

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA ELÉTRICA

ARNALDO SOARES LEDERHANS

**TRATAMENTO DE IMAGEM PARA AQUISIÇÃO DA INFORMAÇÃO DA DATA DE
VALIDADE EM GARRAFAS DE REFRIGERANTES UTILIZANDO PYTHON
OPENCV, TESSERACT E EASYOCR**

São Leopoldo
2022

ARNALDO SOARES LEDERHANS

**TRATAMENTO DE IMAGEM PARA AQUISIÇÃO DA INFORMAÇÃO DA DATA DE
VALIDADE EM GARRAFAS DE REFRIGERANTES UTILIZANDO PYTHON
OPENCV, TESSERACT E EASYOCR**

Projeto de Pesquisa apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação, pelo Curso de Engenharia Elétrica da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. Lucio Rene Prade

São Leopoldo

2022

Dedico este trabalho a Deus que me proporcionou o dom de aprender, para que eu possa replicar todo conhecimento aprendido e a meus filhos, que são minha inspiração para a busca do conhecimento.

AGRADECIMENTOS

Agradeço, acima de tudo, a Deus, que me concedeu a graça de finalizar mais um desafio, ajudando a enfrentar os obstáculos diários e a encontrar as respostas que parecem sem solução.

A minha mãe, pelo apoio e pelo incentivo, tanto pessoal quanto profissional, pela profissão que escolhi.

A minha esposa Fernanda, pelo companheirismo e pela compreensão de sempre e aos amigos e colegas de curso pela união prestada para alcançar o objetivo conjunto da aprovação.

Aos meus filhos Maria Eduarda, João Gabriel e Ana Clara, pela paciência de esperar um pouco mais para brincar e de entenderem que o objetivo final foi sempre dar o melhor para eles.

Ao professor e orientador Prof. Dr. Lúcio Renê Prade, que com suas orientações, me proporcionou a realização deste trabalho.

RESUMO

A qualidade da embalagem é um fator determinante para que o consumidor faça a escolha por um determinado produto. Além de garantir a aparência adequada, as empresas precisam atender os requisitos legais, dentre os quais está compreendida a impressão de determinadas informações nas embalagens. O tempo operacional para execução periódica das verificações de conformidade e o custo com descarte de produtos não conformes é elevado, podendo ser evitado. Este trabalho tem como objetivo investigar a exequibilidade técnica de um sistema baseado em uma placa Raspberry Pi e uma câmera de baixo custo para Raspberry Pi para inspeção automática da impressão da data de validade e lote de produção em garrafas de refrigerantes. Tal sistema consiste na captura da imagem da garrafa na linha de produção, tratamento da imagem, reconhecimento do texto e comparação dos dados extraídos com o padrão de informações definidos para cada tipo de embalagem ou produto, tendo como objetivo substituir a inspeção realizada pelo operador, aumentando a confiabilidade e reduzindo o tempo entre as inspeções. Após as etapas de tratamento e reconhecimento do texto via ferramenta de reconhecimento ótico de caracteres ocorre a classificação do modelo avaliado e a transferência das informações de conformidade da impressão de codificação para o banco de dados, que servirá de base de informação em uma página web, criada consulta do histórico, bem como para ajustes da posição da imagem ou inserção de padrões de informação das codificações. Foram realizadas simulações de captura automática da imagem das garrafas, bem como a avaliação da qualidade das imagens obtidas, submetendo o processo de captura de imagens a variações de velocidade da linha de produção e de iluminação. Os testes de desempenho do algoritmo de tratamento de imagens e de reconhecimento de caracteres foi realizado através da aplicação e coleta de resultados de saída do algoritmo aplicado em diferentes amostras de imagens coletadas sob diferentes condições de velocidade e iluminação. Os resultados de desempenho obtidos foram satisfatórios, tendo em vista as limitações do hardware de captura das imagens e a baixa qualidade da codificação impressas nas amostras. A qualidade das imagens capturadas de forma automática foi considerada adequada quando utilizada iluminação auxiliar e aplicadas velocidades de simulação da linha de produção de 5000 e 12000 garrafas por hora, onde a acuracidade da ferramenta de OCR Tesseract chegou a 93,33%.

Palavras-chave: Tratamento de imagens. Inspeção de qualidade automática.
Raspberry Pi. Visão computacional.

LISTA DE FIGURAS

Figura 1 - Espaço de coordenadas cartesianas	21
Figura 2 - Espaço de coordenadas cartesianas bidimensionais de uma imagem MxN ...	22
Figura 3 - Componentes de um sistema de processamento de imagens de uso geral..	25
Figura 4 - Mapa de cores da representação de uma imagem colorida	26
Figura 5 - Máscara 3x3 pixels com coeficientes arbitrários.....	27
Figura 6 - (a) Imagem original. (b) Imagem com filtro passa-baixas	29
Figura 7 - (a) Imagem original. (b) Imagem com filtro passa-altas	30
Figura 8 - Exemplo do processo de aquisição de imagens.....	32
Figura 9 - Exemplo de aquisição de imagens digitais com obturador global e de rolamento, respectivamente	33
Figura 10 - Raspberry Pi 4 Modelo B	34
Figura 11 - Câmera OV5467	35
Figura 12 - Tela sensível ao toque.....	36
Figura 13 - Sensor Infravermelho reflexivo	37
Figura 14 - Interface de programação do Node-Red.....	40
Figura 15 - (a) Imagem original. (b) Imagem em tom de cinza. (c) Imagem binarizada. (d)Imagem com aplicação de operador morfológico	41
Figura 16 - (a) Aplicação do filtro de Canny. (b) Detecção dos dígitos	42
Figura 17 - Processos de tratamento com caracteres pontilhados.....	42
Figura 18 - Técnicas de um sistema OCR	43
Figura 19 - Visão geral do sistema de aquisição, processamento de imagens e OCR	45
Figura 20 - Montagem do sistema na linha de produção	46
Figura 21 - Estrutura para simulação da linha de produção.....	47
Figura 22 - Fluxograma para aquisição da imagem	48
Figura 23 - (a) Imagem original. (b) Escala de cinza. (c) Filtro Blur	49
Figura 24 - (a) Filtro Blur. (b) Filtro Sharpen.....	50
Figura 25 - (a) Erosão. (b) Morfologia Gradiente. (c) Inversão binária	50
Figura 26 - Parâmetros Tesseract.....	51
Figura 27 - Resultado da avaliação com o OCR Tesseract.....	51
Figura 28 - Trecho de código para execução do EasyOCR	52
Figura 29 - Resultado da avaliação com o OCR EasyOCR	52

Figura 30 - Programa em Node-Red	53
Figura 31 - Fluxograma do programa de tratamento de aplicação de OCR's	54
Figura 32 - Exemplos de imagens capturas durante os testes.....	59
Figura 33 - Imagens avaliadas e respectivos resultados da avaliação OCR.....	60
Figura 34 - Script python para cálculo da similaridade.....	60
Figura 35 - Imagem captura com luz ambiente	64
Figura 36 - Exemplos de Imagens: (a) Velocidade 5000 garrafas/h, (a) Velocidade 12000 garrafas/h, (a) Velocidade 18000 garrafas/h	64
Figura 37 - Imagem da garrafa captura na velocidade de 18000 garrafas por hora..	65

LISTA DE QUADROS

Quadro 1 - Especificações técnicas do Raspberry Pi 4 Modelo B.....	35
Quadro 2 - Matriz de confusão genérica	61
Quadro 3 - Resultados de conformidade obtidos	68

LISTA DE TABELAS

Tabela 1 - Velocidades da linha de produção e de simulação	47
Tabela 2 - Taxas de aproveitamento de imagens captadas de forma automática	66
Tabela 3 - Similaridade entre os resultados encontrados com a predição	67

LISTA DE SIGLAS

ALPR	<i>Automatic Licence Plate Recognition</i> (Reconhecimento Automático de Placas de Veículos)
ANVISA	Agência Nacional de Vigilância Sanitária
ARM	<i>Advanced RISC Machine</i> (Máquina RISC Avançada)
CCD	<i>Charge Coupled Device</i> (Dispositivo de Carga Acoplada)
CMOS	Complementary Metal-Oxide Semiconductor (Semicondutor de Óxido de Metal Complementar)
CSI	<i>Camera Serial Interface</i> (Interface Serial da Camera)
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
DPI	<i>Dots Per Inch</i> (Pontos por Polegada)
GPIO	<i>General Purpose Input/Output</i> (Entrada e saída de uso geral)
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
HDMI	<i>High-Definition Multimedia Interface</i> (Interface Multimídia de Alta Definição)
HTTP	<i>Hyper Text Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
MES	<i>Manufacturing Execution System</i> (Sistema de Execução da Manufatura)
MRZ	<i>Machine Readable Zone</i> (Zona Legível por Máquina)
OCR	Optical Character Recognition (Reconhecimento Óptico de Caracteres)
OEM	<i>OCR Engine Mode</i> (Modo de Mecanismo OCR)
PDI	Processamento digital de imagens
PET	Poli tereftalato de etila
PNG	<i>Portable Network Graphics</i> (Gráficos Portáteis de Rede)
PSM	<i>Page Segmentation Mode</i> (Modo de segmentação de página)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde, Azul)
RISC	<i>Reduced Instruction Set Computer</i> (Computador com um Conjunto Reduzido de Instruções)
SAC	Serviço de Atendimento ao Consumidor

TCO	Custo Total de Propriedade
TIFF	<i>Tag Image File Format</i> (Formato de arquivo de imagem marcada)
ULA	Unidade Lógica Aritmética
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
WEB	<i>World Wide Web</i> (Rede Mundial de Computadores)

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Tema	16
1.2 Delimitação do tema	16
1.3 Definição do problema.....	17
1.4 Objetivos	17
1.4.1 Objetivo geral	17
1.4.2 Objetivos específicos.....	18
1.5 Justificativa.....	18
1.6 Organização do trabalho	19
2 FUNDAMENTAÇÃO TEÓRICA	20
2.1 Definição sobre processamento de imagens digitais	20
2.1.1 Fundamentos de imagens digitais	20
2.1.2 Processamento digital de imagens.....	22
2.1.3 Alguns passos fundamentais para o processamento digital de imagens	23
2.1.4 Componentes de um sistema de processamento de imagens	24
2.1.5 Imagens multibanda ou multiespectrais	26
2.2 Conceitos sobre métodos de tratamento de imagens digitais	27
2.2.1 Kernel ou Máscara	27
2.2.2 Filtros digitais no domínio espacial.....	28
2.2.3 Filtros passa-baixas	28
2.2.4 Filtro média.....	29
2.2.5 Filtro passa-altas	29
2.3 Conceitos sobre identificação e reconhecimento de textos em imagens digitais	30
2.3.1 Identificação de texto em imagens digitais	30
2.3.2 Aplicações de reconhecimento óptico de caracteres	31
2.4 Processos de aquisição de imagens	31
2.4.1 Sensores e sistemas de aquisição de imagens.....	32
2.4.2 Obturador global e de rolamento	32
2.5 Mecanismos de Hardware	34
2.5.1 Raspberry Pi Modelo 4B.....	34
2.5.2 Câmera Raspberry Pi OV5647	35

2.5.3 Monitor de tela sensível ao toque.....	36
2.5.4 Sensor infravermelho reflexivo	36
2.6 Mecanismos de software	37
2.6.1 Debian	38
2.6.2 Biblioteca OpenCV	38
2.6.3 Biblioteca Tesseract	38
2.6.4 Biblioteca EasyOCR.....	39
2.6.5 Node-Red	39
2.7 Trabalhos correlatos	40
2.7.1 Leitura de caracteres de codificação.....	40
2.7.2 Comparação entre algoritmos OCR	43
2.7.3 Aplicação de sistemas de inspeção com Raspberry Pi	43
3 DESENVOLVIMENTO DO SISTEMA DE AQUISIÇÃO, TRATAMENTO DE IMAGENS E OCR	45
3.1 Montagem do sistema de aquisição de imagens na linha de produção.....	45
3.2 Montagem da estrutura para simulação da linha de produção.....	46
3.2.1 Simulação da velocidade da linha de produção e aquisição das imagens.....	47
3.2 Desenvolvimento do algoritmo de captura, processamento e avaliação das imagens.....	48
3.2.1 Algoritmo para aquisição das imagens de forma automática	48
3.2.2 Algoritmo para tratamento da imagem	49
3.2.3 Avaliação com OCR Tesseract	50
3.2.4 Avaliação com OCR EasyOCR	52
3.2.5 Processamento dos resultados via Node-Red	53
4 METODOLOGIA	56
4.1 Qualidade de captura das imagens	56
4.1.1 Velocidade de 5000 garrafas por hora e iluminação natural	56
4.1.2 Velocidade de 5000 garrafas por hora e iluminação auxiliar de led de 800 Lux	57
4.1.3 Velocidade de 5000 garrafas por hora e iluminação auxiliar de led de 1500 Lux	57
4.1.4 Velocidade de 12000 garrafas por hora e iluminação natural	57
4.1.5 Velocidade de 12000 garrafas por hora e iluminação auxiliar de led de 800 Lux	57

4.1.6 Velocidade de 12000 garrafas por hora e iluminação auxiliar de led de 1500 Lux	57
4.1.7 Velocidade de 18000 garrafas por hora e iluminação natural	58
4.1.8 Velocidade de 18000 garrafas por hora e iluminação auxiliar de led de 800 Lux	58
4.1.9 Velocidade de 18000 garrafas por hora e iluminação auxiliar de led de 1500 Lux	58
4.2 Desempenho do reconhecimento ótico de caracteres – OCR Tesseract.....	59
4.3 Critério de conformidade da codificação	61
5 RESULTADOS OBTIDOS	63
5.1 Qualidade das imagens adquiridas de forma automática	63
5.1.1 Variação de velocidade e exposição a iluminação natural	63
5.1.2 Velocidades variadas e exposição a iluminação de 800 Lux.....	64
5.1.3 Velocidades variadas e exposição a iluminação de 1500 Lux.....	65
5.2 Desempenho do reconhecimento ótico de caracteres Tesseract.....	66
5.3 Conformidade da codificação	68
6 CONSIDERAÇÕES FINAIS	70
REFERÊNCIAS.....	72

1 INTRODUÇÃO

O mercado consumidor de produtos industrializados está cada vez mais exigente no que diz respeito a qualidade. Este fato tem motivado as indústrias a investirem cada vez mais nos processos de inspeção de qualidade, por se tratar de um diferencial competitivo.

Podemos definir a qualidade de um produto como sendo o grau de satisfação dos consumidores com os requisitos dados por um conjunto de características intrínsecas, ou seja, de uma combinação de atributos do projeto e da produção, os quais são avaliados constantemente desde a sua aquisição. Além da aparência adequada, outras informações constantes na embalagem são necessárias, como a codificação da data de validade e lote de produção, por exemplo. A inserção dessas informações nos produtos alimentícios é obrigatória de acordo com a Resolução-RDC nº 259 da Agência Nacional de Vigilância Sanitária (ANVISA), que é o órgão aprovador do Regulamento Técnico sobre Rotulagem de Alimentos Embalados. A Resolução 259 estabelece as formatações permitidas e enfatiza que todas as informações devem ser exibidas de forma clara e nítida em todos os produtos. A não execução de qualquer uma das especificações constantes na resolução está sujeita a execução da Lei Federal nº 6437 que prescreve as infrações à legislação sanitária federal. A principal finalidade desta regulamentação é de assegurar a saúde do consumidor, demandando que todas as informações relacionadas ao consumo sejam informadas na embalagem.

Para as indústrias de alimentos a codificação é fundamental no processo de rastreabilidade dos seus produtos. Através da informação do lote impresso é possível rastrear toda a cadeia produtiva do produto e dessa forma tomar as providências necessárias em casos de problemas relacionados a qualidade que envolvam a recolha do lote ou efetuar investigações de anomalias constatadas por meio do Serviço de Atendimento ao Cliente (SAC). Sendo assim, a inspeção da presença e qualidade da codificação nos produtos é uma medida adotada como parte da rotina da operação nas fábricas de refrigerantes.

A visão computacional é algo presente na indústria e contribui significativamente na gestão da qualidade. Uma das áreas mais beneficiadas pelas novas tecnologias em visão é o setor de controle de qualidade das indústrias. Processos complexos de inspeção de qualidade, antes realizados somente com a

ajuda da visão humana, hoje evoluíram para um controle mais preciso e eficaz, elevando o nível de qualidade de produtos a novos patamares. A aplicação dessa tecnologia propicia a análise de fatores de qualidade com acurácia e imparcialidade, substituindo a avaliação visual do operador, de forma amostral, pela avaliação de toda a produção, caso necessário.

A inspeção da qualidade da rotulagem, do tamponamento e do nível de enchimento de garrafas compostas por *poli tereftalato de etila* (PET) são exemplos de onde são aplicados sistemas de visão computacional na indústria de bebidas. A inspeção da codificação em garrafas PET apresenta algumas complexidades que se relacionam com a qualidade da imagem capturada. A formação de bolhas ou espuma na parede da garrafa, nível do líquido próximo a região de codificação e até mesmo a curvatura cilíndrica da garrafa são exemplos dessas complexidades associadas ao processo de inspeção. Tendo como ponto de partida o que foi explanado, pretende-se pesquisar a utilização de visão computacional para inspeção de maneira automática das informações de lote e data de validade impressas em garrafas de refrigerante PET.

