

UNIVERSIDADE DE SANTA CRUZ DO SUL  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Fernando Wartchow

**IDENTIFICAÇÃO E CONTAGEM DE INSETOS ATRAVÉS DE TÉCNICAS DE  
PROCESSAMENTO DE IMAGENS**

Santa Cruz do Sul  
2016

Fernando Wartchow

**IDENTIFICAÇÃO E CONTAGEM DE INSETOS ATRAVÉS DE TÉCNICAS DE  
PROCESSAMENTO DE IMAGENS**

Trabalho de conclusão apresentado ao curso de Ciência da Computação da Universidade de Santa Cruz do Sul para obtenção parcial do título de Bacharel em Ciência da Computação.

Orientador Prof. Dr. Rolf Fredi Molz

Santa Cruz do Sul  
2016

## RESUMO

Este trabalho apresenta um estudo relacionado à utilização de técnicas de processamento de imagens com o auxílio da biblioteca de código aberto chamado OpenCV no desenvolvimento de uma aplicação para realizar a contagem de insetos em armadilhas. A contagem é um procedimento para encontrar e controlar focos de infestações de insetos nos armazéns de empresas de tabaco. Utiliza-se para este fim armadilhas compostas com um feromônio sexual, que atraem e prendem os insetos.

O objetivo é aperfeiçoar o processo de contagem dando mais rapidez e possibilitar maior assertividade das informações para determinar as ações e práticas a serem tomadas para tratar a área infestada.

Quanto aos resultados, pode-se citar que o uso de técnicas de processamento de imagens para a contagem automática de insetos é aplicável para a execução da tarefa. Conforme resultados obtidos nos testes, a aplicação demonstrou-se eficaz com variação de 3% entre a contagem automática e manual em uma amostragem de 722 insetos.

**Palavras chave:** Processamento de Imagens, armadilhas, contagem de insetos.

## ABSTRACT

This final work presents a study related to image processing techniques with the support of an open source library called OpenCV for the development of an application to run an automatic insect counting from traps. The insect counting is a procedure to find and allow infestation control in the tobacco companies' warehouse. Traps that contain a sexual pheromone that attract and glue insects on it are used. The goal is improve the counting process, making it faster and more assertive to determine actions and practices to be taken to treat the infested area.

The outcome analyses indicate that is possible to use image processing techniques to perform an automatic insect counting. According the test results, the application achieved 3% variance between automatic and manual counting in a sample of 722 insects.

**Keywords:** Image processing, traps, insect counting.

## LISTA DE FIGURAS

Figura 1 – Convenção dos eixos para representação de imagens digitais .....	14
Figura 2 – Etapas de um sistema de processamento de imagens .....	14
Figura 3 – Espectro eletromagnético com destaque para sua faixa visível .....	16
Figura 4 – Imagem colorida (RGB) .....	17
Figura 5 – Canais RGB separados ( <i>Red, Green e Blue</i> ) .....	18
Figura 6 – Exemplo de histograma da imagem (veículo) e histograma da imagem (à direita) .....	18
Figura 7 – Histogramas com diferentes contrastes e luminosidades .....	19
Figura 8 – Exemplo de máscara 3 x 3 sobre uma imagem .....	20
Figura 9 – Exemplo de técnica de subtração de imagem .....	23
Figura 10 – Código de cadeia .....	26
Figura 11 – Lista do código de cadeia do contorno .....	26
Figura 12 – Imagem típica contendo moscas e a máscara .....	29
Figura 13 – Etapas de segmentação .....	29
Figura 14 – Fluxo de etapas para detecção de insetos .....	31
Figura 15 – Imagens processadas.....	32
Figura 16 – Imagem de saída após classificador SVM .....	33
Figura 17 – Comparação da imagem 1 com as 5 subsequentes .....	35
Figura 18 – Fluxo de processamento de imagens .....	38
Figura 19 – Imagem original .....	40
Figura 20 – Imagem após aplicação do filtro Bilateral (direita) e imagem original (esquerda) .....	41
Figura 21 – Separação das três bandas da imagem original .....	42
Figura 22 – Imagem borrada e o resultado da subtração .....	43
Figura 23 – Imagem filtrada e histograma .....	44
Figura 24 – Imagem após aplicação do limiar .....	45

