOS	37
2.1	Estúdio de fotos	37
2.2	Algoritmo	38

2.2.1	Identificação das bordas da régua.....	39
2.2.2	Identificação da linha do leite	40
2.2.3	Reconhecimento de caracteres	41
2.2.4	Cálculo do intervalo de escala.....	42
2.2.5	Segmentação das linhas da escala da régua	44
2.3	Validação dos resultados	44
2.3.1	Raiz quadrada do erro médio quadrático	45
2.3.2	Erro médio absoluto	45
2.3.3	Erro médio absoluto percentual.....	45
3	RESULTADOS E DISCUSSÃO	47
3.1	Segmentação da região de interesse	47
3.2	Resultado final.....	49
4	CONCLUSÃO	52
4.1	Trabalhos futuros	53
	REFERÊNCIAS.....	54

1 INTRODUÇÃO

Segundo MAPA (2011) (Ministério da Agricultura, Pecuária e Abastecimento), o processo de coleta de leite cru refrigerado a granel consiste em recolher o produto em caminhões com tanques isotérmicos construídos internamente de aço inoxidável, diretamente da propriedade do produtor. Trata-se de uma revolução nos moldes de produção e de comercialização da matéria-prima em questão, que ocorreu na década de 90, e que permitiu um controle maior de sanidade do rebanho e análises laboratoriais de qualidade, também na plataforma industrial (DORNELAS, 2000).

O aumento da competitividade e o aquecimento do mercado do setor de lácteos, naquela época, submeteu toda a cadeia de produção a critérios de qualidade, redução de custos mais rigorosos e melhoramento de produtividade (DORNELAS, 2000).

As empresas, atentas a estes critérios, promoveram mudanças na operação da coleta de leite, ampliando seus investimentos, para melhorar a logística da produção, trazendo à tona canais redutores de custos nas transações, melhorando ganhos para todas as partes.

Segundo Timm e Oliveira (2011), no ano de 1998, o Ministério da Agricultura, Pecuária e Abastecimento (MAPA) criou um grupo de trabalho para propor medidas visando o aumento da competitividade e a modernização do setor produtivo de leite e derivados no Brasil, que culminou na Instrução Normativa nº 51 de 2002 (IN 51/2002), posteriormente modificada pela IN 62/2011, que entre outros, regulamenta a coleta a granel de leite.

Segundo o Ministério da Agricultura, Pecuária e Abastecimento (2011), o funcionário encarregado da coleta deve receber treinamento básico sobre higiene, análises preliminares do produto e coleta de amostras, podendo ser o próprio motorista do carro-tanque.

Ao funcionário, então, cabe agitar o leite com utensílio próprio, realizar a prova do alizarol 72% e coletar amostras para os testes de laboratório (MAPA, 2011). Além do estabelecido na norma, o funcionário da empresa de coleta também deve determinar o volume de leite contido no tanque de depósito do produtor.

O funcionário que realiza a coleta faz essa medição inserindo uma régua por dentro do tanque, e a pendurando na borda, como mostra a Figura 1. A régua é

então observada, e conforme o nível do leite medido, o funcionário consulta uma tabela elaborada pelo fabricante, que fornece o volume contido no tanque de acordo com a medida.

Figura 1 - Medição do nível do leite



Fonte: Autoria própria (2021).

Este processo é, portanto, sujeito a erros de medição, imprecisão e inexatidão, que podem acarretar perdas tanto para o produtor quanto para a empresa que realiza o transporte do leite. Neste sentido, surge a perspectiva do uso de processamento de imagens para a medição do nível do leite no tanque, de modo a eliminar a subjetividade desta etapa, no processo da coleta do leite.

O processamento de imagens, por sua vez, já é utilizado nas mais variadas aplicações, “não existe praticamente mais nenhuma área de empreendimento técnico que não seja impactada de uma forma ou de outra pelo processamento de imagens” (GONZALEZ; WOODS, 2010).

Portanto, o objetivo deste trabalho é elaborar um algoritmo que faça a leitura da régua de medição, e determine o nível de leite no tanque. Essa leitura será feita em fotos tiradas em campo, com o auxílio de um estúdio de fotos portátil, de fácil manuseio.

1.1 Trabalhos correlatos

No contexto da proposta em questão, alguns trabalhos relacionados já foram elaborados. Bhalerao e Reynolds (2014) desenvolveram um algoritmo, utilizando principalmente a Transformada Discreta de Fourier (DFT), capaz de calcular a inclinação de uma régua, o espaçamento entre as graduações da régua (usando um estimador de frequência, como em Cheveigné e Kawahara (2002)), e a defasagem da escala, tendo como entrada do usuário apenas um ponto interior à régua. A finalidade do trabalho é o escalonamento automático de fotos de investigações forenses de cenas de crimes, em que uma régua é posicionada na foto, como referência para o objeto de interesse. O trabalho analisou diversas imagens, com variadas resoluções e SNR (*Signal to Noise Ratio* – proporção entre sinal e ruído), e obteve erros percentuais inferiores a 0,1%, bem como resolução de escala inferior a 1 pixel.

Herrmann, Zambanini e Kampel (2015) desenvolveram um algoritmo semelhante a Bhalerao e Reynolds (2014), na identificação da escala da imagem. A finalidade do algoritmo é determinar o tamanho real de moedas antigas dos catálogos de museus. Seu método utiliza a DFT, para encontrar a escala da imagem, e a binarização adaptativa para segmentar a moeda do fundo da imagem, para posteriormente determinar o diâmetro real da moeda. Os autores obtiveram um algoritmo robusto e preciso, com erro médio em torno de 1%.

Konovalov et. al (2017) desenvolveram um algoritmo que realiza um processo semelhante aos anteriores, com base na DFT, que detecta a escala da imagem, para realizar o escalonamento automático de fotos de peixes. O algoritmo desenvolvido neste trabalho se diferencia dos demais, pois não necessita de entradas do usuário, analisa quadrantes das imagens e classifica quais são candidatos a conterem um pedaço da régua. O algoritmo atingiu precisão de 95-98%.

Chen, Kojchev e Haschtmann (2015) aplicaram processamento de imagens a radiografias de uma mesma região do corpo de um paciente, que era grande demais para caber em uma só imagem. Seu objetivo foi alinhar diversas radiografias para obter uma só imagem que permitisse que os médicos determinassem se a recuperação foi adequada. Para detectar as linhas e os vértices das régua foi usada a Transformada de Hough, e o método de Harris. Para a determinação dos

números foi treinado um classificador do tipo Random Forest, com o Histograma de Gradientes Orientados extraídos da base de treinamento. Os resultado final foi bastante satisfatório, e pode ser usado em situações reais, por médicos.

Ueda et. al. (2005) desenvolveram um método de detecção do intervalo de escala em imagens com réguas, utilizando a DFT. Seu algoritmo admitia apenas réguas posicionadas horizontalmente, entre outras limitações, mas segundo os autores, estes impedimentos podem ser removidos com facilidade.

Barbosa (2017) desenvolveu um trabalho em que obteve 80% de acerto segmentando e identificando caracteres de placas de carros com a ferramenta Tesseract OCR (*Optimal Character Recognition* – Reconhecimento Ótimo de Caracteres) e outras funções úteis na segmentação.

1.2 Justificativa

A etapa da medição do nível do leite na coleta a granel, da maneira como é feita atualmente, está sujeita a erros humanos, imprecisão e inexatidão, que podem causar prejuízos para o produtor, caso o motorista lance um volume inferior ao correto, ou para os laticínios, caso seja um volume maior. Portanto, o processamento de imagens é uma ótima alternativa garantindo que a leitura da régua seja imparcial, haja vista a situação de uma transportadora de alimentos líquidos local, cujo prejuízo pode chegar a 70 mil reais mensais devido a estas imprecisões.

Além deste motivo, em conjunto com outros projetos de automação, a coleta do leite pode ser completamente automatizada, reduzindo a interferência humana no processo da coleta e, conseqüentemente, a contaminação externa ou adulteração nas medidas, garantindo melhora no produto final, para o consumidor e, principalmente, obedecendo a IN62/2011.

1.3 Objetivos

Implementar um algoritmo que realize a leitura da régua na medição do nível do leite na coleta a granel através de processamento de imagens.

1.3.1 Objetivos específicos

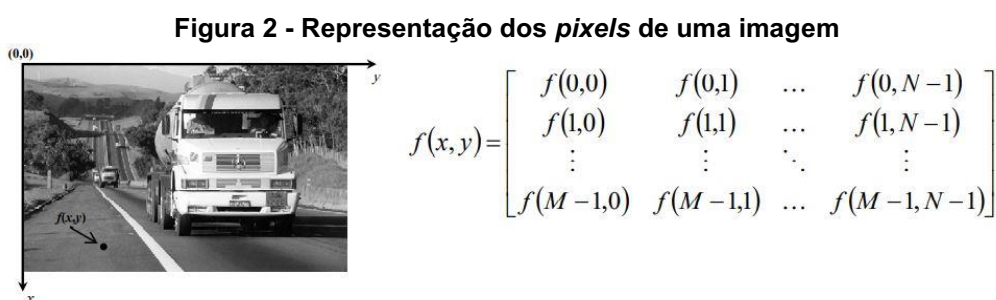
- Confecção de estúdio de fotos;
- Coleta de fotos em campo;
- Determinar o nível do leite na régua, computacionalmente;
- Avaliar erro de medição.

1.4 Fundamentação teórica

Neste tópico serão discutidos conceitos e técnicas básicas de processamento de imagens, que são essenciais para a compreensão deste trabalho.

1.4.1 Espaços de cores

Segundo Gonzalez e Woods (2010), uma imagem pode ser definida como uma função bidimensional, $f(x,y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou nível de cinza da imagem naquele ponto. Cada elemento (x,y) desta imagem é denominado *Picture Element*, ou *pixel*. A Figura 2 exemplifica a convenção de eixos utilizada nas imagens em tons de cinza e sua representação matricial:

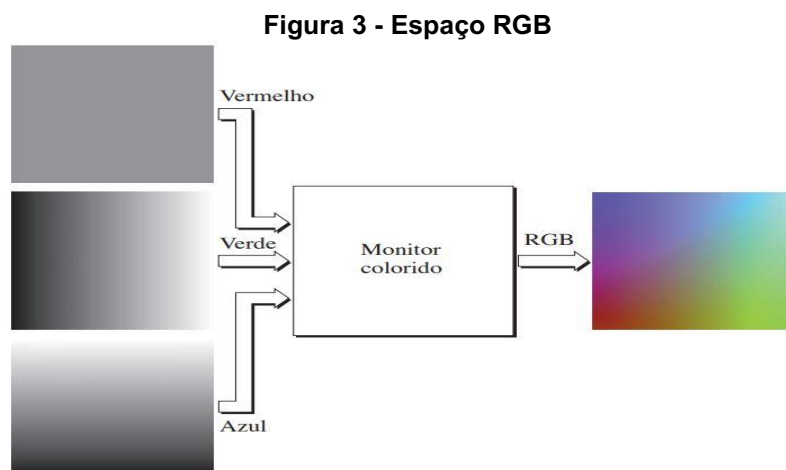


Fonte: Cunha(2013).

