

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

FABRICIO SOARES RODRIGUES

IDENTIFICAÇÃO DE RUPTURA EM PRATELEIRAS DE SUPERMERCADOS
ATRAVÉS DE VISÃO COMPUTACIONAL

São Leopoldo

2019

FABRICIO SOARES RODRIGUES

**IDENTIFICAÇÃO DE RUPTURA EM PRATELEIRAS DE SUPERMERCADOS
ATRAVÉS DE VISÃO COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação, pelo curso de Engenharia da Computação da Universidade do Vale do Rio dos Sinos – UNISINOS

Orientador: Prof. Me. Jean Schmith

São Leopoldo

2019

RESUMO

Um dos principais problemas enfrentados pelo varejo é a ruptura de produtos nas prateleiras — que se refere à falta deles nas prateleiras ou no estoque. Ela é responsável por causar prejuízos tanto financeiros quanto para a imagem da empresa, pois quando o cliente não encontra o produto disponível na prateleira pode deixar de realizar a compra ou até mesmo procurar o concorrente. Com isso, a empresa perde a fidelidade do consumidor. O presente projeto apresenta um estudo para o desenvolvimento de uma solução automatizada, que visa a identificação de ruptura nas prateleiras de supermercados ou hipermercados por meio da visão computacional. Essa técnica seguiu as etapas de aquisição, pré-processamento, segmentação, reconhecimento e identificação para capturar a imagem da prateleira, localizar seus espaços vazios e validar o produto faltante. A validação foi realizada pela técnica de visão computacional, conhecida como *Scale Invariant Feature Transform* (SIFT). Ela extrai os pontos característicos nas imagens do produto faltante e das prateleiras. A partir disso, foi realizada a comparação dos pontos característicos entre as duas imagens para validar se o produto realmente estava em falta. O sistema fornece telas para cadastro de prateleiras, configurações de parâmetros, como alertas e intervalos de monitoramento, que foram desenvolvidas na linguagem de programação C++ utilizando a API *Microsoft Foundation Class* (MFC). Com isso, o objetivo é reduzir o tempo que o responsável leva para verificar e reabastecer as prateleiras, principalmente em áreas de grande movimento, garantindo a fidelidade do cliente com a marca e com o estabelecimento. Por fim, foram aplicados diversos cenários de teste para um conjunto de produtos de diferentes classes nas prateleiras de um grande varejista. O resultado obtido foi de 98% de acerto na identificação da posição e quantidade dos produtos nas prateleiras. No monitoramento de espaços vazios os resultados foram satisfatórios em todos os cenários aplicados, exceto no cenário em que o produto estava no fundo da prateleira.

Palavras-chave: Ruptura. Visão Computacional. Varejo. Processamento de Imagens. SIFT.

LISTA DE FIGURAS

Figura 1 – Visão geral do sistema	11
Figura 2 – Elementos de um sistema de processamento de imagens.....	18
Figura 3 – Sistema de visão computacional e suas principais etapas	19
Figura 4 – Cores primárias e cubo que representa o modelo de cores RGB.....	22
Figura 5 – Cilindro de cores HSV	23
Figura 6 – Processo de Limiarização.....	24
Figura 7 – Operação de Erosão.....	25
Figura 8 – Operação de Dilatação	26
Figura 9 – Exemplo de Histograma	27
Figura 10 – Detecção de bordas com algoritmo <i>Canny</i>	28
Figura 11 – Matriz de células acumuladoras e a representação dos valores θ e ρ	29
Figura 12 – Senoides geradas pela transformada de <i>Hough</i>	29
Figura 13 – Pirâmide de oitavas gaussianas	31
Figura 14 – Imagens resultantes da Diferença de Gaussianas (DoG).....	32
Figura 15 – Comparação de máximos e mínimos no espaço escala.....	33
Figura 16 – Mapa de orientação de gradientes (a) e histograma de orientação (b)	35
Figura 17 – Descritor de um ponto chave	36
Figura 18 – Etapas para identificação de ruptura	40
Figura 19 – Fluxograma da interface e monitoramento de ruptura	44
Figura 20 – Tela de inicialização da aplicação	46
Figura 21 – Tela principal da aplicação	47
Figura 22 – Diálogo para cadastro de produtos.....	48
Figura 23 – Seleção de prateleiras	49
Figura 24 – Seleção do produto.....	49
Figura 25 – Diálogo de configurações gerais da aplicação.....	50
Figura 26 – Resultado utilizando 3 tamanhos de matrizes diferentes para o elemento estruturante.....	53
Figura 27 – Resultado do processamento utilizando a transformada probabilística de Hough para o produto Leite Ninho.....	57
Figura 28 – Resultado do processamento utilizando a transformada probabilística de Hough para o produto Farinha de trigo Garota	57

Figura 29 – Imagens resultantes do cadastro de produtos	59
Figura 30 – Resultados da operação de dilatação.....	61
Figura 31 – Resultados para alguns produtos faltando na gôndola.....	66
Figura 32 – Resultados para a falta de produtos em uma ou mais prateleiras da gôndola.....	68
Figura 33 – Resultados produtos no fundo da prateleira	69
Figura 34 – Resultados para produto indisponível na gôndola	70

LISTA DE TABELAS

Tabela 1 – Elemento Estruturante 5 x 5 em forma de cruz.....	25
Tabela 2 – Resultados da operação de limiarização com valor limiar igual a 45.....	51
Tabela 3 – Resultados da operação de limiarização com valor limiar igual a 60.....	51
Tabela 4 – Resultados da operação de limiarização com valor limiar igual a 75.....	52
Tabela 5 – Resultados da operação de detecção de bordas para valores de limiar 45, histerese mínimo 45 e máximo 140	54
Tabela 6 – Resultados da operação de detecção de bordas para valores de limiar 60, histerese mínimo 60 e máximo 140	54
Tabela 7 – Resultados da operação de detecção de bordas para valores de limiar 75, histerese mínimo 75 e máximo 140	55
Tabela 8 – Resultados da transformada de Hough para valores de limiar 5 e tamanho mínimo de linha 10	55
Tabela 9 – Resultados da transformada de Hough para valores de limiar 10 e tamanho mínimo de linha 10	56
Tabela 10 – Resultados da transformada de Hough para valores de limiar 20 e tamanho mínimo de linha 10	57
Tabela 11 – Resultados obtidos para o cadastro de produtos.....	58
Tabela 12 – Resultado do cadastro de produto utilizando SIFT	60
Tabela 13 – Número de contornos obtidos utilizando elemento estruturante de 3x3 para dilatação	61
Tabela 14 – Número de contornos obtidos utilizando elemento estruturante de 4x4 para dilatação	62
Tabela 15 – Número de contornos obtidos utilizando elemento estruturante de 5x5 para dilatação	62
Tabela 16 – Resultados com produto disponível na gôndola	64
Tabela 17 – Resultados com correção da validação de área dos produtos.....	64
Tabela 18 – Resultados para alguns produtos faltando na gôndola	65
Tabela 19 – Resultados para a falta de produtos em uma ou mais prateleiras da gôndola	67
Tabela 20 – Resultados produtos no fundo da prateleira	68
Tabela 21 – Resultados para produto indisponível na gôndola	69
Tabela 22 – Resultados para reposição de produtos na gôndola.....	70

LISTA DE SIGLAS

API	<i>Application Programming Interface</i> (Interface de Programação de aplicativos)
HSV	<i>Hue, Saturattion, Value</i> (Matiz, Saturação e Valor)
IP	<i>Internet Protocol</i> (Protocolo de Internet)
MFC	<i>Microsoft Foundation Class</i> (Classes Fundamentais da Microsoft)
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde e Azul)
SIFT	Scale Invariant Feature Transform (Transformação de recurso invariante de escala)
SSID	<i>Service Set IDentifier</i> (Identificador de Conjunto de Serviços)
SURF	Speed Up Robust Features
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)

SUMÁRIO

1 INTRODUÇÃO	11
1.1 ESCOPO DO PROJETO	13
1.2 OBJETIVOS.....	13
1.2.1 Objetivos específicos	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 RUPTURA NO VAREJO.....	14
2.2 IMAGENS DIGITAIS	16
2.3 PROCESSAMENTO DE IMAGENS.....	17
2.4 SISTEMAS DE VISÃO ARTIFICIAL.....	18
2.4.1 Domínio do Problema e Resultado.....	19
2.4.2 Aquisição da Imagem	20
2.4.3 Pré-Processamento	20
2.4.4 Técnicas de segmentação	20
2.4.5 Extração de Características	21
2.4.6 Reconhecimento e Interpretação	21
2.4.7 Base de Dados	22
2.5 MODELO DE CORES.....	22
2.6 LIMIAZIZAÇÃO	23
2.7 OPERAÇÕES MORFOLÓGICAS	24
2.7.1 Erosão.....	25
2.7.2 Dilatação.....	25
2.8 HISTOGRAMA.....	26
2.9 DETECTORES DE <i>BLOBS</i>	27
2.10 DETECTOR DE BORDAS	27
2.11 TRANSFORMADA DE HOUGH PARA LINHAS.....	28
2.12 TRANSFORMAÇÃO DE RECURSO INVARIANTE DE ESCALA (SIFT).....	30
2.12.1 Espaço de Escala.....	30
2.12.2 Localização de Pontos Chave.....	33
2.12.3 Orientação dos Descritores	34
2.12.4 Descritor do Ponto Chave	35
2.13 BIBLIOTECA OPENCV.....	36
2.14 CÂMERAS IP.....	37

3 METODOLOGIA.....	39
3.1 LISTA DE MATERIAIS.....	39
3.2 SOFTWARE	40
3.3 CADASTRO DE PRODUTOS.....	41
3.3.1 Pré-Processamento	41
3.3.2 Segmentação	41
3.3.3 Reconhecimento e Interpretação	42
3.4 IDENTIFICAÇÃO DE ESPAÇOS VAZIOS.....	42
3.4.1 Pré-Processamento	42
3.4.2 Segmentação	42
3.5 RECONHECIMENTO E INTERPRETAÇÃO.....	43
3.6 INTERFACE	43
3.6.1 Cadastro de Produtos	44
3.6.2 Configuração de Parâmetros	45
3.6.3 Monitoramento	45
4. RESULTADOS	46
4.1 INTERFACE	46
4.1.1 Tela de Inicialização	46
4.1.2 Tela Principal	47
4.1.2.1 Visualizar Descrição, Quantidade e Status Atual do Produto	47
4.1.2.3 Cadastrar Novos Produtos.....	48
4.1.2.4 Alterar Configurações Gerais da Aplicação e Iniciar/Interromper Monitoramento dos Produtos	50
4.2 PROCESSAMENTO DE IMAGENS NO CADASTRO DE PRODUTOS	50
4.2.1 Aplicação da Operação de Limiarização	51
4.2.2 Aplicação da Operação de Erosão	53
4.2.3 Aplicação da Operação de Detecção de Bordas.....	53
4.2.4 Aplicação da Transformada Probabilística de Hough	55
4.2.5 Identificação da Posição e Quantidade de Produtos.....	58
4.2.6 Identificação de Pontos Chave Utilizando Algoritmo SIFT	59
4.3 PROCESSAMENTO DE IMAGENS NA IDENTIFICAÇÃO DE ESPAÇOS VAZIOS....	60
4.3.1 Operação de Dilatação	60
4.3.2 Aplicação da Operação de Busca de Contornos	61

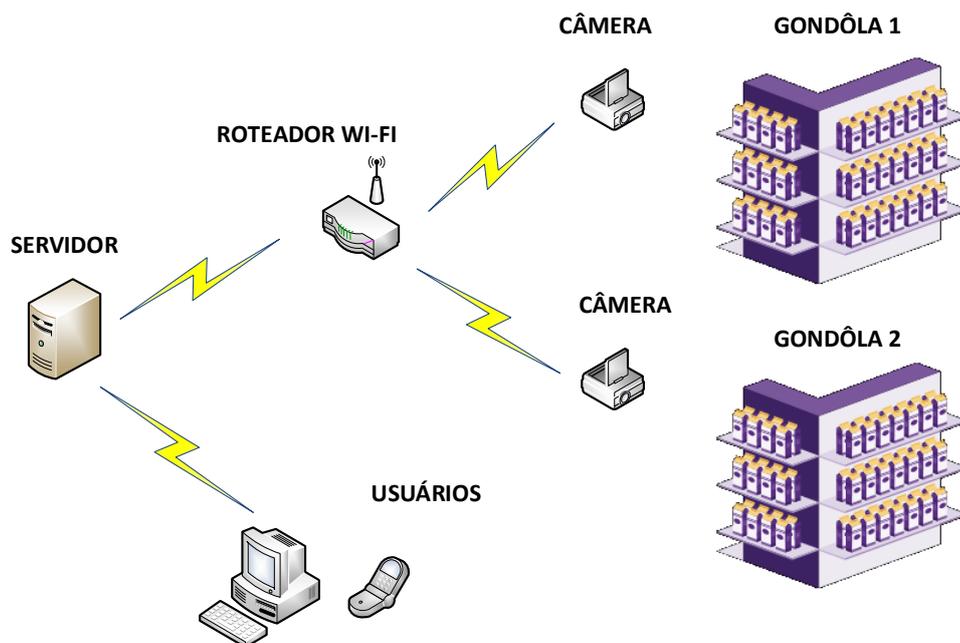
4.4 GERENCIAMENTO DE ALERTAS.....	63
4.4.1 Produto Disponível da Gôndola	63
4.4.2 Alguns Produtos Faltando na Gôndola	65
4.4.3 Produto em Falta em uma ou Mais Prateleiras da Gôndola.....	67
4.4.4 Produto no Fundo da Prateleira	68
4.4.5 Produto Indisponível na Gôndola.....	69
4.4.6 Produto Reposto na Gôndola	70
4.5 SÍNTESE DOS RESULTADOS OBTIDOS	71
5 CONCLUSÃO E TRABALHOS FUTUROS.....	72
REFERÊNCIAS	74

1 INTRODUÇÃO

Um dos grandes problemas enfrentados pelo setor varejista é a ruptura de produtos nas prateleiras — termo utilizado para caracterizar a falta de um produto em estoque ou a indisponibilidade nas prateleiras dos estabelecimentos. Atualmente, a identificação de ruptura é realizada através da auditoria visual, ou seja, o responsável observa as prateleiras de toda a loja, buscando a identificação de um produto faltante. No entanto, esse processo é demorado e ineficaz, pois na grande maioria dos casos o responsável pelo abastecimento não tem conhecimento da localização exata de cada item que deveria estar exposto. Além disso, a ruptura é responsável por causar prejuízos tanto financeiros quanto para a imagem da empresa, pois o cliente pode deixar de realizar a compra ou até mesmo procurar o concorrente, fazendo com que a empresa perca a fidelidade com o consumidor. (MESQUITA; LOOS, 2017).

Diante da problemática da falta de produtos nas prateleiras, se faz necessário o desenvolvimento de um sistema para identificação automática de ruptura, mostrado de forma geral pela figura 1. Por meio dele, tornar-se-á possível gerar um alerta visual aos responsáveis, para que assim realizem o reabastecimento das prateleiras com rapidez e eficiência, garantindo a fidelidade do cliente com a marca e com o estabelecimento.

Figura 1 – Visão geral do sistema



Fonte: Elaborada pelo autor (2018).

O algoritmo utilizado pelo sistema para identificação de ruptura nas prateleiras foi desenvolvido na linguagem de programação C++, utilizando a metodologia de processamento de imagens e visão computacional, disponibilizadas pela biblioteca *Open Source Computer Vision Library* (OpenCV). (OPEN SOURCE COMPUTATION VISION. (OPENCV), 2019). Assim, na imagem da gôndola capturada por meio de uma câmera digital aplicam-se três etapas:

- a) pré-processamento: técnicas de remoção de ruídos e isolamento de regiões (prateleiras) na imagem bruta;
- b) segmentação: de técnicas de separação de objetos, remoção de imperfeições e sistema de cores;
- c) reconhecimento e interpretação: utilização do algoritmo *Scale Invariant Feature Transform* (SIFT), que tem por objetivo identificar pontos chave e descritores de uma cena ou objeto. (GONZÁLES, 2014). O objetivo é validar a existência de ruptura, comparando os descritores do produto e da gôndola. Se não encontrar equivalências, o sistema interpreta como válida a existência de ruptura.