1.1 Tema

Utilização de visão computacional em sistema embarcado para inspeção automática da qualidade e exatidão da impressão da codificação da data de validade em garrafas PET em uma linha de produção de refrigerantes.

1.2 Delimitação do tema

A pesquisa realizada neste trabalho limita-se avaliação e reconhecimento de imagens capturadas em garrafas PET de uma linha de produção de refrigerantes, na qual a impressão da codificação ocorre antes do processo de envase do produto. Essa limitação decorre da facilidade da aquisição da imagem sem que ocorra a interferência do líquido ou da espuma por trás da área de impressão. Além disso, o objeto de pesquisa se restringe a utilização de ferramentas de software e hardware livres como Python e suas bibliotecas OpenCV, EasyOCR, Tesseract e Raspberry Pi, respectivamente. Dentre as motivações em utilizar ferramentas de software e

hardware livres estão o baixo custo do hardware e o desenvolvimento e aprendizado colaborativo relacionado aos softwares.

1.3 Definição do problema

A estrutura do processo de fabricação se dá pelo produtor que deseja minimizar o custo de produção e produzir em maior quantidade, e pelo consumidor que deseja receber produtos de alta qualidade também pelo menor custo. A diminuição dos custos associados aos processos de controle de qualidade com o uso de ferramentas tecnológicas se torna fundamental diante deste cenário.

Uma das etapas de maior importância no processo de controle de qualidade de garrafas de refrigerante PET é a da codificação da data de validade e lote. O tempo destinado pela operação da fábrica para realizar as inspeções, bem como a acurácia da avaliação e qualidade e do conteúdo impresso são fatores relevantes para se investir em métodos automatizados de controle de qualidade utilizando visão computacional. Esses conceitos de inspeção auxiliam e substituem a inspeção humana, trazendo vantagens significativas quando a inspeção exige precisão e velocidade, substituindo a inspeção baseada em critérios técnicos e de amostragem, permitindo ao operador o ganho de tempo para atuar em outras atividades em seu posto de trabalho.

1.4 Objetivos

A seguir, serão relatados os objetivos determinados para realização deste trabalho.

1.4.1 Objetivo geral

Este trabalho tem como objetivo investigar a viabilidade técnica da aplicação de um sistema automático de verificação da qualidade da codificação em garrafas PET em uma indústria de refrigerantes utilizando placa de desenvolvimento microcontrolada Raspberry Pi, câmera de baixo custo e software de interface para configuração e visualização dos dados pertinentes ao processo de inspeção, além da interligação das informações coletadas com o banco de dados da empresa.

1.4.2 Objetivos específicos

Para atingir o objetivo geral estabelecido é necessário o cumprimento dos seguintes objetivos específicos:

- a) elaborar um método de processamento em tempo real do sistema;
- b) propor um hardware auxiliar para a iluminação e um mecanismo de fixação dos dispositivos junto ao transportador de garrafas da linha de produção;
- c) averiguar se os dispositivos de hardware utilizados possuem capacidade de suprir a precisão necessária para o processo de inspeção;
- d) definir uma sequência de mecanismos para processamento de imagens digitais satisfatórios para atendimento dos requisitos básicos de reconhecimento óptico de caracteres (OCR) empregado;
- e) definir as ferramentas de OCR mais apropriadas para detecção e reconhecimento de texto;
- f) definir parâmetros de classificação das inspeções da codificação como “APROVADO” ou “REPROVADO”.

1.5 Justificativa

A eficiência do processo de inspeção da qualidade da codificação resulta na diminuição dos custos operacionais e dos custos com retrabalhos ou descartes de produtos inadequados. Uma vez que este processo seja realizado de forma automática e com recursos de visão computacional, é possível que haja uma melhora na assertividade e uma maior abrangência das amostras avaliadas. Por se tratar de uma atividade repetitiva e que exige um excessivo esforço físico pelo funcionário que a realiza, a inspeção manual acarreta problemas tais como a falta de inspeção em todos os produtos, a falta de precisão nas inspeções e a alta rotatividade desses trabalhadores. Além disso, a efetividade da inspeção humana declina significativamente no decorrer da jornada de trabalho.

A utilização de recursos de hardware e de software de código aberto propiciam mais confiabilidade, maior interoperabilidade e liberdade de personalização, além da redução do custo total de propriedade (TCO).

1.6 Organização do trabalho

Este trabalho está estruturado em 3 capítulos, conforme descrição a seguir:

No primeiro capítulo são apresentadas a introdução e a contextualização do problema e dos objetivos. No capítulo 2 é feita uma revisão bibliográfica onde são descritos alguns conceitos relacionados a obtenção e tratamento de imagens digitais, conceitos sobre identificação e reconhecimento de caracteres em imagens. Neste capítulo também são explorados conceitos referentes aos hardwares utilizados para aquisição e processamentos das imagens, bem como uma caracterização e comparações sobre os métodos de reconhecimento óptico de caracteres EasyOCR e Tesseract OCR e visão computacional com OpenCV (*Open Source Computer Vision*). No capítulo 3 é realizada uma descrição acerca dos passos fundamentais para o desenvolvimento do projeto. No capítulo 4 são explorados os métodos de simulação e avaliação das ferramentas utilizadas e dos resultados obtidos. Por fim, no capítulo 5 é realizada uma discussão sobre os resultados obtidos e no capítulo 6 é apresentada a conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Este trabalho trata da pesquisa de um sistema de visão computacional, em que se estudará a aplicação constante e recorrente de uma câmera embarcada em um computador de placa única em plataforma *Raspberry Pi* com arquitetura *Advanced RISC Machine* (ARM). A câmera em questão será o sensor de maior pertinência, pois toda a pesquisa de manipulação, alinhamento, filtragem de ruídos, interpretação dos dados e tomada de decisões serão aplicados e derivados das imagens capturadas por este sensor. Por esse motivo, é fundamental a aquisição de uma base sólida acerca de técnicas conceituadas de visão computacional. Também serão explorados os conceitos relevantes sobre os fundamentos de imagens digitais, métodos de tratamento de imagens digitais, identificação e reconhecimento de texto em imagens e sensores e sistema de aquisição de imagens.

2.1 Definição sobre processamento de imagens digitais

O processamento digital de imagens (PDI) envolve um conjunto de tarefas interconectadas que se inicia com a captura de uma imagem, a qual, normalmente, corresponde à iluminação que é refletida na superfície dos objetos, realizada através de um sistema de aquisição (QUEIROZ; GOMES, 2016).

2.1.1 Fundamentos de imagens digitais

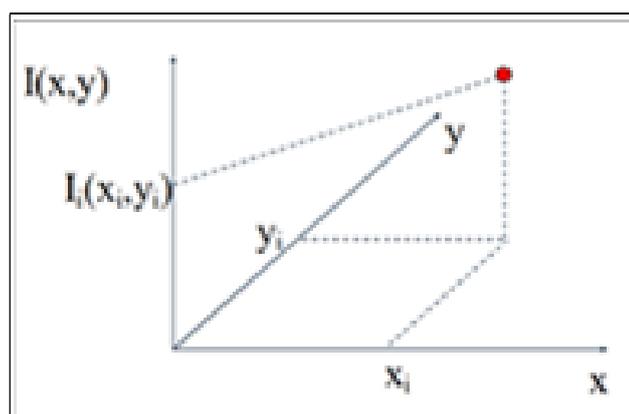
Uma imagem digital pode ser considerada como uma representação discreta de dados que possuem uma dimensão e uma intensidade (PEDRINI; SCHWARTZ, 2008). Conforme Solomon e Brekon (2011), uma imagem digital pode ser definida como uma função bidimensional que representa a resposta de uma matriz de sensores em uma série de posições fixas ($m=1,2,\dots,M; n=1,2,\dots,N$) em coordenadas cartesianas bidimensionais e é derivada do sinal espacial contínuo $I(x,y)$ através de um processo de amostragem regularmente chamado de discretização.

Uma imagem digital difere de uma foto porque os valores x , y e $f(x, y)$ são todos discretos. Normalmente eles assumem apenas valores inteiros, então a imagem mostrada na Figura 1 terá x e y variando de 0 a 255 cada, e os valores de brilho também variam de 0 (preto) a 255 (branco). Uma imagem digital, como vimos

acima, pode ser considerada uma grande variedade de pontos, chamado de pixels, amostrados da imagem contínua, cada um com um brilho quantizado particular. Os pixels ao redor de um dado pixel constituem sua vizinhança. Um bairro pode ser caracterizado por sua forma semelhante a uma matriz: podemos falar, por exemplo, de uma vizinhança 3×3 ou de uma vizinhança 5×7 . Exceto em circunstâncias muito especiais, os bairros têm números ímpares de fileiras e colunas, isso garante que o pixel atual esteja no centro da vizinhança. Se uma vizinhança tiver um número par de linhas ou colunas (ou ambos), pode ser necessário especificar qual pixel na vizinhança é o “pixel atual” (MCANDREW, 2016).

Existem dois modos de geração de imagens digitais. O primeiro é o mapa de bits (ou raster), que é uma matriz de informação que descreve os pontos individuais chamados de pixels. A segunda forma de geração é através de vetores, onde a informação a respeito da imagem é descrita a partir de suas coordenadas no plano cartesiano $f(x, y)$, 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 nesse ponto. Chamamos de imagem digital quando x , y e os valores de intensidade de f são quantidades finitas e discretas (GONZALES; WOODS, 2009). Na Figura 1 podemos observar o gráfico de uma função bidimensional.

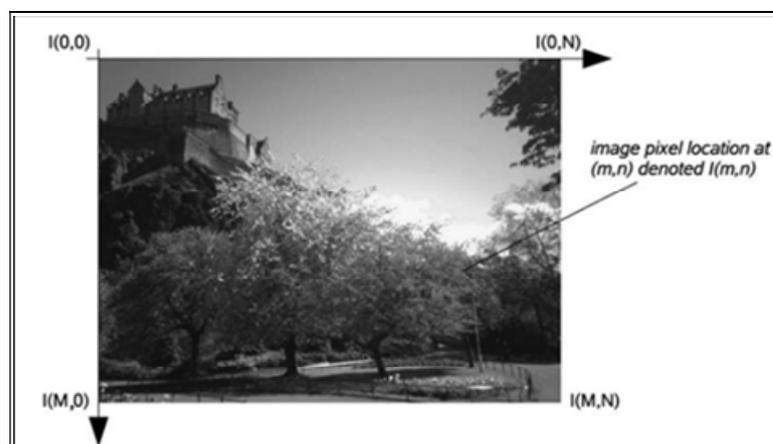
Figura 1 - Espaço de coordenadas cartesianas



Fonte: Solomon e Breckon (2011)

O brilho refletido em pontos $(x, y) \in \mathfrak{R}^2$ de uma dada região do espaço (cena) pode ser amostrado por um arranjo de sensores com espaçamentos (dx, dy) (e.g., $dx = dy = 1mm$) e quantizado em valores inteiros $I(x, y)$, formando uma imagem cinza, conforme ilustrado na Figura 2.

Figura 2 - Espaço de coordenadas cartesianas bidimensionais de uma imagem $M \times N$



Fonte: Solomon e Breckon (2011)

2.1.2 Processamento digital de imagens

O processamento de imagem envolve alterar a natureza de uma imagem com o objetivo de melhorar suas informações pictóricas para interpretação humana e para torná-la mais adequada para percepção de máquina autônoma. Não se trata de uma tarefa simples, pois está relacionada a um conjunto de tarefas interconectadas. O processo inicia com a captura da imagem, que normalmente corresponde à iluminação que é refletida na superfície dos objetos, por um sistema de aquisição (MCANDREW, 2016).

A área do processamento digital de imagens vem evoluindo ao longo dos anos, com um aumento significativo de estudos que envolvem morfologia matemática, processamento de imagens coloridas, redes neurais, reconhecimento de imagens e sistemas de análise baseado em conhecimento discretas (GONZALES; WOODS, 2009).

O processamento de imagens por um computador digital conceitua o campo do processamento digital de imagens. Uma imagem digital é composta de um número finito de elementos, cada um com localização e valor específicos. Tais elementos são chamados de *pixels*, que é o termo mais utilizado para representar os elementos de uma imagem digital. Os primeiros computadores poderosos o suficiente para realizar tarefas de processamento de imagens significativas foram

desenvolvidos no início da década de 1960. Deste período até os dias de hoje, a área de processamento de imagens cresceu rapidamente. Além das aplicações na medicina e em programas espaciais, as técnicas de processamento digital de imagens, atualmente, são utilizadas em uma ampla variedade de aplicações. Procedimentos computacionais são utilizados para realçar o contraste ou para codificar os níveis de intensidade em cores para facilitar a interpretação de imagens radiográficas e outras imagens utilizadas na indústria, na medicina e nas ciências biológicas. Uma das principais áreas de aplicação do processamento de imagens digitais é a resolução de problemas referentes à percepção por máquinas, onde o interesse se concentra em procedimentos para extrair informação de uma imagem de uma forma adequada para o processamento computacional. O reconhecimento automático de caracteres costuma utilizar técnicas de processamento de imagens (GONZALES; WOODS, 2009).

2.1.3 Alguns passos fundamentais para o processamento digital de imagens

Após a aquisição por um processo de digitalização a imagem precisa estar representada em uma forma adequada para ser analisada por um sistema computacional. O primeiro passo satisfatório do processamento é chamado de pré-processamento, como por exemplo o redimensionamento da imagem, a filtragem de ruídos e correções geométricas causados pelo sensor. Posteriormente, Gonzales e Woods (2009), cita o processo de realce de imagens como passo fundamental do processamento digital de imagens. Este processo visa manipular uma imagem de forma que o resultado seja mais adequado que o original para uma aplicação específica. A restauração da imagem também é uma área que lida com a melhora visual de uma imagem e se baseia em modelos matemáticos ou probabilísticos de degradação de imagens. A compressão de imagens lida com as técnicas de redução do armazenamento necessário para salvar uma imagem. O processamento morfológico tem relação com ferramentas para a extração de componentes da imagem úteis na representação e descrição da forma. Os procedimentos de segmentação dividem uma imagem em suas partes ou nos objetos que a constituem. O procedimento de segmentação é um processo crítico que, se, realizado da forma correta, aumenta as chances de resolução de problemas que envolvam o reconhecimento de objetos. Já o processo de descrição, também

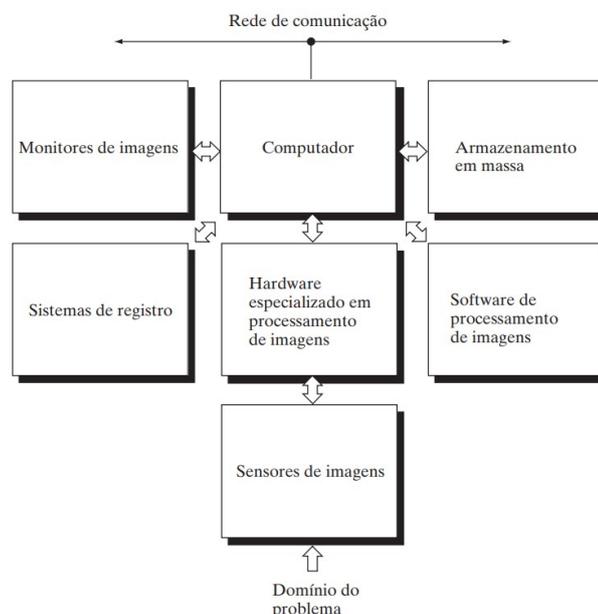
chamado de seleção de características, lida com a extração de atributos que resultam em alguma informação quantitativa de interesse ou que possam ser utilizados para diferenciar uma classe de objetos de outra. Por fim, o reconhecimento é o processo que atribui um rótulo a um objeto com base em seus atributos.

É conveniente subdividir diferentes algoritmos de processamento de imagem em subclasses amplas. Há algoritmos diferentes para tarefas e problemas diferentes, e muitas vezes gostaríamos de distinguir a natureza da tarefa do processamento da imagem. Melhorar a imagem se refere ao processamento de uma imagem para que o resultado seja mais adequado para uma determinada aplicação. Exemplos desse tratamento incluem alterar a nitidez ou desfoque de uma imagem fora de foco, o destaque de bordas, melhorar o contraste da imagem ou clarear a mesma e a remoção de ruído. Além disso, a restauração, segmentação e registro também são fundamentais no tratamento (MCANDREW, 2016).

2.1.4 Componentes de um sistema de processamento de imagens

Conforme Gonzales e Woods (2009), apesar de os sistemas de processamento de imagens em grande escala ainda serem comercializados para grandes aplicações envolvendo imagens, como por exemplo o processamento de imagens de satélite, a tendência continua sendo a miniaturização e a combinação de pequenos computadores de uso geral com hardware para processamento de imagens. A Figura 3 mostra os componentes básicos que constituem um sistema de uso geral típico para o processamento digital de imagens. A função de cada componente é brevemente descrita nos tópicos a seguir, começando com o sensoriamento de imagens.