No entanto, para imagens coloridas, cada valor de $f(x,y)$ possui um valor diferente para cada plano de cor, formando um espaço (em alguns casos, hiperespaço) de cor. Dessa maneira, diferentemente do exemplificado na Figura 2, as imagens passam a ser compostas por não uma, mas três matrizes diferentes. O espaço de cor mais convencional é o RGB (*Red, Green, Blue*), em que há um plano

vermelho, um plano verde e um plano azul, que em conjunto, formam uma imagem colorida, como pode ser observado na Figura 3.

Existem inúmeros outros espaços de cores, entre eles: RGB, $RGB\alpha$, CMYK, HSL, HSV.



Fonte: Gonzalez e Woods (2010).

1.4.1.1 RGB e $RGB\alpha$

O canal RGB, assim como o canal BGR, é um espaço de cores aditivas, em que as cores se somam para que se forme a cor final, como ocorre quando se misturam tintas. O canal RGB é essencialmente análogo ao canal BGR, com a única diferença de a ordem das matrizes serem alteradas.

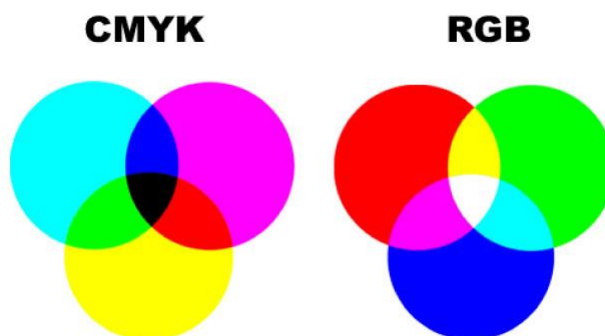
O hiperespaço $RGB\alpha$, no entanto possui um canal de cores a mais, que representa a transparência, e é um canal bastante utilizado em *softwares* de edição de fotos e afins.

1.4.1.2 CMYK

O espaço de cores CMYK, formado pelos canais *Cyan* (ciano), *Magenta* (magenta), *Yellow* (amarelo), *Key* (preto), e é diferente dos anteriores, pois é um hiperespaço de cores subtrativas, no qual as cores se formam pela absorção da luminosidade, e é utilizado em impressoras. A Figura 4 mostra a comparação dos espaços de cores RGB e CMYK.

A principal diferença pode ser notada no centro, em CMYK, a junção de todas as cores forma o preto, e em RGB, a junção de todas as cores forma o branco.

Figura 4 - Espaços de cores RGB e CMYK



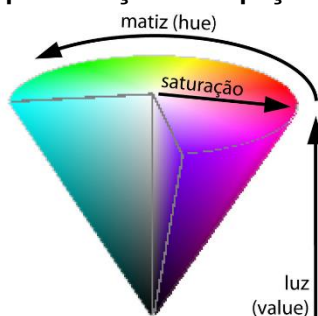
Fonte: Lang (2013).

1.4.1.3 HSV e HSL

Segundo Cunha (2013), o espaço HSV é representado pela combinação de matiz (Hue), saturação (S) e valor (V) ou intensidade de luz.

O espaço HSV é representado por coordenadas cilíndricas, como mostra a Figura 5:

Figura 5 - Representação do espaço de cores HSV



Fonte: Ribeiro & Gomes (2012).

O espaço de cores HSV é particularmente interessante por representar a cor efetivamente em apenas uma coordenada, facilitando a segmentação de elementos com cores discrepantes, como discutido em Silva et. al. (2020).

1.4.2 Limiarização de imagens

Segundo Silva et. al. (2020), filtros de limiar são comumente utilizados na segmentação de imagens, separando objetos ou regiões da imagem original para uma imagem binária em preto e branco.

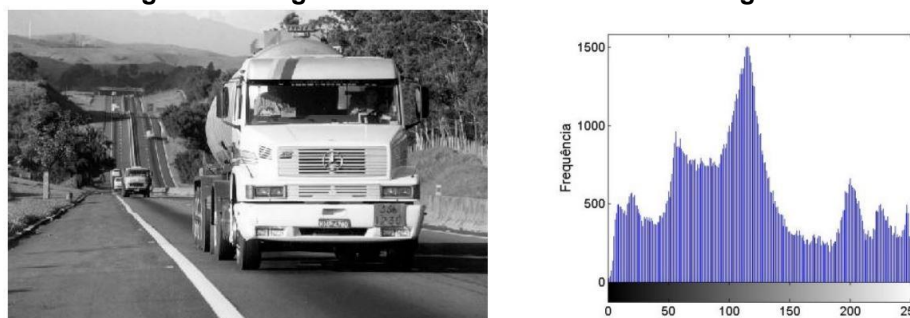
Uma das ferramentas que é bastante utilizada no processo de limiarização das imagens é o histograma.

1.4.2.1 Histograma de uma imagem

O histograma de uma imagem digital é uma função discreta $h(t) = n$, onde t é uma intensidade dentro do intervalo da imagem, e n o número de *pixels* com a intensidade t (GONZALEZ E WOODS, 2010).

Segundo Cunha (2013), o histograma proporciona um melhor entendimento da imagem, facilitando a visualização de contraste e luminosidade. A Figura 6 apresenta uma imagem em escala de cinza e seu respectivo histograma.

Figura 6 - Imagem em escala de cinza e seu histograma



Fonte: Cunha(2013).

1.4.2.2 Limiarização

A função de limiarização ou *thresholding* consiste em comparar, individualmente, os pixels da imagem com um valor arbitrário t , conforme Equação 1.

$$T(x,y) = \begin{cases} 0, & \text{se } f(x,y) < t \\ 1, & \text{se } f(x,y) \geq t \end{cases} \quad (1)$$

Esse valor de “*threshold*” pode ser escolhido por tentativa e erro (verificando qual é mais adequado para a ocasião), porém algumas técnicas, como a mostrada na próxima seção, utilizam alguns cálculos locais em torno do ponto a ser limiarizado.

O resultado dessa operação pode ser observado na Figura 7.

Figura 7 - Imagem Binarizada



Fonte: Cunha(2013).

1.4.2.3 Limiarização adaptativa

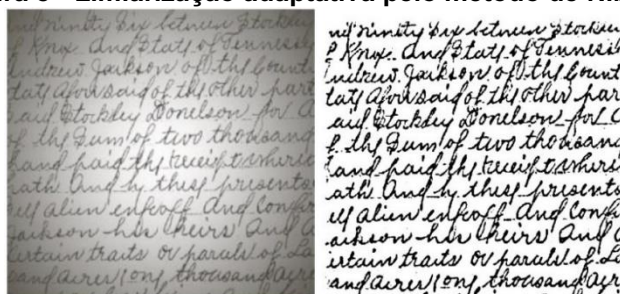
A limiarização adaptativa consiste em considerar a imagem original como uma composição de outras janelas menores de tamanho $M \times N$ pixels. Cada uma dessas imagens é binarizada individualmente, tornando o método local e adaptativo.

Segundo Falcão e Menotti (2014) existem vários métodos existentes. Os mais populares são propostos por Niblack (1986) e Sauvola e Pietikainen (2000), cujas modelagens calculam o limiar com base em medidas estatísticas de primeira ordem (média e desvio padrão), levando em consideração o contraste local da janela. Seus trabalhos originais podem ser consultados para detalhes adicionais.

O resultado da binarização adaptativa pelo método proposto por Niblack (1986) pode ser observado na Figura 8.

Na imagem, pode ser observada, à esquerda, a imagem original, possivelmente uma foto tirada com flash, devido ao fato de o centro da imagem ter mais brilho do que as bordas, e à direita a imagem após a binarização adaptativa.

Figura 8 - Limiarização adaptativa pelo método de Niblack

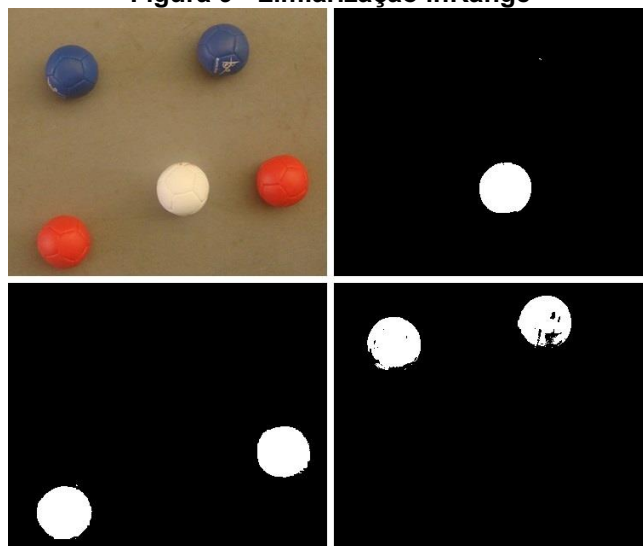


Fonte: Falcão & Menotti (2014).

1.4.2.4 Inrange

A limiarização *inRange* é expressivamente menos complexa, uma vez que simplesmente trata-se de, dentro de um determinado espaço de cor, escolher um limiar inferior t_i e um superior t_s para cada canal. Dessa forma, o pixel $T(x,y)$ será 1 caso $f(x,y)$ esteja compreendido entre os limiares inferiores e superiores de cada canal individualmente. A Figura 9 exemplifica a aplicação deste método em uma imagem:

Figura 9 - Limiarização *inRange*



Fonte: Autoria própria (2021).

No canto superior esquerdo, pode ser observado (a) a imagem original, e as demais imagens são exemplos da aplicação da binarização *inRange*, segmentando em cada imagem, as bolas vermelhas, azuis e a branca.

Essa binarização teve essencialmente 2 etapas, das quais a primeira foi verificar, para cada cor de bola, alguns valores de cor que aparecem na imagem. Na

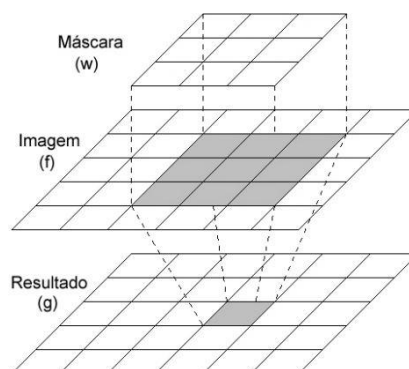
segunda etapa, a imagem é submetida à binarização *inRange*, cujo valor de entrada é uma faixa que engloba os valores encontrados anteriormente, para cada bola, respectivamente.