A interface da aplicação foi desenvolvida para o sistema operacional Windows 10, utilizando a API *Microsoft Foundation Class* (MFC), disponibilizada pela *Microsoft*. (MICROSOFT, 2016). Por meio da aplicação, além do monitoramento de gôndolas, o usuário poderá realizar o cadastro das prateleiras, assim como a configuração de parametrização da aplicação, tanto para o algoritmo de identificação quanto para a forma de comunicação dos responsáveis quando ocorrerem os eventos de ruptura.

Os resultados obtidos com o desenvolvimento da aplicação e dos algoritmos de processamento de imagens demonstrou que o problema de ruptura pode ser automatizado utilizando a visão computacional. Foram aplicados diversos cenários de teste em seis produtos de diferentes classes. Os testes aplicados foram: remoção de produtos da prateleira; prateleiras vazias; produto em falta; produtos ao fundo da prateleira; e, por fim, sua reposição na gôndola. Assim, chegou-se à conclusão de que a técnica de análise por posição dos produtos e aplicação da limiarização é a mais eficaz para a solução do problema. Já o algoritmo SIFT mostrou-se ineficaz, pois o método *match* não consegue obter resultados satisfatórios comparando duas imagens com os mesmos produtos capturadas em momentos distintos. Porém, sua utilização pode ser repensada para tornar-

se uma ferramenta complementar ao método de limiarização para validar se o produto encontra-se ou não disponível na sua respectiva posição.

1.1 Escopo do projeto

Pertencem ao escopo deste projeto as atividades referentes ao desenvolvimento de um software para identificação de ruptura nas gôndolas do varejo. O software primeiramente analisou fotos estáticas capturadas com uma câmera genérica. Além disso, foi utilizado um computador comum que assumiu o papel de servidor e também de dispositivo de análise dos alertas pelo usuário. Sendo assim, está fora de escopo o projeto de infraestrutura, que abrange todo o sistema para captura de imagens das gôndolas.

1.2 Objetivos

O principal objetivo deste trabalho, é desenvolver um sistema automatizado para identificação de ruptura nas gôndolas de supermercados ou hipermercados reduzindo o tempo que o responsável leva para verificar e reabastecer as prateleiras, principalmente em áreas e horários de grande movimento.

1.2.1 Objetivos específicos

Visando realizar o objetivo geral, divide-se nos seguintes objetivos específicos:

- a) desenvolver um sistema analisando imagens para identificação de ruptura nas prateleiras das gôndolas;
- b) incorporar o uso das técnicas de processamento de imagens e visão computacional para garantir a confiabilidade dos resultados;
- c) substituir o método tradicional de auditoria visual por um sistema automatizado.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a revisão de textos, artigos e livros sobre visão computacional, com o foco nas técnicas utilizadas para solucionar o problema de ruptura no varejo.

2.1 RUPTURA NO VAREJO

O comércio varejista tem realizado algumas mudanças para tornar-se competitivo, atrativo e inovador, objetivando a busca de novos adeptos e a garantia de fidelidade dos clientes já existentes. E é nesse propósito que os gestores focam, pois suas ações estratégicas e operacionais buscam atender as expectativas dos consumidores, alcançar a satisfação de suas necessidades e proporcionar uma experiência agradável de compra, o que acaba contribuindo para os negócios, além de gerar competitividade de mercado. (MESQUITA; LOOS, 2017).

Ainda assim, uma das maiores dificuldades enfrentadas pelo setor varejista, em todo o mundo, tem sido a ruptura de produtos nas gôndolas. Cabe destacar que gôndola é um termo utilizado para caracterizar um conjunto de prateleiras existentes em um supermercado ou hipermercado, na qual os produtos ficam expostos para que possam ser adquiridos e posteriormente consumidos pelos clientes. Ruptura, por sua vez, é empregada para caracterizar a falta de um produto em estoque, ou ainda quando ele existe, mas não se encontra disponível na gôndola. Dessa forma, esse problema vem sendo um dos principais motivos de descontentamento dos consumidores, o que compromete o nível de fidelização, tanto do estabelecimento quanto da marca, e influencia na perda de vendas e lucratividade do setor. Por esse motivo, é de suma importância que as gôndolas estejam sempre abastecidas, disponibilizando toda a variedade de produtos existentes aos clientes.

Os fatores responsáveis pelas causas de ruptura são decorrentes das práticas ineficientes nas operações de loja, sendo de responsabilidade da atuação interna. Segundo Mesquita e Loss (2017), as causas da ruptura no varejo podem ser classificadas em três grandes grupos, que são: problemas de planejamento, de processamento de pedido e de reposição.

Encontra-se dificuldade em mensurar os custos da ruptura no setor varejista, pois é algo que está totalmente voltado ao comportamento, satisfação e poder de decisão dos

clientes. De acordo com Mesquita e Loss (2017), ao ocorrer uma ruptura, o cliente pode optar por substituir o item desejado, sair da loja e esquecer-se da compra, procurá-lo em um outro estabelecimento ou nem mesmo comprá-lo novamente, o que acaba interferindo na perda de vendas e evidenciando a dificuldade de estimar o custo da ruptura, devido a impossibilidade de avaliar e quantificar cada uma das decisões escolhidas pelos clientes. A partir disso, entende-se que a ruptura é responsável por causar um impacto ao fabricante e ao varejista, assim como em toda a cadeia de abastecimento, complicando a avaliação das perdas.

Buscando identificar quais são os métodos empregados pelos gestores dos supermercados para quantificar o nível de ruptura de seus estabelecimentos, Soares et al. (2018) interrogou empresas do ramo varejista de Sousa, Paraíba, a respeito de como elas quantificam a falta de produtos nas prateleiras. Para tanto, foi utilizada uma lista com sete opções de métodos, através dos quais o avaliado deveria assinalar uma ou mais alternativas. Dentre elas, destacam-se:

- a) auditoria visual da gôndola com horários e dias determinados por parte da equipe interna;
- b) auditoria visual da gôndola com horários e dias determinados por parte da equipe externa;
- c) estoque do produto zerado durante um período;
- d) vendas do produto zerado durante um período;
- e) perguntas diretas aos clientes sobre a falta de produtos quando estes se dirigiam ao caixa;
- f) informações levantadas com os fornecedores;
- g) outras (especificar);
- h) a empresa não quantifica a falta de produtos nas gôndolas.

A partir dos dados obtidos pela pesquisa de Soares et al. (2018), nota-se que o principal método utilizado pelos gestores é a auditoria visual da gôndola, na qual o gestor da loja ou o responsável pelo abastecimento busca a identificação dos itens faltantes visualmente. Porém, essa metodologia apresenta algumas falhas, pois muitas vezes o varejista não possui a relação de itens que deveriam estar expostos em determinado local, além de demandar certo tempo até a identificação do item faltante. Esse período pode ser

o suficiente para que o cliente não encontre o produto desejado ou até mesmo nem realize outras compras.

Para analisar as causas de ruptura, Araújo e Martins (2010) realizaram uma pesquisa, utilizando o Supermercado San Center, do ramo alimentício, localizado na Paraíba, a fim de verificar o índice de ruptura e ainda o comportamento dos clientes quanto a falta dos itens nas gôndolas. Para a identificação de ruptura foi utilizada a observação direta intensiva, na qual busca-se verificar os produtos faltantes nas gôndolas durante o período examinado. Constatou-se, então, que a falha no departamento de compras e na reposição dos itens são os principais motivos de ruptura. A falha na reposição significa que o produto estava disponível no estoque, mas não foi colocado na gôndola. Ainda de acordo com a análise realizada por Araújo e Martins (2010), foram verificadas as consequências da ruptura em relação ao comportamento do consumidor, identificando-se que 34% dos clientes buscavam o produto em um outro supermercado. Sendo assim, ocorria a perda da venda, o que causa um prejuízo para o varejista e para a indústria. Em outro cenário, foi constatado que 14% dos clientes adiavam suas compras, resultando, portanto, na perda da credibilidade frente ao consumidor.

Perante os dados levantado por Soares et al. (2018) e Araújo e Martins (2010), é possível identificar que o principal método empregado na validação da disponibilidade de um item na gôndola é a auditoria visual e que a falta de produtos nas gôndolas, mesmo quando está em estoque, é um dos principais causadores da ruptura no varejo.

Por esse motivo, a técnica de visão computacional será empregada no estudo em questão, objetivando a substituição da auditoria visual por um sistema inteligente através do processamento de imagens capturadas das gôndolas. Assim, a técnica irá reduzir o percentual de falta dos produtos, os custos do processo e ainda aumentará a fidelidade dos clientes com a marca e com o estabelecimento. O tópico 2.2 irá apresentar o conceito de imagens digitais que será o objeto de entrada do sistema para análise através das técnicas de visão computacional.

2.2 IMAGENS DIGITAIS

O termo imagem digital refere-se a uma imagem $f(x,y)$ discretizada, tanto espacialmente quanto em brilho. Sendo assim, ela pode ser vista como uma matriz, cujas linhas e colunas identificam um ponto na imagem, e o valor do elemento da matriz

corresponde ao nível de cinza naquele ponto. Os elementos dessa matriz digital são chamados de pixels, que significam elementos de figura. (FILHO; NETO, 1999). O tópico 2.3 apresenta os elementos para o processamento de imagens digitais.

2.3 PROCESSAMENTO DE IMAGENS

A área de processamento de imagens vem se destacando, pois permite diversas aplicações em duas categorias bem distintas. A primeira está associada ao aprimoramento das informações das imagens para a interpretação do ser humano. Já a segunda refere-se a análise de informações que uma cena contém através de um computador. (FILHO; NETO, 1999).

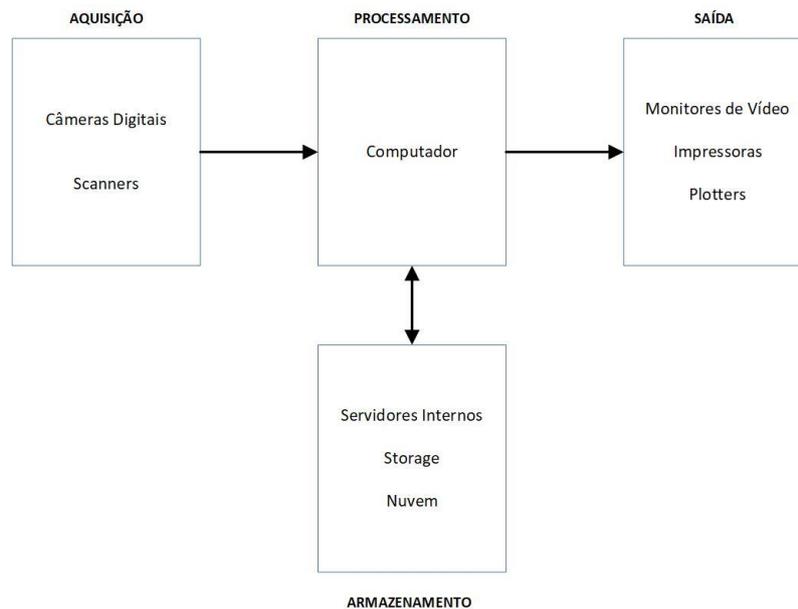
Para a realização das referidas aplicações, normalmente é necessário um sistema de processamento de imagens, que possui cinco principais elementos:

- a) aquisição: adquire-se cenas do mundo real (tridimensional) através de câmeras ou sensores que capturam a incidência de luz. A função da aquisição é converter uma cena em uma representação numérica, apropriada para o processamento digital;
- b) armazenamento: existem basicamente três tipos de armazenamento de dados:
 - servidores internos: possuem aplicações e arquivos comuns, que são acessados por meio de computadores conectados;
 - dispositivos de *Storage*: são exclusivamente dedicados ao armazenamento de dados e arquivos. Funcionam como HDs externos e não utilizam uma interface (sistema operacional, por exemplo);
 - serviços de armazenamento em nuvem: os arquivos, dados e aplicações são armazenados nos data centers de um fornecedor especializado;
- c) processamento: é uma etapa responsável pela realização de processos expressos por meio de algoritmos, tendo a maioria de suas funções realizadas via software. O processamento de imagens tem por objetivo realizar operações específicas. Assim, algumas técnicas podem ser muito úteis em determinada área, mas podem não funcionar em outras;

- d) comunicação: etapa que apresenta a comunicação entre dois ou mais sistemas de processamento de imagens, realizada por meio de protocolos de rede existentes, tais como TCP, UDP, etc;
- e) exibição: etapa que apresenta o resultado do processamento de imagens em dispositivos de exibição, como os monitores, por exemplo.

A figura 2 apresenta um diagrama com os elementos mencionados acima, os quais permitem descrever desde sistemas de baixo custo até sistemas complexos, envolvendo intenso uso de imagens. No presente estudo, serão utilizados todos os elementos: aquisição, utilizando câmeras digitais para obtenção das imagens da gôndola; processamento de imagens; armazenamento de imagens dos produtos e das gôndolas; e por fim, a visualização dos alertas e imagens da gôndola em monitores de vídeo.

Figura 2 – Elementos de um sistema de processamento de imagens



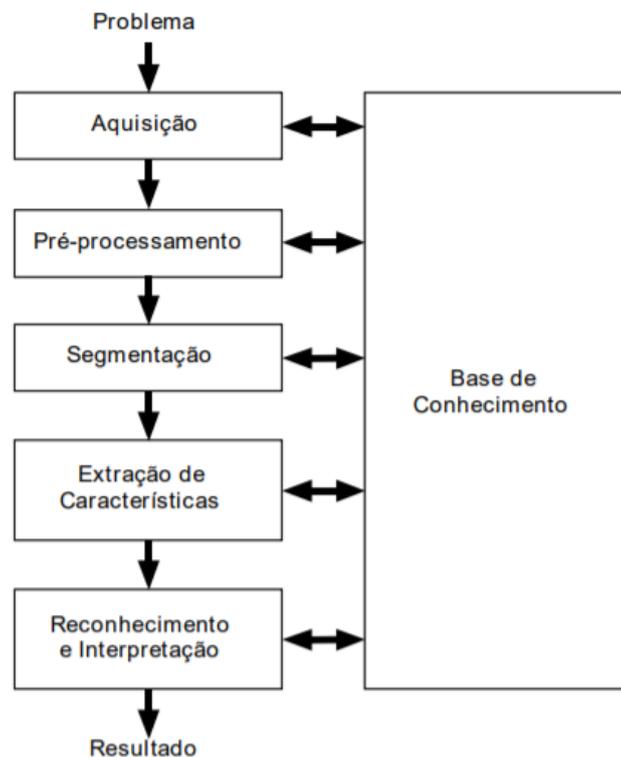
Fonte: Adaptado de Filho e Neto (1999).

2.4 SISTEMAS DE VISÃO ARTIFICIAL

Um sistema de visão artificial é um sistema computadorizado, responsável por adquirir, processar e interpretar imagens correspondentes a cenas reais. (FILHO; NETO,

1999). A figura 3 apresenta suas principais etapas, sendo que cada uma delas será explicada utilizando o exemplo prático para identificação de ruptura na gôndola.

Figura 3 – Sistema de visão computacional e suas principais etapas



Fonte: Filho e Neto (1999).

Um sistema de visão artificial é formado pelas etapas de aquisição, pré-processamento, segmentação, extração de características, reconhecimento e interpretação e por fim uma base de conhecimento. Todas essas etapas serão explicadas nos tópicos a seguir.

2.4.1 Domínio do Problema e Resultado

O domínio do problema, no estudo em questão, consiste na identificação da existência de ruptura em uma gôndola de supermercado. O resultado aguardado é uma imagem com a localização exata da falta desse produto e a identificação do produto que se encontra em falta na gôndola.

2.4.2 Aquisição da Imagem

Em um primeiro momento, é necessária a aquisição da imagem da gôndola. Para isso, será utilizado um dispositivo de captura de imagem, que pode ser uma câmera digital. Ela será responsável pela conversão da informação ótica e pela realização da digitalização da imagem. Ao final, essa etapa terá como resultado uma imagem digitalizada da gôndola.