Figura 3 - Componentes de um sistema de processamento de imagens de uso geral



Fonte: Adaptada de Gonzales e Woods (2009)

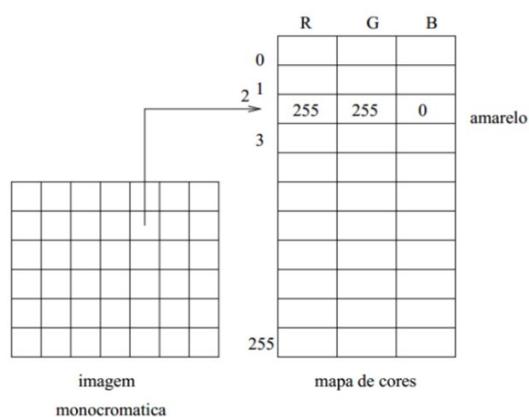
- Sensoriamento: No que se refere ao sensoriamento, são necessários dois elementos para a aquisição de imagens digitais. O primeiro é um dispositivo físico sensível a energia irradiada pela imagem a ser capturada. O segundo, chamado digitalizador, é um dispositivo utilizado para converter a saída do dispositivo físico de sensoriamento em formato de dados digitais;
- Hardware especializado em processamento de imagens: Normalmente consiste no digitalizador mencionado anteriormente, além de um hardware capaz de desempenhar outras operações primárias, como uma unidade lógica de aritmética (ULA), que realiza operações aritméticas e lógicas em paralelo em toda a imagem;
- Computador: Equipamento necessário para o processamento de imagens que pode ser um computador pessoal ou supercomputadores. O software para o processamento de imagens consiste em módulos especializados que realizam tarefas específicas;
- Capacidade de armazenamento em massa: Indispensável em aplicações de processamento de imagens. Responsável pelo armazenamento dos arquivos de imagens. O armazenamento digital para aplicações de processamento de imagens se divide basicamente em armazenamento de curto prazo para utilização durante o processamento, armazenamento on-line para acesso relativamente rápido e armazenamento em arquivo para acesso com pouca frequência;

- Monitores de imagem: Os monitores são controlados pelas placas de vídeo (gráficas ou de imagens), que são parte integral de um sistema computacional;
- Sistemas de registro: Incluem impressoras a laser, filmes fotográficos, impressora térmica e mídias digitais;
- Rede de comunicação: É um componente padrão de qualquer sistema de computacional hoje em dia. Em razão do grande volume de dados inerente às aplicações de processamento de imagens, a principal preocupação na transmissão de imagens é a largura de banda.

2.1.5 Imagens multibanda ou multiespectrais

Objetos que emitem luz visível são percebidos em função da soma das cores espectrais emitidas. Uma imagem tridimensional pode ser representada como uma sequência de imagens monocromáticas ou multibandas, tal que cada imagem é conhecida como banda. Neste caso, o valor de cinza de cada pixel na imagem é um índice para uma entrada no mapa de cores, sendo que a entrada do mapa de cores contém o valor das componentes vermelha (R, red), verde (G, green), e azul (B, blue), referente a cor do pixel (PEDRINI; SCHWARTZ, 2008). A Figura 4 representa a relação entre cada pixel de uma imagem digital e suas bandas de cores RGB.

Figura 4 - Mapa de cores da representação de uma imagem colorida



Fonte: Adaptada de Pedrini e Schwartz (2008)

A combinação das cores vermelho, verde e azul, duas a duas e em igual intensidade, produz as cores secundárias Ciano, Magenta e Amarelo.

2.2 Conceitos sobre métodos de tratamento de imagens digitais

Em problemas de classificação e reconhecimento de padrões, uma ferramenta fundamental é a extração de características. Dessa forma, torna-se ainda mais importante o conhecimento e aplicação de técnicas de tratamento de imagens digitais para atingir objetivos específicos. A seguir serão mostrados alguns conceitos fundamentais pertinentes a aplicação dessas técnicas neste projeto de pesquisa.

2.2.1 Kernel ou Máscara

Em processamento de imagens são necessárias operações de filtragem a fim de extrair informações de interesse na imagem. Essas operações podem ser realizadas tanto no domínio do espaço quanto no domínio da frequência. A filtragem no domínio espacial, isto é, aquele realizado diretamente sobre o conjunto de pixels que compõe a imagem, geralmente é realizada por meio de matrizes denominadas “máscaras”. A máscara mais simples tem tamanho 3x3, e a cada posição da máscara está associado um valor numérico, chamado de peso ou coeficiente, conforme Figura 5.

Figura 5 - Máscara 3x3 pixels com coeficientes arbitrários

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fonte: Adaptada de Pedrini e Schwartz (2008)

A aplicação da máscara com centro na coordenada (x, y) consiste na substituição do valor do pixel na posição (x, y) por um novo valor, que depende dos valores dos pixels vizinhos e dos pesos da máscara. Os coeficientes do filtro são multiplicados pelos níveis de cinza dos pixels correspondentes e então somados, substituindo o nível de cinza do pixel central (PEDRINI; SCHWARTZ, 2008), conforme Equação 2.1.

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i \quad (2.1)$$

A máscara é movida para a próxima posição de pixel na imagem, e o processo é repetido até que a máscara passe por todas as posições da imagem. A aplicação deste filtro em cada pixel da imagem é chamada de correlação. Formalmente, a correlação é definida na Equação 2.2. A convolução consiste em um processo similar, porém o filtro deve sofrer uma reflexão, ou rotação de 180 graus antes de ser aplicado à imagem (PEDRINI; SCHWARTZ, 2008).

$$w \cdot f(x, y) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} w(i, j) f(x + i, y + j) \quad (2.2)$$

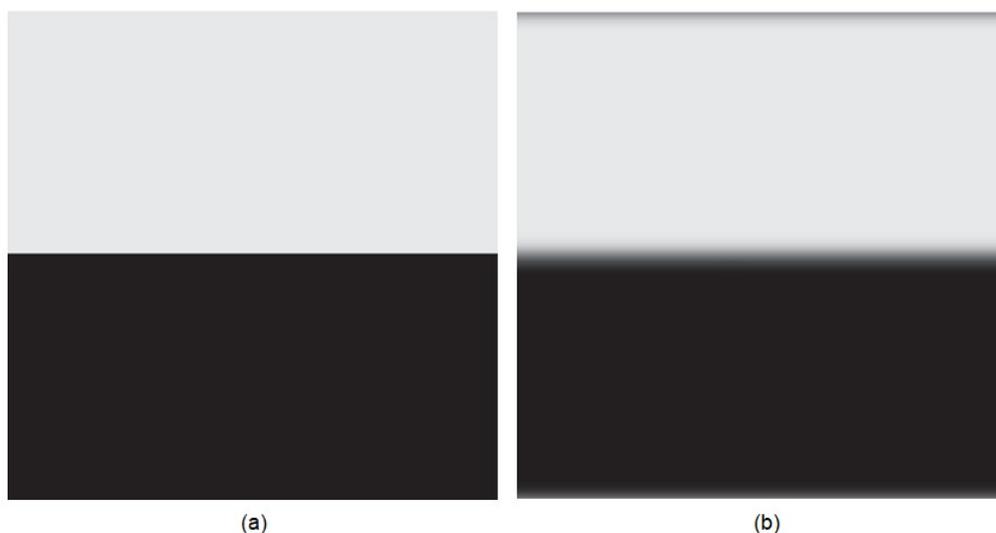
2.2.2 Filtros digitais no domínio espacial

Segundo Pedrini e Schwartz (2008), o domínio espacial faz referência ao plano da imagem, em outras palavras, ao conjunto de pixels que compõe uma imagem. Conhecidos como operadores locais ou filtros locais, os filtros no domínio espacial são técnicas baseadas na convolução matrizes ou tuplas (conjunto de pixels). Uma grande variedade de filtros pode ser implementada no domínio do espaço. Convolução é o processo de calcular a intensidade de um determinado pixel em função da intensidade de seus vizinhos. O cálculo é baseado em ponderação, isto é, utilizam-se pesos diferentes para pixels vizinhos diferentes (GONZALES; WOODS, 2009).

2.2.3 Filtros passa-baixas

O efeito do filtro passa-baixa é o de suavização da imagem, visto que as frequências elevadas que correspondem às transições abruptas na imagem são atenuadas. Além disso, o filtro passa-baixa tende a reduzir os efeitos do ruído em imagens, entretanto, devido ao borramento ou embassamento causado pela filtragem passa-baixa, detalhes finos podem ser removidos da imagem (GONZALES; WOODS, 2009). Na Figura 6 podemos observar o efeito da aplicação do filtro passa-baixas.

Figura 6 - (a) Imagem original. (b) Imagem com filtro passa-baixas



Fonte: Adaptada de Gonçalves e Woods (2010)

Ao borrar a imagem original, o filtro passa-baixa gera uma estimativa irregular da imagem do fundo. Os filtros Gaussianos são um tipo de filtro passa-baixa, linear no domínio espacial (SOLOMON; BREKON, 2011).

2.2.4 Filtro média

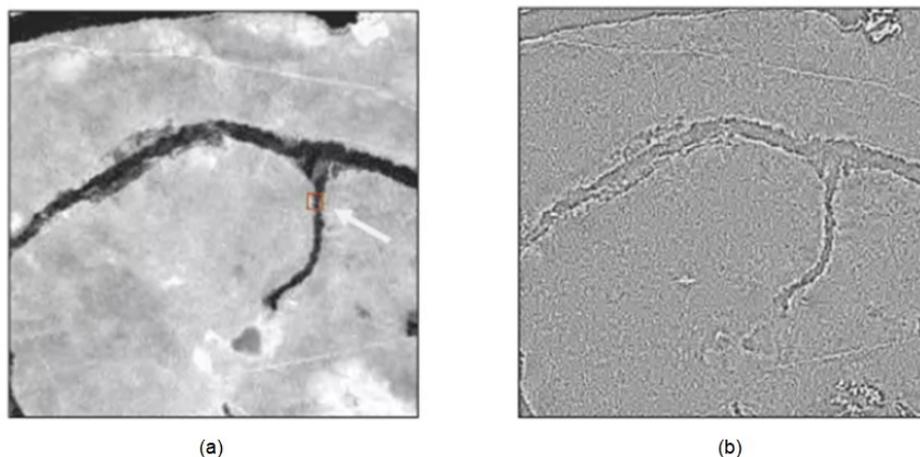
Os filtros de média podem ser usados como um método para suprimir o ruído em uma imagem, embora o filtro de mediana geralmente execute essa função com melhor qualidade. É usado normalmente na etapa de processamento preliminar para suavizar a imagem para que algumas operações de processamento se tornem mais eficazes. A redução do ruído sem prejuízo a qualidade da imagem depende do tamanho do kernel utilizado, ou seja, kernels maiores suprimirão mais ruído, porém, resultarão maior degradação da qualidade da imagem (SOLOMON; BREKON, 2011).

2.2.5 Filtro passa-altas

Os filtros passa-altas podem ser usados para realçar certas características presentes nas imagens, tais como bordas, linhas e regiões de interesse, produzindo uma "agudização" ("*sharpering*") da imagem, isto é, as transições entre regiões diferentes tornam-se mais nítidas. Estes filtros podem ser usados para realçar certas

características presentes na imagem, tais como bordas, linhas curvas ou manchas, mas enfatizam o ruído existente na imagem (PEDRINI; SCHWARTZ, 2008). Podemos observar um exemplo do processo de filtragem passa-alta na Figura 7.

Figura 7 - (a) Imagem original. (b) Imagem com filtro passa-altas



Fonte: Adaptada de Gonzales e Woods (2009)

2.3 Conceitos sobre identificação e reconhecimento de textos em imagens digitais

A seguir serão apresentados alguns conceitos referentes ao reconhecimento de caracteres em imagens digitais, bem como alguns de seus processos e aplicações.

2.3.1 Identificação de texto em imagens digitais

O reconhecimento óptico de caracteres, ou OCR, é usado para descrever algoritmos e técnicas (eletrônicas e mecânicas) para conversão de textos em imagens digitais para linguagem de máquina. Normalmente pensamos em OCR em termos de software. Ou seja, são sistemas que primeiramente tem um uma imagem de entrada (digitalizada, fotografada ou gerada por computador), detecção automática do texto e “leitura” como um humano faria e, por fim, a conversão do texto em um formato legível por máquina para que possa ser pesquisado, indexado e processado dentro do escopo de um sistema de visão computacional maior (ROSEBROCK, 2020).

A identificação de texto em imagens tipicamente envolve duas etapas. A primeira delas é detectar a região onde está contido o texto, obtendo as coordenadas da imagem que contém o texto, comumente chamadas de *bounding box*. A segunda etapa consiste em reconhecer os caracteres desta região, e para isso podem ser utilizados algoritmos de OCR (SANTIAGO, 2017).

2.3.2 Aplicações de reconhecimento óptico de caracteres

Podemos destacar diversas aplicações do OCR, cuja original era criar máquinas de leitura para os cegos. As aplicações de OCR evoluíram significativamente desde então, incluindo:

- a) reconhecimento automático de placas de veículos (ALPR);
- b) reconhecimento de sinais de trânsito;
- c) analisar CAPTCHAs (Teste de Turing público completamente automatizado para distinguir entre computadores e pessoas) em sites;
- d) extração de informações de cartões de visita;
- e) leitura automática da zona legível por máquina (MRZ) e outras partes relevantes de um passaporte;
- f) análise do número de roteamento, do número da conta e do valor da moeda de um cheque bancário;
- g) compreender o texto em cenas naturais, como as fotos capturadas de telefones celulares.

Se houver texto em uma imagem digital, provavelmente será possível aplicar o OCR para identificação e reconhecimento do texto, desde que sejam utilizadas as técnicas corretas (ROSEBROCK, 2020).

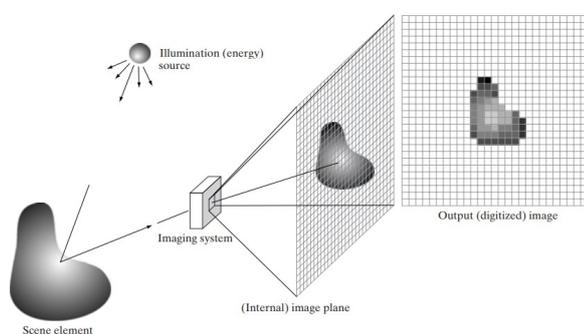
2.4 Processos de aquisição de imagens

Nesta seção serão apresentados conceitos referentes ao processo de aquisição de imagens digitais como digitalização, sensores, influência do obturador e da iluminação na obtenção de imagens.

2.4.1 Sensores e sistemas de aquisição de imagens

A maioria das imagens nas quais estamos interessados é gerada pela combinação de uma fonte de iluminação e a reflexão ou absorção de energia dessa fonte pelos elementos da cena cuja imagem está sendo gerada. Um sensor de imagem digital, agindo como a retina dos olhos, capta a luminosidade das imagens que são projetadas sobre ele continuamente e dá início ao processo de captura de uma instância ou de uma sequência de instâncias da imagem consecutivamente. O arranjo de sensores, situado no plano focal, produz uma tensão elétrica proporcional à integral da energia recebida em cada sensor. Circuitos digitais e analógicos convertem essa tensão em um sinal analógico, que então é digitalizado por outra parte do sistema de aquisição de imagem. O resultado é a aquisição de uma imagem digital (GONZALES; WOODS, 2009). A Figura 8 mostra os componentes de um único sensor de aquisição de imagens.

Figura 8 - Exemplo do processo de aquisição de imagens



Fonte: Adaptada de Gonzales e Woods (2009)

Um sensor de imagem *Charge Coupled Device* (CCD) pode conter dezenas de milhões de transdutores fotossensíveis que convertem a energia luminosa em sinais de tensão. Câmeras digitais tipicamente utilizam arranjos destes transdutores em formato de uma matriz 2D.

2.4.2 Obturador global e de rolamento

O obturador é um mecanismo usado pela câmera para controlar o tempo de exposição à luz para capturar a imagem. É uma parte importante da câmera, e sua estrutura, forma e função são um fator importante para medir a qualidade da

câmera. O obturador global expõe a cena inteira ao mesmo tempo. Todos os pixels do sensor são expostos a luz ao mesmo tempo. O obturador de rolamento, ao contrário do obturador global, é conseguido pelo método progressivo da exposição do sensor. No início da exposição, o sensor faz a varredura linha a linha executando a exposição até que todos os pixels estejam expostos (COATES; JUVAN, 2020).

O método de captura através de obturador global consiste em permitir a exposição à luz, armazenando a energia de cada sensor no seu respectivo acumulador, e, após o tempo determinado, interromper o acúmulo de energia em todos os sensores simultaneamente. Este processo pode ser encontrado em sensores do tipo CCD (*Charge Coupled Device*), e em alguns sensores do tipo CMOS (*Complementary Metal-Oxide Semiconductor*). Neste método, o processo de interrupção do acúmulo de energia nos sensores não é feito simultaneamente, e sim, linha a linha do arranjo. Esta forma de leitura das cargas acumuladas dos sensores pode causar o chamado efeito *rolling shutter*, caso o objeto capturado esteja em movimento e a velocidade do obturador não seja pequena o suficiente para assegurar o congelamento da imagem (COATES; JUVAN, 2020). A Figura 9 ilustra um exemplo do efeito *rolling shutter*.