1.4.3 Transformações locais

Segundo Gonzalez e Woods (2010), a filtragem espacial envolve o conceito de vizinhança do pixel ao realizar qualquer operação, permitindo uma variedade de funções de processamento. Ao contrário das transformações pontuais discutidas até o momento, estas transformações consideram a vizinhança do pixel também.

Cunha (2013) afirma que para tanto, a formulação de filtros espaciais é baseada no uso de máscaras (também chamadas de *kernels*, filtros, janelas), que são matrizes bidimensionais cujos pesos determinam a natureza do processo. A Figura 10 exemplifica a filtragem de uma imagem f por uma máscara w , que resulta em uma imagem g :

Figura 10 - Transformações locais



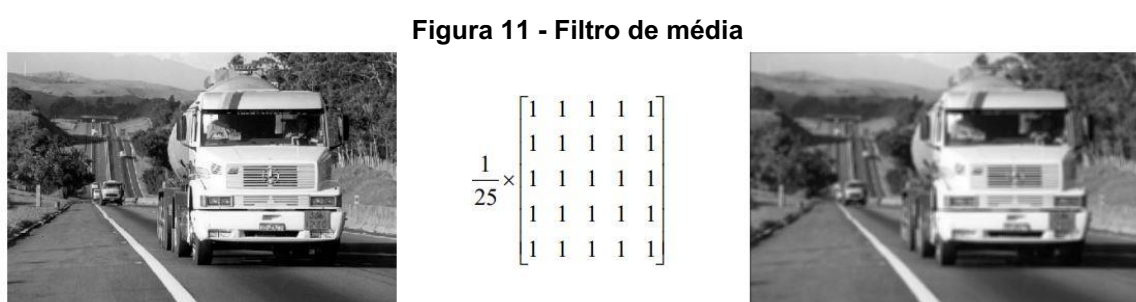
Fonte: Cunha (2013).

Essas transformações podem ser usadas principalmente para dois fins: suavização (borramento) e aguçamento (realce de detalhes).

Segundo Gonzalez e Woods (2010), a ideia por trás dos filtros de suavização é direta.

1.4.3.1 Filtro de média

A filtragem de média consiste em um filtro espacial linear (suavização), pois pode escrita como uma equação linear cujos pesos são os elementos do filtro e cujas variáveis são os respectivos pixels da imagem. Um exemplo de *kernel* de filtragem de média pode ser observado na Figura 11.

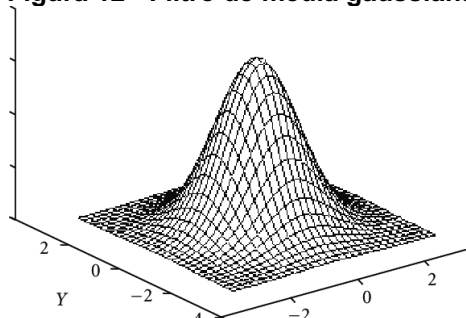


Fonte: Cunha (2013).

Observa-se na imagem, que o filtro aplicado tem dimensão 5x5, portanto possui 25 elementos, portanto trata-se de uma média simples.

Também existem filtros de média ponderada, como é o caso da média gaussiana. O filtro deste tipo de média observa os valores de uma curva gaussiana 3D, como pode ser observado na Figura 12, ou seja, dá pesos maiores para o elemento central e menos peso quanto maior a distância do centro. Pode ser observado o filtro de média gaussiana 5x5, também.

Figura 12 - Filtro de média gaussiana



Fonte: Dinesh & Radhakrishnan (2006).

Segundo Gonzalez e Woods (2010), é difícil ver as diferenças entre as imagens suavizadas utilizando filtros de média simples e gaussiana, visto que a área coberta por essas mascaras em qualquer posição de uma imagem é pequena.

1.4.3.2 Filtro de mediana

O filtro de mediana é um filtro de estatística de ordem e é um filtro não linear que se baseia na ordenação dos elementos da imagem cobertos pelo filtro. Ou seja, o filtro de mediana substitui o valor do pixel pela mediana de sua vizinhança, incluindo o próprio pixel.

Segundo Gonzalez e Woods (2010), os filtros de mediana são especialmente populares por que proporcionam excelentes resultados na remoção de ruído aleatório. Também são particularmente eficazes na presença de ruído impulsivo, também conhecido como ruído sal e pimenta, caracterizado pela presença de pontos brancos e pretos na imagem. O efeito do filtro de mediana pode ser observado na Figura 13:

Figura 13 - Filtro de mediana



Fonte: Adaptado de Soares & Silva (2013).

Na Figura 13, a imagem à esquerda é a imagem original, a imagem do centro é a imagem original com ruído sal e pimenta com SNR 40% (*Signal to Noise Ratio* 40%), o que significa que 40% dos pixels da imagem são substituídos por ruído, e a imagem da direita é a imagem com ruído sal e pimenta, após ser tratada com filtro de mediana 5x5 (autoria própria).

1.4.3.3 Detector de bordas canny

Dentre inúmeras técnicas de detecção de borda, uma das principais e mais populares é o Detector de bordas Canny (CANNY, 1986). Segundo Ding & Goshtasby (2001), o método suaviza a imagem com um filtro gaussiano apropriado,

determina a magnitude e ângulo do gradiente de cada pixel, classifica entre fundo ou borda, de acordo com pixels vizinhos e remove bordas duvidosas através de limiarização com histerese, isto é: elementos dentro da zona de histerese serão consideradas borda se estiverem conectadas a um elemento borda fora da zona. Para informações adicionais sobre o método, consultar Canny (1986) e Ding & Goshtasby (2001).

A Figura 14 mostra um exemplo de aplicação do detector de bordas Canny:

Figura 14 - Detector de bordas Canny



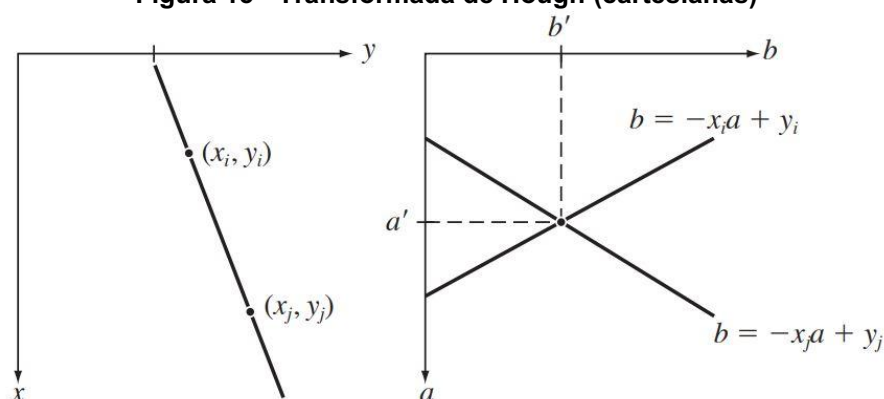
Fonte: OpenCV Docs (2002).

1.4.3.3.1 Transformada de hough

Hough (1962) propôs uma abordagem para encontrar retas, comumente chamada de transformada de Hough, com base nas bordas de uma imagem, que podem ser encontradas com o Canny edge detector ou qualquer outro método.

A transformada se baseia na seguinte formulação: dado um ponto (x_i, y_i) , existem infinitos conjuntos (a, b) que satisfazem $y_i = ax_i + b$. Dessa maneira, pode-se escrever a equação da reta no espaço de parâmetros: $b = -ax_i + y_i$. O que também é verdadeiro para o ponto (x_j, y_j) , obtendo $b = -ax_j + y_j$. A Figura 15 exemplifica a formulação, e demonstra que a interseção das duas retas no espaço de parâmetros equivale ao conjunto (a, b) da reta que passa por (x_i, y_i) e (x_j, y_j) .

Figura 15 - Transformada de Hough (cartesianas)



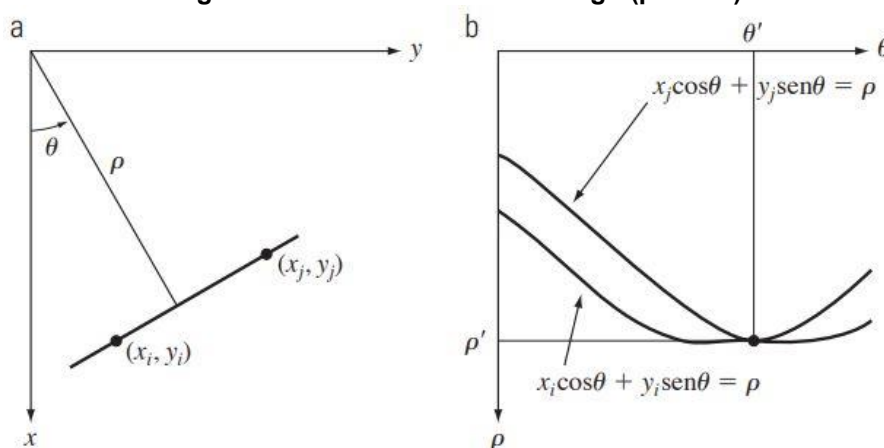
Fonte: Gonzalez e Woods (2010)¹³.

Esta formulação, porém, encontra problemas ao passo que o valor a se aproxima de infinito conforme a reta se aproxima da inclinação vertical. Este problema é solucionado substituindo para o sistema de coordenadas polares, onde $x\cos(\theta) + y\sin(\theta) = \rho$.

Desta maneira, os diagramas da Figura 15 são substituídos pelos diagramas apresentados na Figura 16.

Assim, as retas no espaço de parâmetros passam a ser curvas senoidais, e analogamente, as regiões com maior número de intercessões no espaço de parâmetros correspondem ao conjunto (ρ, θ) que formam as retas que englobam mais pontos dos contornos encontrados pelo detector de bordas.

Figura 16 - Transformada de Hough (polares)



Fonte: Gonzalez e Woods (2010).

1.4.3.3.2 Find contours

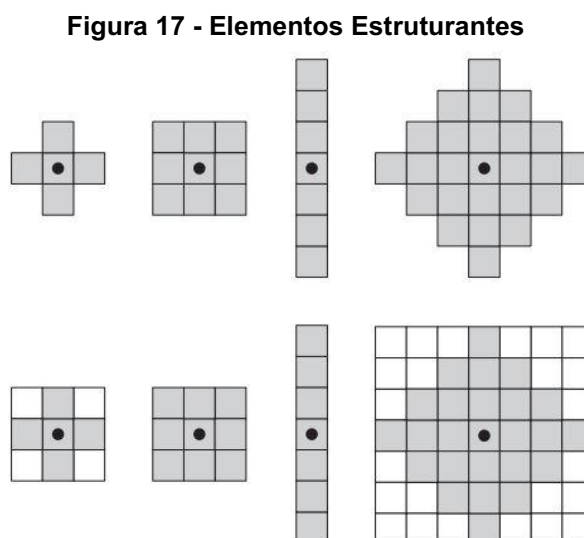
O método findContours (OPENCV, 2008), da biblioteca OpenCV, recebe como entrada uma imagem sobre a qual estão desenhadas apenas bordas, e sua função é detectar todos os contornos possíveis. É válido salientar que este método também é capaz de detectar a hierarquia, ou seja, contornos que englobam outros.