Figura 25 – Imagem após dilatação .....	46
Figura 26 – Imagem após erosão .....	47
Figura 27 – Representação dos contornos e a medida do perímetro .....	48
Figura 28 – Imagens para análise visual .....	51
Figura 29 – Dados descritivos dos objetos .....	52
Figura 30 – Traças de fumo sobrepostas .....	52
Figura 31 – Resultados obtidos na primeira execução .....	53
Figura 32 – Diminuição do perímetro devido o reflexo da luz .....	54
Figura 33 – Novos atributos descritivos de cada objeto .....	55
Figura 34 – Imagem de objetos com a forma circular .....	56
Figura 35 – Novos resultados após otimização .....	57
Figura 36 – Análise variação da mosca do fumo .....	58
Figura 37 – Análise variação do bicho do fumo .....	58

## LISTA DE TABELAS

Tabela 1 – Diferença entre contagem manual e contagem pelo ImageJ .....	34
Tabela 2 – Comparativo entre os trabalhos relacionados .....	36
Tabela 3 – Dados descritivos das imagens .....	49
Tabela 4 – Lista das imagens adquiridas .....	50

## LISTA DE ABREVIATURAS

PDI	–	Processamento Digital de Imagens
SVM	–	<i>Support Vector Machine</i>
HOG	–	<i>Histogram Oriented Gradient</i>
GMM	–	<i>Gaussian Mixture Model</i>



## SUMÁRIO

1	INTRODUÇÃO .....	11
1.1	Objetivos.....	12
1.2	Estrutura do trabalho .....	12
2	FUNDAMENTAÇÃO TEÓRICA.....	13
2.1	Representação de Imagens Digitais .....	13
2.2	Etapas de um Sistema de Processamento de Imagens.....	14
2.3	Imagens coloridas e o modelo RGB .....	15
2.3.1	Separação dos canais RGB.....	16
2.4	Histograma.....	18
2.5	Filtros.....	20
2.5.1	Filtros de Suavização (Passa-Baixa).....	21
2.5.2	Filtros de Realce (Passa-Alta) .....	21
2.6	Subtração de imagens .....	22
2.7	Limiarização ( <i>Thresholding</i> ) .....	24
2.8	Operadores Morfológicos.....	24
2.9	Representação e Descrição dos Objetos.....	25
3	TRABALHOS RELACIONADOS .....	28
3.1	Método automatizado para contagem de mosca branca em folhas de soja utilizando Processamento de Imagens Digitais. ....	28
3.2	Detecção Precoce de Pragas em Folhas usando <i>Support Vector Machine</i> . 30	
3.3	Novas Técnicas para Contagem de Mosquitos Adultos: Usando <i>ImageJ Software</i> para Estimar o Número de Mosquitos Adultos em uma Armadilha.....	33
3.4	Considerações sobre os trabalhos relacionados .....	35
4	SOLUÇÃO DESENVOLVIDA .....	37
4.1	Visão Geral.....	37

4.2	Etapas de Processamento de Imagens.....	39
4.2.1	Aquisição da Imagem.....	39
4.2.2	Filtragem para Remoção de Ruídos – Primeira etapa .....	40
4.2.3	Separação das bandas .....	41
4.2.4	Subtração e Correção do Fundo .....	42
4.2.5	Filtragem para Remoção de Ruídos – Segunda etapa .....	43
4.2.6	Segmentação via Limiar .....	44
4.2.7	Operação Morfológica.....	45
4.2.8	Representação e descrição dos contornos .....	47
4.3	Validação do Trabalho .....	50
4.3.1	Coleta das imagens.....	50
4.3.2	Calibração da aplicação .....	51
4.3.3	Primeiros resultados obtidos.....	53
4.3.4	Otimização dos resultados .....	54
4.3.5	Resultados obtidos após otimização .....	57
5	CONCLUSÃO.....	59
	REFERÊNCIAS.....	60

## 1 INTRODUÇÃO

Nos dias atuais, em virtude da alta concorrência nacional e internacional, as empresas de tabaco vêm buscando melhorias nos seus processos internos para garantir produtividade e qualidade no seu produto final.