2.4.3 Pré-Processamento

A imagem digital resultante pode apresentar diversas imperfeições, como presença de pixels ruidosos, contraste e/ou brilho inadequado, entre outros. E é por isso que essa etapa tem como função aprimorar a qualidade da imagem, buscando o aumento das chances de sucesso nas próximas etapas.

As intervenções realizadas no pré-processamento são de baixo nível, pois trabalham diretamente com valores de intensidade dos pixels. Como exemplos das técnicas utilizadas, destacam-se o realce de contrastes, a remoção de ruído e o isolamento de regiões. Como resultado, obtém-se uma imagem digitalizada da gôndola, na qual se verifica uma qualidade superior comparada a original. No estudo em questão, serão utilizados a remoção de brilho e o isolamento de regiões, este último servindo para dividir a gôndola em prateleiras.

2.4.4 Técnicas de segmentação

Segmentação é o processo que tem por objetivo realizar a subdivisão de uma imagem em regiões ou em objetos de interesse que a compõem. Existem diversas dificuldades dentro do processo de segmentação, tais como: a falta de nitidez para consideração de bordas ou fronteiras das regiões, as variações de iluminação e a existência de sombras ou de movimentos que tendem a gerar uma falsa impressão da região a ser segmentada. Por isso se torna importante controlar o ambiente de captura e a utilização das técnicas de pré-processamento, o que irá facilitar a interpretação da imagem. (FILHO; NETO, 1999).

Conforme Gonzalez e Woods (2013), os algoritmos de segmentação, para imagens monocromáticas, têm como base duas propriedades básicas de níveis de cinza: a descontinuidade e a similaridade. Na descontinuidade, a imagem é dividida, baseando-se

nas mudanças bruscas dos níveis de cinza, tendo como foco as técnicas de detecção de pontos isolados e de linhas e bordas em uma imagem. Já a similaridade tem como base as técnicas de limiarização, crescimento de regiões e divisão e fusão de regiões. Ambos os conceitos podem ser aplicados em imagens fixas ou em vídeos.

Devido as técnicas de segmentação abrangerem uma grande área dentro dos sistemas de processamento de imagens, serão abordados somente os itens pertinentes ao escopo desse projeto, que são: limiarização, detecção de bordas, detecção de linhas, sistema de cores *Red, Green, Blue* (RGB) e *Hue, Saturation, Value* (HSV), detectores de *blobs* e identificação de contornos.

2.4.5 Extração de Características

Dentro do processo, essa etapa busca extrair características das imagens resultantes da segmentação. A entrada ainda será uma imagem, mas o resultado é um conjunto de dados correspondentes àquela imagem. (FILHO; NETO, 1999). Suponha-se que para a identificação do produto em falta na gôndola, seja necessário extrair os seus pontos característicos através do algoritmo SIFT e realizar a busca na gôndola para validar a existência de ruptura.

2.4.6 Reconhecimento e Interpretação

A última etapa do sistema é o reconhecimento e interpretação. O reconhecimento tem como denominação o processo de atribuição de um rótulo a um objeto, tendo em base suas características, que são traduzidas pelos dados resultantes da etapa anterior. (GONZALEZ; WOODS, 2013). No exemplo de ruptura, seria o reconhecimento dos espaços vazios na gôndola, sinalizados através de contornos. Já a interpretação procura dar um significado a um grupo de objetos que já foram reconhecidos. (GONZALEZ; WOODS, 2013). Retornando ao exemplo, esta etapa realiza a busca do produto faltante na gôndola para interpretar se realmente existe ruptura.

2.4.7 Base de Dados

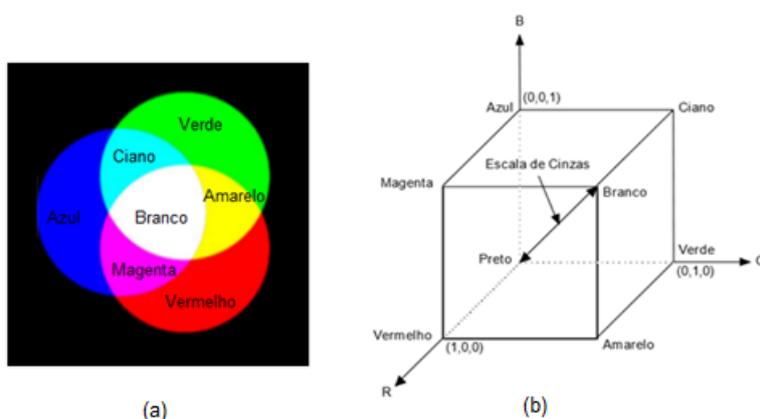
Todas as etapas mencionadas anteriormente necessitam do conhecimento sobre o problema que deve ser resolvido, armazenado em uma base de dados, que poderá variar em tamanho e complexidade. Ela pode ser simples, ao ponto de apresentar as regiões de interesse na imagem de uma gôndola — no caso das prateleiras de produtos — ou ainda ser complexa, a ponto de ter uma base com todas as informações necessárias, como a localização de um produto na gôndola e suas principais características, a fim de facilitar a identificação do produto faltante.

A integração através da base de dados ainda é um objetivo de difícil alcance e não está presente na maioria dos sistemas de visão artificial existentes atualmente. (GONZALEZ; WOODS, 2013).

2.5 MODELO DE CORES

O principal objetivo de um modelo de cores é uma padronização de aceite geral. Existem diversos modelos de cores, porém o tópico atual ficará restrito ao RGB e HSV, que serão utilizados para o desenvolvimento do projeto.

Figura 4 – Cores primárias e cubo que representa o modelo de cores RGB



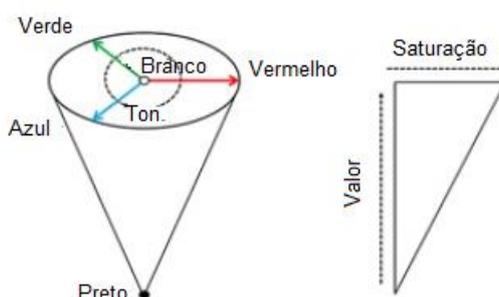
Fonte: Adaptado de Filho e Neto (1999).

O modelo RGB é baseado em coordenadas cartesianas, representado por um cubo, conforme figura 4 (b), na qual três vértices são as cores primárias (vermelho, verde e azul) e as outras três, secundárias (magenta, ciano e amarelo). Já o preto está no vértice junto a

origem, e o branco é o mais afastado em relação a origem. No mesmo modelo, outras cores são obtidas pela soma de porções das três cores primárias: vermelho, verde e azul (Figura 4a).

O modelo HSV é formado pelos componentes de tonalidade (hue), saturação (saturation) e valor (value). O HSV utiliza um cilindro de cores (Figura 5), no qual o ângulo do círculo é uma escala de cinza e representa a tonalidade, a distância para o centro refere-se à saturação e a altura representa o brilho. Os componentes matiz e saturação estão ligados à percepção humana das cores e essas características o tornam uma ótima ferramenta para o desenvolvimento de algoritmos de processamento de imagens. (GARCÍA-LAMONT; CUEVAS; NIÑO, 2016).

Figura 5 – Cilindro de cores HSV



Fonte: Adaptado de Garcia-Lamont, Cuevas e Niño (2016, p. 80).

2.6 LIMIAZIZAÇÃO

Quando há a necessidade de separar ou de extrair os objetos do fundo de uma imagem, a limiarização tem sido uma das operações de segmentação mais utilizadas. Para realizar essa separação, é necessária a seleção de um limiar T que separe os dois grupos. Dessa forma, cada ponto no plano cartesiano (x, y) , tal que $f(x, y) > T$, é denominado um ponto do objeto; caso contrário, o ponto é denominado fundo.

A técnica utiliza um limiar único T , que busca realizar o particionamento do histograma da imagem. A segmentação deve ser realizada, passando pixel a pixel da imagem e validando se o nível do pixel é maior ou menor que T . Sendo assim, uma imagem limiarizada $g(x, y)$ é definida como:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) > T \\ 0, & \text{se } f(x,y) \leq T \end{cases} \quad (1)$$

Dessa forma, pixels definidos como 1 correspondem ao objeto. Já aqueles definidos como 0 correspondem ao fundo. Cabe ressaltar que quando T depender apenas de $f(x,y)$, o limiar será chamado global.

A figura 6 apresenta o processo de limiarização após a separação dos pixels, conforme o limiar aplicado. No resultado, foi aplicado um valor 1 (preto) para o fundo e 0 (branco) para o objeto. (BRADSKY; KAEHLER, 2008).

Na parte superior, verifica-se a imagem original; inferior, uma imagem limiarizada com $T = 4$.

Figura 6 – Processo de Limiarização



Fonte: Elaborada pelo autor (2018).

A técnica de limiarização é premissa para aplicação das próximas etapas de processamento de imagens. O próximo tópico apresenta operação para remoção de imperfeições geradas por esta técnica.

2.7 OPERAÇÕES MORFOLÓGICAS

As operações morfológicas são muito utilizadas para remoção de imperfeições na estrutura de imagens que passam por operações de limiarização. Para a sua realização é necessária a utilização de uma imagem limiarizada ou em escala de cinza e um elemento estruturante/*kernel* (Tabela 1). As principais operações morfológicas são: Erosão e Dilatação.

Tabela 1 – Elemento Estruturante 5 x 5 em forma de cruz

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

Fonte: Adaptada de Gonzalez e Woods (2013).

2.7.1 Erosão

A operação de erosão tem como objetivo remover as estruturas que são menores que o elemento estruturante. (RAVI; KHAN, 2013). Este, por sua vez, desliza sobre todos os pixels de uma imagem, analisando seus vizinhos. Caso todos os pixels sobre o elemento sejam 1, o pixel é validado; caso contrário, é modificado para 0. O tamanho do objeto será, portanto, reduzido. A operação é muito utilizada para remoção de ruídos ou separação e destaque de objetos em primeiro plano.

A figura 7 apresenta exemplo da operação de erosão. A imagem à esquerda (a) traz o resultado de uma operação de limiarização, e a direita (b), da operação de erosão sobre ela. É possível observar a redução e pixels ruidosos e uma maior separação entre os objetos em primeiro plano.

Figura 7 – Operação de Erosão



Fonte: Elaborada pelo autor (2019).

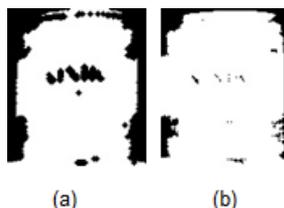
2.7.2 Dilatação

A operação de dilatação tem como objetivo aumentar o número de pixels em primeiro plano e reduzir o número daqueles com valor zero. Também é utilizado para preencher pixels vazios em um objeto contínuo. (RAVI; KHAN, 2013). Para isso, um elemento de pixel

será 1 se pelo menos um pixel sob o elemento estruturante for igual a 1. A operação de dilatação faz com que o objeto em primeiro plano fique maior.

A figura 8 apresenta o resultado da operação de dilatação em uma imagem. A imagem à esquerda (a) traz o resultado de uma operação de erosão, e à direita (b), o da operação de dilatação. É possível observar o aumento de pixels tornando maior o objeto em primeiro plano.

Figura 8 – Operação de Dilatação



Fonte: Elaborada pelo autor (2019).

2.8 HISTOGRAMA

O histograma, em imagens digitais, é uma função discreta, que busca representar a distribuição de valores de pixels, de acordo com seus níveis de cinza. São representados em forma de gráficos, capazes de informar um número ou percentual a cada nível de cinza na imagem. Seu objetivo principal é fornecer a qualidade da imagem, quanto a contraste e quanto ao seu brilho. (FILHO; NETO, 1999). A equação 2 representa a função discreta:

$$p(r_k) = \frac{n_k}{n} \quad (2)$$

Onde:

r_k = k-ésimo nível de cinza

n_k = número de pixels na imagem na qual o nível de cinza corresponde r_k

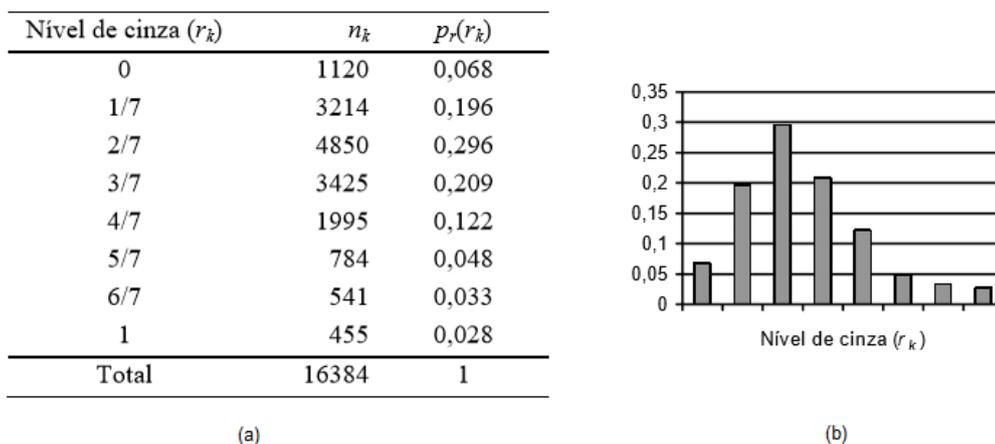
n = número total de pixels da imagem

$k = 0, 1, 2 \dots L-1$, onde L é o número de valores de pixel da imagem digitalizada

p = resultado da função discreta

A figura 9(a) demonstra os cálculos realizados para a obtenção dos níveis de cinza (r_k). A figura 9(b) apresenta o gráfico gerado por esses níveis de cinza.

Figura 9 – Exemplo de Histograma



Fonte: Filho e Neto (1999).

2.9 DETECTORES DE *BLOBS*

Em processamento de imagens, *blob* é definido como uma região de uma imagem onde todos os pixels são interpretados como semelhantes. Atualmente os *blobs* também são chamados de operadores de ponto de interesse ou operadores de região de interesse. Os descritores de *blobs* podem indicar pontos na imagem que são informativos para o reconhecimento de objetos com base em históricos locais. (SILVA, 2017). Neste estudo, os detectores *blobs* são utilizados no algoritmo SIFT utilizado para validação de produtos nas gôndolas.

2.10 DETECTOR DE BORDAS

Uma borda está localizada entre duas regiões que possuem níveis diferentes de cinza em uma imagem (GONZALEZ; WOODS, 2013). O objetivo de um detector de bordas é identificar os contornos de objetos dentro de uma imagem, através dos quais é possível obter informações importantes como tamanho, formato e até mesmo a sua localização.

Um dos mais populares algoritmos utilizados para detecção e bordas é o *Canny Edge Detector*. O detector *Canny* realiza uma sequência de etapas para se obter uma boa detecção de bordas. A primeira delas é realizar a remoção de ruídos da imagem, o que é feito através de uma convolução na imagem utilizando uma máscara gaussiana. Nessa etapa, o resultado é uma imagem levemente desfocada da original. (OPENCV, 2015).

A segunda etapa é realizar o cálculo dos gradientes. Como a borda pode estar em vários sentidos, são utilizadas 4 máscaras para detectar bordas horizontais, verticais e diagonais. A partir da aplicação realiza-se a convolução da imagem original com cada uma das máscaras. Sendo assim, para cada pixel é determinado o maior valor de gradiente e o sentido da máscara que produziu essas bordas. Com os valores obtidos, é realizada uma análise utilizando o limiar de histerese mínima e máxima. Os valores abaixo do mínimo são descartados e os valores acima do máximo são considerados bordas seguras. Já os que ficam entre os dois limiares são avaliados, e todos aqueles que de certa forma estão conectados a bordas seguras são considerados. Os demais são descartados. (OPENCV, 2015).

Figura 10 – Detecção de bordas com algoritmo *Canny*



Fonte: Elaborada pelo autor (2019).