A saída do método é um vetor de comprimento N (número de contornos), que em cada posição, por sua vez recebe um vetor com todos os pixels da imagem que formam aquele contorno.

1.4.4 Morfologia Matemática

Segundo Gonzalez e Woods (2010), a morfologia matemática é uma ferramenta para extrair componentes das imagens que são úteis na representação e na descrição da forma de uma região, como fronteiras e esqueletos. Também são interessantes as técnicas morfológicas para pré ou pós-processamento, como a filtragem morfológica.

As operações morfológicas são baseadas nos elementos estruturantes, que por sua vez são “pequenos conjuntos ou subimagens usadas para examinar uma imagem buscando propriedades de interesse (GONZALEZ; WOODS, 2010)”. A Figura 17 mostra alguns exemplos de elementos estruturantes:



Fonte: Gonzalez e Woods (2010).

Na imagem pode ser observado, na linha superior, 4 exemplos de elementos estruturantes, uma cruz, um quadrado, um retângulo e um diamante. O ponto preto indica o centro do elemento estruturante, que é onde a operação acontecerá. Na linha de baixo são apresentados os formatos que os elementos estruturantes devem ter para serem usados efetivamente: retangular, com lados iguais ou não. Nesse caso, os *pixels* em cinza no elemento estruturante recebem o valor 1 e os *pixels* em branco, 0.

1.4.4.1 Erosão

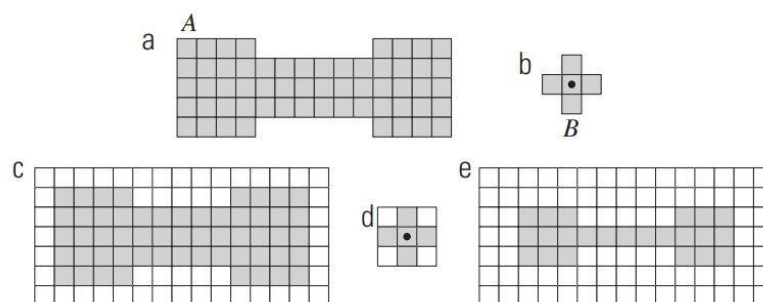
Segundo Gonzalez e Woods (2010), dada uma imagem A e um elemento estruturante B, a erosão é uma operação morfológica denotada pela Equação 2:

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2)$$

Ou seja, a erosão de A pelo elemento estruturante B é o conjunto de todos os elementos z tal que B na posição z_k seja igual ou subconjunto de A (esteja contido em A).

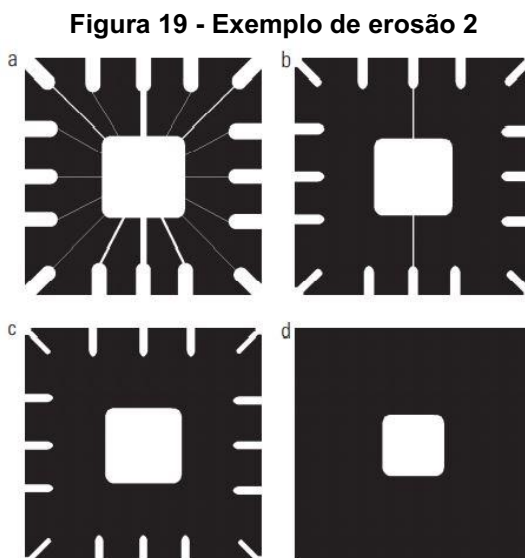
A Figura 18 mostra um exemplo simples de erosão. A erosão pode ser interpretada de uma maneira mais simplista: B fará uma varredura por A, e em cada posição que B estiver contido em A, a origem de B, naquela posição fará parte da nova imagem. Nota-se na Figura 18 que as bordas do objeto desapareceram.

Figura 18 - Exemplo de erosão 1



Fonte: Gonzalez e Woods (2010).

A Figura 19 mostra como a erosão pode ser utilizada para remover elementos de uma imagem:



Fonte: Gonzalez e Woods (2010).

Sendo (a) a imagem original, (b), (c) e (d) são a imagem original erodida por elementos estruturantes quadrados de tamanho 11x11, 15x15 e 45x45, respectivamente.

1.4.4.2 Dilatação

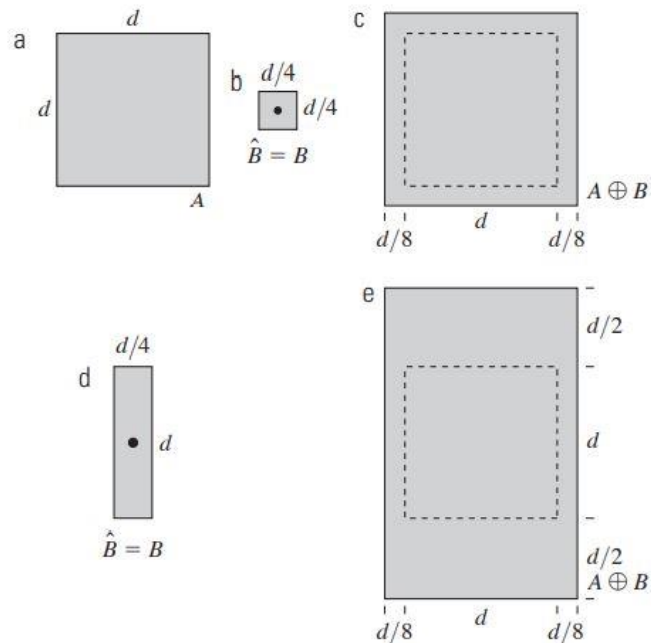
Segundo Gonzalez e Woods (2010), dados uma imagem A, um elemento estruturante B, a dilatação de A por B é dada pela Equação 3:

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \phi\} \quad (3)$$

Ou seja, a dilatação de A por B é o conjunto de elementos z tal que B refletido em torno de sua origem e transladado por z sobreponha A por pelo menos um elemento.

A Figura 20 mostra um exemplo de dilatação. Na figura, (a) é a imagem original, (b) o primeiro elemento estruturante, (c) a dilatação de (a) por (b). (d) é o segundo elemento estruturante e (e) a dilatação de (a) por (d). Nota-se que ao contrário da erosão, as bordas dos objetos aumentaram.

Figura 20 - Dilatação



Fonte: Gonzalez e Woods (2010).

1.4.5 Transformada de Fourier

Segundo Najarian & Splinter (2012), dentre todas as transformadas usadas em processamento de imagens, a transformada de Fourier é provavelmente a mais utilizada. Os autores afirmam que um sinal pode ser expresso em diferentes domínios. Sinais no domínio do tempo respondem perguntas de “quando” algo ocorre. Sinais no domínio da frequência respondem perguntas “com que frequência” algo ocorre.

A transformada de Fourier é, portanto, uma ferramenta naturalmente capaz de detectar padrões repetitivos dentro de uma imagem.

A Transformada Discreta de Fourier 2D é dada pela Equação 4:

$$G(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y) e^{-j \frac{2\pi(ux+vy)}{N}} \quad (4)$$

Onde u e v são os eixos de frequência ao longo dos quais $G(u, v)$ é descrita.

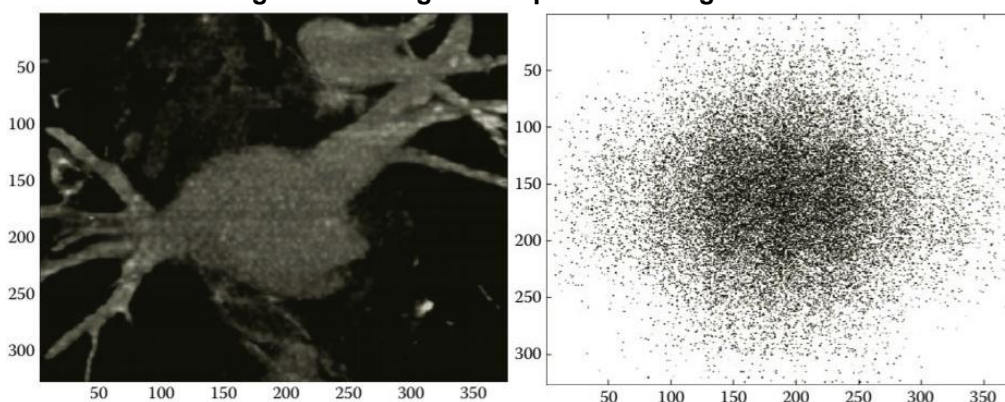
A transformada inversa, por sua vez é dada pela Equação 5:

$$g(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} G(u, v) e^{-j \frac{2\pi(ux+vy)}{N}} \quad (5)$$

A função 2D DFT, portanto recebe uma imagem como entrada, e retorna um espectro de magnitude em função da frequência, a partir do qual podem ser feitas algumas operações, entre elas: detectar um padrão específico desejado, filtragem passa-baixa, filtragem passa-alta.

A Figura 21 apresenta um exemplo de uma imagem e seu espectro de magnitude:

Figura 21 - Imagem e Espectro de magnitude



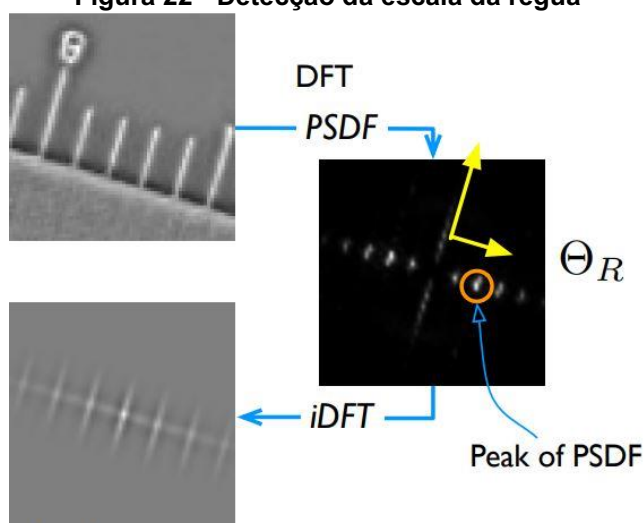
Fonte: Najarian & Splinter (2012).

Para informações adicionais sobre a 2D DFT, consultar Najarian & Splinter (2012) e Conci (2017).