Um destes processos é o controle de infestação, cujo intuito é o monitoramento dos insetos que comprometem a qualidade e a sanidade do tabaco, referindo-se tanto ao tabaco cru quanto ao já processado, que se encontram armazenados em seus estoques. Geralmente o tabaco processado tende a ficar de 1 a 3 anos estocado para melhorar seu aroma e outras características de qualidade, ficando assim exposto ao risco de infestação.

Praticamente todo o tabaco produzido no Brasil é exportado para diversos países. Na maioria dos países é necessária a apresentação de documentos que comprovam tratamentos fitossanitários realizados no país de origem. Caso as empresas não executem um trabalho de forma eficiente, isto pode implicar em custos extras, redução da qualidade do tabaco e conseqüentemente a insatisfação dos consumidores [BEUREN, 2013].

Conforme dados da Coresta (Centro de cooperação científica e pesquisa relativas ao tabaco) estima-se que houve perda em torno de quatrocentos milhões de dólares no ano de 2014, que equivaleu a 1% da produção de tabaco desde a propriedade do produtor rural até a fábrica de cigarro.

Tal processo é realizado com a visita aos pontos de controles, onde se encontram “armadilhas” que contém um feromônio sexual para atrair os insetos com finalidade de identificar focos de infestação. Em cada ponto é realizada uma contagem manual e a quantidade encontrada é preenchida em um formulário para posteriormente incluir os resultados em planilhas eletrônicas.

Durante o processo, devido à contagem e tabulação ser realizada de forma manual pode-se gerar informações incorretas. Além disso, devido às buscas por melhorias nos processos internos e também a otimização dos seus recursos, torna-se importante obter informações exatas e no menor tempo possível.

Após a consolidação destes dados, são realizadas análises dos indicadores por seção ou departamento e com base nos resultados é possível investigar as causas

e tomar ações para o controle de infestação. Conforme Beuren (2013), a realização de um controle e monitoramento confiável permite avaliar a eficácia das técnicas e atividades empregadas para a diminuição dos focos de insetos.

## **1.1 Objetivos**

Assim, o objetivo deste trabalho é utilizar técnicas de processamento de imagens para auxiliar no controle de infestações.

Para isso, consideram-se os seguintes objetivos específicos:

- Estudar as espécies de insetos que contaminam o tabaco para entender e extrair as características visuais.
- Pesquisar e compreender as variadas técnicas de processamento de imagens.
- Estudar a biblioteca OpenCV e seus algoritmos de processamento de imagens para auxiliar no desenvolvimento do trabalho.
- Desenvolver uma ferramenta que possibilite a coleta de dados através de fotos tiradas com câmera fotográfica, e que realize contagem dos insetos presentes nas armadilhas.
- Validar a ferramenta desenvolvida.

## **1.2 Estrutura do trabalho**

O trabalho está estruturado da seguinte forma: o capítulo 2 aborda a fundamentação teórica sobre Processamento de Imagens e as características principais, conceitos e técnicas. O capítulo 3 traz os trabalhos relacionados, que são os fundamentos dos assuntos abordados. No capítulo 4 é apresentada a solução proposta a ser implementada, e por fim o capítulo 5 contém as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

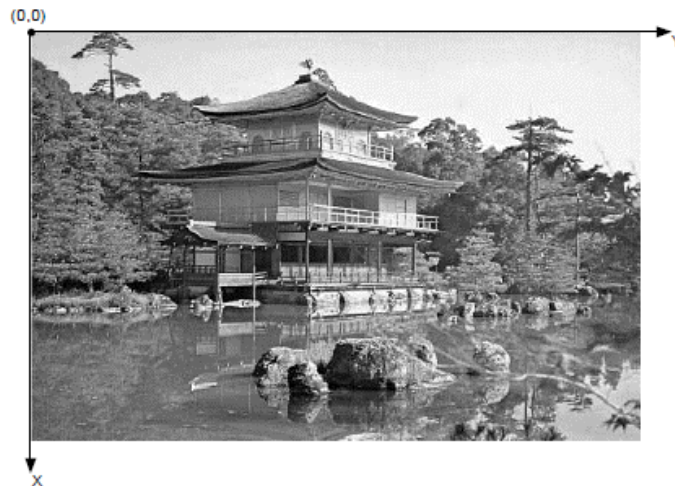
A área de processamento de imagens e visão computacional ainda é um desafio devido à complexidade do sistema visual humano. O objetivo é a obtenção de informações suficiente para distinguir diferentes objetos de modo confiável, de maneira a auxiliar na solução de problemas imitando a habilidade visual dos seres humanos. De forma simplificada, podem-se separar em dois níveis as tarefas envolvidas: processamento de imagens (baixo nível) e análise de imagens (alto nível) [PEDRINI,2008]. O processamento de imagens transforma as imagens para melhorar e facilitar a extração de informações. Já a análise de imagens baseia-se em características essenciais como formas, cor ou textura para distinguir objetos.