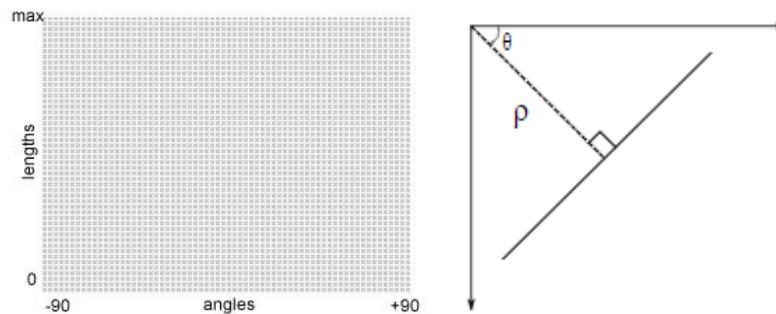
A figura 10 apresenta o resultado da detecção com bordas utilizando o algoritmo *Canny*. Essa imagem irá servir de entrada para o próximo tópico onde é realizada a identificação de linhas verticais.

2.11 TRANSFORMADA DE HOUGH PARA LINHAS

Trata-se de uma técnica para identificar formas em uma imagem, mesmo que ela esteja distorcida ou quebrada, sendo elas: linhas, círculos, elipse, entre outras. Em resumo, a transformada é aplicada após realizar um pré-processamento na imagem original através dos métodos de detecção de bordas e de limiarização. Após isto, o algoritmo percorre todos os pixels da imagem de contornos e realiza uma “votação” para dar maior peso aos pixels que já estão em uma linha. (SINHA, (2010?)). Em outras palavras a transformada de Hough converte cada pixel da imagem original em curvas no espaço de parâmetros.

Para identificar as linhas em uma imagem, o algoritmo realiza a montagem de uma matriz de células acumuladoras, representada na figura 11. O eixo horizontal mostra os diferentes valores de ângulo θ e o vertical, de distância ρ .

Figura 11 – Matriz de células acumuladoras e a representação dos valores θ e ρ



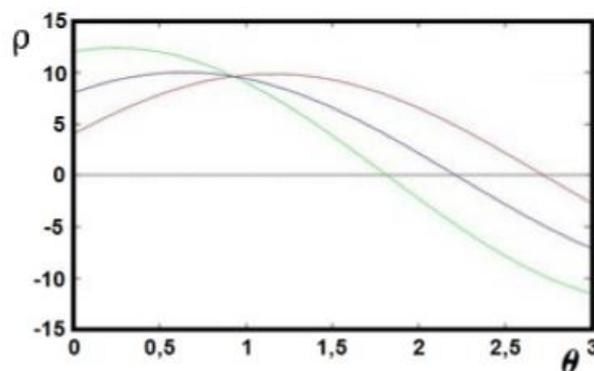
Fonte: Sinha (2010?).

Cada posição da matriz contém uma célula representada pelos valores (θ, ρ) . O algoritmo percorre cada pixel diferente de zero da imagem e para cada variação do ângulo θ é obtido um novo valor de distância ρ . Através da equação 3, uma curva senoidal é gerada no espaço de Hough para cada pixel da imagem original. A intersecção entre essas senoides formam a linha.

$$\rho = x \cos\theta + y \sin\theta \quad (3)$$

Para entendimento será utilizado um exemplo com três pontos de uma imagem: (4,9), (8,6) e (12,6). Variando o parâmetro θ e pelo correspondente cálculo de ρ , utilizando a equação 3, obtém-se três senoides no espaço de Hough, representados na figura 12. (JÚNIOR, 2016).

Figura 12 – Senoides geradas pela transformada de *Hough*



Fonte: Junior (2016, p. 37).

O ponto de intersecção entre as curvas indica os valores dos parâmetros θ e ρ , que formam a reta que passa pelos respectivos pontos (4, 9), (8, 6) e (12, 6).

No projeto do software para identificação de ruptura, será utilizada a transformada probabilística de *Hough* para identificação da posição e quantidade de produtos expostos para o cliente na gondola. A transformada probabilística utiliza o mesmo conceito da transformada normal, porém, ao invés de realizar o cálculo para cada pixel resultante das bordas, ela seleciona randomicamente esses pontos. Isso traz ganhos no processamento computacional e no descarte de pontos ruidosos da imagem.

2.12 TRANSFORMAÇÃO DE RECURSO INVARIANTE DE ESCALA (SIFT)

Dentro da visão computacional, muitas vezes é necessário encontrar pontos de equivalência em diferentes imagens, pois, sabendo dos pontos em comum entre elas, é possível extrair diversos tipos de informações. De uma maneira geral, eles referem-se às características que podem ser reconhecidas facilmente nas imagens. Dentre os atributos possíveis a serem encontrados em uma imagem destacam-se: arestas, cantos e blobs, que são classificados como regiões de interesse.

Existem diversos algoritmos utilizados para encontrar pontos de equivalência ou de interesse em imagens e objetos, no entanto, um bom algoritmo deve ser robusto quanto à iluminação, rotação, escala, entre outros. (COUTINHO, 2013). Sendo assim, o algoritmo SIFT, transformada de características invariantes a escala, será utilizado para encontrar produtos nas gôndolas por meio da comparação dos pontos da imagem do produto e da gôndola.

Cabe destacar que esse algoritmo é composto pelos seguintes estados: detecção de pontos no espaço de escala, aquisição, localização e descrição de pontos chave. Seguem as etapas do algoritmo nos próximos tópicos.

2.12.1 Espaço de Escala

O objetivo desta etapa é realizar a busca de pontos chave, que são invariantes a mudanças de escala da imagem. Para atingir esse objetivo, o primeiro passo é a construção de uma pirâmide de imagens, ou seja, a partir de uma imagem $I(x, y)$ novas serão geradas

por meio da aplicação incremental de um filtro de suavização gaussiano $G(x, y, \sigma)$. A função é representada pela equação 4:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4)$$

Onde:

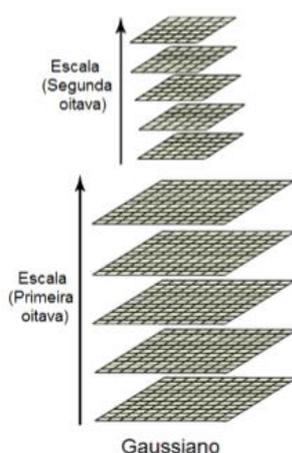
σ representa o desvio padrão do filtro gaussiano e define a escala da imagem filtrada

$L(x, y, \sigma)$

* representa a convolução em x e y

O resultado da aplicação incremental do filtro gera um conjunto de imagens que formam uma oitava. Segundo Lowe (2004), recomenda-se utilizar de 4 a 5 oitavas para um bom funcionamento do algoritmo. Então, para gerar uma nova oitava, uma dessas imagens tem seu tamanho reduzido pela metade e serve de entrada para novamente ser aplicado o filtro Gaussiano. Esse processo é repetido até atingir o número de oitavas recomendadas.

Figura 13 – Pirâmide de oitavas gaussianas



Fonte: Gonzáles (2014, p. 31).

A utilização do filtro gaussiano tem como objetivo obter amostras de imagens dos produtos e da gôndola, suavizar ou remover detalhes indesejados e ruídos e realçar suas principais características.

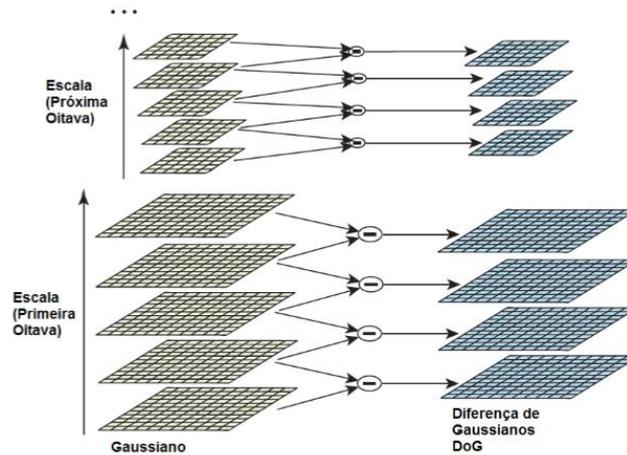
Com a pirâmide de gaussiano formada, é aplicada a técnica de Diferença de Gaussianas (DoG), através da qual calcula-se a distinções entre imagens filtradas com

escalas variadas por uma constante k . A equação 5 demonstra que o resultado da função DoG é obtido pela subtração das imagens filtradas em escalas próximas dentro de uma mesma oitava:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (5)$$

A figura 14, em seu lado direito, apresenta as imagens resultantes da diferença do gaussiano.

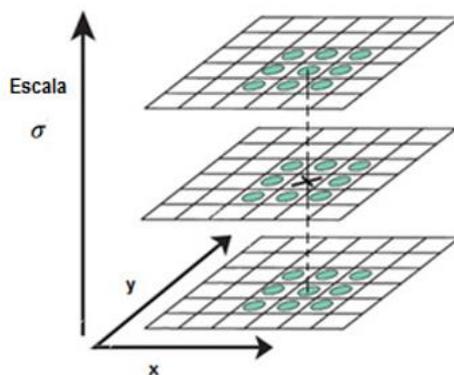
Figura 14 – Imagens resultantes da Diferença de Gaussianas (DoG)



Fonte: Adaptado de Lowe (2004, p. 6).

Com as DoG calculadas, as imagens são varridas em busca de seus máximos e mínimos no espaço de escalas. Esses valores são obtidos comparando a intensidade de cada ponto com os oito pixels vizinhos na escala atual. Assim, com os seus nove vizinhos na escala inferior, e outros nove na superior, conforme a figura 15, o “X” representa o pixel que está sendo analisado.

Figura 15 – Comparação de máximos e mínimos no espaço escala



Fonte: Adaptado de Lowe (2004, p. 7).

Caso o pixel seja um extremo local (possui maior ou menor valor dentre todos os 27 pixels vizinhos), o mesmo é considerado um potencial ponto chave da imagem do produto ou gôndola.

2.12.2 Localização de Pontos Chave

Após obtenção dos potenciais pontos chave, é necessário que seja realizado um refinamento, buscando resultados mais precisos, com o objetivo de descartar pontos gerados por ruídos ou pontos que não sejam considerados bons.

Os pontos com maior contraste (de maior frequência) são selecionados em torno do ponto chave aproximado, buscando a obtenção dos valores de intensidade dos sub pixels, ou seja, se a magnitude for menor que um determinado valor, o ponto chave é descartado.

Além de descartar os pontos de baixo contraste, o algoritmo também descarta potenciais pontos chave mal localizados ao longo de arestas, calculando os gradientes nos pontos. Com base na imagem em torno do ponto chave, existem três possibilidades:

- a) uma região plana: nessa região ambos os gradientes serão pequenos;
- b) uma aresta: nela um dos gradientes terá um valor alto (perpendicular à aresta) e o outro, um valor pequeno (ao longo da aresta);
- c) um canto: ambos os gradientes terão um alto valor.

Os cantos são os principais pontos chave, se ambos os gradientes forem obtidos e tiverem um valor satisfatório, o ponto chave é aceito. (GONZÁLES, 2014).

2.12.3 Orientação dos Descritores

O próximo passo é a identificação da orientação dos descritores, isto é, calcular orientações dos gradientes e magnitudes ao redor de cada ponto chave. Esses cálculos garantem invariância a rotações na imagem.

Calcula-se, então, para cada amostragem da imagem na escala, a magnitude utilizando a equação 6 e a orientação do gradiente utilizando a equação 7, usando a diferença de pixels, sendo m a magnitude, e θ a orientação dos gradientes:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (6)$$

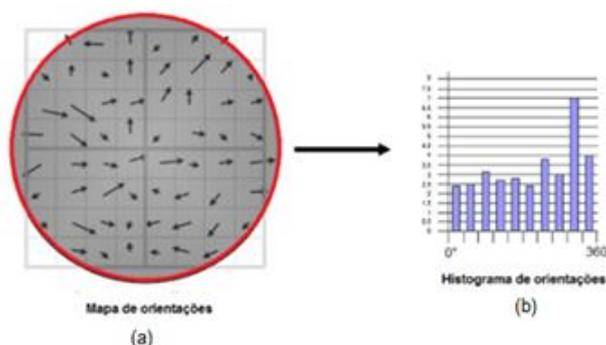
$$\theta(x, y) = \text{tang}^{-1} \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \quad (7)$$

A magnitude e a orientação são calculadas para todos os pixels ao redor do ponto chave analisado. Através desse resultado, é criado um histograma, no qual os 360° de orientação são divididos em 10 campos. Na montagem dos histogramas, a contribuição de cada ponto da vizinhança é classificada por pesos. (GONZÁLES, 2014). São eles:

- a) distância normalizada entre a orientação dos pixels e a orientação do ponto chave;
- b) valor da magnitude do gradiente;
- c) janela gaussiana circular com desvio padrão 1,5 vezes maior do que a escala do ponto chave.

Com esses pesos, o histograma é atualizado. A figura 16 apresenta o mapa de orientações representada pelas setas na imagem (a) e o histograma de orientação (b), sendo que ambas as partes exibem as direções dominantes dos gradientes.

Figura 16 – Mapa de orientação de gradientes (a) e histograma de orientação (b)



Fonte: Gonzáles (2014, p. 36).

Além do pico máximo, qualquer um acima de 80% é convertido em novos pontos chave, podendo assim um ser dividido em vários outros novos com suas respectivas orientações. Nesse caso, os pontos se tornam mais estáveis para futuras identificações. Com todas essas etapas realizadas, cada ponto chave do produto ou gôndola possui, portanto, posição x e y, magnitude e orientação.

2.12.4 Descritor do Ponto Chave

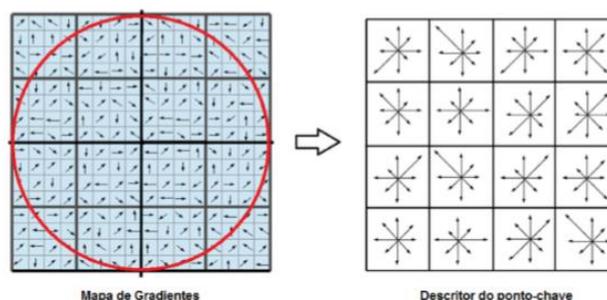
Neste tópico, o algoritmo irá atribuir a cada ponto chave um descritor invariante à iluminação e ponto de vista 3D. São necessários dois passos para obter os descritores, que serão utilizados para identificação de produtos na gôndola: cálculo dos gradientes da vizinhança de cada ponto chave e montagem do histograma.

Em relação ao primeiro passo, para entendimento, os gradientes ao redor do ponto chave foram demonstrados como pequenas setas no lado esquerdo da figura 16. Sendo assim, é utilizada uma janela de suavização gaussiana, com desvio padrão igual à metade da janela do descritor, o que dará peso à magnitude do gradiente de cada ponto vizinho. O filtro gaussiano é utilizado para não afetar o descritor em caso de pequenas mudanças na posição da janela e para reduzir a ênfase nos gradientes mais afastados, que são os mais afetados por erros. (GONZÁLES, 2014).

Já no segundo passo, é realizada a suavização dos gradientes. A vizinhança em torno do ponto chave é dividida em $n \times n$ regiões de $m \times m$ pixels. Para cada região, é montado um histograma de 8 direções, com base nos valores de magnitude dos pixels.

A figura 17 apresenta um exemplo de montagem do histograma. Há um conjunto de 4x4 de histogramas com 8 campos de acumuladores e, a partir disso, é gerado um vetor resultante de 128 elementos para cada ponto chave. (GONZÁLES, 2014).

Figura 17 – Descritor de um ponto chave



Fonte: Gonzáles (2014, p. 38).

É demonstrado na figura 17, um conjunto de 4x4 de histogramas com 8 campos de acumuladores, a partir disso, é gerado um vetor resultante de 128 elementos para cada ponto chave. (GONZÁLES, 2014).

É importante destacar que as orientações dos gradientes são medidas em relação à orientação do ponto chave. Dessa forma, o descritor se torna invariante quanto à rotação. No entanto, duas imagens contendo um mesmo objeto podem ter variações de luminosidade que podem afetar os descritores obtidos. Para resolver esse problema, é realizado uma normalização do vetor, impondo um limite máximo para que as direções e magnitudes muito elevadas não prevaleçam na montagem do descritor. Após isso, ele é normalizado novamente obtendo invariância à iluminação. (GONZÁLES, 2014).

Como resultado, são obtidos um conjunto de descritores robustos para identificar unicamente cada ponto do produto e da gôndola. Sendo assim, é possível reconhecer e localizar o produto na gôndola através da comparação dos descritores gerados.