1.4.5.1 Detecção da escala da régua

A Figura 22 mostra um trecho do algoritmo desenvolvido por Bhalerao & Reynolds (2014). Para sua aplicação, apenas interessava uma frequência específica: a frequência da escala da régua. Como as escalas da régua são elementos que se repetem constantemente inúmeras vezes, é natural que esta frequência se sobressaia no espectro de magnitude da transformada de Fourier. Portanto, aos autores coube identificar em qual frequência se situava os picos do espectro de magnitude:

Figura 22 - Detecção da escala da régua



Fonte: Bhalerao & Reynolds (2014).

1.4.5.2 Filtragem passa-baixa

A Figura 23 mostra uma imagem e seu espectro de magnitude, que serão utilizados para filtragem passa baixa e passa alta, em seguida:

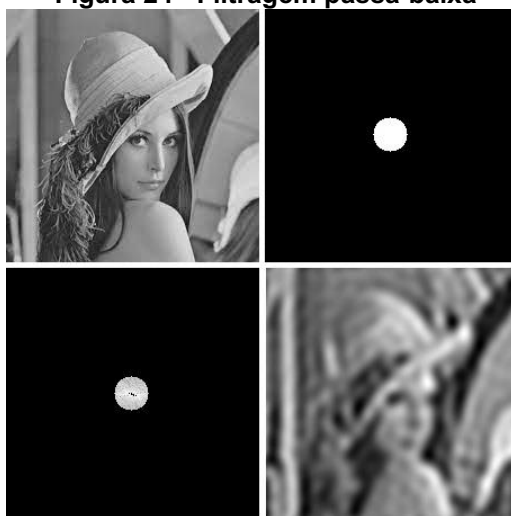
Sabendo que em imagens, altas frequências são regiões em que há transições mais abruptas de tonalidade de cor, e que as frequências baixas são as regiões em que há transições mais suaves, o filtro passa-baixa pela transformada de Fourier é um filtro suavizador, assim como filtros de média, mediana, etc.

A Figura 24, por sua vez, mostra um exemplo de filtragem passa-baixa utilizando a transformada de Fourier.

Figura 23 - Imagem e espectro de magnitude



Fonte: Autoria própria (2021).

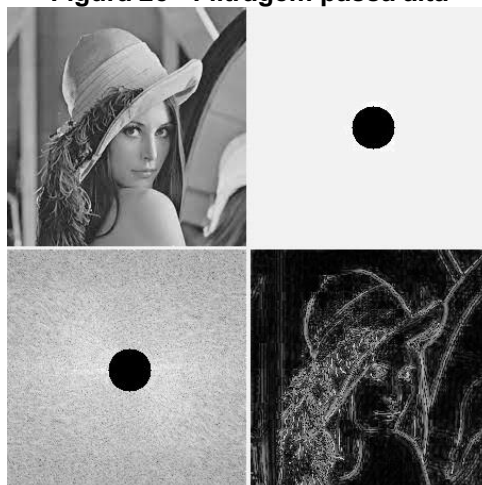
Figura 24 - Filtragem passa-baixa

Fonte: A autoria própria (2021).

A partir da imagem original (superior esquerdo), e seu espectro de magnitude, é aplicada a máscara da imagem (superior direito) ao espectro de magnitude. O espectro resultante é a imagem no canto inferior esquerdo, e a transformada inversa é a imagem no canto inferior direito.

1.4.5.3 Filtragem passa-alta

Levando em consideração que as frequências altas estão distantes do centro da figura, basta fazer o inverso da situação anterior, como apresentado na Figura 25:

Figura 25 - Filtragem passa alta

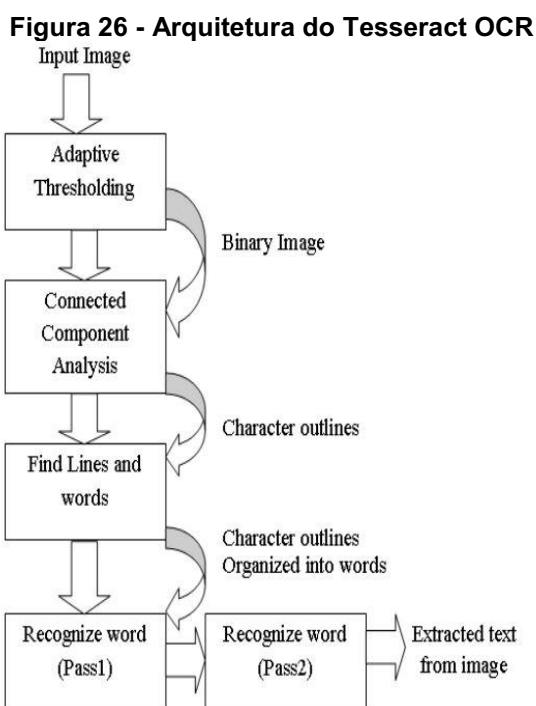
Fonte: A autoria própria (2021).

Na figura, podem ser observadas a imagem original (superior esquerdo), a máscara utilizada para a filtragem passa alta (superior direito), o espectro de magnitude após aplicada a máscara (inferior esquerdo) e a transformada inversa (inferior direito).

1.4.6 Tesseract OCR

Segundo Smith (2013), o *Engine Tesseract OCR* (acrônimo para *Optimal Character Recognition*) é um sistema *open source* desenvolvido pela Hp entre 1985 e 1995, atualmente desenvolvido majoritariamente na Google. Sua acurácia atingiu top 3 no teste UNLV em 1995, e com o recente trabalho, está alcançando as engines comerciais de OCR.

A arquitetura do Tesseract OCR é apresentada na Figura 26:



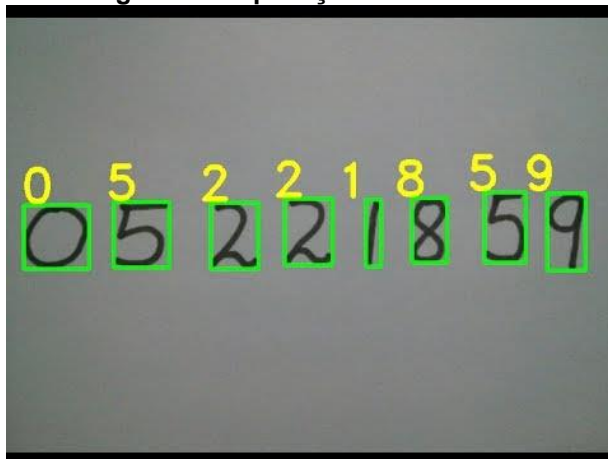
Fonte: Patel, Patel e Patel (2012).

A primeira etapa se trata de limiarização adaptativa. A segunda etapa é a análise de componentes conectadas, para detectar contornos de caracteres. Os contornos são em seguida convertidos em blocos, que são organizados em linhas de texto, que são analisados. O texto é então separado em palavras usando espaços

definidos e espaços *Fuzzy*. A classificação do texto é realizada por último, usando classificadores adaptativos.

A Figura 27 mostra um exemplo de aplicação da ferramenta:

Figura 27 - Aplicação do Tesseract



Fonte: Maret (2019).

2 MATERIAL E MÉTODOS

A metodologia está separada em duas partes: o estúdio de fotos e o algoritmo, e serão descritas nesta ordem, a seguir.

2.1 Estúdio de fotos

As técnicas de processamento de imagens utilizadas neste trabalho, e especificamente no algoritmo desenvolvido, são bastante sensíveis a variações nas fotos, portanto faz-se necessário a utilização de algum equipamento ou estrutura que padronize as fotos, neste caso, um estúdio de fotos, fazendo com que todas as fotos tenham o mesmo fundo branco, e em todas as fotos, a régua esteja com aproximadamente o mesmo tamanho. Adicionalmente o estúdio é capaz de limitar a posição da régua, contribuindo para que ela fique em uma posição mais próxima da vertical.

O estúdio de fotos se trata de uma caixa de MDF, com iluminação interna, e uma base, que pode ser observada na Figura 28:

Figura 28 - Estúdio de fotos



Fonte: Autoria própria (2021).

A caixa possui 21 centímetros de comprimento, 9 centímetros de largura e 13,5 centímetros de altura. A face superior da caixa possui um recorte de 6x4 centímetros, de maneira que quando um celular seja colocado sobre a caixa, sua câmera tenha visão do interior, e possibilite tirar fotos.

As faces frontal e traseira possuem um recorte na parte de baixo, de 0,5x4 centímetros, para que a caixa possa ser apoiada sobre a régua.

A iluminação interior é feita com fitas de LED, alimentadas por um adaptador AC/DC, modelo DYP-1210 (100V ~ 240V AC / 12V DC). Também possui um difusor de iluminação em papel branco.

A base é uma chapa de MDF com acabamento branco, 22x10cm, um pouco maior do que a caixa, para facilitar seu manuseio.

O estúdio de fotos é utilizado da seguinte maneira: a régua deve ser inserida no tanque de leite, conforme procedimento padrão, em seguida, é retirada e colocada sobre a base, sobre os quais são colocados o estúdio e um celular, e então a foto pode ser tirada.

Uma das fotos tiradas em uma visita em campo pode ser observada na Figura 29:

Figura 29 - Foto tirada com estúdio de fotos



Fonte: Autoria própria (2021).

2.2 Algoritmo

O algoritmo foi desenvolvido na linguagem de programação *Python* (3.7), no ambiente de desenvolvimento *Spyder* (v4.2.1), utilizando as bibliotecas *OpenCV* (v.3.4.2), *NumPy* (v. 1.20.0), *TesseractOCR* (4.1.1).

Como exposto anteriormente, as fotos tiradas com auxílio do estúdio são tiradas com um celular. Neste trabalho as fotos foram transferidas para um PC, onde o algoritmo foi desenvolvido e testado.

O algoritmo pode ser separado em 5 etapas, sendo elas: Detecção da linha do leite, Detecção das bordas da régua, Reconhecimento dos caracteres, Cálculo do

intervalo de escala, Segmentação das linhas da escala da régua. Essas etapas serão organizadas conforme a Figura 30, e serão exemplificadas com base na Figura 29, para melhor esclarecimento do funcionamento do algoritmo. As fotos tiradas em campo, com uso do estúdio de fotos serão apresentadas na seção “RESULTADOS E DISCUSSÃO”.

A Figura 30 apresenta um fluxograma que representa as etapas realizadas pelo algoritmo, que possui uma entrada (a imagem), 5 etapas, e uma saída (o resultado final).



Uma descrição de cada uma das 5 etapas é apresentada a seguir.

2.2.1 Identificação das bordas da régua

A primeira e a segunda etapa, detecção das bordas da régua e detecção da linha do leite, têm como objetivo separar a região de interesse na imagem.

Portanto o primeiro passo deve identificar, na foto, onde se encontram as bordas da régua, para que a imagem possa ser recortada.

A imagem passa por filtros passa baixa e erosão (morfologia matemática), seguidos pelo detector de bordas Canny e transformada de Hough. O algoritmo busca na lista de retas a que está mais à direita e a que está mais à esquerda, e que tiverem inclinações próximas a vertical.