Figura 9 - Exemplo de aquisição de imagens digitais com obturador global e de rolamento, respectivamente



Fonte: Taylor Kavanaugh (2022)

Segundo o estudo realizado por Silva (2020), a velocidade do obturador causa um escurecimento nas imagens adquiridas, à medida que o tempo para captação dos fótons é menor. Também, velocidades menores contribuem para a estabilização da imagem, minimizando tremidos e borrões.

2.5 Mecanismos de Hardware

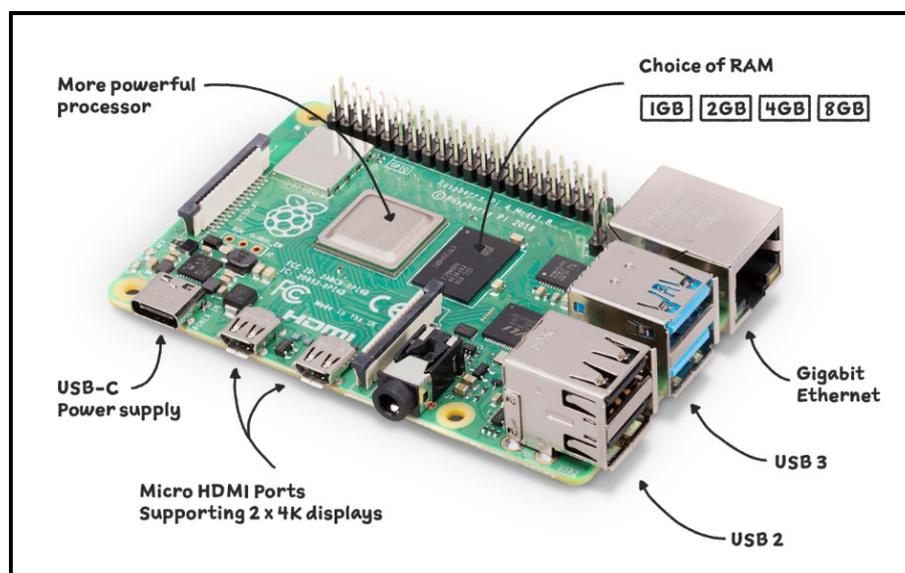
Para os experimentos realizados no presente trabalho serão utilizados hardwares de baixo custo, normalmente utilizados para prototipagem. A seguir será feita uma breve revisão sobre os conceitos e características técnicas dos dispositivos utilizados.

2.5.1 Raspberry Pi Modelo 4B

Pi é um computador de que possui tamanho equivalente ao de um cartão de crédito. Começou a ser desenvolvido em 2012 por uma Fundação sem fins lucrativos chamada Raspberry Pi, situada no Reino Unido, que visava promover o ensino da computação básica em escolas, além da inclusão e empoderamento social dos alunos (RASPBERRY PI FOUNDATION, 2022).

O modelo mais atual, o Raspberry Pi 4 modelo B é o primeiro de uma nova geração de computadores Raspberry Pi que suportam até 8GB de memória RAM e que possui desempenho de CPU e GPU aprimorados. Sua escolha para o desenvolvimento deste trabalho é justificada pelo baixo custo, por ser *open-source* e por possuir, além de outras interfaces, entrada específica para câmera digital e para display touchscreen e pinos de entrada e saída de uso geral (RASPBERRY PI FOUNDATION, 2022).

Figura 10 - Raspberry Pi 4 Modelo B



Fonte: Raspberry Pi Foundation (2022)

Um resumo das principais especificações técnicas do Raspberry Pi 4B são apresentadas no Quadro 1.

Quadro 1 - Especificações técnicas do Raspberry Pi 4 Modelo B

Processador Broadcom 2711 Quad-core Cortex-A72 64-bit SoC @ 1,5 GHz
8GB de memória RAM
WiFi 2,4 GHz / 5,0 GHz IEEE 802.11.b/g/n/ac
Bluetooth 5.0
2 portas USB 2.0
2 portas USB 3.0
True Gigabit Ethernet over USB 3.0
GPIO com 40 pinos
2 portas micro HDMI, vídeo de 4k
Interface para display (DSI)
Interface para câmera (CSI)
Conector P2 para saída de áudio e vídeo
Slot para cartão micro SD
Alimentação 5V, 3A via conector USB tipo C

Fonte: Adaptado de Raspberry Pi Foundation (2022)

2.5.2 Câmera Raspberry Pi OV5647

OV5647 é um módulo de câmera com ângulo de visão mais amplo e de baixo custo, compatível com a versão computador Raspberry Pi 4 B. Possui em sensor com resolução de 5 megapixels, ângulo de abertura de 75° e lente de 3.6mm. É conectada ao Raspberry Pi através da interface CSI via cabo *flat* (TECHNOLOGIES, 2009).

Figura 11 - Câmera OV5647



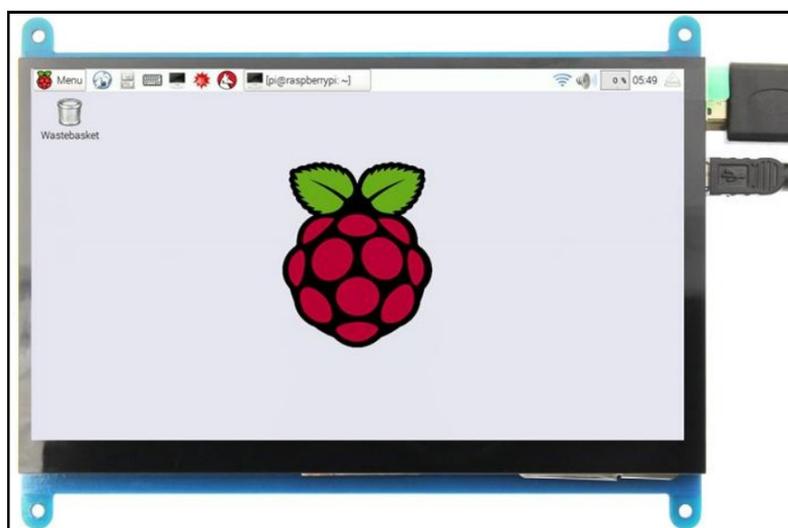
Fonte: Adaptada de Technologies (2009)

Para utilizar a câmera OV5467 no Raspberry é necessário realizar a habilitação desta através das configurações de hardware do Raspberry Pi. Comandos da câmera via python são possíveis com a instalação da biblioteca picamera. Essa biblioteca permite executar a captura de fotos e vídeos, ajustar o contraste de -100 a 100, o brilho de 0 a 100, a velocidade do obturador de 0 (automático) até 20000 ms, além de outras funcionalidades.

2.5.3 Monitor de tela sensível ao toque

O monitor de tela sensível ao toque de 7" para Raspberry Pi possibilita a prototipação sem a necessidade de utilização de teclado ou mouse. Permite a interface do usuário com o projeto embarcado. Possui uma tela com resolução de 1024 x 600 que é conectada através da porta HDMI e recebe alimentação de 5V pela porta USB, com consumo de 0,62A. Possui dimensões de 19,2 x 11 x 2cm, com tela visível de 15,5 x 8,6cm. O *touch* é de toque capacitivo de até 5 pontos. (RASPBERRY PI FOUNDATION, 2022).

Figura 12 - Tela sensível ao toque



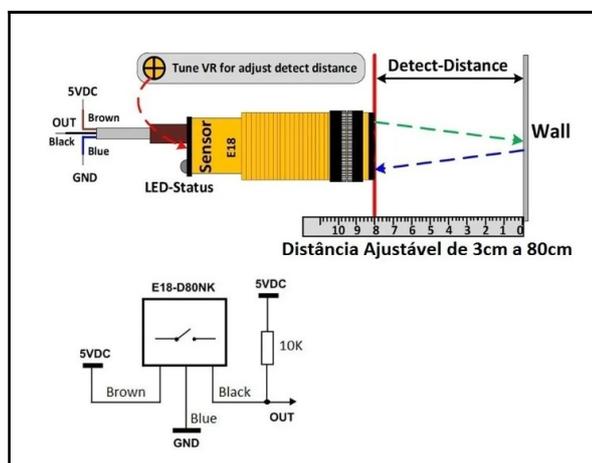
Fonte: Raspberry Pi Foundation (2022)

2.5.4 Sensor infravermelho reflexivo

O E18-d80nk é um sensor óptico reflexivo difuso que possui emissor e receptor infravermelho que possibilita a detecção de objetos a sua frente. Possui

ajuste da distância de detecção que varia de 3 a 80 cm, além de led indicador de status do sinal de saída (DATASHEETSPDF, 2022).

Figura 13 - Sensor Infravermelho reflexivo



Fonte: Adaptada de Datasheetpdf (2022)

Os sensores difusos possuem o componente emissor da luz e o componente receptor da luz em um mesmo corpo (carcaça). Este tipo de fotocélula depende somente da propriedade reflexiva da superfície do material ou objeto em que irá detectar. No sensor do tipo retro reflexivo os componentes emissor e receptor também ficam na mesma carcaça, porém, a fotocélula depende exclusivamente de um espelho prismático para a reflexão da luz emitida (BALLUFF, 2019).

2.6 Mecanismos de software

Com o propósito de tornar o projeto economicamente viável foram utilizadas ferramentas *open source* para o desenvolvimento dos programas. Além do cunho econômico, o grande número de publicações a respeito da utilização dessas ferramentas foi fator fundamental para a escolha. A linguagem Python será utilizada em conjunto com o editor de código Visual Studio Code. A seguir, é feita uma revisão bibliográfica acerca dos conceitos básicos relacionados as ferramentas de software utilizados no projeto.

2.6.1 Debian

O Debian é um sistema operacional de software livre e de código aberto baseado no sistema operacional Linux e desenvolvido para uma ampla variedade de dispositivos, incluindo laptops, desktops e servidores. É fornecido com uma configuração padrão para cada pacote, tendo atualizações de segurança regulares durante a vida útil dos pacotes. O Debian oferece suporte a uma longa lista de arquiteturas de CPU, incluindo amd64, i386, e várias versões dos processadores ARM e MIPS, POWER7, POWER8, IBM System z e RISC-V (DEBIAN, 2022).

2.6.2 Biblioteca OpenCV

Desenvolvida inicialmente pela Intel, o OpenCV (*Open Source Computer Vision*) é uma biblioteca que atende diversas plataformas de software e que possui um intenso cerne para o processamento digital de imagens em tempo real. Sua primeira versão foi lançada em 2000 e até hoje suas funcionalidades vêm sendo desenvolvidas pela comunidade científica. A biblioteca OpenCV conta com várias centenas de algoritmos de visão computacional, o que a torna uma ferramenta poderosa no campo da visão computacional (BAGGIO, 2015). Possui código aberto escrito em C e C++ que roda em Linux, Windows e Mac OS X, fornecendo blocos de construção para experimentos e aplicações de visão computacional, com interfaces de alto nível para capturar, processar e apresentar dados de imagem. Há desenvolvimento ativo em interfaces para Python, Ruby, Matlab e outras linguagens. Um dos objetivos do OpenCV é fornecer uma infraestrutura de visão computacional de uso relativamente simples, para auxiliar as pessoas na criação de aplicativos de visão bastante sofisticados de forma eficiente. Contém mais de 500 funções que abrangem muitas áreas de visão, inspeção de produto, imagens médicas, segurança, interface do usuário, calibração de câmera, visão e robótica (BRADSKI; KAEHLER, 2008)

2.6.3 Biblioteca Tesseract

O mecanismo de OCR Tesseract foi desenvolvido na HP entre os anos de 1984 e 1994, porém, foi lançado para código aberto ao final de 2005. A partir de

2006 passou a ser desenvolvido pela empresa Google. Possui suporte para UTF-8 (*8-bit Unicode Transformation Format*) e pode reconhecer mais de 100 idiomas. Suporta vários formatos de imagem, incluindo PNG, JPEG e TIFF. A ferramenta pode ser treinada para reconhecer outros idiomas (TESSERACT, 2022).

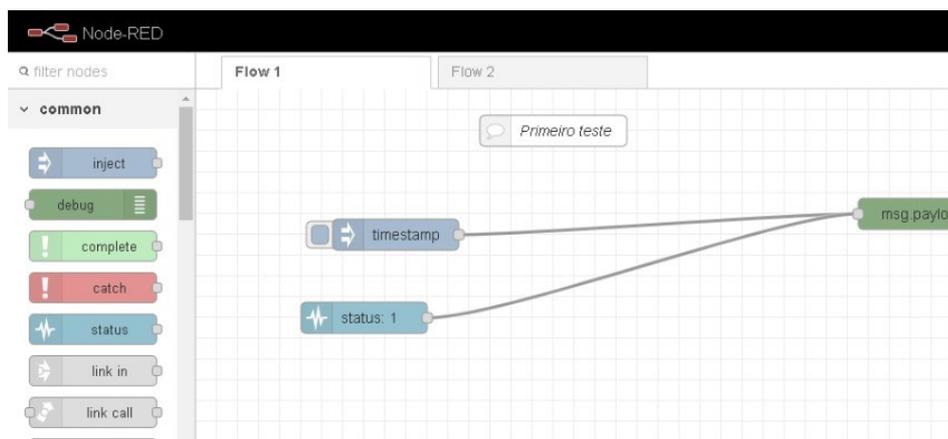
2.6.4 Biblioteca EasyOCR

EasyOCR é uma biblioteca OCR de código aberto para python pertencente a empresa Jaided AI, que serve para extrair textos de imagens. É um OCR geral que pode ler texto de cena natural e texto denso em documentos. Atualmente, oferece suporte a mais de 80 idiomas. Possui mecanismos para treinar o reconhecimento de modelos personalizados e novos idiomas (JAIDED AI, 2022).

2.6.5 Node-Red

O Node-Red é uma ferramenta de programação baseada no fluxo de dados desenvolvida pela equipe de Serviços de Tecnologia Emergente da IBM, que tem como propósito principal a conexão de dispositivos de hardware, APIs e serviços online. Consiste em um editor baseado em um navegador web, com um ambiente visual acessível. Utiliza os chamados *nodes*, que são blocos de códigos com propósito definido, que recebe os dados, processa essas informações e passa essas informações para o próximo nó, onde ocorre um novo processamento. A rede é a responsável pelo fluxo de dados entre os nós. O Node-Red é baseado no software Node.js, para o qual é apontado um navegador web para acessar o editor de fluxo. Neste navegador, é possível criar as aplicações arrastando os nós disponíveis para um espaço de trabalho, onde esses nós podem ser conectados (OPENJS FOUNDATION, 2022).

Figura 14 - Interface de programação do Node-Red



Fonte: Autor

Posicionados na coluna da esquerda da Figura 14 temos os *nodes* (nós) que ao serem arrastados para o centro da área de trabalho poderão ser configurados conectados em outros nós. Na coluna da esquerda temos diversas ferramentas como *deploy*, *debug*, edição, instalação de novos nós, ajuda e importação e exportação de *flows*, dentre outras funcionalidades. Na próxima seção serão abordados alguns trabalhos relacionados que servirão de base para o tema deste projeto.

2.7 Trabalhos correlatos

Nesta seção serão apresentadas pesquisas que possuem relação com o objetivo deste trabalho, seja pela utilização de ferramentas técnicas semelhantes, seja com o uso de Raspberry Pi e algoritmos de tratamento de imagem e OCR. A partir desta revisão, foram selecionados alguns dos métodos utilizados no desenvolvimento do trabalho, bem como definidos os principais parâmetros de ajuste para os tratamentos de imagem utilizados.

2.7.1 Leitura de caracteres de codificação

Zaafouri, Sayadi e Fnaiech (2015) propõe etapas para a construção de um sistema de visão para reconhecimento dos caracteres da data de validade de produtos industriais, orientado exclusivamente na verificação da data de validade. O

modelo foi testado em produtos variados com diversos formatos. São propostos quatro estágios de processamento: pré-processamento da imagem, segmentação da data de validade, extração das características e a classificação dos caracteres. A etapa de pré-processamento consiste em converter a imagem do espaço de cores RGB para o espaço de escala de cinza. Em seguida, é realizada a correção de ângulo, a operação de limiarização e aplicação da operação morfológica de erosão, responsável por diminuir a espessura dos dígitos, buscando desconectar caracteres. Na etapa seguinte, o *thresholding* é empregado para a binarização da imagem em nível de cinza e assim separar o objeto de interesse do fundo. Após a binarização, o algoritmo morfológico é empregado no módulo de desbaste. O resultado dos módulos de binarização e desbaste do código de validade sem duplicação é mostrado Figura 15.

Figura 15 - (a) Imagem original. (b) Imagem em tom de cinza. (c) Imagem binarizada. (d) Imagem com aplicação de operador morfológico



Fonte: Adaptada de Zaafouri et al (2015).