2.13 BIBLIOTECA OPENCV

A OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de código aberto, voltada para visão computacional e inteligência artificial. Ela possui suporte aos principais sistemas operacionais de mercado, tais como Windows, Linux, Android e Mac

OS e suporta as linguagens de programação C ++, Python, Java e MATLAB. (OPENCV, 2019).

A biblioteca possui diversos algoritmos voltados para visão computacional e processamento de imagens. Para esse projeto, serão utilizados recursos gratuitos, como: captura de imagens, *threshold*, identificação de contornos, modelo de cores, entre outros. Também serão utilizados recursos liberados para estudantes, como os métodos para identificação de objetos através de pontos de interesse, como o SIFT.

2.14 CÂMERAS IP

Sistema de monitoramento IP é a tecnologia de captura de imagens, através da qual a câmera atua como um componente computacional, enviando vídeo digital, sem necessidade de conversões para visualização ou armazenamento. Por se tratar de um recurso computacional, sua conexão se dá através de cabos de rede, existindo também a opção de fornecimento de energia pelo mesmo cabo.

Trata-se de um sistema mais flexível, expansível e fácil de administrar, quando comparado com o sistema analógico. Permite acesso tanto de imagens ao vivo e gravadas, podendo oferecer registros de forma mais rápida, pois os dados já estão na forma digital, além de possibilitar maior tempo de gravação, uma vez que as tecnologias de codificação de vídeo são mais atuais. E por último, a qualidade da imagem digital é superior, além de não sofrer degradações com armazenamento.

Os sistemas digitais buscam diminuir os gastos de operação, resultando em um melhor custo e benefício. Um fator que impulsiona esse ganho é a possibilidade de conexão em rede, reduzindo a infraestrutura para instalação (eliminação do cabo coaxial), permitindo o acesso local ou remoto a qualquer momento, além do gerenciamento de permissões de acessos, histórico de eventos, entre outros.

A câmera IP pode ser descrita como uma câmera e computador combinados em um único dispositivo. Os principais componentes de uma câmera IP incluem uma lente, sensor de imagem, um ou mais processadores e memória. Os processadores são utilizados para o processamento de imagem, compressão, análise de vídeo e funcionalidades de rede. A memória é utilizada para armazenar o firmware da câmera e para armazenar curtos períodos de gravação de vídeo.

Em uma estrutura em que exista uma aplicação servidora, as câmeras enviam a esse servidor as imagens capturadas para que elas sejam gerenciadas. Também existe a possibilidade de armazenamento direto das imagens registradas, embora esse formato seja desaconselhado devido à ausência de organização e funcionalidades previstas pelas aplicações servidoras.

3 METODOLOGIA

Para o desenvolvimento do sistema de identificação de ruptura em gôndolas do varejo, optou-se pela metodologia apresentada neste capítulo. A arquitetura macro do sistema utilizada no presente estudo foi ilustrada na figura 1. Cabe ressaltar, no entanto, que a arquitetura completa não foi aplicada, portanto, preservou-se no tempo verbal futuro os elementos que fazem parte de projetos posteriores.

A aplicação de identificação e tratamento de ruptura nas gôndolas será executada no servidor de aplicação, que comunica-se com as câmeras IP por meio da rede WI-FI. Ela fornece uma interface para cadastro e configuração das gôndolas, assim como suas respectivas câmeras. Cada gôndola pode ter 1 (uma) ou mais câmeras de monitoramento. A identificação de ruptura realizar-se-á através das técnicas de processamento de imagem, utilizando a biblioteca OpenCV. O algoritmo identificará os espaços vazios na imagem da gôndola, e caso detectada uma ruptura, irá gerar o alerta com todas as informações no servidor. A aplicação de interface com usuário consumirá o alerta do servidor, apresentando para o usuário a imagem da(s) gôndola(s), destacando: a sua identificação, a descrição do produto em falta e o local da ruptura.

Cabe reforçar que o escopo deste projeto está limitado as atividades referentes ao desenvolvimento de um software para identificação de ruptura nas prateleiras das gôndolas do varejo. O software primeiramente analisou fotos estáticas capturadas com uma câmera genérica. Além disso, foi utilizado um computador comum que assumiu o papel de servidor e também de dispositivo de análise dos alertas pelo usuário. Sendo assim, está fora de escopo o projeto de infraestrutura, que abrange todo o sistema para captura de imagens das gôndolas como: a fixação, distribuição e configurações das câmeras e instalação da rede WI-FI.

3.1 LISTA DE MATERIAIS

Considerando o escopo do projeto, o presente tópico apresenta os materiais necessários para o seu desenvolvimento:

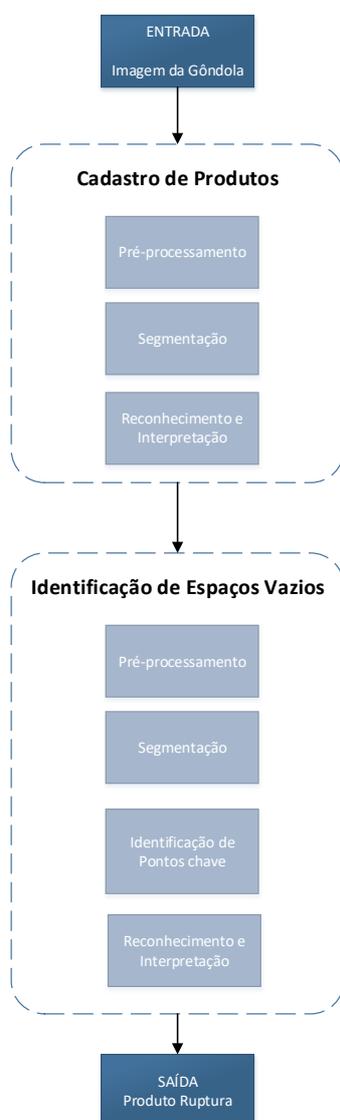
- a) câmera genérica;
- b) computador com sistema operacional Windows 10;
- c) visual Studio 2017 com biblioteca MFC;

d) biblioteca OpenCV versão 3.4.1.0.

3.2 SOFTWARE

A identificação de ruptura foi implementada utilizando técnicas de processamento de imagens, disponibilizadas pela biblioteca de visão computacional OpenCV. A aplicação foi desenvolvida inicialmente para o sistema operacional Windows 10, utilizando o compilador Visual Studio 2017 e a linguagem de programação C++. A API empregada para construção da aplicação e da interface foi MFC. A figura 18 apresenta todas as etapas que realizadas para a obtenção do resultado desejado.

Figura 18 – Etapas para identificação de ruptura



Fonte: Elaborada pelo autor (2018).

Na primeira etapa da estrutura do algoritmo desenvolvido para identificação de ruptura, realizou-se o cadastro do produto com o objetivo de identificar a sua quantidade, suas respectivas posições na gôndola e a quantidade total de pontos chave de cada prateleira completa. Todas essas informações foram armazenadas para utilização nas próximas etapas.

Realizado o cadastro do produto, analisou-se, em seguida, cada posição para validar se o produto estava ou não na prateleira. Para isso utilizou-se dois tipos de verificações: por limiarização e por identificação de pontos chave. Após verificações, classificou-se o nível de alerta a ser enviado aos responsáveis, pois um produto pode estar faltando em uma determinada localização e disponível, em número considerável, em local próximo. Os tópicos seguintes irão apresentar em detalhes todas as etapas realizadas até o resultado.

3.3 CADASTRO DE PRODUTOS

Este tópico irá apresentar, em detalhes, todas as etapas realizadas para o cadastro de produtos.

3.3.1 Pré-Processamento

A etapa de pré-processamento é realizada sobre a imagem bruta capturada pela câmera. Foi feito o isolamento das prateleiras da gôndola através do corte da imagem original, utilizando as coordenadas selecionadas pelo usuário.

3.3.2 Segmentação

Realizada a etapa de pré-processamento, e utilizando os parâmetros de configuração da prateleira, aplicou-se na imagem resultante a técnica de limiarização seguida de duas erosões. Em seguida efetuou-se a detecção de bordas e, posteriormente, aplicou-se a transformada probabilística de *Hough* para identificação de todas as linhas verticais na imagem. Na biblioteca OpenCV, as funções que correspondem a essas técnicas são: *threshold*, *erode* e *HoughLinesP* respectivamente.

3.3.3 Reconhecimento e Interpretação

O objetivo foi identificar nas imagens geradas pelas etapas anteriores a quantidade de produtos disponíveis em cada prateleira e as suas respectivas posições. Para o segundo nível de verificação de ruptura, contabilizou-se a quantidade de pontos-chaves que correspondem a prateleira completa.

3.4 IDENTIFICAÇÃO DE ESPAÇOS VAZIOS

Este tópico irá apresentar, em detalhes, todas as etapas realizadas para identificação de espaços vazios.

3.4.1 Pré-Processamento

Consiste em realizar duas operações sob a imagem capturada pela câmera. A primeira foi a realização do isolamento das prateleiras da gôndola através do corte da imagem original, utilizando as coordenadas que foram configuradas pelo usuário no momento do cadastro da gôndola. Obtida a imagem de cada prateleira, a segunda etapa foi realizar novamente um corte na imagem de acordo com a posição de cada produto obtida através da transformada de *Hough*.

3.4.2 Segmentação

Realizada a etapa de pré-processamento, e utilizando os parâmetros de configuração da prateleira e posição de cada produto, foi aplicada em cada posição a técnica de limiarização. A função responsável por essa técnica na biblioteca OpenCV é chamada de *threshold* e nela são informados os parâmetros do valor limiar e a definição de valores de coloração para pixels maiores que o limiar e menores que o limiar definido. O valor limiar é definido de acordo com a iluminação de cada loja onde o sistema será implementado.

Com a imagem limiarizada, utilizou-se o método *findcountors*, da OpenCV, que analisou a área de produto. Por fim, realizou-se uma comparação da área identificada com a conhecida para o produto no intuito de validar a sua presença ou não.

No segundo nível de validação, utilizou-se o algoritmo SIFT para identificar os pontos chave da prateleira atual e original com o objetivo de montar para cada imagem o seu vetor de descritores. A montagem dos pontos chave e dos descritores foi realizada pelo método *detectAndCompute* da biblioteca OpenCV.

3.5 RECONHECIMENTO E INTERPRETAÇÃO

Esta etapa tem como objetivo identificar os produtos faltantes e ainda classificar o tipo de alerta que será apresentado para o usuário.

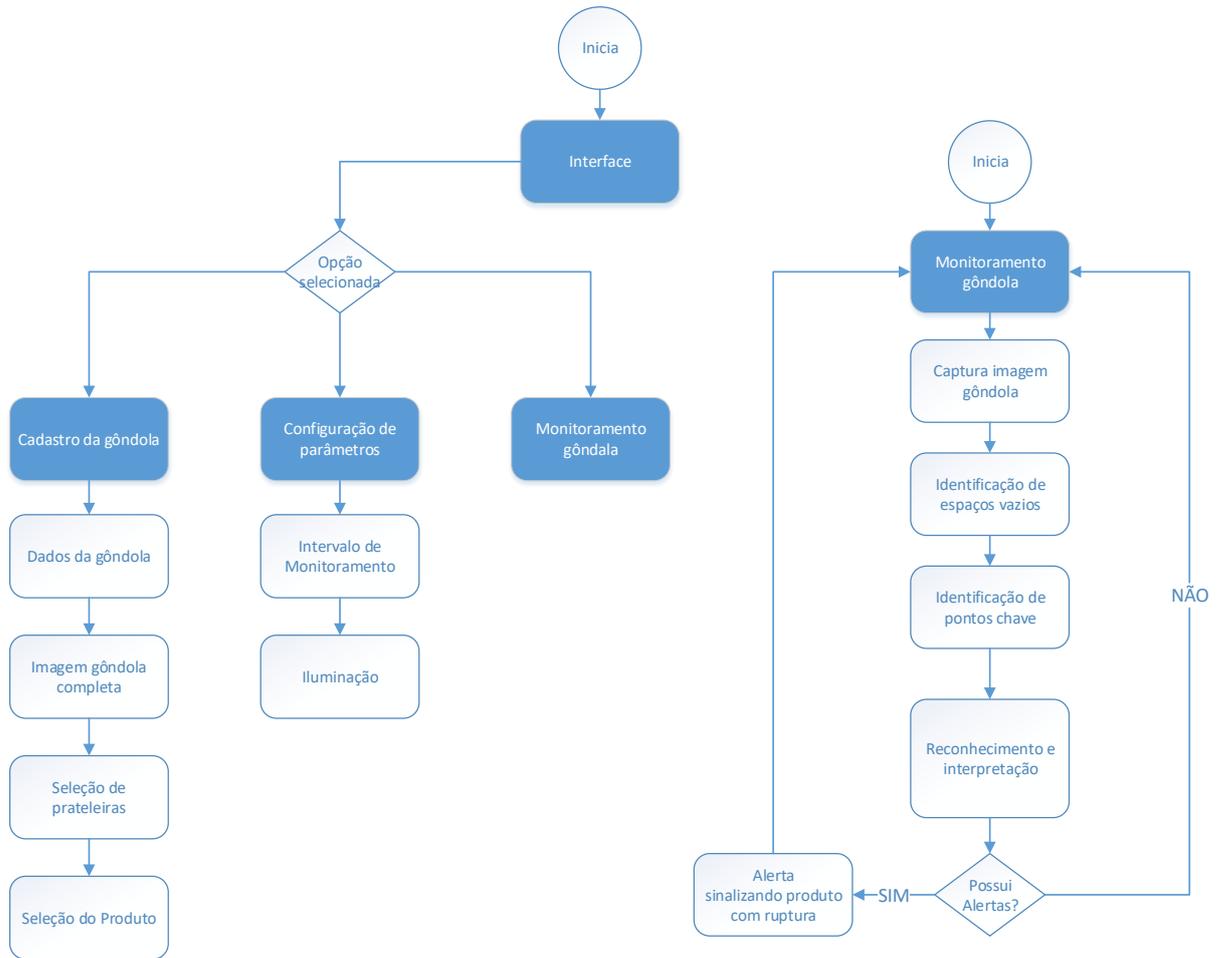
Com as etapas anteriores realizadas, foi possível identificar a quantidade de produtos frontais disponíveis em cada prateleira da gôndola. Na sequência, utilizou-se o algoritmo SIFT para complementar o método anterior e validar se o resultado obtido representava realmente uma ruptura. O método *knnMatch* da biblioteca OpenCV foi empregado para retornar as melhores correspondências entre os pontos — avaliadas de acordo com os parâmetros de distância e tamanho —, que foram testadas e apresentadas no capítulo de apresentação de resultados. Se um número N de pontos válidos fosse localizado, considerar-se-ia identificado o produto na gôndola.

Por fim, foi realizada a interpretação dos resultados de ambos os métodos e classificado o nível de alerta para o usuário, sendo eles: produto disponível na gondola; alguns produtos faltando na gôndola; produto faltando em uma das prateleiras da gôndola; e produto indisponível (ruptura).

3.6 INTERFACE

A interface do software foi desenvolvida na linguagem de programação C++, utilizando a API MFC da *Microsoft*. A interface deve disponibilizar, para o usuário, botões e telas para cadastro de produtos e prateleiras, configurações gerais como intervalo de monitoramento, iluminação e ainda opções de iniciar e interromper o monitoramento de ruptura nas gôndolas. A figura 19 apresenta o fluxograma da interface do software com o usuário.

Figura 19 – Fluxograma da interface e monitoramento de ruptura



Fonte: Elaborada pelo autor (2018).

3.6.1 Cadastro de Produtos

A opção foi desenvolvida para que o usuário possa realizar o cadastro de um produto. O sistema irá solicitar informações como: descrição, código de barras e número da gôndola. A próxima informação requisitada é a imagem da gôndola onde o produto está localizado — as prateleiras devem estar completas para que o usuário possa selecioná-las. Por fim, a área um dos produtos na gôndola, pois o algoritmo precisa de suas dimensões para o processo de segmentação das imagens.

3.6.2 Configuração de Parâmetros

Foram instituídas para que usuário realize as configurações de monitoramento das gôndolas. As configurações são: intervalo de monitoramento e o valor de limiar para aplicação da limiarização. Esse valor corresponde à iluminação da loja.