O resultado pode ser observado na Figura 31.

Figura 31 - Identificação das bordas da régua



Fonte: Autoria própria (2022).

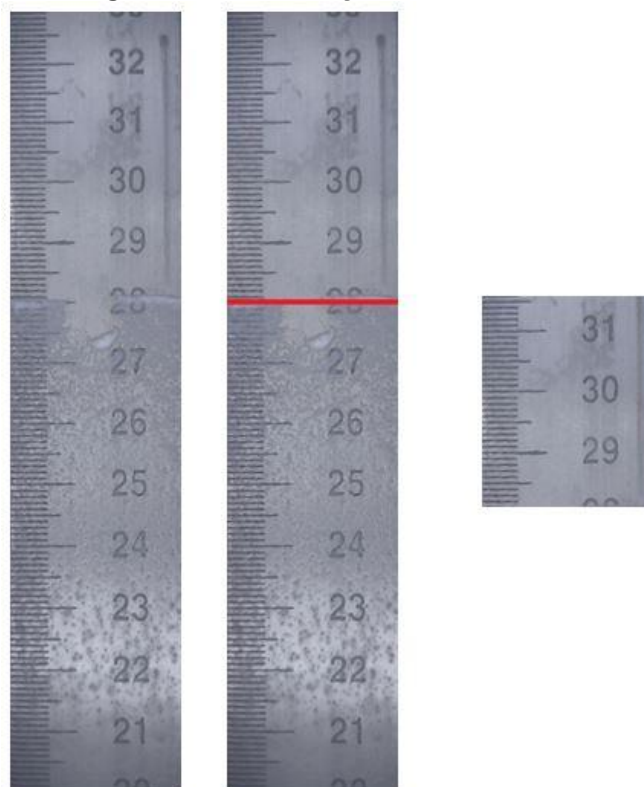
Na figura, pode ser observada, à esquerda, a imagem original, e à direita, a imagem após a aplicação dos passos descritos.

2.2.2 Identificação da linha do leite

Este trecho do algoritmo recebe a imagem à direita da Figura 31 como entrada, que passa por uma conversão do espaço de cor BGR para HSV, seguida de uma binarização *inRange*. Em seguida, são aplicados o detector de bordas Canny e a transformada de Hough, que retorna uma lista de retas candidatas, das quais o algoritmo seleciona a que tem a inclinação mais próxima da horizontal. A

foto então é recortada, eliminando-se o que está abaixo da linha. O resultado pode ser observado na Figura 32:

Figura 32 - Identificação da linha do leite



Fonte: Autoria própria (2022).

Na imagem, observa-se à esquerda a mesma imagem que a Figura 31 apresenta, ao centro, a imagem com destaque para a linha identificada pelo código, e à direita, a imagem resultante, já recortada.

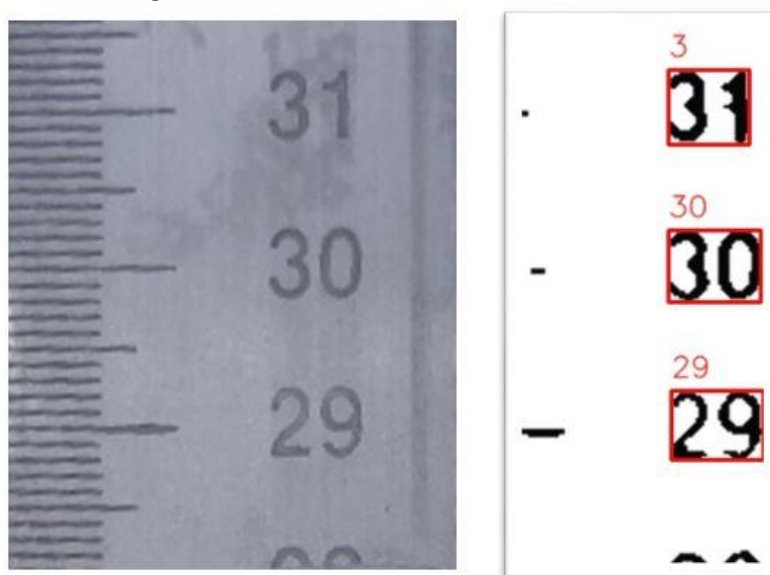
Esta etapa conclui a segmentação da Região de Interesse (ROI). Dessa maneira, resta na imagem apenas a região que efetivamente possui informações úteis, e essa imagem, exatamente um recorte da imagem original, será usada como entrada para as demais etapas do algoritmo.

2.2.3 Reconhecimento de caracteres

Nesta etapa, a região de interesse passa por uma conversão de espaço de cores BGR para HSV, binarização `inRange`, filtros passa-baixa e morfologia matemática. Para melhorar o desempenho da ferramenta, também é recortada uma faixa do lado esquerdo, removendo as linhas da escala da régua.

Levando-se em consideração que existem régua em que os números estão inclinados para a direita, ou para a esquerda, o algoritmo deve testar a região de interesse em 3 posições diferentes: original, inclinado 90° para a direita e 90° para a esquerda. Em seguida, o algoritmo usa a ferramenta TesseractOCR¹ em cada uma das 3 imagens, para identificar os caracteres na região de interesse, como pode ser observado na Figura 33:

Figura 33 - Reconhecimento de caracteres



Fonte: Autoria própria (2022).

Em seguida, o algoritmo escolhe a imagem em que foram encontrados mais números em sequência, retornando os números e suas caixas englobantes.

2.2.4 Cálculo do intervalo de escala

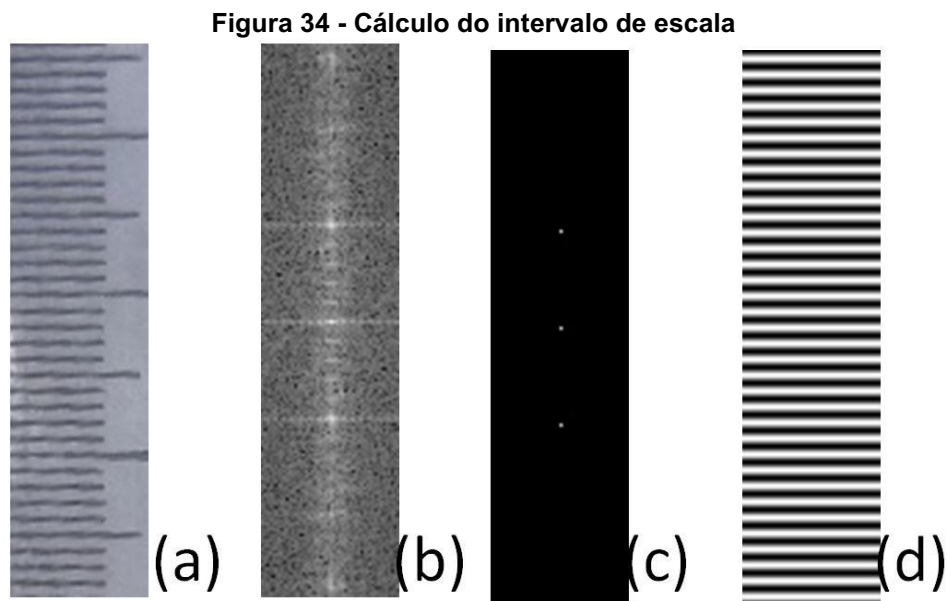
O cálculo do intervalo de escala será através da Transformada Discreta de Fourier. A partir do espectro de magnitudes da transformada de Fourier, podem ser determinadas quais as frequências de maior intensidade no espectro. Selecionado apenas estes pontos do espectro de magnitude de Fourier, o período do intervalo de escala será calculada matematicamente com base na distância dos picos ao centro, no espectro de magnitude (u). Segundo Ueda et. al. (2005), dada uma imagem de

¹ Mais informações sobre o TesseractOCR: <<https://github.com/tesseract-ocr/tesseract>>

altura M *pixels*, e a distância u , o período d pode ser calculado conforme a Equação 7:

$$d = \frac{M}{u} \quad (7)$$

Os resultados da transformada discreta de Fourier 2D podem ser observados na Figura 34, que está dividida em 4 imagens:



Fonte: Autoria própria (2022).

Na Figura 34, a imagem (a) é a região da régua que contém as escalas, a imagem (b), o espectro de magnitude da transformada de Fourier de (a). A imagem (c) é o espectro de magnitude da DFT da imagem original (b) após passar por uma binarização (intencionalmente de maneira que restassem apenas os 3 pontos de maior intensidade), e a imagem (d) sua transformada inversa. Segundo Conci (2017), o espectro (c) equivale ao espectro de um sinal senoidal puro, portanto, o sinal na imagem (d) é efetivamente um sinal senoidal puro, cuja frequência é a frequência da escala da régua. Portanto, pode-se calcular o intervalo de escala conforme a equação (7). Caso o leitor deseje se aprofundar mais no assunto, recomenda-se que consulte Najarian & Splinter (2012) e Conci (2017), e caso deseje familiarizar-se com o cálculo feito acima, consultar Ueda et. Al. (2005).

2.2.5 Segmentação das linhas da escala da régua

Nesta etapa, são segmentadas as linhas de graduação da régua, usando o detector de bordas Canny e a função *FindContours*. A imagem passa primeiramente, por um processo de agrupamento de cores *Kmeans*, em seguida por binarização *inRange*, para depois ser utilizado método Canny de detecção de bordas e o detector de contornos "*findContours*". Cada um dos contornos das linhas de graduação da régua tem seu centroide calculado, e é associado à caixa englobante do número mais próximo, obtida com o *Tesseract*, e pode ser observado na Figura 35:

Figura 35 - Segmentação das linhas da escala



Fonte: Autoria própria (2022).

O resultado final é calculado, então, determinando quantos pixels há, abaixo do centroide da linha, valor que é convertido em centímetros, através do valor do intervalo de escala e é subtraído do número lido pelo *Tesseract*.

2.3 Validação dos resultados

Para a avaliação dos resultados do modelo, algumas métricas estatísticas amplamente usadas como validação de modelos de regressão em *machine learning*

foram escolhidas: Erro médio, Raiz quadrada do erro médio quadrático, Erro médio absoluto, Erro médio absoluto percentual. Estas métricas são brevemente introduzidas e suas equações apresentadas nos subtópicos seguintes.

2.3.1 Raiz quadrada do erro médio quadrático

Segundo Chai (2014), a raiz quadrada do erro médio quadrático tem sido utilizada como métrica estatística padrão para mensurar a performance de vários modelos matemáticos. O autor também afirma que é uma métrica adequada e apropriada quando o erro segue uma distribuição normal. Todavia, sua sensibilidade em relação a *outliers* é uma preocupação comum no meio científico.