Métodos de baixo nível realizam operações como redução de ruído, aumento de contraste, extração de bordas e a compressão de imagens, que são comumente representadas em uma matriz retangular com coordenadas  $(x, y)$  onde cada elemento representa valores de brilho da imagem. Entretanto, o método de alto nível corresponde as atividades de segmentação e descrição que facilita o processo de reconhecimento e classificação dos objetos através dos contornos ou bordas, medidas, cores, textura, dentre outras [PEDRINI, 2008].

Neste capítulo serão abordados conceitos sobre imagens digitais iniciando na descrição da representação no plano cartesiano, fluxo de processamento de imagens, e demais técnicas e métodos até a obtenção dos dados descritivos das imagens.

### 2.1 Representação de Imagens Digitais

A imagem é representada em um plano com coordenadas de um ponto  $x$  e  $y$  correlacionada com uma função  $f(x, y)$  sendo que  $f$  é o valor da intensidade da luz ou brilho (ou também nível de cinza) no ponto  $(x, y)$ . A imagem digital é considerada uma matriz onde linhas e colunas identificam o ponto na imagem e valor do elemento naquele ponto identifica o nível de cinza. Na literatura os pontos são chamados de elementos de imagens, elementos da figura, *pixels* dentre outros [GONZALEZ, 2000]. Por convenção os eixos do plano cartesiano de uma imagem são representados como na figura 1:



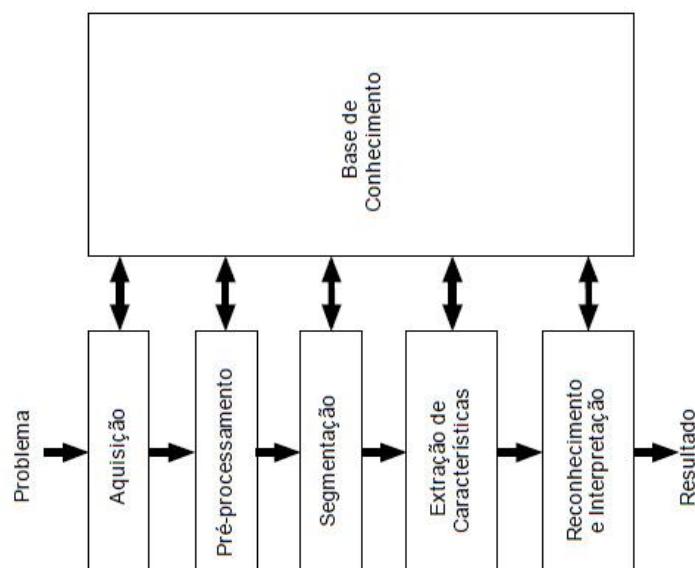
**Figura 1 – Convenção dos eixos para representação de imagens digitais**

Fonte: [FILHO, 1999]

Conforme acordo a origem da imagem é situada no canto superior esquerdo, inclusive é a mesma lógica da biblioteca OpenCV que é utilizada neste trabalho.

## 2.2 Etapas de um Sistema de Processamento de Imagens

Para a solução de problemas que envolvem processamento digital de imagens, geralmente são seguidas as etapas ilustradas na figura 2. Dependendo do domínio do problema, nem sempre é necessário executar todos os módulos. Cada etapa contempla técnicas de processamento de imagens.