3.6.3 Monitoramento

Elaborado para realizar o processo de identificação de ruptura através das técnicas apresentadas nos itens 3.4 e 3.5. Existindo um caso de ruptura, enviar-se-á para a interface principal do software todas as informações pertinentes, assim como ficará disponível no servidor através de arquivos que podem ser consumidos por outros dispositivos.

4. RESULTADOS

Serão apresentados os resultados obtidos através da análise de produtos de diferentes classes e formatos. As imagens dos produtos e gôndolas foram obtidas em um grande varejista, localizado na cidade de Canoas, Rio Grande do Sul. A câmera utilizada foi do smartphone da fabricante *Apple*, modelo *Iphone XR*, utilizando uma resolução personalizada de 1440x1080 megapixels.

4.1 INTERFACE

Este tópico apresenta os resultados referentes ao desenvolvimento da interface da aplicação para identificação de ruptura. Os sub tópicos a seguir apresentam as telas desenvolvidas.

4.1.1 Tela de Inicialização

A tela foi desenvolvida para que o usuário possa visualizar todas as etapas de inicialização da aplicação. Elas envolvem o carregamento de informações necessárias para o funcionamento da aplicação, como produtos e parâmetros configurados.

A figura 20 apresenta a tela de inicialização da aplicação. No centro da tela foi utilizado o componente *Listbox* da biblioteca MFC, que se trata de uma lista na qual são exibidas todas as etapas da inicialização, possíveis erros e a versão do software.

Figura 20 – Tela de inicialização da aplicação

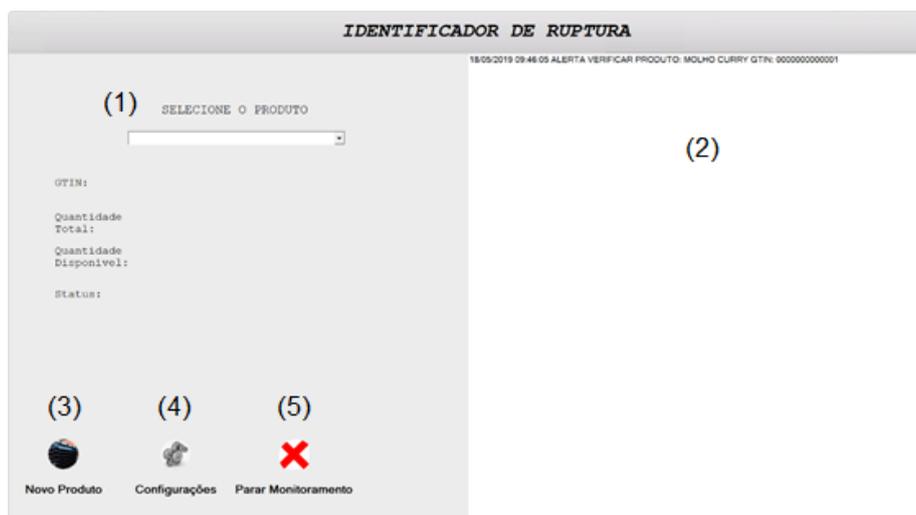


Fonte: Elaborada pelo autor (2019).

4.1.2 Tela Principal

Ao finalizar a inicialização, será apresentada para o usuário a tela principal da aplicação, a principal interação entre o usuário e as funcionalidades do software proposto.

Figura 21 – Tela principal da aplicação



Fonte: Elaborada pelo autor (2019).

Os itens abaixo apresentam as operações que podem ser realizadas na tela principal e um detalhamento de como cada uma funciona.

4.1.2.1 Visualizar Descrição, Quantidade e Status Atual do Produto

O objetivo foi disponibilizar para o usuário uma forma rápida para verificar as informações de quantidade e status de um determinado produto. Para isso foi utilizado o componente *combobox* figura 21 (1) da biblioteca MFC para que o usuário selecione o produto do qual deseja as informações. Após a seleção, a aplicação irá buscar o status atual do mesmo e exibir esses dados na tela.

4.1.2.2 Visualizar Alertas

A visualização de alertas é uma das principais interações que o usuário irá realizar com a aplicação. Foi utilizado o componente *ListBox* figura 21 (2) da biblioteca MFC, que se trata de uma lista com todos os alertas que necessitam de uma intervenção. Além disso,

o usuário terá a opção de dar dois cliques no alerta para abrir a imagem da gôndola na qual estarão sinalizados com um quadrado verde os locais onde o produto está em falta.

4.1.2.3 Cadastrar Novos Produtos

Ao clicar no botão “Novo Produto” figura 21 (3), será aberto pela aplicação um diálogo para preenchimento de informações sobre o produto. As informações são: descrição, código de barras e o número da gôndola em que o produto está localizado. Essas informações são importantes para identificação e localização em caso de alertas.

Figura 22 – Diálogo para cadastro de produtos

A imagem mostra uma janela de software intitulada "CADASTRO DE NOVO PRODUTO". Ela contém os seguintes elementos:

- Um campo de texto "Descrição de Produto" com o valor "Leite Ninho".
- Um campo de texto "Código GTIN" com o valor "7891000142202".
- Um campo de texto "Numero Gôndola" com o valor "10".
- Um campo "Selecione a imagem da gôndola" com um ícone de câmera.
- Um botão "Prateleiras" para selecionar as prateleiras.
- Um botão "Produto" para selecionar o produto.
- Um botão "Finalizar Cadastro" para concluir o processo.
- Um checkbox "Utilizar filtro por cor" desativado e um botão "Configurar Filtro" ao lado dele.
- Textos de instrução: "Selecione a prateleira onde se encontra o produto a monitorar e aperte *ENTER* para selecionar a proxima. Para concluir aperte *ESC*" e "Selecione apenas um produto de qualquer prateleira e aperte *ENTER*".

Fonte: Elaborado pelo autor (2019).

Realizado o preenchimento das informações do produto, o usuário deve informar para o software onde está localizada a imagem da gôndola completa, ou seja, com todos os produtos repostos e em suas respectivas posições.

Após informar o local, deve-se clicar no botão “prateleiras” figura 22, onde será aberta a imagem da gôndola para seleção de todas as prateleiras que se deseja monitorar.

A figura 23 apresenta a seleção de prateleiras onde está localizado o produto Leite Ninho. O quadro azul na imagem representa a seleção de uma das prateleiras onde o produto Leite Ninho está localizado na gôndola. Após seleção, deve ser pressionada a tecla “ENTER” e partir para seleção das demais prateleiras. Para finalizar deve ser pressionada a tecla “ESC”.

Figura 23 – Seleção de prateleiras



Fonte: Elaborada pelo autor (2019).

Na última etapa, o usuário deve selecionar um dos produtos de qualquer prateleira. O objetivo é informar para a aplicação as dimensões do produto.

A figura 24 apresenta a seleção de um produto Leite Ninho em uma das prateleiras da gôndola. O quadro azul na imagem, representa a seleção das dimensões de apenas um produto na gôndola. A tecla “ENTER” deve ser pressionada para finalização do cadastro.

Figura 24 – Seleção do produto



Fonte: Elaborada pelo autor (2019).

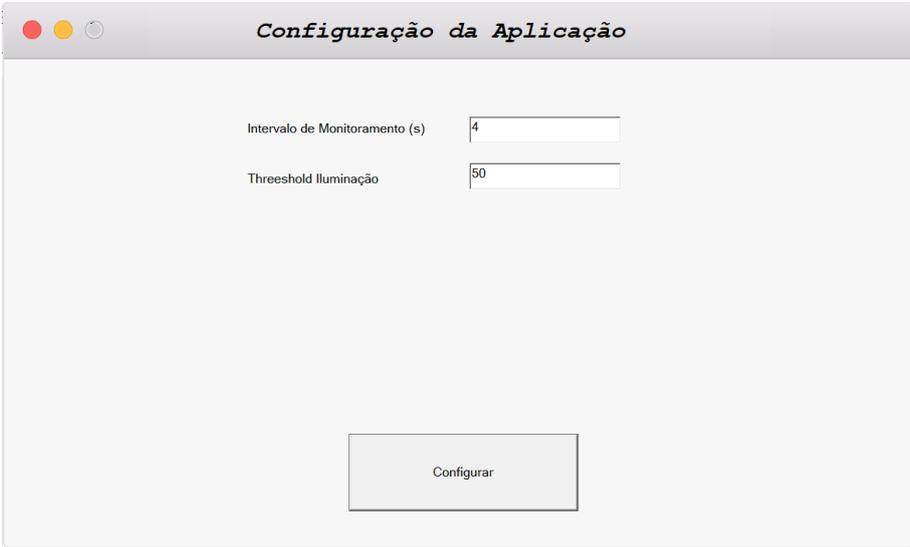
Como todas as informações preenchidas, o próximo passo é clicar no botão “Finalizar Cadastro”.

4.1.2.4 Alterar Configurações Gerais da Aplicação e Iniciar/Interromper Monitoramento dos Produtos

Ao clicar no botão “Configurações” figura 21 (4), será aberto pela aplicação um diálogo em que é possível alterar o tempo de monitoramento dos produtos e o valor limiar de iluminação que utilizado em todas as etapas de processamento de imagens.

A figura 25 apresenta o diálogo de configurações gerais da aplicação. Para alteração dos valores padrões, os novos valores devem ser preenchidos nas caixas disponíveis e por fim clicar no botão “Configurar”.

Figura 25 – Diálogo de configurações gerais da aplicação



Configuração da Aplicação

Intervalo de Monitoramento (s) 4

Threshold Iluminação 50

Configurar

Fonte: Elaborada pelo autor (2019).

Por fim, o usuário pode utilizar o botão caso deseje interromper/iniciar figura 21 (5) o monitoramento de produtos. Por padrão, a aplicação inicializa com o monitoramento ativo.

4.2 PROCESSAMENTO DE IMAGENS NO CADASTRO DE PRODUTOS

O cadastro de produtos realiza todas as etapas de processamento de imagem e teve por objetivo identificar na imagem da gôndola completa a posição de cada produto, sua quantidade e, por fim, o número total de pontos chave de cada prateleira (idem item 3.4). Nos próximos tópicos, serão apresentadas as principais etapas do processamento e seus respectivos resultados para cada produto.

4.2.1 Aplicação da Operação de Limiarização

A operação de limiarização é realizada sobre a imagem de cada prateleira selecionada para o produto e tem como pré-requisito a imagem em escala de cinza. Essa operação é fortemente influenciada pelas características de iluminação de cada prateleira. Para a identificação desse valor, foi considerado que a loja do Park Shopping Canoas possui uma iluminação padrão, ou seja, com baixa variação de luminosidade nos corredores nas quais as gôndolas estão localizadas. Desta forma, analisou-se os valores das tabelas 2,3, e 4 para a utilização do método *threshold* da biblioteca OpenCV.

Tabela 2 – Resultados da operação de limiarização com valor limiar igual a 45

Produto	Valor Limiar	Tipo	Número de Pixels diferentes de zero
Leite Ninho			34743
Classe: Leites e derivados			
Óleo Girassol VITALIV			39205
Classe: Óleos vegetais			
Molho Curry Terra Rica			12009
Classe: Temperos e condimentos	45	THRESH_BINARY	
Farinha de Trigo Garota			52945
Classe: Farinhas			
Coca-Cola Tradicional			144815
Classe: Refrigerantes			
Achocolatado Pirakids			13522
Classe: Achocolatados			

Fonte: Elaborada pelo autor (2019).

Tabela 3 – Resultados da operação de limiarização com valor limiar igual a 60

(continua)

Produto	Valor Limiar	Tipo	Número de Pixels diferentes de zero
Leite Ninho	60	THRESH_BINARY	33446
Classe: Leites e derivados			
Óleo Girassol VITALIV			26525
Classe: Óleos vegetais			

Produto	Valor Limiar	Tipo	Número de Pixels diferentes de zero
Molho Curry Terra Rica			11024
Classe: Temperos e condimentos			
Farinha de Trigo Garota	60	THRESH_BINARY	36421
Classe: Farinhas			
Coca-Cola Tradicional			85416
Classe: Refrigerantes			
Achocolatado Pirakids			13287
Classe: Achocolatados			

Fonte: Elaborada pelo autor (2019).

Tabela 4 – Resultados da operação de limiarização com valor limiar igual a 75

Produto	Valor Limiar	Tipo	Número de Pixels diferentes de zero
Leite Ninho			31611
Classe: Leites e derivados			
Óleo Girassol VITALIV			32814
Classe: Óleos vegetais			
Molho Curry Terra Rica			11656
Classe: Temperos e condimentos	75	THRESH_BINARY	
Farinha de Trigo Garota			44514
Classe: Farinhas			
Coca-Cola Tradicional			201319
Classe: Refrigerantes			
Achocolatado Pirakids			12534
Classe: Achocolatados			

Fonte: Elaborada pelo autor (2019).

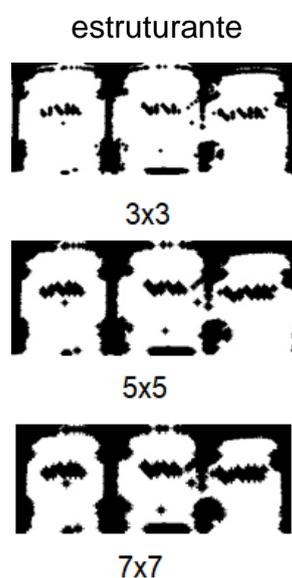
Com a aplicação da operação de limiarização para 6 produtos localizados em diferentes corredores da loja, verificou-se que existe uma diferença de iluminação entre as gôndolas, principalmente na da Coca-Cola. Sendo assim, para a decisão do melhor valor limiar a ser utilizado, foi analisada a quantidade de número de pixels diferente de zero para

cada produto. O melhor, levando em conta a média de todos os produtos analisados, foi do de limiar igual a 75. Utilizando esse valor, foi aplicada a técnica de erosão, que será avaliada no próximo item.

4.2.2 Aplicação da Operação de Erosão

O objetivo da operação de erosão na identificação dos produtos foi eliminar pequenas diferenças internas, pois isso facilita a detecção das bordas dos produtos na prateleira. Para a escolha do melhor elemento estruturante para a erosão, foram utilizados os seguintes valores 3x3, 5x5 e 7x7 de matrizes em forma de cruz.

Figura 26 – Resultado utilizando 3 tamanhos de matrizes diferentes para o elemento



Fonte: Elaborada pelo autor (2019).

Analisando os resultados na figura 26, foi possível verificar que a matriz 3x3 eliminou os pequenos ruídos internos da imagem e foi a melhor relação entre manter o tamanho do objeto e separá-los dos demais. A partir da matriz 7x7 notou-se uma redução considerável no seu tamanho, o que torna inviável a sua utilização.

4.2.3 Aplicação da Operação de Detecção de Bordas

Realizada a operação de erosão, foi aplicada a detecção de bordas utilizando o método *canny* da biblioteca OpenCV. O método recebe os parâmetros limiar, histerese mínimo e máximo. Sendo assim, foram analisados os valores nas tabelas 5,6 e 7.

Tabela 5 – Resultados da operação de detecção de bordas para valores de limiar 45, histerese mínimo 45 e máximo 140

Produto	Valor Limiar	Histerese Mínimo	Histerese Máximo	Número de pixels
Leite Ninho				1453
Classe: Leites e derivados				
Óleo Girassol VITALIV				2366
Classe: Óleos vegetais				
Molho Curry Terra Rica				417
Classe: Temperos e condimentos	45	45	140	
Farinha de Trigo Garota				2282
Classe: Farinhas				
Coca-Cola Tradicional				15436
Classe: Refrigerantes				
Achocolatado Pirakids				405
Classe: Achocolatados				

Fonte: Elaborada pelo autor (2019).

Tabela 6 – Resultados da operação de detecção de bordas para valores de limiar 60, histerese mínimo 60 e máximo 140

Produto	Valor Limiar	Histerese Mínimo	Histerese Máximo	Número de pixels
Leite Ninho				1851
Classe: Leites e derivados				
Óleo Girassol VITALIV				2314
Classe: Óleos vegetais				
Molho Curry Terra Rica				441
Classe: Temperos e condimentos	60	60	140	
Farinha de Trigo Garota				4666
Classe: Farinhas				
Coca-Cola Tradicional				14379
Classe: Refrigerantes				
Achocolatado Pirakids				765
Classe: Achocolatados				

Fonte: Elaborada pelo autor (2019).