Santiago (2017), propõe métodos de extração de dígitos de um hidrômetro analógico com o objetivo de extrair os dados de consumo instantâneo e estimar a vazão de água proveniente por meio de OCR utilizando linguagem python. Primeiramente foi realizada a detecção dos dígitos em duas etapas: Na primeira foi realizado o alinhamento horizontal da imagem, utilizando a transformada de Hough para encontrar o visor do hidrômetro, baseando-se em seu formato geométrico. Na segunda, foi realizada a detecção de bordas por Canny, onde foram filtrados os dígitos com base nas dimensões dos contornos e algumas regras específicas de proporção e distâncias. Como componentes do sistema de processamento de imagens foram utilizados um kit Raspberry Pi 2 integrado a uma *webcam*. A Figura 16 ilustra as etapas de detecção de bordas e dos dígitos.

Figura 16 - (a) Aplicação do filtro de Canny. (b) Detecção dos dígitos



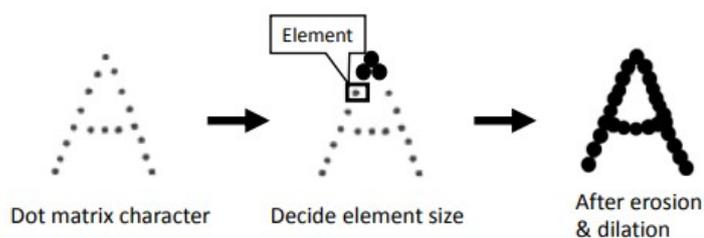
Fonte: Adaptada de Santiago (2018)

Com o objetivo de construir um sistema de gestão para acompanhar a data de validade dos alimentos dentro da geladeira, Hosozawa (2018), desenvolveu uma aplicação que utiliza um telefone celular com câmera para aquisição da imagem correspondente a área de interesse da validade dos produtos. As funções do aplicativo incluem um leitor de código de barras e um leitor capaz de reconhecer a data de validade em uma variedade de embalagens de alimentos. Foram avaliados três OCRs de código aberto: Tesseract-OCR, Nhocr e OCRopus, sendo que o Tesseract-OCR obteve a maior confiabilidade.

Além da influência dos espaços em branco ao redor da região de interesse, que leva a erros na identificação e reconhecimento de caracteres, as influências do nível de brilho e de contraste também foram analisadas, onde foi observado que para valores de contraste e brilho entre -40%, 0% (padrão) e +40% não ocorreram erros de reconhecimento.

Segundo os autores, datas de validade com caracteres impressos em matrizes pontilhadas, ou seja, por codificadores a jato de tinta, não tiveram seus caracteres reconhecidos em nenhuma das amostras. A Figura 17 ilustra este tipo de caractere, semelhante aos impressos para codificação nas garrafas estudadas neste trabalho.

Figura 17 - Processos de tratamento com caracteres pontilhados



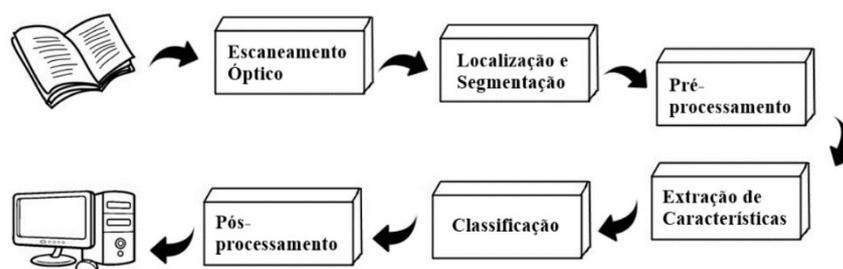
Fonte: Adaptada de Hosozawa (2018)

Da mesma forma, foi constatado que mesmo com uma pequena quantidade de luz refletida na superfície dos objetos, os caracteres não foram reconhecidos.

2.7.2 Comparação entre algoritmos OCR

Cunha (2018) contextualiza a história do desenvolvimento de algoritmos de OCR, desde o surgimento da tecnologia, mudanças nos processos, principais desafios enfrentados, até chegar à forma como era utilizado a época. Apoiando-se principalmente em redes neurais a autora realizou um estudo comparativo da qualidade em função do tempo de diversas ferramentas de OCR, para servir de apoio à tomada de decisão de qual solução aplicar a um determinado negócio. O estudo explica que um sistema de OCR tipicamente aplica as seguintes técnicas descritas no esquema da Figura 18.

Figura 18 - Técnicas de um sistema OCR



Fonte: Adaptada de Cunha (2018)

No estudo foram avaliadas 18 ferramentas através de testes com 108 imagens, a fim de verificar a acurácia e eficiência para diferentes tipos de imagens de documentos. Os resultados demonstraram que as ferramentas OCR Web Service e Tesseract apresentaram os melhores resultados.

2.7.3 Aplicação de sistemas de inspeção com Raspberry Pi

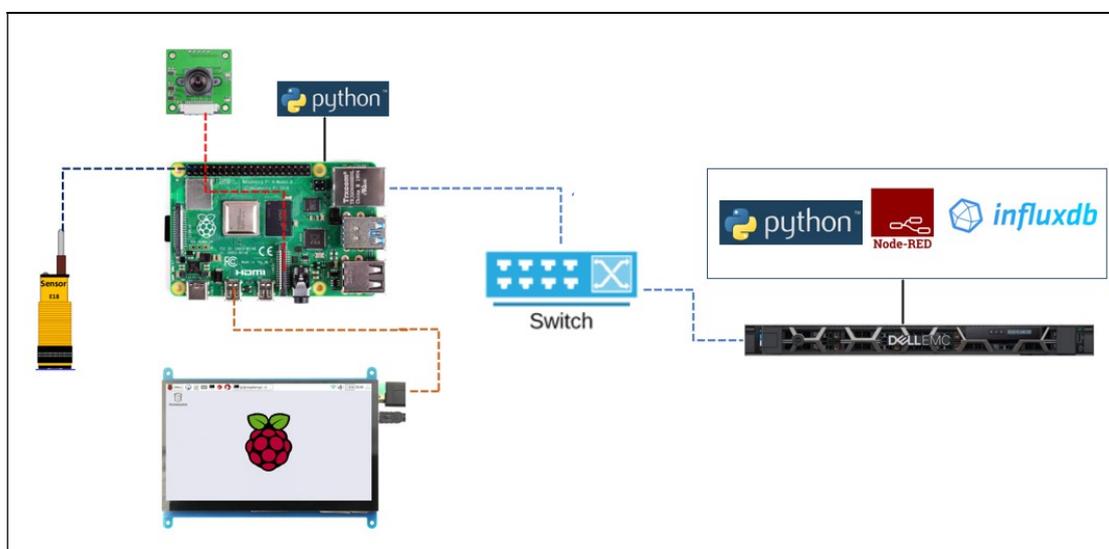
Segundo Machado (2019), o aumento da capacidade produtiva, bem como a ocorrência de mercados consumidores mais exigentes, tem impulsionado o uso de sistemas de inspeção visual automática destinados ao controle de qualidade. Por se tratar de sistemas com custo elevado e inviáveis a empresas de pequeno porte foi desenvolvido e avaliado um sistema de inspeção visual automática de frascos

utilizando dispositivos de baixo custo, visando a detecção de presença de tampa em dois modelos de frascos diferentes, através de imagens capturadas em uma linha de envase a velocidade nominal. O hardware utilizado foi um Raspberry Pi, onde obteve-se uma acurácia de 100% na inspeção de frascos brancos, com tempo de máximo de inspeção de 311,9 ms. Para os frascos verdes a acurácia também alcançou 100%, entretanto, o tempo de inspeção máximo foi de 460,3 ms, ultrapassando o limite tolerável para a linha de produção. Dessa forma, foi observado que imagens com menor variação de cor e intensidade entre as regiões de interesse demandam algoritmos de processamento mais robustos e, como consequência, ocorre o aumento do tempo de inspeção.

3 DESENVOLVIMENTO DO SISTEMA DE AQUISIÇÃO, TRATAMENTO DE IMAGENS E OCR

Diferentes técnicas e metodologias podem ser empregados para a resolução de determinado problema. Neste capítulo, demonstra-se a aplicação prática do trabalho, onde serão apresentadas as informações referentes a metodologia utilizada para o desenvolvimento do projeto de pesquisa, incluindo o *hardware*, o *software*, as técnicas de processamento de imagens, ferramentas de OCR, dentre outras. A Figura 19 mostra uma visão geral da estrutura do sistema.

Figura 19 - Visão geral do sistema de aquisição, processamento de imagens e OCR



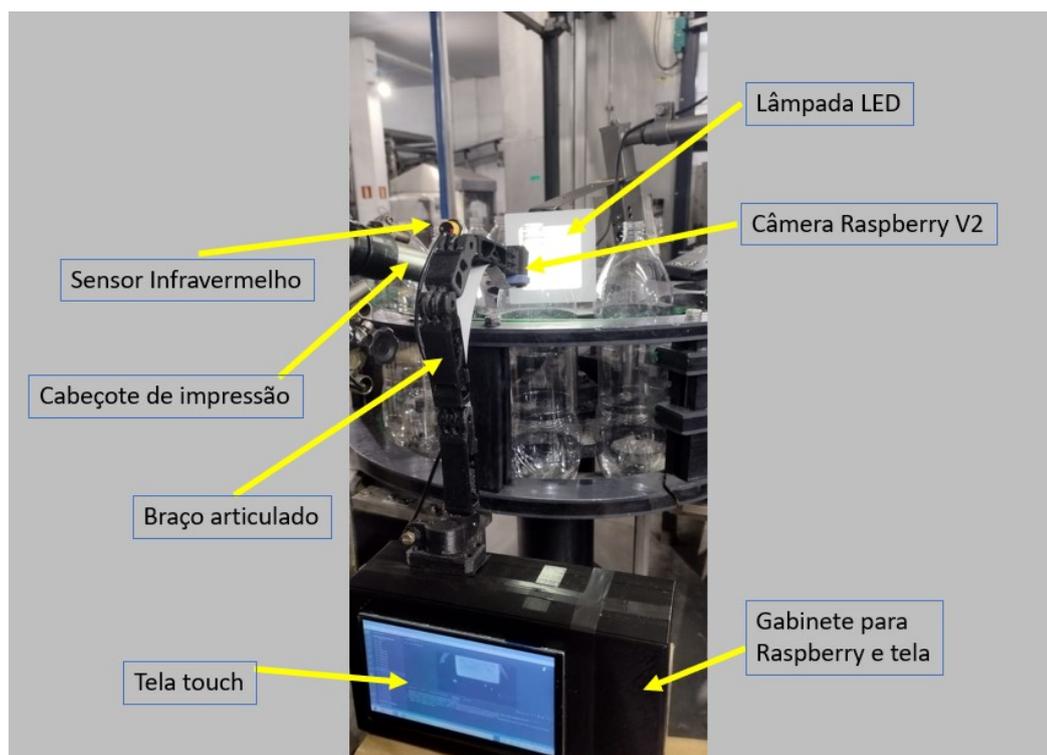
Fonte: Autor

3.1 Montagem do sistema de aquisição de imagens na linha de produção

O sistema utilizado para aquisição das imagens no transportador de garrafas da linha de produção de refrigerantes consiste em uma placa microprocessada Raspberry Pi Modelo B, câmera para Raspberry e sensor infravermelho conectado a uma entrada digital do Raspberry Pi. Para acomodação dos dispositivos raspberry e dipasy projetado um dispositivo mecânico, que posteriormente foi manufaturado em uma impressora 3D. Na mesma impressora 3D foi impresso um braço articulado para acomodação da câmera, com o intuito de facilitar o ajuste horizontal, vertical e de profundidade. O sensor infravermelho, com suporte para ajustes de posição, foi fixado ao lado da câmera. A posição escolhida para montagem da estrutura possui

relação com a posição do cabeçote de impressão da data de codificação, pois, ambos devem estar o mais próximo possível, evitando a variação da posição da garrafa e conseqüentemente a variação da posição do texto da impressão até a captura da foto. A Figura 20 ilustra a montagem e os componentes do sistema na linha de produção.

Figura 20 - Montagem do sistema na linha de produção



Fonte: Autor

3.2 Montagem da estrutura para simulação da linha de produção

Para simular o transportador da linha de produção foi utilizado um mecanismo motorizado com ajuste de velocidade acoplado a um disco com diâmetro de 100 cm. Na extremidade deste disco foram dispostas oito garrafas, separadas uma da outra por uma distância semelhante à que ficam separadas na linha de produção, que é de aproximadamente 10 cm. Esse dispositivo possibilitou a avaliação da aquisição da imagem sob diferentes cenários de velocidade e luz de fundo, sem a necessidade de intervenções na linha de produção. O dispositivo é mostrado na Figura 21.

Figura 21 - Estrutura para simulação da linha de produção



Fonte: Autor

3.2.1 Simulação da velocidade da linha de produção e aquisição das imagens

As velocidades da linha de produção onde o projeto de captura e avaliação da codificação para o qual o sistema foi simulado variam conforme a Tabela 1. Da mesma forma, na mesma tabela, estão indicadas a velocidade mínima e máxima que o dispositivo para simulação pode variar.

Tabela 1 - Velocidades da linha de produção e de simulação

LINHA DE PRODUÇÃO		DISPOSITIVO DE SIMULAÇÃO	
Modo de operação	Velocidade (Garrafas/hora)	Ajuste	Velocidade (Garrafas/h)
Manual (JOG)	5000	Mínimo	4428
Acúmulo	12000	Médio	17000
Máxima	18000	Máximo	32620

Fonte: Autor

Com base na Tabela 1 podemos observar que é possível variar a velocidade da simulação abrangendo todas as velocidades em que a linha de produção opera. Para as simulações foi inserida apenas uma garrafa no dispositivo, ajustando a velocidade conforme cada um dos 3 modos de operação da linha. Foram realizados testes com dispositivos de iluminação frontal direta e de fundo, normalmente utilizados para acentuar as formas do objeto inspecionado, produzindo um contraste adequado. A iluminação incidente na parte traseira da garrafa variou em três níveis de intensidade, sendo a iluminação natural do ambiente (100 Lux), lâmpada de led 1 (800 Lux) e lâmpada de led 2 (1500 Lux). Nas condições de diferentes velocidades e

diferentes intensidades de luz de fundo, foram avaliados aspectos referentes a aquisição das imagens, bem como qualidade e posição do texto nas imagens adquiridas.

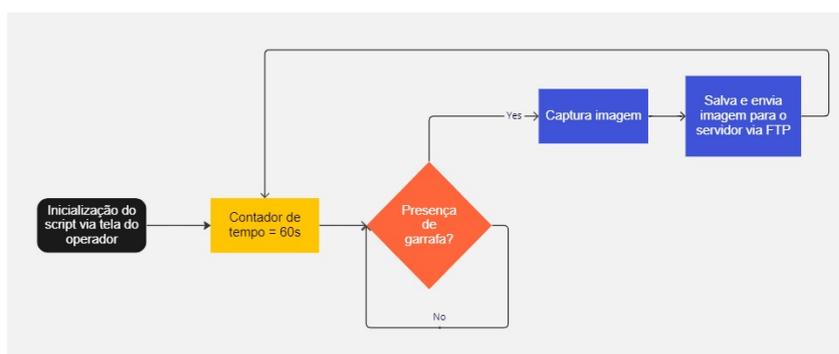
3.2 Desenvolvimento do algoritmo de captura, processamento e avaliação das imagens

Nos tópicos seguintes abordaremos os processos de desenvolvimento dos algoritmos para aquisição automática das imagens, de tratamento das imagens com OpenCV e para os testes de similaridade e acurácia do sistema.

3.2.1 Algoritmo para aquisição das imagens de forma automática

Foi desenvolvido um algoritmo específico para aquisição e armazenamento das fotos, onde é possível ajustar parâmetros de velocidade do obturador (*shutter speed*), brilho, contraste, fator ISO e resolução da imagem. A captura da imagem estará condicionada a habilitação da função de captura da foto pelo algoritmo a intervalor de 60 segundos e ao sinal enviado pelo sensor infravermelho após a detecção da garrafa na área de inspeção. Tal sistemática se fez necessário para permitir que o sistema de processamento execute as funções de captura, processamento, avaliação OCR e comparação do resultado em tempo hábil. Além disso, as avaliações nesse intervalo de tempo são suficientes para atender a necessidade de substituição da tarefa de inspeção visual pelo operador, que, segundo o padrão técnico deste processo deve ocorrer em intervalos de 30 minutos. A Figura 22 mostra o fluxograma do algoritmo.