Sua equação é dada pela Equação 9:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (9)$$

2.3.2 Erro médio absoluto

Segundo Chai (2014), o erro médio absoluto é outra medida útil amplamente utilizada na validação de modelos de regressão, porém diferentemente da raiz quadrada do erro médio quadrático, que penaliza erros grandes, o erro médio absoluto dá o mesmo peso para todos os erros.

O erro médio absoluto é calculado pela Equação 10:

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (10)$$

2.3.3 Erro médio absoluto percentual

Segundo Myttenaere et. Al. (2016) o erro médio absoluto percentual é geralmente utilizado na prática por ser de interpretação bastante intuitiva se tratando

de erro. O uso dessa métrica é amplo nas áreas das finanças, uma vez que os ganhos e perdas são geralmente medidos em valores relativos, e é utilizado quando os valores que serão calculados estão sempre acima de zero.

O erro médio absoluto percentual pode ser calculado pela Equação 11:

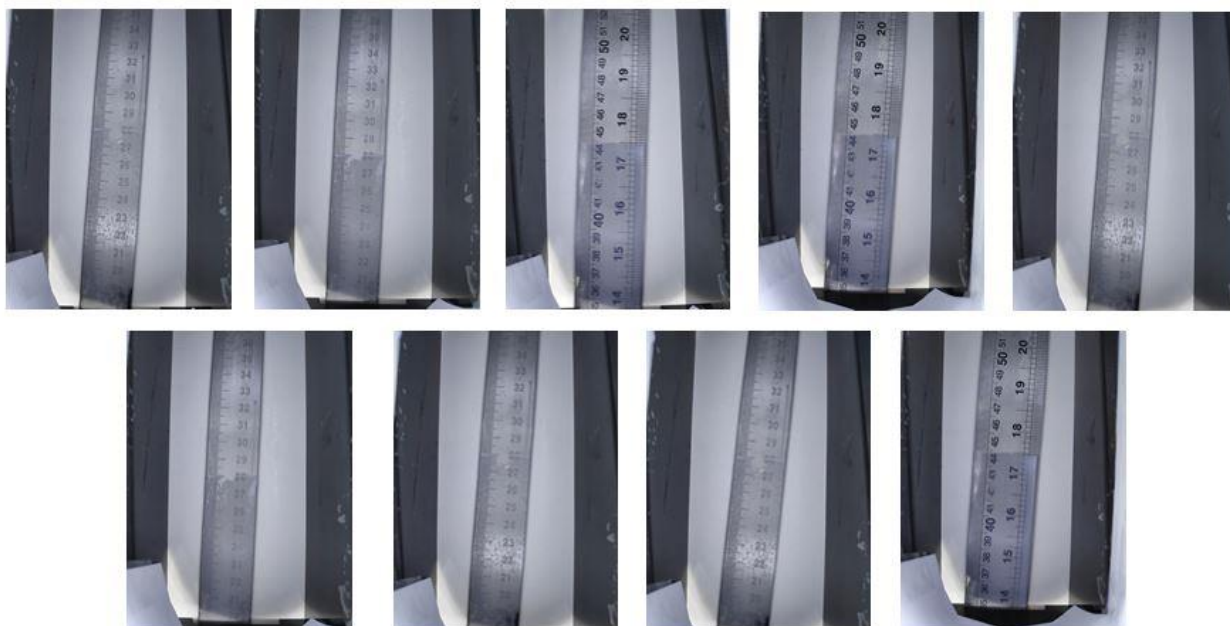
$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - g(x)_i}{y_i} \right| \quad (10)$$

Sendo y_i o valor esperado e $g(x)_i$ o valor calculado.

3 RESULTADOS E DISCUSSÃO

A Figura 36 mostra as fotos tiradas em campo em visita a produtores rurais de leite a granel, durante uma das coletas:

Figura 36 - Imagens obtidas durante a coleta



Fonte: Autoria própria (2021).

As fotos foram obtidas em visita a 2 produtores rurais distintos, logo: dois tanques diferentes e duas réguas diferentes. Portanto, pode-se verificar que são 6 fotos, diferentes entre si, da primeira régua e 3 fotos da segunda régua, também diferentes entre si.

Esta seção será dividida em duas partes, ao longo das quais serão apresentados alguns resultados parciais durante o processamento das imagens, cujos resultados finais serão apresentados e discutidos no tópico seguinte.

3.1 Segmentação da região de interesse

O resultado das duas primeiras etapas do algoritmo podem ser observadas na Figura 37:

Figura 37 - Região de interesse durante execução



Fonte: Autoria própria (2022).

São as duas primeiras etapas do algoritmo descrito pelo fluxograma da Figura 30: Detecção da linha do leite e Detecção das bordas da régua.

Em seguida, o algoritmo utiliza as imagens apresentadas anteriormente para extrair outros elementos necessários para a determinação do nível do leite dentro do tanque em questão. São eles: os números que se encontram na imagem, o intervalo de escala da régua e as linhas de graduação.

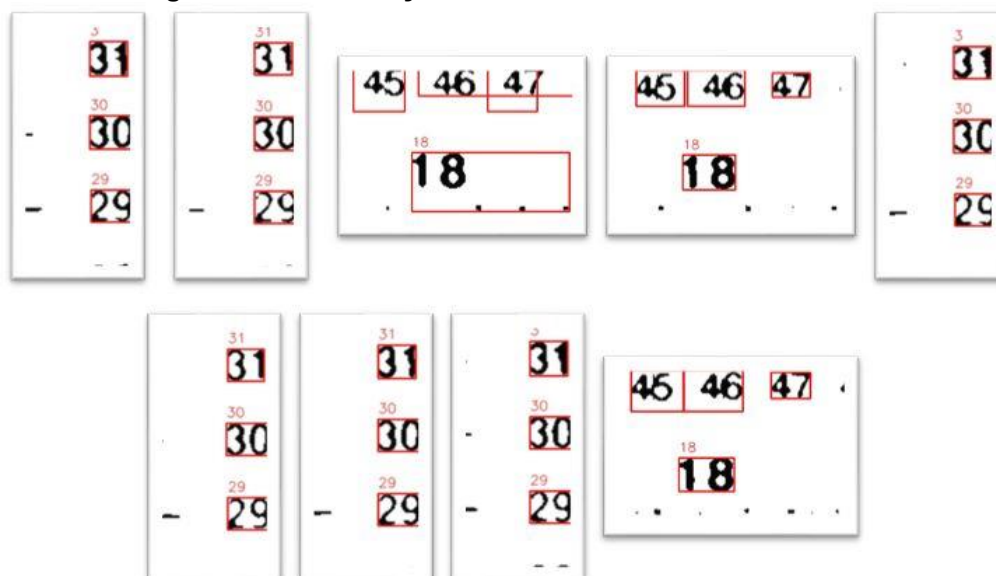
Algumas das imagens ao final desses processos são mostradas na Figura 38 e na Figura 39:

Figura 38 - Segmentação das linhas da graduação da régua



Fonte: Autoria própria (2022).

Figura 39 - Identificação de caracteres com TesseractOCR



Fonte: Autoria própria (2022).

3.2 Resultado final

O algoritmo obteve êxito em calcular a medida em todas as 9 imagens coletadas. As medidas obtidas são apresentadas na Tabela 1:

Tabela 1 - Resultados obtidos pelo algoritmo

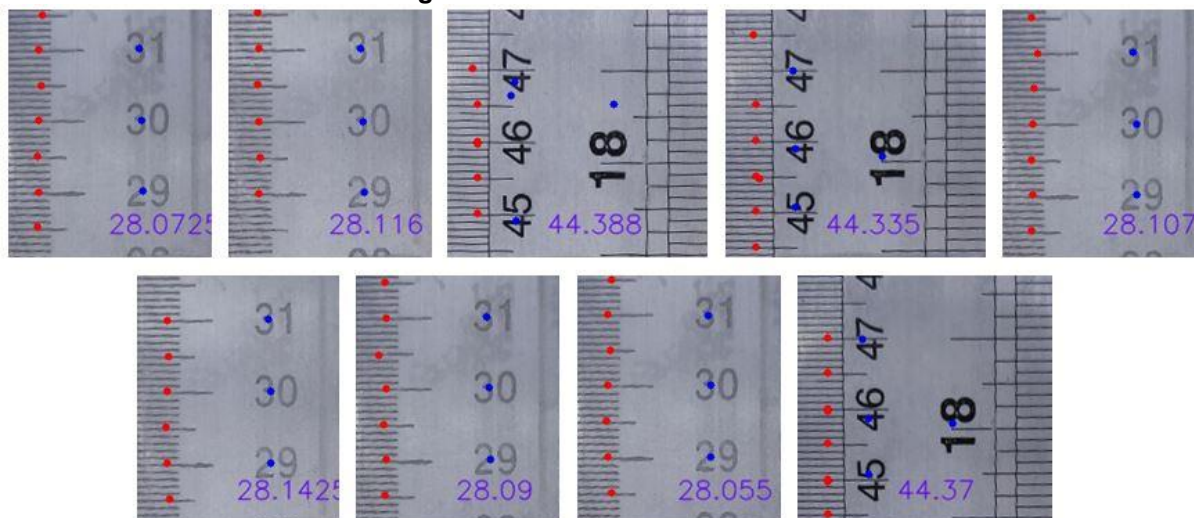
	Resultado esperado	Resultado Obtido
1	28,0	28,075
2	28,0	28,116
3	44,4	44,388
4	44,4	44,335
5	28,0	28,107
6	28,0	28,142
7	28,0	28,090
8	28,0	28,055
9	44,4	44,387

Fonte: Autoria própria (2022).

Os resultados obtidos podem ser melhor observados na Figura 40, que mostra o resultado final das Regiões de interesse obtidas para cada uma das

imagens, bem como o centro das caixas englobantes de cada um dos números encontrados, as linhas da graduação das régulas e o resultado final calculado:

Figura 40 - Resultado final obtido



Fonte: Autoria própria (2022).

Para avaliar o funcionamento do algoritmo, algumas métricas foram calculadas a respeito dos resultados obtidos. Os resultados apresentaram boa precisão e acurácia, aproximando-se bastante dos resultados esperados.

A raiz do erro quadrático médio (RMSE) consiste na média do erro das previsões ao quadrado. Pelo fato de elevar o erro ao quadrado, resultados distantes do esperado aumentam o valor da medida com facilidade. Esse algoritmo obteve $0,086 \text{ cm}^2$ de RMSE.

O erro médio absoluto obtido foi de $0,075 \text{ cm}$. Ao contrário do anterior, é mais brando com *outliers*, mesmo assim, sendo confiável para validação de modelos, além de sua visualização e interpretação serem mais intuitivas e simples.

O erro médio absoluto percentual, também de interpretação mais intuitiva, consiste no cálculo da porcentagem do valor esperado que o erro representa. Neste caso, o algoritmo erra por $0,255\%$ do valor esperado.