Tabela 7 – Resultados da operação de detecção de bordas para valores de limiar 75, histerese mínimo 75 e máximo 140

Produto	Valor Limiar	Histerese Mínimo	Histerese Máximo	Número de pixels
Leite Ninho				2366
Classe: Leites e derivados				
Óleo Girassol VITALIV				3068
Classe: Óleos vegetais				
Molho Curry Terra Rica				505
Classe: Temperos e condimentos	75	75	140	
Farinha de Trigo Garota				5290
Classe: Farinhas				
Coca-Cola Tradicional				13358
Classe: Refrigerantes				
Achocolatado Pirakids				1621
Classe: Achocolatados				

Fonte: Elaborada pelo autor (2019).

Analisando o conjunto de produtos, os valores utilizados na tabela 7 apresentaram os melhores resultados, pois identificaram em média a maior quantidade de bordas, tornando mais precisa a identificação da posição dos produtos pela transformada de Hough.

4.2.4 Aplicação da Transformada Probabilística de Hough

Aplicada a operação de detecção de bordas, utilizou-se a transformada de Hough para identificar todas as linhas verticais na imagem. O objetivo foi identificar a localização de cada produto na gôndola e a quantidade frontal exposta para os clientes.

Tabela 8 – Resultados da transformada de Hough para valores de limiar 5 e tamanho mínimo de linha 10

(continua)

Produto	Valor Limiar	Tamanho Min Linha	Número de Linhas Verticais
Leite Ninho			123
Classe: Leites e derivados	5	10	

Produto	Valor Limiar	Tamanho Min Linha	Número de Linhas Verticais
			194
Óleo Girassol VITALIV			
Classe: Óleos vegetais			
Molho Curry Terra Rica			19
Classe: Temperos e condimentos			
Farinha de Trigo Garota			372
Classe: Farinhas			
Coca-Cola Tradicional			1109
Classe: Refrigerantes			
Achocolatado Pirakids			129
Classe: Achocolatados			

Fonte: Elaborada pelo autor (2019).

Tabela 9 – Resultados da transformada de Hough para valores de limiar 10 e tamanho mínimo de linha 10

Produto	Valor Limiar	Tamanho Min Linha	Número de Linhas Verticais
Leite Ninho			60
Classe: Leites e derivados			
Óleo Girassol VITALIV			142
Classe: Óleos vegetais			
Molho Curry Terra Rica	10	10	10
Classe: Temperos e condimentos			
Farinha de Trigo Garota			243
Classe: Farinhas			
Coca-Cola Tradicional			691
Classe: Refrigerantes			
Achocolatado Pirakids			54
Classe: Achocolatados			

Fonte: Elaborada pelo autor (2019).

Tabela 10 – Resultados da transformada de Hough para valores de limiar 20 e tamanho mínimo de linha 10

Produto	Valor Limiar	Tamanho Min Linha	Número de Linhas Verticais
Leite Ninho Classe: Leites e derivados			13
Óleo Girassol VITALIV Classe: Óleos vegetais			25
Molho Curry Terra Rica Classe: Temperos e condimentos	20	10	3
Farinha de Trigo Garota Classe: Farinhas			50
Coca-Cola Tradicional Classe: Refrigerantes			76
Achocolatado Pirakids Classe: Achocolatados			5

Fonte: Elaborada pelo autor (2019).

Analisando os resultados, os valores utilizados na tabela 8 identificaram o maior número de linhas. As figuras 27 e 28 apresenta alguns resultados considerando a largura dos produtos.

Figura 27 – Resultado do processamento utilizando a transformada probabilística de Hough para o produto Leite Ninho



Fonte: Elaborada pelo autor (2019).

Figura 28 – Resultado do processamento utilizando a transformada probabilística de Hough para o produto Farinha de trigo Garota



Fonte: Elaborada pelo autor (2019).

4.2.5 Identificação da Posição e Quantidade de Produtos

Com a posição de cada produto identificada, foi realizada a contagem deles em cada prateleira. A tabela 10 apresenta os resultados obtidos e realiza uma classificação quanto ao seu posicionamento na gôndola, ou seja, através da observação visual de o quanto a identificação foi precisa na localização do produto. As classificações criadas foram: ruim, para quando ocorreram erros na determinação da posição e quantidade de produtos; regular, para erros de posicionamento que não interferiram na quantidade; e, por fim, bom, para os produtos que tiveram sua localização completamente acertadas.

Tabela 11 – Resultados obtidos para o cadastro de produtos

Produto	Quantidade Esperada	Quantidade Identificada	Percentual	Posicionamento
1 Leite Ninho Classe: Leites e derivados	9	9	100%	Bom
2 Óleo Girassol VITALIV Classe: Óleos vegetais	12	12	100%	Regular
3 Molho Curry Terra Rica Classe: Temperos e condimentos	1	1	100%	Bom
4 Farinha de Trigo Garota Classe: Farinhas	24	24	100%	Bom
5 Coca-Cola Tradicional Classe: Refrigerantes	19	19	100%	Regular
6 Achocolatado Pirakids Classe: Achocolatados	42	39	92,85%	Ruim

Fonte: Elaborada pelo autor (2019).

A figura 29 demonstra visualmente a classificação realizada na tabela 11.

Figura 29 – Imagens resultantes do cadastro de produtos



Fonte: Elaborada pelo autor (2019).

Analisando os resultados da tabela 11 e da figura 29, observou-se que alguns produtos obtiveram o resultado “regular” porque o quadro não ficou totalmente centralizado no produto e acabou invadindo espaço de outro na prateleira. Porém, isso não afetou a identificação da quantidade correta exposta para o cliente. Já para o produto Achocolatado Pirakids, apesar de estar em grande quantidade, não foi possível identificar 3 posições, o que acabou levando à classificação “ruim”. Na média dos produtos cadastrados, a aplicação acertou em 98% sua quantidade.

4.2.6 Identificação de Pontos Chave Utilizando Algoritmo SIFT

A última etapa do cadastro foi analisar a quantidade de *matches* obtidos entre as mesmas imagens da prateleira — uma segunda validação para identificação de ruptura. O objetivo no cadastro foi obter o número de pontos chave máximos da imagem de configuração de cada prateleira.

Tabela 12 – Resultado do cadastro de produto utilizando SIFT

Produto	Quantidade	Número total de Pontos Chave
Leite Ninho	9	1901
Classe: Leites e derivados		
Óleo Girassol VITALIV	12	2261
Classe: Óleos vegetais		
Molho Curry Terra Rica	1	109
Classe: Temperos e condimentos		
Farinha de Trigo Garota	24	3200
Classe: Farinhas		
Coca-Cola Tradicional	19	3500
Classe: Refrigerantes		
Achocolatado Pirakids	42	2096
Classe: Achocolatados		

Fonte: Elaborada pelo autor (2019).

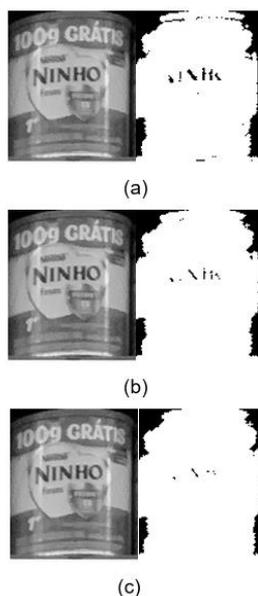
4.3 PROCESSAMENTO DE IMAGENS NA IDENTIFICAÇÃO DE ESPAÇOS VAZIOS

A identificação de espaços vazios realiza todas as etapas de processamento de imagens e tem por objetivo identificar na imagem atual da gôndola a quantidade de produtos visíveis para o cliente. As operações de limiarização e de erosão utilizam o mesmo estudo realizado para o cadastro de gôndolas (itens 4.2.1 e 4.2.2), por isso não constam nesse tópico. Nos próximos sub tópicos serão apresentados as principais etapas e seus respectivos resultados.

4.3.1 Operação de Dilatação

A operação de dilatação é utilizada logo após a de erosão, pois esta remove ruídos e acaba de certa forma reduzindo o tamanho do objeto. Como o objetivo neste tópico foi somente identificar a presença do objeto, a operação de dilatação facilita o processo, já que, além de aumentar o objeto, ela também ajuda a unir possíveis partes separadas. Sendo assim, foram utilizados os seguintes valores de matrizes em forma de cruz: 3x3, 4x4 e 5x5.

Figura 30 – Resultados da operação de dilatação



Fonte: Elaborada pelo autor (2019).

A figura 30 apresenta os resultados da operação de dilatação utilizando valores de matrizes em forma de cruz: 3x3 (a), 4x4 (b) e 5x5 (c).

4.3.2 Aplicação da Operação de Busca de Contornos

Com a operação de erosão e dilatação realizadas, foi avaliado o número de contornos identificados, relacionando com os parâmetros utilizados nos métodos anteriores.

Tabela 13 – Número de contornos obtidos utilizando elemento estruturante de 3x3 para dilatação

(continua)

Produto	Elemento estruturante	Número de contornos
Leite Ninho		34
Classe: Leites e derivados		
Óleo Girassol VITALIV		35
Classe: Óleos vegetais		
Molho Curry Terra Rica	3x3	8
Classe: Temperos e condimentos		

Produto	Elemento estruturante	Número de contornos
Farinha de Trigo Garota		36
Classe: Farinhas		
Coca-Cola Tradicional		66
Classe: Refrigerantes	3x3	
Achocolatado Pirakids		18
Classe: Achocolatados		

Fonte: Elaborada pelo autor (2019).

Tabela 14 – Número de contornos obtidos utilizando elemento estruturante de 4x4 para dilatação

Produto	Elemento estruturante	Número de contornos
Leite Ninho		17
Classe: Leites e derivados		
Óleo Girassol VITALIV		19
Classe: Óleos vegetais		
Molho Curry Terra Rica		8
Classe: Temperos e condimentos	4x4	
Farinha de Trigo Garota		23
Classe: Farinhas		
Coca-Cola Tradicional		48
Classe: Refrigerantes		
Achocolatado Pirakids		8
Classe: Achocolatados		

Fonte: Elaborada pelo autor (2019).

Tabela 15 – Número de contornos obtidos utilizando elemento estruturante de 5x5 para dilatação

(continua)

Produto	Elemento estruturante	Número de contornos
Leite Ninho		11
Classe: Leites e derivados	5x5	

Produto	Elemento estruturante	Número de contornos
Óleo Girassol VITALIV		20
Classe: Óleos vegetais		
Molho Curry Terra Rica		7
Classe: Temperos e condimentos		
Farinha de Trigo Garota		23
Classe: Farinhas		
Coca-Cola Tradicional		38
Classe: Refrigerantes		
Achocolatado Pirakids		10
Classe: Achocolatados		

Fonte: Elaborada pelo autor (2019).

Analisando os resultados das tabelas 13, 14 e 15, o melhor resultado apresentado foi a matriz 4x4. Na matriz 3x3, apesar de apresentar um maior número de contornos, houve prejuízo na identificação das áreas, pois o objeto pode ser quebrado em pequenas áreas, dificultando os processos de validação utilizados pela aplicação.

4.4 GERENCIAMENTO DE ALERTAS

O gerenciamento de alertas tem como objetivo alcançar os resultados de todas as etapas anteriores e interpretá-los para verificar o tipo de alerta e em qual o produto se encontra (Item 3.6).

4.4.1 Produto Disponível da Gôndola

Neste tópico serão analisados os 6 produtos cadastrados com as prateleiras completas na gôndola.

Tabela 16 – Resultados com produto disponível na gôndola

Produto	Quantidade Cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
Leite Ninho	9	9	LIMIAR:9	Produto
Classe: Leites e derivados			SIFT: 9	disponível
Óleo Girassol VITALIV	12	12	LIMIAR:12	Produto
Classe: Óleos vegetais			SIFT: 12	disponível
Molho Curry Terra Rica	1	1	LIMIAR:1	Produto
Classe: Temperos e condimentos			SIFT: 1	disponível
Farinha de Trigo Garota	24	24	LIMIAR:24	Produto
Classe: Farinhas			SIFT: 24	disponível
Coca-Cola Tradicional	19	19	LIMIAR:0	Produto
Classe: Refrigerantes			SIFT: 19	Indisponível
Achocolatado Pirakids	39	39	LIMIAR:39	Produto
Classe: Achocolatados			SIFT: 39	disponível

Fonte: Elaborada pelo autor (2019).

Analisando os resultados da tabela 16, ocorreu uma falha ao realizar o processo de identificação de ruptura para o produto Coca-Cola. Mesmo o produto estando completo na gôndola, a aplicação exibiu que a quantidade estava zerada. Isso ocorreu porque o conteúdo da Coca-Cola possui cor preta, e o algoritmo identificou apenas a cor do rótulo, que não foi considerada uma área suficiente na verificação comparada à área selecionada pelo usuário.

Analisando essa situação, foi realizada uma correção na obtenção do valor de área. O algoritmo passa a aplicar o processo de limiarização, erosão e dilatação para identificar a área útil de um produto.

Tabela 17 – Resultados com correção da validação de área dos produtos

(continua)

Produto	Quantidade Cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
Leite Ninho	9	9	LIMIAR: 9	Produto
Classe: Leites e derivados			SIFT: 9	disponível
Óleo Girassol VITALIV	12	12	LIMIAR: 12	Produto
Classe: Óleos vegetais			SIFT: 12	disponível

Produto	Quantidade Cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
Molho Curry Terra Rica Classe: Temperos e condimentos	1	1	LIMIAR: 1 SIFT: 1	Produto disponível
Farinha de Trigo Garota Classe: Farinhas	24	24	LIMIAR: 24 SIFT: 24	Produto disponível
Coca-Cola Tradicional Classe: Refrigerantes	19	19	LIMIAR: 19 SIFT: 19	Produto disponível
Achocolatado Pirakids Classe: Achocolatados	39	39	LIMIAR: 39 SIFT: 39	Produto disponível

Fonte: Elaborada pelo autor (2019).

Analisando a tabela 17, conclui-se que a correção foi realizada com sucesso e todos os produtos obtiveram os resultados esperados.

4.4.2 Alguns Produtos Faltando na Gôndola

Este tópico tem por objetivo avaliar o comportamento da aplicação deixando espaços vazios nas prateleiras, ou seja, alguns produtos foram consumidos, porém todas as prateleiras possuem produtos disponíveis para o consumidor. Nesse caso, a aplicação deve identificar que a quantidade de produtos expostos reduziu e alterar seus status.

Tabela 18 – Resultados para alguns produtos faltando na gôndola

(continua)

Produto	Quantidade Cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
1 Leite Ninho Classe: Leites e derivados	9	8	LIMIAR: 8 SIFT: 0	Alguns produtos indisponíveis na gôndola
2 Óleo Girassol VITALIV Classe: Óleos vegetais	12	10	LIMIAR: 10 SIFT: 1	Alguns produtos indisponíveis na gôndola

Produto	Quantidade Cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
3 Farinha de Trigo Garota Classe: Farinhas	24	22	LIMIAR: 22 SIFT: 5	Alguns produtos indisponíveis na gôndola
4 Coca-Cola Tradicional Classe: Refrigerantes	19	16	LIMIAR: 16 SIFT: 4	Alguns produtos indisponíveis na gôndola

Fonte: Elaborada pelo autor (2019).

Verificou-se que o método principal de validação acertou em 100% dos casos. Porém, também se constatou que o método de identificação por pontos chave (SIFT) falhou na identificação, isto porque o algoritmo falha ao obter a mesma quantidade de pontos da imagem original, não fechando com o cálculo realizado. Por esse motivo, o método será utilizado apenas para validação de ruptura total na gôndola e não será mais considerado nas próximas análises que envolvem contagem de produtos. A figura 31 apresenta as imagens geradas pela aplicação.

Figura 31 – Resultados para alguns produtos faltando na gôndola



Fonte: Elaborada pelo autor (2019).