Figura 22 - Fluxograma para aquisição da imagem



Fonte: Autor

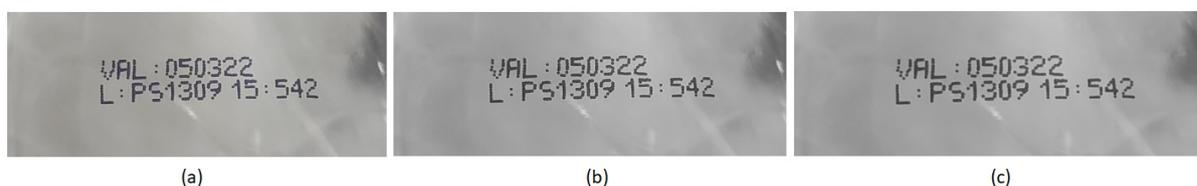
As imagens capturas são salvas em uma máquina virtual hospedada no servidor de automação da fábrica, ficando disponíveis para análise através de um algoritmo desenvolvido em python, responsável pelo processamento de imagem, aplicação do OCR e envio dos dados para o Node-Red.

3.2.2 Algoritmo para tratamento da imagem

Após o processo de aquisição das imagens de forma automática foram realizadas etapas de testes de sequências e métodos de processamento com diferentes imagens utilizando um algoritmo desenvolvido em python. Os resultados obtidos foram analisados e a partir deles foram realizados os ajustes necessários para melhorar a qualidade da classificação dos códigos.

Como as imagens são adquiridas em diferentes condições de iluminação e de posição da área impressa, se faz necessário um pré-processamento para melhorar as etapas seguintes. O pré-processamento reduz o ruído da imagem e aumenta a velocidade e precisão do processamento do OCR. Primeiramente a imagem foi convertida em escala de cinza e, devido aos caracteres da codificação serem compostos por pontos desconectados, foi aplicado primeiramente um filtro passa-baixa (Gaussian Blur), que reduz as componentes de alta frequência, tornando a imagem com um aspecto opaco. Na Figura 23 são mostradas imagens com os pré-tratamentos descritos acima.

Figura 23 - (a) Imagem original. (b) Escala de cinza. (c) Filtro Blur



Fonte: Autor

A partir da imagem gerada com a aplicação do filtro Gaussiano (Blur) foi aplicado um procedimento para tornar a imagem mais nítida. Foi utilizado um *kernel* para destacar cada pixel específico e aprimorar a cor que ele emite. É muito semelhante ao processo de desfoque, exceto que agora, em vez de criar uma matriz para a média de cada intensidade de pixel, foi criada uma matriz que fará com que as intensidades de cada pixel sejam maiores e, portanto, mais proeminentes ao olho

humano. Esse procedimento, na biblioteca OpenCV, é chamado de *sharpen*. O resultado desta operação é mostrado na Figura 24.

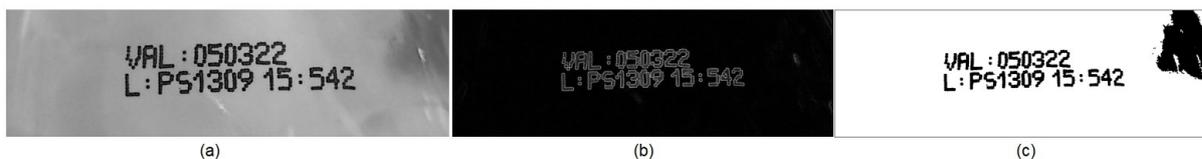
Figura 24 - (a) Filtro Blur. (b) Filtro Sharpen



Fonte: Autor

Nas etapas seguintes foram executadas transformações morfológicas de erosão, gradiente, filtro de mediano. Além disso, foi atribuída uma função para inversão da cor preta pela cor branca, chamada inversão binária. A sequência de imagens de saída dos tratamentos realizados é mostrada na Figura 25.

Figura 25 - (a) Erosão. (b) Morfologia Gradiente. (c) Inversão binária



Fonte: Autor

3.2.3 Avaliação com OCR Tesseract

Segundo Rosebrock (2020), para obtenção de um bom resultado no reconhecimento de caracteres, algumas recomendações devem ser seguidas, conforme listado abaixo:

- a) alinhamento do texto na horizontal;
- b) nitidez do texto;
- c) Imagem digitalizada em preto e branco ou em tons de cinza;
- d) fundo do texto limpo;
- e) resolução mínima de 300 DPI (número de pontos por polegada na imagem).

O tratamento de imagens mostrado na seção anterior teve como premissa seguir essas recomendações. Os testes e ajustes no tratamento da imagem tiveram

como objetivo atender ao máximo a eficiência na transformação da imagem original em uma imagem com características adequadas para o OCR, entretanto, na medida em que as características de posição, nitidez, contraste, ou distância entre os pontos que formam as letras impressas da imagem são alteradas, os parâmetros de tratamento poderão sofrer alterações, de forma a se obter uma maior acurácia.

Algumas configurações na chamada da função do OCR Tesseract no python são necessárias para que a ferramenta obtenha uma melhor assertividade. Os parâmetros de PSM e OEM são o modo de segmentação de página e o modo do “motor” do OCR, respectivamente. Também foi configurada uma lista específica com os caracteres que poderão ser encontrados no texto, chamada *whitelist*. A Figura 26 mostra o trecho do código em python para execução da ferramenta, bem como os parâmetros utilizados. Para os casos em que é utilizada uma lista *whitelist*, o parâmetro OEM deve ser configurado com valor zero.

Figura 26 - Parâmetros Tesseract

```
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
img_tess = cv2.imread('tratamento2.png')
texto_extraido = pytesseract.image_to_string(img_tess, config="-c tessedit_char_whitelist=0123456789SPETPCRVAl: --psm 11")
#PSM 11 --> Sparse text. Find as much text as possible in no particular order
#OEM 0 --> Legacy engine only
print("Resultado do TesseractOCR:")
print (texto_extraido)
print("PROGRAMA EXECUTADO COM SUCESSO")
```

Fonte: Autor.

Para o ajuste da ferramenta Tesseract utilizamos a imagem após o tratamento de binarização mostrada na Figura 25. Com essa imagem obtivemos o melhor resultado nos testes preliminares, conforme ilustrado na Figura 27.

Figura 27 - Resultado da avaliação com o OCR Tesseract



Fonte: Autor

3.2.4 Avaliação com OCR EasyOCR

Para ajuste da ferramenta de reconhecimento óptico de caracteres EasyOCR foi utilizada a mesma sistemática de processamento de imagens citada no tópico 3.2.2. O EasyOCR é executado através do método *readtext* no python, que é o método principal para a classe de leitura. Os parâmetros de idioma, parágrafo, GPU e *allowlist* (lista dos caracteres possíveis de serem encontrados no texto) foram configurados conforme mostrado na Figura 28.

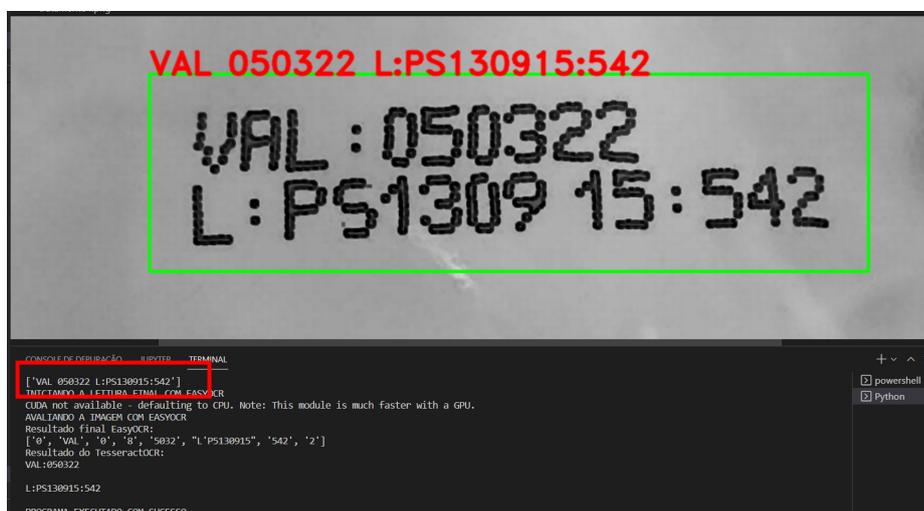
Figura 28 - Trecho de código para execução do EasyOCR

```
print("INICIANDO A LEITURA FINAL COM EASYOCR")
reader = easyocr.Reader(['en'], gpu=True)
#gpu = true --> Utiliza processamento da GPU
kernel = np.ones((3,3),np.uint8)
lista = []
print("AVALIANDO A IMAGEM COM EASYOCR")
result = reader.readtext(caminho,paragraph=False,
                        x_ths=1,y_ths=2,
                        allowlist="'VAL':'PET':'PCR'-1234567890'L'PS:')")
#paragraph = False --> Combina os resultados encontrados em uma lista
#allowlist --> Lista dos caracteres específicos que poderão ser encontrados na avaliação
img = cv2.imread(caminho)
```

Fonte: Autor.

Durante o ajuste preliminar da ferramenta EasyOCR foi utilizada a imagem (c) da Figura 25. A partir dessa imagem, e com as configurações mostradas na Figura 26 obtivemos o melhor resultado, conforme ilustrado na Figura 29.

Figura 29 - Resultado da avaliação com o OCR EasyOCR

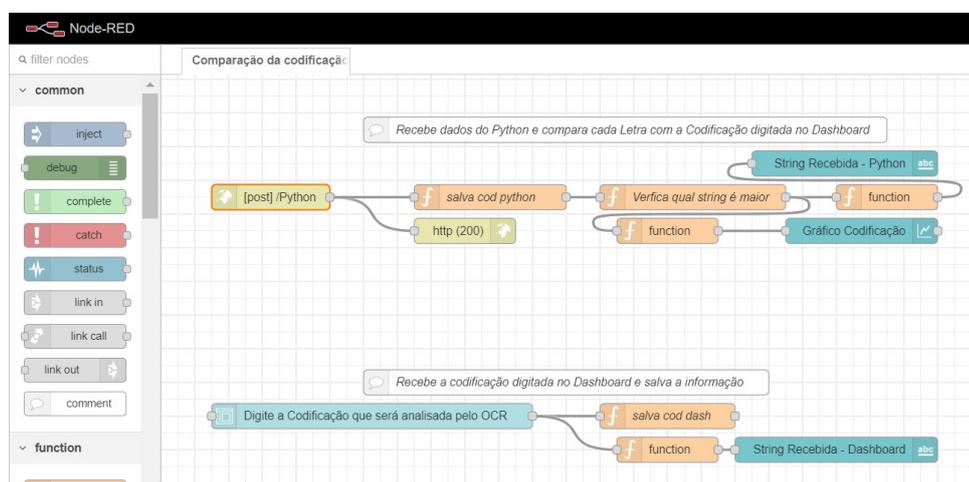


Fonte: Autor.

3.2.5 Processamento dos resultados via Node-Red

Após a execução da verificação do texto pelo OCR Tesseract e EasyOCR os resultados são enviados para o Node-Red através de uma requisição HTTP, onde serão comparados com um padrão previamente inserido pelo usuário através da interface web disponível no display *touchscreen*. Na plataforma Node-Red foi programada uma lógica para comparação do resultado inserido pelo operador com o resultado obtido na avaliação OCR. A Figura 30 mostra a programação em Node-Red responsável pela criação da interface web (dashboard) e pela comparação e envio dos resultados para o banco de dados.

Figura 30 - Programa em Node-Red

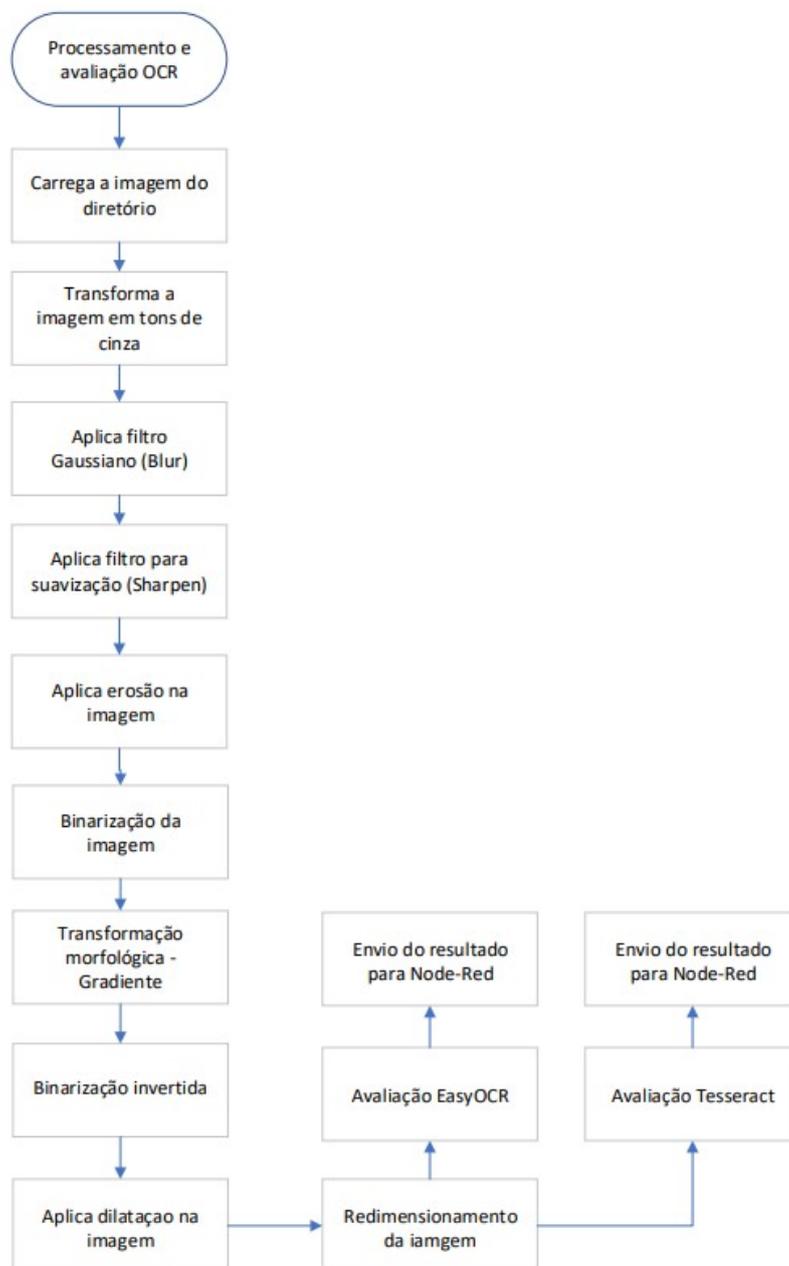


Fonte: Autor

A interface web, construída no software Node-Red, utilizando os recursos de dashboard, permite a inserção de informações de parâmetros de configuração do sistema de inspeção, botão de início do script de ajuste da câmera, além de informações estatísticas dos resultados do sistema de inspeção. As fotos e resultados ficarão salvos no banco de dados Influx-DB.

O fluxograma ilustrado na Figura 31 mostra a sequência do programa desde o carregamento da imagem até o envio do resultado da avaliação OCR para a ferramenta Node-Red.

Figura 31 - Fluxograma do programa de tratamento de aplicação de OCR's



Fonte: Autor

O primeiro passo no processo foi a aquisição da imagem, com a formação e digitalização da imagem. Após a obtenção da imagem o próximo passo consiste no pré-processamento da imagem, tendo como objetivo atribuir características a imagem que facilitem as etapas seguintes, como transformação em tons de cinza, redução de ruído através do filtro Blur e suavização da imagem através do filtro *sharpen*. A etapa de segmentação tem por objetivo separar o objeto de interesse, neste caso o texto impresso na garrafa. Essa etapa compreende a binarização,

erosão, transformação morfológica gradiente e dilatação. O redimensionamento ocorre com base nas matrizes referentes ao texto encontrado, que servem para delimitar a área que será recortada e redimensionada.

O capítulo a seguir apresenta a metodologia utilizada para as simulações, testes e análise dos resultados dos algoritmos e do sistema.

4 METODOLOGIA

Neste tópico será abordada a metodologia utilizada para a realização das simulações propostas para a captura das imagens de forma automática e para a avaliação do OCR. Os testes se concentraram principalmente em dois aspectos. O primeiro, em identificar a melhor condição física e lógica para obter, de forma automática, uma imagem com qualidade suficiente para ser tratada e ter o texto da codificação extraído com a aplicação da ferramenta de OCR. O segundo teste consistiu em avaliar a eficácia do tratamento da imagem via Open-CV e conseqüentemente a acuracidade do reconhecimento ótico de caracteres.

4.1 Qualidade de captura das imagens

Para avaliação da qualidade das imagens capturadas do código de validade e lote, dentre outras informações presentes na codificação impressa, foram utilizados três cenários diferentes, variando a incidência de luz e a velocidade com que a garrafa passa pelo sensor infravermelho e pela câmera. Para todos os cenários de velocidade e iluminação foram necessários ajustes na posição do sensor e de parâmetros da câmera, como brilho, contraste, sensibilidade ISO e *shutter speed*, que é o tempo que o obturador da câmera fica aberto expondo a luz no sensor. Em cada diferente cenário foram capturas 350 imagens.