**Figura 2 – Etapas de um sistema de processamento de imagens**

Fonte: [GONZALEZ, 2000]

A etapa de aquisição ocorre por meio da captura de uma imagem por um hardware, tais como: máquina fotográfica, *scanner*, satélite, tomógrafo médico, dentre outros. A imagem adquirida é convertida para a etapa subsequente. A imagem capturada pode conter algumas imperfeições decorrentes, por exemplo, das condições de iluminação ou a própria característica do dispositivo. Sendo assim, o objetivo da etapa de pré-processamento é melhorar a imagem através da aplicação de técnicas para redução de ruídos, correção de contraste ou brilho. A terceira etapa é de segmentação que divide uma imagem em unidades significativas, ou seja, realiza a identificação de áreas de interesse como, por exemplo, detecção de descontinuidades (bordas) [PEDRINI, 2008] [GONZALEZ, 2002] [FILHO, 1999].

As próximas etapas são consideradas de alto nível e visam extrair características dos objetos presentes na imagem que geralmente são descritas por atributos numéricos, sendo assim, a entrada ainda é uma imagem, já a saída passa a ser um conjunto de dados referentes àquela imagem (descritores).

A última etapa abrange o Reconhecimento e Interpretação dos objetos conforme suas características descritivas extraídas na etapa anterior, isso corresponde ao processo de atribuição de um rótulo para cada objeto de acordo com seus descritores. O último módulo da figura 2 é a base do conhecimento. Este módulo está vinculado com todas as etapas de um sistema de processamento de imagens, serve como guia de comunicação entre os módulos e é profundamente dependente da aplicação e do resultado que se busca atingir [PEDRINI, 2008] [GONZALEZ, 2002] [FILHO, 1999].

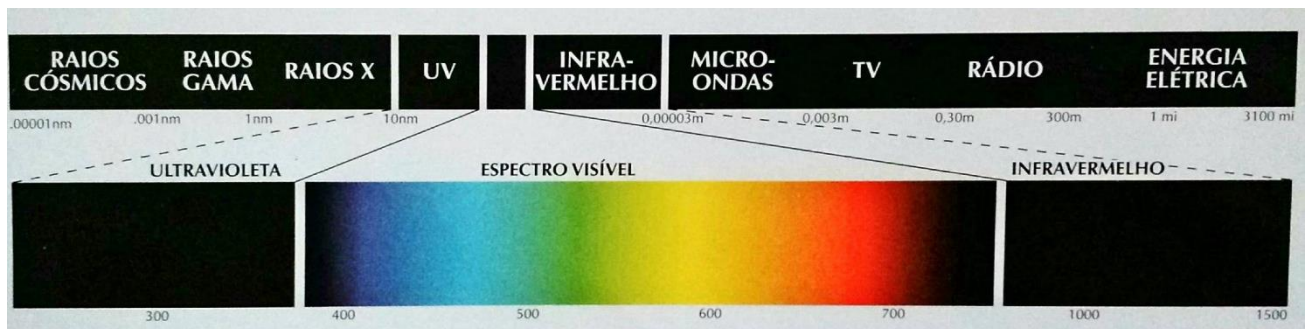
### **2.3 Imagens coloridas e o modelo RGB**

Apesar de praticamente todas as técnicas de processamento de imagens antes da aplicação dos algoritmos converterem uma imagem colorida em tons de cinza, é importante ressaltar sobre os fundamentos das imagens coloridas. Embora a percepção das cores pelo cérebro humano não esteja completamente compreendida pela sociedade científica.

Basicamente, os olhos humanos podem distinguir as cores em milhares de tonalidades e intensidades diferentes e tal percepção em um objeto é determinada

pela luz refletida no mesmo. Essa faixa de luz perceptível é composta de uma banda de frequência muito pequena, conforme pode ser visto na figura 3.

Naturalmente, devido à estrutura do olho humano, todas as cores são vistas como combinações variadas de três cores primárias: vermelho, verde e azul. Isto originou no modelo de cores mais utilizados em dispositivos atualmente, esse modelo é conhecido como RGB (R do inglês “red”, G do inglês “green” e B do inglês “blue”) [PEDRINI, 2008].