Embora o resultado tenha sido satisfatório, o método utilizado é bastante sensível a quaisquer modificações nas fotos pelo fato de envolver muitos processos diferentes realizados computacionalmente, e pelo fato de ser uma tarefa complexa, com essencialmente uma entrada dada pelo usuário: a foto. Levando em consideração pesquisas utilizadas como referência neste trabalho, o resultado foi bastante satisfatório. Em Bhalerao & Reynolds(2014), por exemplo, o algoritmo

construído recebe duas entradas: a foto de investigação forense, e um ponto da imagem dentro da escala da régua. Essa abordagem é tecnicamente inviável neste caso, tendo em vista a estrutura disponível no momento da coleta.

Outro ponto relevante no desenvolvimento deste projeto é o tempo de execução gasto no processamento das imagens, haja vista que a viabilidade da implementação do equipamento a ser desenvolvido que seja capaz de realizar todo o processamento em campo depende que o processo seja realizado em tempo hábil que não atrase as coletas e as rotas do motorista. O tempo de execução deste código é de não mais do que 20 segundos, variando entre 17 e 18, se utilizadas as 9 imagens coletadas. Sendo utilizada apenas uma imagem, como o algoritmo será utilizado em campo, o tempo de execução do código é reduzido proporcionalmente, levando em torno de 2 segundos.

4 CONCLUSÃO

Este trabalho apresenta um sistema, que, com base em uma foto de uma régua, é capaz identificar os elementos de uma régua e identificar em seu comprimento, até que ponto do instrumento foi submerso em leite, durante o processo da coleta. Esse processo de identificação, contudo, exige que a régua seja retirada do tanque e posicionada junto ao estúdio de fotos e que a foto seja tirada com um celular, para que esta imagem possa ser analisada computacionalmente.

Tratando-se de processamento de imagens, é evidente que a abordagem oferece praticidade e segurança tanto para o produtor quanto para a empresa que realiza a coleta, pois se trata de uma abordagem que pode minimizar os erros inerentes à interferência humana em um processo manual.

O projeto, dentro de suas limitações, logrou êxito, atingindo resultados muito próximos dos resultados esperados, na leitura da régua, na coleta de leite a granel, com base em fotos tiradas em campo, com auxílio de um estúdio de fotos produzido para o trabalho.

As leituras obtidas pelo algoritmo atingiram métricas de erro bastante baixas, como, por exemplo, o Erro Absoluto Médio Percentual, de 0,255% e o Erro Médio Absoluto de 0,086 cm, considerados pequenos, dada a complexidade de se realizar todo o processo computacionalmente.

O resultado se mostrou satisfatório também, considerando que um dos fatores necessários é que não haja grande acréscimo de tempo comparado ao processo original da leitura manual, pelo motorista, quando este protótipo for implementado, esperando-se que o tempo necessário para a leitura completa da régua com a utilização deste protótipo não tome tempo excessivo e esforço do operador.

Adicionalmente, apresentam-se perspectivas promissoras quanto à implementação de um novo protótipo cuja utilização seja facilitada, tendo em vista a o baixo custo computacional desse algoritmo.

4.1 Trabalhos futuros

Para trabalhos futuros recomenda-se a confecção de um novo estúdio de fotos, que tenha uma utilização mais prática, facilitando o manuseio pelo operador.

Também se sugere que as análises e processamento sejam realizados no local da coleta, sendo exigido para tal um laptop, ou então uma placa *Raspberry Pi*. Dessa maneira, incoerências em relação à medida podem ser solucionadas antes de a coleta acontecer.

Outra melhoria que pode ser realizada é a generalização deste código, para que seja capaz de realizar a leitura em mais réguas, de modelos diferentes.

Por fim, recomenda-se a implementação de um banco de dados que armazene os dados dos tanques de cada produtor e seja capaz de calcular o volume de leite contido no tanque, com base na medida obtida, ao invés de retornar a medida da régua.

REFERÊNCIAS

BARBOSA, Alexandre Henrique. C.; SILVA, Bruno Ramon de Almeida e; BANDEIRA, Junior Marcos. **PROCESSAMENTO DIGITAL DE IMAGENS PARA O RECONHECIMENTO DE PLACAS DE VEICULOS**. 2014. 25 f. TCC (Graduação) - Curso de M Sistemas de Informação, Unibalsas, Balsas, 2017.

BHALERAO, Abhir. REYNOLDS, Gregory. (2014). **Ruler Detection for Autoscaling Forensic Images**. International Journal of Digital Crime and Forensics. 6. 9-27. 10.4018/ijdcf.2014010102.

CANNY, John. **A computational approach to edge detection**. IEEE Transactions on pattern analysis and machine intelligence, n. 6, p. 679-698, 1986.

CHAI, Tianfeng; DRAXLER, Roland R. **Root mean square error (RMSE) or mean absolute error (MAE)**. Geoscientific Model Development Discussions, v. 7, n. 1, p. 1525-1534, 2014.

CHEN, C., KOJCHEV, R., HASCHTMANN, D. *et al.* **Ruler Based Automatic C-Arm Image Stitching Without Overlapping Constraint**. *J Digit Imaging* 28, 474–480 (2015). <https://doi.org/10.1007/s10278-014-9763-3>

CHEVEIGNÉ, A. KAWAHARA, H. (2002). **YIN, a fundamental frequency estimator for speech and music**. Journal of the Acoustical Society of America, 111(4)

CONCI, Aura. **Transformada de Fourier**. Niterói: Curso Pisb, 2017. 92 slides, color. Disponível em: <http://www.ic.uff.br/~aconci/Fourier2017.pdf>. Acesso em: 03 maio 2021.

CUNHA, André Luiz Barbosa Nunes da. **Sistema automático para obtenção de parâmetros do tráfego veicular a partir de imagens de vídeo usando OpenCV**. 2013. Tese de Doutorado. Universidade de São Paulo.

DINESH, S.; RADHAKRISHNAN, P. **Linear and nonlinear approach for DEM smoothing**. Discrete Dynamics in Nature and Society, v. 2006, 2006.

DING, Lijun; GOSHTASBY, Ardeshir. **On the Canny edge detector**. Pattern Recognition, v. 34, n. 3, p. 721-725, 2001.

DORNELAS, Stivilane. **A coleta a granel de leite refrigerado e suas implicações sobre as relações contratuais entre produtor e indústria: um estudo de caso**. 2000. Tese de Doutorado. Universidade de São Paulo.

FALCÃO, Alexandre Xavier; MENOTTI, David. **Segmentação por limiarização**. Campinas: Instituto de Computação, 2014. 52 slides, color. Disponível em: <https://www.ic.unicamp.br/~afalcao/mo443/index.html>. Acesso em: 04 maio 2021.

GONZALEZ, Rafael C.; WOODS, Richard E.. **Processamento digital de imagens**. 3. ed. São Paulo: Pearson, 2010. 644 p.

HERRMANN, M., ZAMBANINI, S., KAMPEL, M. (2009). **Image Based Measurement of Ancient Coins**. Proc. of the 37th Conference on Computer Applications and Quantitative Methods in Archaeology, Williamsburg, USA, p. 117–121.

HOUGH, Paul VC. **Method and means for recognizing complex patterns**. U.S. Patent n. 3,069,654, 18 dez. 1962.

KONOVALOV, D. A. DOMINGOS, J. A. BAJEMA, C. WHITE, R. D. JERRY, D. R. 2017. **Ruler Detection for Automatic Scaling of Fish Images**. Em International Conference on Advances in Image Processing (ICAIP 2017) Proceedings. Association for Computing Machinery, New York, NY, USA, 90–95. DOI:<https://doi.org/10.1145/3133264.3133271>

LANG, Marco. **Diferença entre CMYK e RGB?** 2013. Disponível em: <https://estampaweb.com/diferenca-entre-cmyk-e-rgb/>. Acesso em: 04 maio 2021.

LATICÍNIOS DAMARE. **Normas técnicas de coleta de leite a granel e de amostras da embraça**. Montanha - ES: Laticínios Damare, 2010. Color.

MARET, Nicholas. **Ocr Machine Learning Python**. 2019. Disponível em: <https://quantumcomputingtech.blogspot.com/2019/03/ocr-machine-learning-python.html>. Acesso em: 04 maio 2021.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. **Instrução Normativa nº 62**, de 29/12/2011. Diário Oficial da União, Brasília, 20 set. 2011b. Seção I.

MYTTENAERE, Arnaud de, et al. **Mean absolute percentage error for regression models**. Neurocomputing, v. 192, p. 38-48, 2016.

NAJARIAN, Kayvan; SPLINTER, Robert. **Biomedical signal and image processing**. Taylor & Francis, 2012.

NIBLACK, Wayne. **An introduction to digital image processing**. Prentice-Hall, Inc., 1986, p115-116.

OPENCV. **Canny Edge Detection**. 2002. Disponível em: https://docs.opencv.org/master/da/d22/tutorial_py_canny.html. Acesso em: 03 maio 2021.

OPENCV. **Contours : Getting Started**. 2008. Disponível em: https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html. Acesso em: 18 maio 2022.

PATEL, Chirag; PATEL, Atul; PATEL, Dharmendra. **Optical character recognition by open source OCR tool tesseract: A case study**. International Journal of Computer Applications, v. 55, n. 10, p. 50-56, 2012.

RIBEIRO, M. M. G.; GOMES, Abel. **Adaptação de cor para dicromatas na visualização de imagens**. 20.º Encontro Português de Computação Gráfica, 2010.

SAUVOLA, Jaakko; PIETIKÄINEN, Matti. **Adaptive document image binarization**. Pattern recognition, v. 33, n. 2, p. 225-236, 2000.

SILVA, Guilherme B. da et al. **Estratégia de Detecção de Faixas de Trânsito Baseada em Câmera Monocular para Sistemas Embarcados**. Anais da Sociedade Brasileira de Automática, v. 2, n. 1, 2020.

SOARES, Pablo Luiz Braga; SILVA, José Patrocínio da. **Redes neurais aplicadas na redução de ruído impulsivo de imagens digitais**. Revista Brasileira de Computação Aplicada, v. 5, n. 1, p. 2-13, 2013.

SMITH, Ray W. **History of the Tesseract OCR engine: what worked and what didn't**. In: Document Recognition and Retrieval XX. International Society for Optics and Photonics, 2013. p. 865802.

TIMM, Cláudio Dias; OLIVEIRA, Daniela dos Santos de. **Nova legislação do leite no Brasil**. Ciência & Tecnologia Veterinária. Disponível em: <http://www.ufpel.tche.br/veterinaria/inspleite/documentos/prelo/legisla.pdf>. Acesso em, v. 16, n. 01, 2011.

UEDA, K., BABA, T., NAKAGAWA, Y., AMANO, K. (2005). **Detection of Scale Intervals in Digital Images**. em Data Engineering Workshops, 2005. 21st International Conference, p. 1232–1237.