4.1.1 Velocidade de 5000 garrafas por hora e iluminação natural

Com a execução do algoritmo de captura de imagens de forma automática, a primeira simulação ocorreu com a aquisição de fotos sob iluminação natural, de aproximadamente 100 Lux, ajustando a velocidade do simulador da linha de produção para 5000 garrafas por hora. A distância do sensor até a câmera foi ajustada para 63 mm, a fim de obter a centralização do texto da codificação na imagem capturada. A velocidade do obturador foi parametrizada em 500 ms. O contraste e o brilho foram ajustados em 80 e 70, respectivamente. A sensibilidade ISO foi ajustada para 800, que é o máximo valor possível.

4.1.2 Velocidade de 5000 garrafas por hora e iluminação auxiliar de led de 800 Lux

Nesta simulação foi mantida a velocidade de 5000 garrafas por hora, sendo aplicada iluminação auxiliar de led de cor branca com intensidade de 800 Lux, na parte traseira da garrafa. A distância do sensor até a câmera foi mantida em 63 mm. A velocidade do obturador foi alterada para 300 ms. O fator ISO, contraste e brilho não foram alterados.

4.1.3 Velocidade de 5000 garrafas por hora e iluminação auxiliar de led de 1500 Lux

Nesta simulação foi aplicada iluminação auxiliar de led de cor branca com intensidade de 1500 Lux, na parte traseira da garrafa. A distância do sensor até a câmera foi mantida em 63 mm. A velocidade do obturador foi alterada para 300 ms. O contraste e o brilho não foram alterados. O fator ISO foi alterado para 400.

4.1.4 Velocidade de 12000 garrafas por hora e iluminação natural

Com a alteração da velocidade do simulador para 12000 garrafas por hora foi necessário realizar ajustes na posição do sensor e nos parâmetros de *shutter speed* para 150 ms e de contraste para 60. A distância do sensor até a câmera foi alterada para 97 mm.

4.1.5 Velocidade de 12000 garrafas por hora e iluminação auxiliar de led de 800 Lux

Nesta simulação foi mantida a velocidade de 12000 garrafas por hora, sendo aplicada iluminação auxiliar de led de cor branca com intensidade de 800 Lux. A distância do sensor até a câmera foi mantida em 97 mm e velocidade do obturador foi alterada para 200 ms. O fator ISO foi alterado para 600. O contraste e o brilho não foram alterados.

4.1.6 Velocidade de 12000 garrafas por hora e iluminação auxiliar de led de 1500 Lux

Para este cenário foi introduzida uma iluminação auxiliar de led de cor branca com intensidade de 1500 Lux na parte traseira da garrafa. A distância do sensor até

a câmera foi mantida em 97 mm. A velocidade do obturador foi alterada para 200 ms. O contraste, brilho e o fator ISO permaneceram inalterados.

4.1.7 Velocidade de 18000 garrafas por hora e iluminação natural

A velocidade de 18000 garrafas por hora é a velocidade nominal de operação da linha de produção de refrigerantes simulada. Neste caso os ajustes de posição do sensor e de parâmetros da câmera foram testados até que fosse possível a aquisição de uma imagem centralizada, entretanto, com iluminação natural não foi possível capturar a imagem do texto impresso na garrafa sem que houvesse prejuízo a formação da imagem. O sensor ficou distante 123 mm com relação a câmera. A velocidade do obturador foi alterada para 10 ms e o fator ISO para 800.

4.1.8 Velocidade de 18000 garrafas por hora e iluminação auxiliar de led de 800 Lux

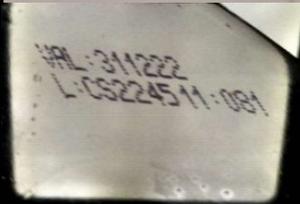
Nesta condição de simulação foi aplicada uma iluminação auxiliar de led de cor branca com intensidade de 800 Lux na parte traseira da garrafa. A distância do sensor até a câmera foi mantida em 123 mm. A velocidade do obturador foi alterada para 80 ms. O fator ISO foi alterado para 600. O contraste e o brilho não foram alterados.

4.1.9 Velocidade de 18000 garrafas por hora e iluminação auxiliar de led de 1500 Lux

Para simular essa condição de operação foi alterada a iluminação auxiliar de 800 para 1500 Lux. Além disso, foram alterados os parâmetros de *shutter speed* para 80 ms, de brilho para 90 e contraste para 80. O fator ISO foi desabilitado. A distância entre o sensor e a câmera não sofreu alteração.

A qualidade de todas as imagens capturas durante os testes foi avaliada com base em uma inspeção visual e posteriormente, para algumas amostras pré-selecionadas, com a utilização do algoritmo de tratamento e avaliação OCR. Alguns exemplos de imagens capturas em diferentes condições de luz e velocidade são mostradas na Figura 32:

Figura 32 - Exemplos de imagens capturas durante os testes.

		ILUMINAÇÃO		
VELOCIDADE (Garrafas por hora)	Natural	intermediária	Forte	
5000				
12000				
18000				

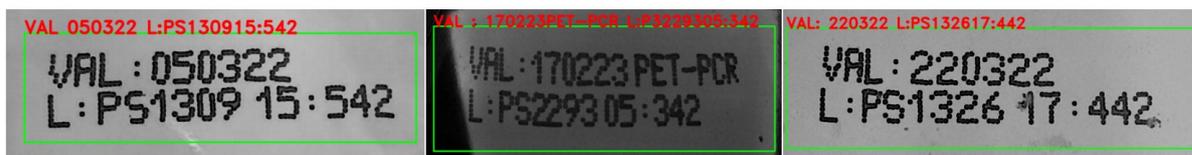
Fonte: Autor

Foram observadas variações da posição da garrafa no instante da captura das imagens em distancias fixas do sensor infravermelho em relação a câmera, enquanto mantidos os parâmetros de configuração de captura de imagem da câmera.

4.2 Desempenho do reconhecimento ótico de caracteres – OCR Tesseract

Para a avaliação do desempenho e eficiência do tratamento e avaliação do OCR, foi necessário definir métricas para medir o grau de semelhança do texto extraído pela ferramenta de reconhecimento ótico com o texto impresso na garrafa, para assim, após a comparação com o texto padrão, atestar a acuracidade do sistema através de uma matriz de confusão. Exemplos de saída da análise OCR via Tesseract são mostradas na Figura 33.

Figura 33 - Imagens avaliadas e respectivos resultados da avaliação OCR



Fonte: Autor

As *strings* geradas como resultado da aplicação do algoritmo de OCR nas imagens são escritas em um arquivo de texto de forma sequencial. Dessa forma, foi possível avaliar o histórico dos resultados de todas as inspeções realizadas, utilizando 15 imagens de três modelos de codificação diferentes, capturadas na condição de velocidade de 5000 garrafas por hora e com iluminação de led forte. Para a avaliação da similaridade as *strings* geradas como resultado da avaliação OCR foram comparadas através de um código escrito em python. Na Figura 34 é mostrado parte do código em python para avaliação da similaridade, no qual o resultado da similaridade foi de 96,29%.

Figura 34 - Script python para cálculo da similaridade

```

venv > teste_similaridade.py > ...
5  def similar(str1, str2):
6      return SequenceMatcher(None, str1, str2).ratio()
7
8  test_string1 = 'VAL : 050322 L:PS1309 15:542' #imagem A
9
10 test_string2 = 'VAL:050322 L:PS1309 15:542'
11 res = similar(test_string1, test_string2)
12 print ("The similarity between 2 strings is : " + str(res))

```

CONSOLE DE DEPURAÇÃO JUPYTER TERMINAL

TERMINAL

```

v/Scripts/python.exe" "c:/Users/user/Desktop/TCC_2/Projetos python/TCC/venv/test
The similarity between 2 strings is : 0.9629629629629629

```

Fonte: Autor

A similaridade informada como saída do script pode variar de 0 a 1, sendo 0 para nenhuma similaridade e 1 para completa similaridade. Quando ocorre a completa similaridade temos uma de maior eficiência nos processos de obtenção, tratamento da imagem e reconhecimento de texto pelo OCR.

4.3 Critério de conformidade da codificação

Para simular o desempenho do algoritmo de classificação de conformidade foi utilizada uma matriz de confusão. Nesta etapa, foram avaliadas imagens que apresentaram boa qualidade, onde o programa de reconhecimento OCR obteve acurácia igual a 1. Metade das 30 imagens avaliadas tiveram parte do texto da codificação apagado ou borrado. As predições do sistema foram classificadas com saída binária em quatro situações: verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP) e falso negativo (FN). Para utilizar a matriz foi definido que a classificação “Conforme” é tratada como Positiva e “Não-conforme” como Negativa. Os valores VP referem-se as amostras como conforme quando ela realmente está de acordo, ou seja, o algoritmo leu o código correto, comparou com o valor padrão e classificou a amostra como “Conforme”. VN ocorre nos casos em que a amostra apresenta algum defeito e o algoritmo lê corretamente, compara e classifica a amostra como “Não-conforme”. O FP ocorre quando a codificação deveria ser avaliada como “Não-conforme”, porém o algoritmo classificou a amostra como “Conforme”. Classificações FN ocorrem quando a codificação da garrafa é apontada como “Não-conforme”, mas deveria ser classificada como “Conforme”. O Quadro 2 mostra o modelo de matriz de confusão utilizada para avaliação:

Quadro 2 – Matriz de confusão genérica

Condição Atual		<i>P</i>	<i>N</i>
		Teste	
<i>p</i>	VP	FP	
<i>n</i>	FN	VN	

Fonte: Autor

Para calcular a acurácia e a especificidade da avaliação de desempenho, tendo como base a matriz de confusão, são utilizadas as Equações 4.1 e 4.2, respectivamente, (MATOS *et al.*, 2009).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.1)$$

$$Especificidade = \frac{VN}{VN + FP} \quad (4.2)$$

Conforme Matos *et al.* (2009), acurácia avalia o percentual de acertos, ou seja, a razão entre a quantidade de predições corretas e o total de entrada do sistema. A Especificidade avalia a capacidade do sistema detectar corretamente os resultados classificados como negativos. Espera-se obter uma alta acurácia e principalmente, uma especificidade de 100%, que significa que nenhuma amostra com defeito recebeu classificação “CONFORME”.

No próximo tópico serão discutidos os resultados obtidos com a aplicação dos métodos de simulação.

5 RESULTADOS OBTIDOS

Este tópico tem por objetivo descrever os resultados obtidos a partir dos testes e simulações dos algoritmos para aquisição de imagens de forma automática e para tratamento de imagens para posterior aplicação de ferramenta de OCR.

5.1 Qualidade das imagens adquiridas de forma automática

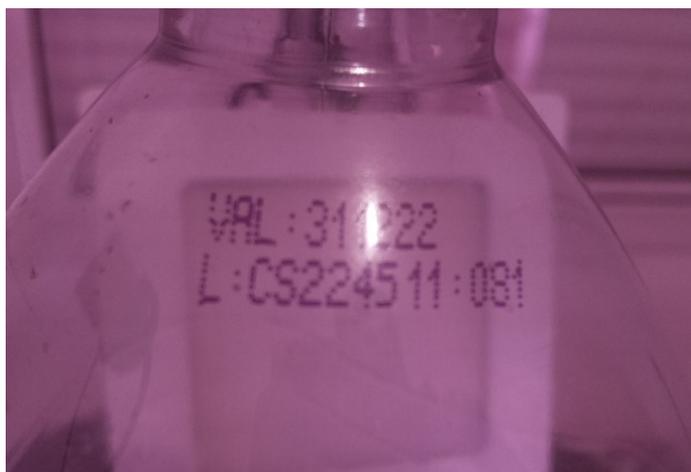
Este tópico trata da análise dos resultados obtidos com os testes de captura automática de imagens, bem como avaliação da qualidade das imagens capturadas, conforme caracterizado na metodologia.

5.1.1 Variação de velocidade e exposição a iluminação natural

As imagens adquiridas com iluminação natural, captadas inicialmente com a garrafa passando a uma velocidade de 5000 garrafas, apresentaram uma qualidade muito baixa, com aproveitamento de 18% das imagens para fins de tratamento e avaliação OCR. Foram necessários diversos ajustes a fim de obter uma imagem com nitidez adequada. Um dos problemas encontrados foram ruídos na imagem ocasionados por reflexão da luz ambiente na garrafa. Nos testes com iluminação natural e velocidades de 12000 e 18000 nenhuma imagem captada pode ser aproveitada, pois a quantidade de luz captada pelo sensor da câmera não foi suficiente. Para tentar corrigir este problema foi aumentado o parâmetro de *shutter speed*, contudo, as imagens geradas ficaram escuras, sem a nitidez.

Segundo Gonzales & Woods (2020), os tipos de imagens são gerados pela combinação de uma fonte de iluminação e a reflexão ou absorção de energia pelos elementos da cena que está sendo fotografada, entretanto, quando estamos interessados na região de contorno em imagens, é indicada a aplicação de retroiluminação, que produzirá uma imagem binária. A Figura 35 mostra uma imagem capturada em uma velocidade de 5000 garrafas por hora e com iluminação natural de aproximadamente 100 Lux.

Figura 35 - Imagem captura com luz ambiente



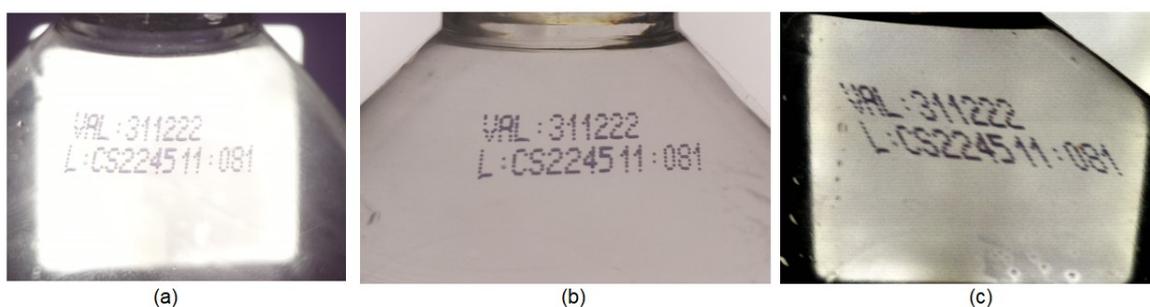
Fonte: Autor

5.1.2 Velocidades variadas e exposição a iluminação de 800 Lux

Os resultados obtidos nas condições em que foi inserida retroiluminação de led na garrafa foram satisfatórios, quando a velocidade de simulação foi fixada em 5000 e 12000 garrafas, onde, após os ajustes de parâmetros da câmera todas as imagens capturadas foram classificadas como adequadas para a etapa de tratamento e avaliação OCR. Na velocidade de 5000 garrafas por hora o aproveitamento ficou em 100% e em 12000 garrafas por hora em 91%.

As imagens adquiridas com o simulador fixado na velocidade de 18000 garrafas por hora não apresentaram uma qualidade adequada, com um baixo aproveitamento de 37%. Na Figura 36 são mostrados exemplos de imagens capturadas nas diferentes velocidades, quando submetidas a retroiluminação de 800 Lux.

Figura 36 - Exemplos de Imagens: (a) Velocidade 5000 garrafas/h, (a) Velocidade 12000 garrafas/h, (a) Velocidade 18000 garrafas/h



(a)

(b)

(c)

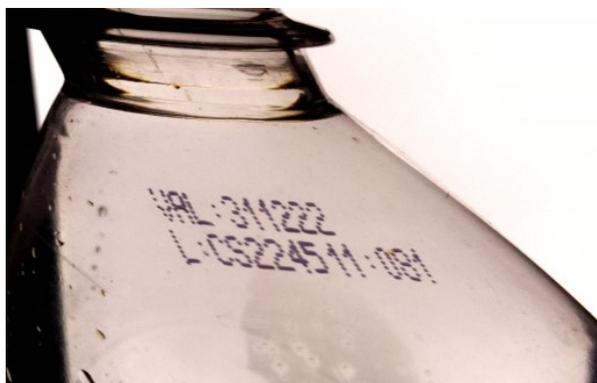
Fonte: Autor

Os defeitos mais significativos foram a descentralização da garrafa na imagem, baixo contraste e a incidência do efeito *rolling shutter*. Segundo o estudo realizado por Silva (2020), a velocidade do obturador causa um escurecimento nas imagens adquiridas, à medida que o tempo para captação dos fótons é menor. Também, velocidades menores contribuem para a estabilização da imagem, minimizando tremidos e borrões.

5.1.3 Velocidades variadas e exposição a iluminação de 1500 Lux

Com o incremento da intensidade luminosa projetada por trás da garrafa obteve-se o melhor resultado na qualidade das imagens para as velocidades de 5000 e 12000 garrafas por hora, com resultado de 100% de aproveitamento. A Figura 37 mostra uma imagem capturada com a velocidade do simulador fixada em 18000 garrafas por hora e iluminação de 1500 Lux.

Figura 37 - Imagem da garrafa captura na velocidade de 18000 garrafas por hora



Fonte: Autor

Para a velocidade de 18000, ocorreram muitos problemas de descentralização da imagem e *rolling shutter*, mesmo após o ajuste do parâmetro de *shutter speed* para o mínimo valor possível. As variações de posição da garrafa na imagem podem ter relação com a limitação de hardware e imprecisão do sensor infravermelho. Segundo a folha de dados do sensor infravermelho utilizado no experimento, variações na reposta de saída podem ter relação com o material do dispositivo a ser detectado e a fontes de luz adjacentes. A Tabela 2 apresenta um compilado das taxas de aproveitamento das garrafas nos diferentes cenários.