4.4.3 Produto em Falta em uma ou Mais Prateleiras da Gôndola

Este tópico tem por objetivo avaliar o comportamento da aplicação para casos onde um produto está em falta em uma ou mais prateleiras. Esse status não se trata de uma ruptura total, ou seja, considera-se que ainda existem outras prateleiras com o produto disponível para o consumidor, porém ele é considerado um alerta a ser exibido para o usuário.

Tabela 19 – Resultados para a falta de produtos em uma ou mais prateleiras da gôndola

Produto	Quantidade cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
1 Leite Ninho Classe: Leites e derivados	9	6	6	Produto indisponível em uma ou mais prateleiras da gôndola
2 Óleo Girassol VITALIV Classe: Óleos vegetais	12	4	4	Produto indisponível em uma ou mais prateleiras da gôndola
3 Farinha de Trigo Garota Classe: Farinhas	24	18	18	Produto indisponível em uma ou mais prateleiras da gôndola
4 Achocolatado Pirakids Classe: Achocolatados	39	30	30	Produto indisponível em uma ou mais prateleiras da gôndola

Fonte: Elaborada pelo autor (2019).

Analisando a tabela 19, verificou-se que o algoritmo acertou em 100% dos casos, gerando os alertas esperados. A figura 32 apresenta as imagens geradas pela aplicação.

Figura 32 – Resultados para a falta de produtos em uma ou mais prateleiras da gôndola



Fonte: Elaborada pelo autor (2019).

4.4.4 Produto no Fundo da Prateleira

Este tópico tem por objetivo avaliar o comportamento da aplicação nos casos em que os produtos frontais são consumidos, ficando expostos somente os produtos ao fundo da prateleira. Por experiência de varejo, eles devem ser considerados em falta, gerando um alerta para que sejam reposicionados para a parte frontal, facilitando a venda ao consumidor.

Tabela 20 – Resultados produtos no fundo da prateleira

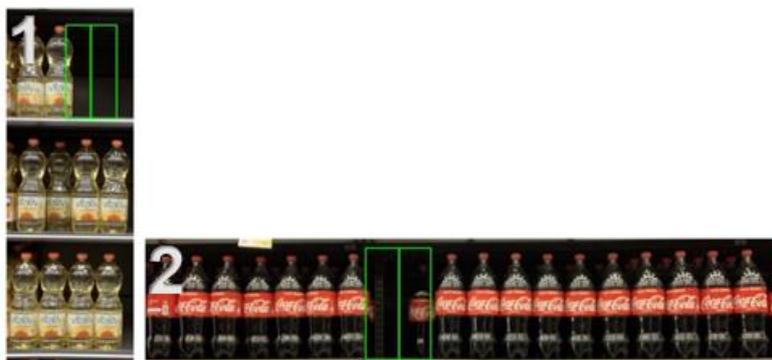
Produto	Quantidade cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
1 Óleo Girassol VITALIV Classe: Óleos vegetais	12	10	10	Alguns produtos indisponíveis na gôndola
2 Coca-Cola Tradicional Classe: Refrigerantes	19	18	17	Alguns produtos indisponíveis na gôndola

Fonte: Elaborada pelo autor (2019).

Verificou-se que na análise do produto Coca-Cola, a aplicação identificou a falta de dois produtos, porém em uma das posições ele estava disponível. Isso ocorreu devido a distância e a pequena área de detecção da Coca Cola. Uma solução para o problema é

realizar a aproximação da câmera, fazendo com que aumente a área de detecção. A figura 33 apresenta as imagens geradas pela aplicação.

Figura 33 – Resultados produtos no fundo da prateleira



Fonte: Elaborada pelo autor (2019).

4.4.5 Produto Indisponível na Gôndola

Este tópico tem por objetivo avaliar o comportamento da aplicação nos casos de ruptura do produto na gôndola, ou seja, quando ele está indisponível para venda ao consumidor final. Nesse caso, a aplicação deve gerar um alerta na lista da tela principal informando que deve ser realizada a reposição do produto.

Tabela 21 – Resultados para produto indisponível na gôndola

Produto	Quantidade cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
1 Leite Ninho	9	0	LIMIAR: 0	Produto
Classe: Leites e derivados			SIFT: 0	Indisponível
2 Óleo Girassol VITALIV	12	0	LIMIAR: 0	Produto
Classe: Óleos vegetais			SIFT: 0	Indisponível
3 Molho Curry Terra Rica	1	0	LIMIAR: 0	Produto
Classe: Temperos e condimentos			SIFT: 0	Indisponível

Fonte: Elaborada pelo autor (2019).

Constatou-se que o algoritmo acertou em 100% dos casos, gerando os alertas esperados. A figura 34 apresenta as imagens geradas pela aplicação.

Figura 34 – Resultados para produto indisponível na gôndola



Fonte: Elaborada pelo autor (2019).

4.4.6 Produto Reposto na Gôndola

Este tópico tem por objetivo avaliar o comportamento da aplicação nos casos em que é realizada a reposição completa do produto na gôndola.

Tabela 22 – Resultados para reposição de produtos na gôndola

Produto	Quantidade cadastro	Quantidade prevista	Quantidade identificada	Alerta Gerado
Leite Ninho	9	9	9	Produto
Classe: Leites e derivados				Disponível
Óleo Girassol VITALIV	12	12	12	Produto
Classe: Óleos vegetais				Disponível
Molho Curry Terra Rica	1	1	1	Produto
Classe: Temperos e condimentos				Disponível
Farinha de Trigo Garota	24	24	24	Produto
Classe: Farinhas				Disponível
Coca-Cola Tradicional	19	19	19	Produto
Classe: Refrigerantes				Disponível
Achocolatado Pirakids	39	39	39	Produto
Classe: Achocolatados				Disponível

Fonte: Elaborada pelo autor (2019).

Apurou-se que o algoritmo acertou em 100% dos casos, retornando a exibir o status de produto disponível e quantidade máxima identificada no processo de cadastro do produto.

4.5 SÍNTESE DOS RESULTADOS OBTIDOS

Em resumo, a aplicação desenvolvida possui três etapas: o cadastro de produtos, a identificação de espaços vazios e a validação por pontos chave.

Para o cadastro, os produtos classificados como “bons” foram: Leite Ninho, Molho Curry Terra Rica e a Farinha de Trigo Garota. Eles foram categorizados assim porque cada produto está praticamente centralizado dentro do quadro com as coordenadas de sua localização na prateleira. Já os outros produtos — Azeite Girassol VITLIV e a Coca-Cola Tradicional — foram considerados regulares, pois alguns quadros não cobriram toda a dimensão dos produtos. Porém, isso não afetou os resultados e a sua quantidade. Por fim, o produto Achocolatado Pirakids foi classificado como “ruim” por ter falhado na identificação de 3 produtos.

Na identificação de espaços vazios, os resultados esperados foram atingidos em praticamente todas as etapas, exceto no cenário de produto no fundo da prateleira para a Coca-Cola tradicional em função do produto apresentar coloração preta e ter uma pequena área de detecção: seu rótulo.

A etapa de identificação de pontos chave — que era um método alternativo — não atendeu a sua funcionalidade para o projeto, atuando apenas em casos de falta total do produto na prateleira.

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo do projeto foi resolver um dos grandes problemas enfrentados pelo setor varejista que é a indisponibilidade de produtos para venda em suas prateleiras. Esse problema causa prejuízos tanto financeiros quanto para a imagem da empresa. Na grande maioria dos varejos, essa identificação é realizada através da auditoria visual, ou seja, o responsável observa as prateleiras de toda a loja, buscando a identificação de um produto faltante. Diante da problemática da falta de produtos nas prateleiras, desenvolveu-se um software para identificação automática de ruptura, que gera alertas visuais aos responsáveis para que realizem o reabastecimento com rapidez e eficiência.

O software para identificação de ruptura foi desenvolvido na linguagem de programação C++ e sua interface foi construída utilizando o conceito de telas e diálogos. As telas desenvolvidas foram: inicialização e principal. A primeira realiza o carregamento de todos os dados da aplicação e mostra o seu status em uma lista; a segunda tem a missão de realizar toda a interação com usuário. Nesta são cadastrados todos os produtos, exibição de alertas e visualização rápida dos produtos registrados. Os diálogos foram construídos de forma que o usuário não precise sair da sua tela de trabalho principal e possa realizar as operações de cadastro produtos e alterações de parâmetros da aplicação. De forma geral, a interface da aplicação atendeu aos objetivos propostos, entregando agilidade para o usuário realizar o monitoramento de todos os produtos cadastrados.

Para o teste do software, foram capturadas imagens de produtos com diversas variações, sendo eles: Leite Linho, Óleo de Girassol, Molho Curry Terra, Farinha de Trigo Garota, Coca-Cola Tradicional e Achocolatado Pirakids. As imagens foram realizadas em um grande varejista, localizado na cidade de Canoas, Rio Grande do Sul. Devido às restrições impostas pela companhia, as imagens foram capturadas utilizando a câmera frontal do smartphone *Apple* modelo XR e um sistema de fixação adaptado no carrinho de compras.

Com as imagens capturadas, foram aplicado os testes em três etapas: o cadastro de produtos, a identificação de espaços vazios e a validação por pontos chave. Para a etapa de cadastros, os resultados para o conjunto de produtos selecionado foram satisfatórios — considerando as condições em que as imagens foram realizadas —, mesmo para produtos que não foram classificados como “bons”. O posicionamento foi suficiente, realizando 98% de acerto em média na identificação de sua quantidade. Ainda deve-se considerar que essa etapa é mais complexa de todo o processo, pois é fortemente influenciada por fatores como

iluminação das prateleiras, tamanho e forma do produto e principalmente o posicionamento e qualidade da câmera.

Na identificação de espaços vazios, os resultados esperados foram atingidos em praticamente todas as etapas, exceto no cenário em que o produto estava no fundo da prateleira, como o da Coca-Cola tradicional. Isso ocorreu devido ao produto possuir coloração preta e ter uma pequena área de detecção: seu rótulo.

Por fim, a etapa de identificação de pontos chave, utilizado como método alternativo, não atendeu a sua funcionalidade para o projeto, pois o método *match* possui bom desempenho quando o produto é exatamente o mesmo do cadastro (mesma imagem). Porém, isto não é viável porque mesmo que prateleiras não sejam alteradas, tratam-se de imagens diferentes e, nesse caso, o método não conseguiu manter um padrão aceitável de identificação.

Para trabalhos futuros, o algoritmo SIFT deve ser repensado para ser utilizado no auxílio da identificação ou não da presença do produto no seu quadro, ou seja, respectiva posição na prateleira. Isso iria auxiliar nos casos em que o produto não seja identificado pela limiarização, como por exemplo: produtos com fundo preto, produtos muito ao fundo da prateleira e produtos virados. Para finalizar os trabalhos futuros de software, pode-se realizar o desenvolvimento de identificação de ruptura no setor de hortifrúti, que não possui os mesmos tipos de prateleiras do departamento de mercearia. Uma maneira seria utilizar filtro de cores para essa detecção e, inclusive, o software desenvolvido neste projeto já foi construído para suportar essa estrutura.

Por fim, toda a parte de infraestrutura deve ser projetada para utilização do menor número de câmeras possível, reduzindo os custos de implementação do sistema em uma loja piloto para maiores avaliações.

REFERÊNCIAS

ARAÚJO, Samara Barbosa; MARTINS, Eliane Ferreira. Análise da ruptura de produtos nas gôndolas supermercadistas: um problema enfrentado por um supermercado de pequeno porte. In: XXX ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2010, São Paulo. **Anais...** São Paulo: [s.n.], 2010.

BRADSKY, Gary; KAEHLER, Adrian. **OpenCV – Computer vision with the OpenCV Library**. United States of America: O’Reilly Média, 2008.

COUTINHO, Daniel Pinto. SIFT - Scale Invariant Feature Transform. **LCG Computer Graphics Lab**, Rio de Janeiro, 2013. Disponível em: <<https://www.lcg.ufrj.br/marroquim/courses/cos756/trabalhos/2013/daniel-coutinho/daniel-coutinho-report.pdf>>. Acesso em: 28 nov. 2018.

FILHO, Ogê Marques; NETO, Hugo Vieira. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

GARCÍA-LAMONT, Farid; CUEVAS, A.; NIÑO, Y. Segmentation of color images by chromaticity features using self-organizing maps. **Ingeniería e Investigación**, v. 36, n. 2, p. 78-89, 2016. Disponível em: <<http://www.scielo.org.co/pdf/iei/v36n2/v36n2a11.pdf>>. Acesso em: 16 set. 2018.

GONZÁLES, Giancarlo Luis Gómez. **Medição de deformações elastoplásticas em regiões de concentração de tensões utilizando métodos sem malha e visão computacional**. 2014. 128 f. Tese (Doutorado em Engenharia Mecânica) - Programa de Pós-Graduação em Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, 2014. Disponível em: <http://meggi.usuarios.rdc.puc-rio.br/teses/PhD14_Giancarlo_Gonzales.pdf>. Acesso em: 18 set. 2018.

GONZALEZ, Rafael C.; WOODS, Richard. **Processamento de Imagens Digitais**. 5. ed. São Paulo: Edgard Blücher, 2013.

JÚNIOR, Francisco Alves de Oliveira. **Identificação e classificação de sinalização horizontal em autovias utilizando OpenCV**. 2016. 156 f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Paraíba, 2016. Disponível em: <http://repositorio.ifpb.edu.br/jspui/bitstream/177683/266/1/IDENTIFICA%C3%87%C3%83O%20E%20CLASIFICA%C3%87%C3%83O%20DE%20SINALIZA%C3%87%C3%83O%20HORIZONTAL%20EM%20AUTOVIAS%20UTILIZANDO%20OPENCV_Francisco_Alves.pdf>. Acesso em: 13 set. 2018.

LOWE, David G. Distinctive Image Features from Scale-Invariant Keypoints. **International Journal of Computer Vision**, [S.l.], v. 60, n. 2, p. 9-100, 2004. Disponível em: <<https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94>>. Acesso em: 15 set. 2018.

MESQUITA, Francisco Wagner da Silva; LOOS, Maurício Johnny. Controle de perdas causadas por rupturas nas gôndolas em uma rede de supermercado no estado do Ceará. **Exacta**, São Paulo, v. 15, n. 2, p. 275-285, 2017. Disponível em: <<https://www.redalyc.org/html/810/81052202008/index.html>>. Acesso em: 14 out. 2018.

MICROSOFT MSDN. **MFC Desktop Applications**. [S.I.], 2016. Disponível em: <<https://msdn.microsoft.com/pt-br/library/d06h2x6e.aspx>>. Acesso em: 30 nov. 2018.

OPEN SOURCE COMPUTATION VISION (OPENCV). **About**. [S.I.], 2019. Disponível em: <<https://opencv.org/about.html>>. Acesso em: 10 nov. 2018.

_____. **Canny Edge Detection**. [S.I.], 2015. Disponível em: <https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html>. Acesso em: 20 maio. 2018.

RAVI, Srisha; KHAN, Am. Morphological Operations for Image Processing: Understanding and its Applications. In: NATIONAL CONFERENCE ON VLSI, SIGNAL PROCESSING & COMMUNICATIONS, 2., 2013, India. **Anais...** India: (s.n.), 2013.

SILVA, Bruno Felipe de Nadai da. **Comparação de técnicas de correspondência de pontos chave para algoritmo ORB**. 2017, 63 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Curso de Ciência da Computação, Centro Universitário Eurípides de Marília, São Paulo, 2017. Disponível em: <<https://aberto.univem.edu.br/bitstream/handle/11077/1648/Bruno%20Felipe%20de%20Nadai%20da%20Silva.pdf?sequence=1&isAllowed=y>>. Acesso em: 19 nov. 2018.

SINHA, Utkarsh. SIFT: Theory and Practice. **Ai Shack**, [S.I.], (2010?). Disponível em: <<http://aishack.in/tutorials/sift-scale-invariant-feature-transform-log-approximation/>>. Acesso em: 10 out. 2018.

SOARES, Raíza Carla de Oliveira et al. Estudo sobre ruptura de estoque na indústria supermercadista. **Revista Científica Semana Acadêmica**, Fortaleza, v.1, n. 129, 2018. Disponível em: <<https://semanaacademica.org.br/artigo/estudo-sobre-ruptura-de-estoque-na-industria-supermercadista>>. Acesso em: 29 nov. 2018.