**Figura 3 – Espectro eletromagnético com destaque para sua faixa visível**

Fonte: [GONZALEZ, 2000]

Modelos de cores existem para manter especificações padronizadas para atender diferentes tipos de dispositivos gráficos que manipulam cores. O Modelo RGB é muito importante para o processamento de imagens, pois é utilizado em diversos dispositivos como monitores e câmeras de vídeo. Consiste em três planos independentes, ou seja, um para cada cor primária (vermelho, verde e azul) que misturadas podem formar várias cores com diferentes intensidades [GONZALEZ,2002] [PEDRINI, 2008].

### 2.3.1 Separação dos canais RGB

Imagens em nível de cinza são comumente armazenadas em 8 bits, ou seja, 256 *pixels* e cada *pixel* pode conter um valor de intensidade que vai de 0 a 255. Como já mencionado anteriormente, as imagens na etapa de pré-processamento são geralmente convertidas para tons de cinza, que facilita o processamento pois a imagem contém agora menos pixels comparada com uma imagem colorida.



As imagens coloridas do tipo RGB são compostas de três canais com as cores primárias vermelha, verde e azul. Cada um destes três canais ocupa 8 bits então uma imagem colorida precisa de 24 bits para armazenar todo o conjunto de informações. A figura 4 é uma imagem RGB com três canais.



**Figura 4 – Imagem colorida (RGB)**

Fonte: [<https://sites.google.com/site/learnimagej/tutorials/converting-image-formats>]

A função de intensidade é representada como:

$I_{RGB} = (F_R, F_G, F_B)$ , onde  $F_R(x, y)$  é a intensidade do pixel  $(x, y)$  no canal vermelho,  $F_G(x, y)$  é a intensidade do pixel  $(x, y)$  no canal verde e  $F_b(x, y)$  é a intensidade do pixel  $(x, y)$  no canal azul [KUMAR, 2010].

Como pode ser analisado, a imagem original (figura 4) contém cores que se aproximam da cor primária vermelha, sendo assim a intensidade dos *pixels* na banda vermelha é maior e acaba gerando uma imagem com maior contraste, conforme figura 5. Pode-se ver que as penas do chapéu que é de cor mais azulada também ficaram mais contrastadas no canal azul.

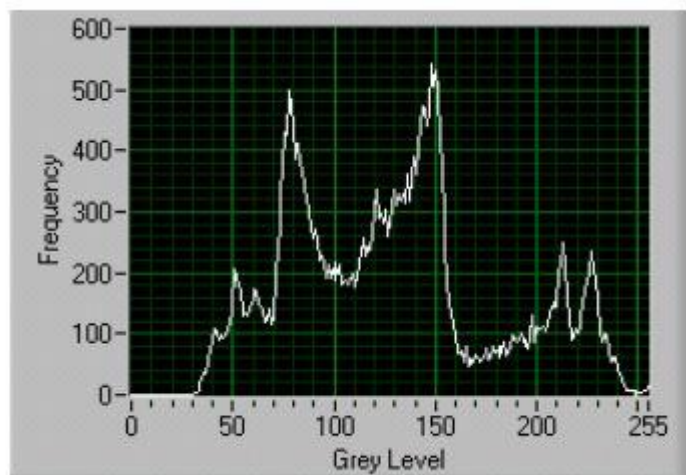


**Figura 5 – Canais RGB separados (*Red, Green e Blue*)**

Fonte: [<https://sites.google.com/site/learnimagej/tutorials/converting-image-formats>]