Tabela 2 - Taxas de aproveitamento de imagens captadas de forma automática

VELOCIDADE (Garrafas por hora)	ILUMINAÇÃO		
	Natural	Intermediária	Forte
5000	88%	100%	100%
12000	30%	71%	100%
18000	0%	63%	82%

Fonte: Autor

O desempenho do sistema de aquisição de imagens de forma automática teve o desempenho comprometido devido aos efeitos de *rolling speed*, bem como a efeitos de pouca luminosidade. As imagens capturadas com iluminação intermediária e forte, com o sistema simulado a velocidade de 5000 garrafas por hora apresentaram qualidade adequada para utilização nos testes de avaliação OCR.

5.2 Desempenho do reconhecimento ótico de caracteres Tesseract

Os resultados obtidos a partir da sequência de métodos de tratamento de imagens aplicada, conforme mostrado na Figura 32, e com a aplicação da ferramenta de reconhecimento ótico Tesseract foram tabulados, conforme Tabela 3. Para a comparação de similaridade entre os textos foram considerados os espaços em branco entre os caracteres, sendo um dos fatores que contribuiu para a diminuição dos resultados de similaridade.

Tabela 3 - Similaridade entre os resultados encontrados com a predição

GARRAFA 1 VAL : 050322 L:PS1309 15:542			GARRAFA 2 VAL : 220322 L:PS1326 17:442			GARRAFA 3 VAL : 170223 PET-PCR L:PS2293 05:342		
	Texto lido pelo OCR	Similaridade	Texto lido pelo OCR	Similaridade	Texto lido pelo OCR	Similaridade		
Imagem 1	VAL 050322 L:PS1309 15:542	0,961538462	VAL:220322 L:PS132617:442	0,943396226	VAL:170223 PET-PCR L:PS729305:342	0,927536232		
Imagem 2	VAL 050322 L:PS130915 542	0,901960784	VAL : 220322 L:PS132047:442	0,909090909	VAL:170223PET-TCR L:PS 29305:342	0,909090909		
Imagem 3	VAL 050322 L:PS130915 542	0,941176471	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223 PET-PCR L:PS29305:342	0,927536232		
Imagem 4	VAL 050322 L:PS130915 542	0,901960784	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223 PET-PCR L:PS729305:342	0,927536232		
Imagem 5	VAL : 050322 L:PS1309 15:542	1,000000000	VAL:220322 L:PS132617:442	0,943396226	VAL:17023PET-PCR L:PS29305:342	0,878787879		
Imagem 6	VAL : 050322 L:PS1309 15:542	1,000000000	VAL:220322 L:PS132617:442	0,943396226	VAL:170223PET-TCR L:PS 29305:342	0,882352941		
Imagem 7	VAL : 050322 L:PS1309 15:542	1,000000000	VAL:220322 L:PS132617:442	0,943396226	VAL:170223PETPCR LP5229305:342	0,878787879		
Imagem 8	VAL 050322 L:PS1309 15 542	0,925925926	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223PET-TCR L:PS 29305:342	0,882352941		
Imagem 9	VAL 050322 L:PS1309 15 542	0,925925926	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223PET-PCR PS229305:342	0,909090909		
Imagem 10	VAL 050322 L:PS1309 15 542	0,925925926	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223PET-PCR L:PS 29305:342	0,911764706		
Imagem 11	VAL : 050322 L:PS1309 15:542	1,000000000	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223PET-PCR L:PS 29305:342	0,911764706		
Imagem 12	VAL 050322 L:PS1309 15 542	0,925925926	VAL: 220322 L:PS132617:142	0,962962963	VL :170223PET-PCR L:PS79305:342	0,865671642		
Imagem 13	VAL : 050322 L:PS130915:542	0,981818182	VAL: 220322 L:PS132617:442	0,962962963	VAL :170223 PET-PCR L:PS229305 : 342	0,916666667		
Imagem 14	VAL : 050322 L:PS130915:542	0,981818182	VAL: 220322 L:PS132617:442	0,962962963	VAL17022PETPR LP229305:34	0,819672131		
Imagem 15	VAL : 050322 L:PS130915:542	0,905660377	VAL: 220322 L:PS132617:442	0,962962963	VAL:170223PET-PCR L:PS27305:342	0,865671642		
Média		0,95		0,95		0,89		

Fonte: Autor

Os maiores problemas encontrados nos resultados da aplicação das ferramentas de tratamento de imagem foram a baixa homogeneidade entre as imagens e codificações. Como o ajuste dos parâmetros dos filtros e kernel escolhidos foram fixados para cada tipo de garrafa, a variação entre os tons de cinza e nitidez do texto nas diferentes imagens avaliadas tiveram forte interferência no resultado. Uma alternativa para minimizar essas distorções seria aumentar a qualidade da codificação e padronizar o processo de captura de fotos, mantendo o objeto o mais centralizado possível. A média das similaridades encontradas entre os textos na amostra “GARRAFA 1” foi de 95%, tendo 4 amostras com reconhecimento de texto completamente similares. Se desconsiderarmos os espaços em branco e o caractere “:”, o reconhecimento de texto completo sobe para 13, das 15 amostras. Trocas das letras “S” pelo número “5” também ocorreram em duas 2 amostras.

Nas avaliações realizadas com a amostra “GARRAFA 2” os resultados de similaridade se mantiveram quando comparados com o da primeira amostra, apesar do fato dos caracteres 1 e 2 apresentarem borrão, conforme mostrado na Tabela 4.

A amostra “GARRAFA 3” obteve o pior desempenho de similaridade entre os textos. A não detecção de algum dígito ou a detecção imprecisa em casos de troca de caracteres parecidos foi mais acentuada nessa amostra. Os erros mais comuns foram a troca de caracteres, como o “S” pelo “5” e “2” pelo “7”. A menor eficiência da aplicação de OCR ocorreu na amostra “GARRAFA 3” e pode ser explicada pelo fato de os caracteres impressos estarem muito próximos um do outro, dificultando o reconhecimento dos caracteres pelo OCR.

5.3 Conformidade da codificação

Os resultados obtidos com a sequência otimizada foram expressos em forma de matriz de confusão, bem como foram calculadas as métricas de Acurácia e Especificidade, conforme apresentado no Quadro 3.

Quadro 3 - Resultados de conformidade obtidos

		Classificação Prevista	
		P	N
Classificação Real	P	13	0
	N	2	15

Acurácia	93,33%
Especificidade	100%

Fonte: Autor

A Acurácia do sistema ficou em 93,93%, se mantendo dentro do esperado, visto que alguns caracteres são “confundidos” pelo OCR, como a troca do “P” pelo “5”. Considerando o escopo deste trabalho, é justificável considerar que é mais significativo o algoritmo classificar uma garrafa conforme como não-conforme (FP) do que o contrário. Desta forma, os resultados de especificidade estão dentro do esperado. O algoritmo se mostrou bastante sensível a pequenas mudanças de ângulo e iluminação.

Os resultados obtidos demonstram que o sistema de baixo custo proposto é capaz de obter uma especificidade de 100%, desde que a velocidade não ultrapasse 5000 garrafas por hora e com iluminação adequada. Visto que essa velocidade de 5000 garrafas por hora está bem abaixo da velocidade nominal da linha de produção simulada, que é de 18000 garrafas por hora, o sistema não estaria apto a operar.

O mecanismo de fixação e ajuste de posição da câmera e do sensor, associado a retroiluminação auxiliar aplicada na garrafa melhorou substancialmente a qualidade das imagens, permitindo também, utilizar mais recursos de captura da câmera.

A partir dos testes verificou-se que os dispositivos de hardware utilizados não possuem capacidade suficiente para suprir as características de processamento necessárias para o processo de inspeção. As limitações acerca dos recursos da câmera e do sensor infravermelho limitaram a aquisição de imagens com qualidade quando simulada a velocidade nominal da linha de produção.

A sequência de métodos de tratamento de imagens se mostrou eficiente, melhorando os resultados de saída do OCR Tesseract em comparação com a ferramenta aplicada nas imagens originais. A ferramenta Tesseract obteve melhor desempenho em comparação a EasyOCR. As limitações encontradas e as contribuições com este trabalho serão destacadas na próxima seção.

6 CONSIDERAÇÕES FINAIS

O objetivo deste projeto envolve o estudo da viabilidade técnica da aplicação de um sistema de aquisição e tratamento de imagens para obtenção das informações da data de validade e lote em garrafas de bebidas em uma linha de produção. Desta forma, foram realizados testes referentes a aquisição das imagens de forma automática, simulando a velocidade de operação da linha de produção. Além disso, foram avaliadas as ferramentas para tratamento de imagens e o algoritmo para reconhecimento óptico de caracteres utilizando a ferramenta de OCR Tesseract.

Este trabalho demonstrou que o hardware utilizado no sistema proposto apresenta limitações de desempenho, o que afetou a precisão na captura das imagens e conseqüentemente, tornando a aplicação insatisfatória nas condições reais de operação da linha de produção. Melhorias no hardware como substituição do sensor infravermelho por um modelo retro reflexivo com espelho prismático, utilização de encoder rotativo para realimentação da velocidade da linha, correção de parâmetros de captura e utilização de uma câmera de melhor desempenho teriam potencial de fornecer uma solução para os problemas enfrentados com a captura de imagens nas velocidades nominais simuladas da linha de produção.

A aplicação dos testes de tratamento das imagens através do OpenCV permitiu concluirmos que são necessários, além do conhecimento das técnicas inerentes ao tratamento da imagem específica da codificação, conhecimentos sobre a obtenção de imagens digitais, bem como dos parâmetros envolvidos no processo fotográfico, como abertura do diafragma, fator ISO e condições de luminosidade. A obtenção de imagens de qualidade e a segmentação nítida do texto é primordial para a melhoria dos processos envolvidos neste projeto, como a aplicação da ferramenta de reconhecimento óptico de caracteres. A inclusão de processos de ajustes da imagem final a ser tratada pelo OCR através da alteração de parâmetros através da tela do operador com pré-visualizações seriam de fundamental importância para o sistema atender diferentes cenários, sendo objeto de estudo para trabalhos futuros.

A utilização e simulações do OCR Tesseract e EasyOCR possibilitou a compreensão da versatilidade e robustez dessas ferramentas. O Tesseract se mostrou, na maioria dos casos, mais preciso, comparado ao EasyOCR. A precisão

do OCR Tesseract aplicado neste projeto poderia ser melhorada significativamente com a aplicação de técnicas de treinamento da ferramenta. Os resultados de Acuracidade do sistema, que compreende o processo de captura automática da imagem, tratamento, aplicação OCR, comparação dos resultados via Node-Red e classificação dos resultados foi de 93,33%, quando simulado e limitado a velocidade de 5000 garrafas por hora.

A utilização de softwares livres possibilitou a simulação de resultados e a criação de interface para inserção de padrões de codificação e amostragem dos resultados, apresentando um desempenho satisfatório, o que permitiu concluir que a aplicação destes recursos em ambiente industrial é factível.

Em função das limitações de hardware para aquisição de imagens de qualidade para o tratamento adequado e posterior aplicação de OCR testadas neste trabalho, recomenda-se para trabalhos futuros a aplicação de tecnologias de geração de imagens com hardware compatível que atendam as condições de operação da linha de produção. A utilização de um sensor de presença de garrafas mais preciso e a inclusão de um dispositivo de realimentação da velocidade da linha para compensação do tempo correto de captura da imagem poderiam minimizar os erros de descentralização da garrafa na imagem. A aplicação de técnicas e ferramentas de aprendizagem do OCR considerando as características do texto extraído das imagens também poderão ser objeto de estudos futuros.

Por fim, sugere-se a incorporação de uma interface com o usuário mais robusta, que permita a inclusão de parâmetros inerentes aos algoritmos utilizados, flexibilizando a utilização do sistema.

REFERÊNCIAS

- BALLUFF (São Paulo) (comp.). Sensores industriais: 3 tecnologias da linha de sensores ópticos da Balluff. 2019. Disponível em: <https://balluffbrasil.com.br/sensores-industriais-3-tecnologias-da-linha-de-sensores-opticos-da-balluff/>. Acesso em: 18 jun. 2022.
- BRADSKI, Gary; KAEHLER, Adrian. Learning OpenCV. Sebastopol: O’reilly Media, 2008. 555 p.
- COATES, C.; JUVAN-BEAULIEU, I. How to choose between a Rolling Shutter & Global Shutter camera mode. Miramar Communications Ltd, 2020.
- CUNHA, R. C. Estudo Comparativo Sobre Ferramentas de Reconhecimento Óptico de Caracteres. [s. l.] Instituto Federal de Minas Gerais, 2018.
- DATASHEETSPDF (comp.). E18-D80NK-N Datasheet, Equivalent, Sensor Switch: manual of ir sensor switch. Manual of IR Sensor Switch. 2022. Disponível em: <https://datasheetpdf.com/pdf/1311838/ETT/E18-D80NK-N/1>. Acesso em: 13 set. 2022.
- DEBIAN (org.). The Universal operating system: razões para escolher o debian. Razões para escolher o Debian. 2022. Disponível em: https://www.debian.org/intro/why_debian. Acesso em: 13 abr. 2022.
- GONZALES, R. C; WOODS, R.C. Processamento digital de imagens. 3° Ed. - São Paulo: Pearson Prentice Hall, 2009.
- HOSOZAWA, K.; WIJAYA, R. H.; LINH, T. D.; et al. Recognition of Expiration Dates Written on Food Packages with Open Source OCR. International Journal of Computer Theory and Engineering, v. 10, n. 5, p. 170–174, 2018.
- MACHADO, U. T. Inspeção Automática de Frascos Utilizando Visão Computacional em Sistema Embarcado. [s.l.] Universidade de Caxias do Sul, 2019.
- MATOS, Pablo Freire *et al.* Relatório Técnico “Métricas de Avaliação”: projeto .:um ambiente para análise de dados da doença anemia falciforme::. São Carlos: Ufscar, 2009. 16 p. Disponível em: <https://sites.icmc.usp.br/tasparado/TechReportUFSCar2009a-MatosEtAl.pdf>. Acesso em: 13 set. 2022.
- MCANDREW, Alasdair. A Computational Introduction to Digital Image Processing. 2. ed. Melbourne: Crc Press, 2016.
- OPENJS FOUNDATION (Estados Unidos). Node-Red. Disponível em: <https://nodered.org/about/>. Acesso em: 10 fev. 2022.
- PEDRINI, H.; SCHWARTZ, W. R. Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações. 1 Ed. São Paulo: Cengage, 2008.

QUEIROZ, José Eustáquio Rangel de; GOMES, Herman Martins. Introdução ao Processamento Digital de Imagens. *Revista Rita: Instrução para preparação de documentos em Word*, Campina Grande, v. 8, n. 1, p. 1-30, 2016. Disponível em: <http://www.dsc.ufcg.edu.br/~hmg/disciplinas/graduacao/vc-2016.2/Rita-Tutorial-PDI.pdf>. Acesso em: 11 abr. 2022.

RASPBERRY PI FOUNDATION (Inglaterra). Raspberry Pi 4: your tiny, dual-display, desktop computer. *Your tiny, dual-display, desktop computer*. 2022. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. Acesso em: 10 maio 2022.

ROSEBROCK, Adrian *et al.* OCR with OpenCV, Tesseract, and Python. Filadélfia: Pyimageeach, 2020.

SANTIAGO, Gaubert Vinícius. Método Baseado em Visão Computacional para Reconhecimento de Dígitos Visando a Leitura de Consumo em Hidrômetros com Indicação Analógica e Digital. 2017. 127 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, Brasília, 2017.

SILVA, A. J. DE J. Metodologia de ajuste de parâmetros para aquisição de imagens de boa qualidade utilizando uma câmera fotográfica semiprofissional comercial. *Revista Sítio Novo*, v. 4, n. 4, p. 202, 2020.

SOLOMON, Chris; BRECKON, Toby. *Fundamentals of Digital Image Processing: a practical approach with examples in matlab*. Chichester: Wiley-Blackwell, 2011.

TAYLOR KAVANAUGHY (Estados Unidos). Get the footage without the rolling shutter effect. 2022. Disponível em: <https://www.adobe.com/creativecloud/video/discover/rolling-shutter-effect.html>. Acesso em: 13 out. 2022.

TECHNOLOGIES, Omnivision. 2009. Disponível em: https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf. Acesso em: 13 abr. 2022.

TESSERACT (Estados Unidos) (comp.). Tesseract OCR. 2022. Disponível em: <https://github.com/tesseract-ocr/tesseract>. Acesso em: 07 mar. 2022.

ZAAFOURI, A.; SAYADI, M.; FNAIECH, F. A vision approach for expiry date recognition using stretched gabor features. *International Arab Journal of Information Technology*, v. 12, n. 5, p. 448–455, 2015.