## 2.4 Histograma

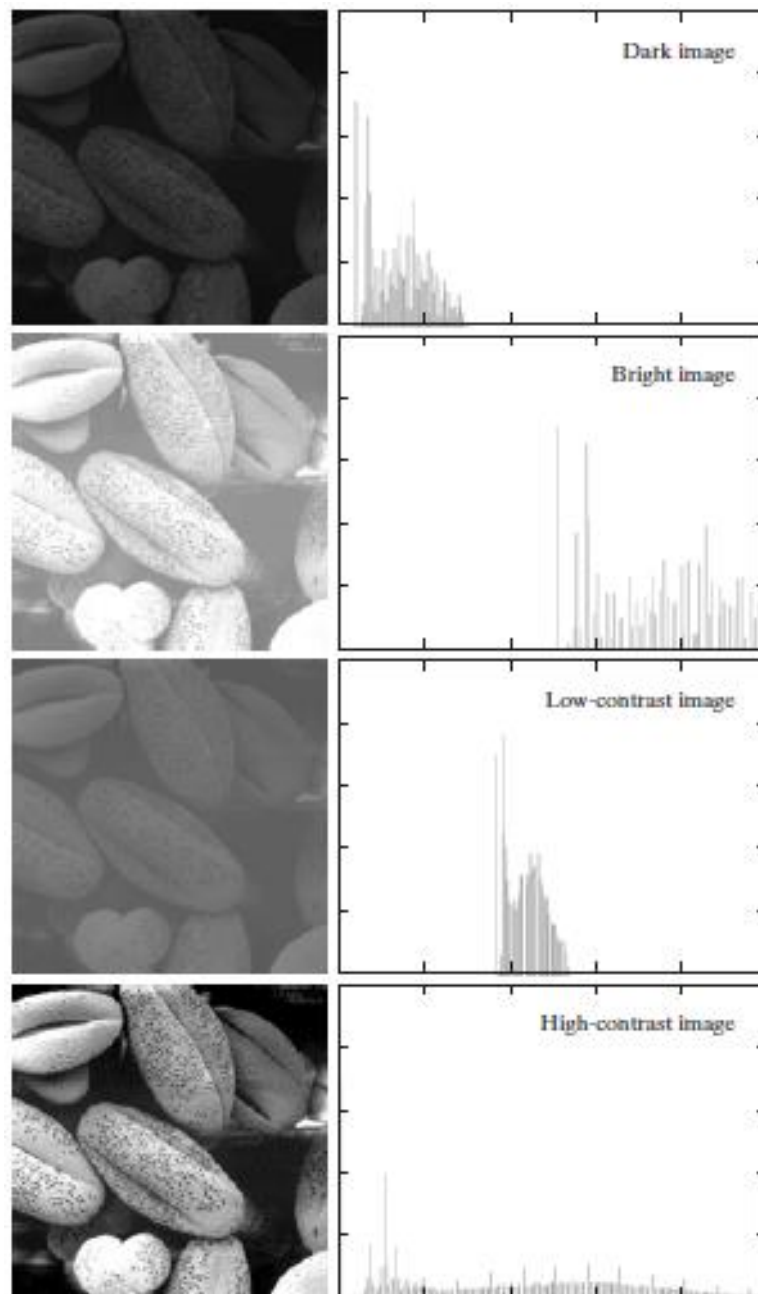
O histograma de uma imagem é utilizado para demonstrar a distribuição dos níveis de cinza. É representado na forma de um gráfico sendo que o eixo x mostra as diferentes intensidades do nível de cinza (0 a 255) e o eixo y a quantidade de *pixels* de cada intensidade (figura 6 à direita). Esta quantidade do eixo y é na realidade uma distribuição discreta da probabilidade, pois cada imagem pode conter um número diferente de *pixels*.



**Figura 6 – Exemplo de histograma da imagem (veículo) e histograma da imagem (à direita)**

Fonte: [ERTÜRK, 2003]

Com o histograma é possível avaliar o contraste da imagem diferenciando em baixo contraste quando os níveis de cinza se encontram comprimidos no gráfico, e por alto contraste quando os níveis de cinza estão mais bem distribuídos. Na figura 7 pode-se observar quatro imagens iguais, porém com diferente distribuição de contraste e luminosidade.



**Figura 7 – Histogramas com diferentes contrastes e luminosidades**

Fonte: [GONZALEZ, 2002]

## 2.5 Filtros

Operações com filtros criam uma nova imagem do mesmo tamanho, porém o *pixel* destino não depende apenas do *pixel* origem, mas sim do conjunto da vizinhança de *pixels* da imagem original. O conjunto vizinhança é conhecido como máscara, *template*, janela ou *kernel* e pode ter variados tamanhos, como por exemplo, 3 x 3, 5 x 5, 7 x 7, etc. Cada ponto da máscara pode conter um valor numérico com peso/coeficiente diferente. O propósito é mover a máscara sobre a imagem e ir modificando o *pixel* destino com base na vizinhança delimitada pela máscara aplicada, conforme demonstrado na figura 8:

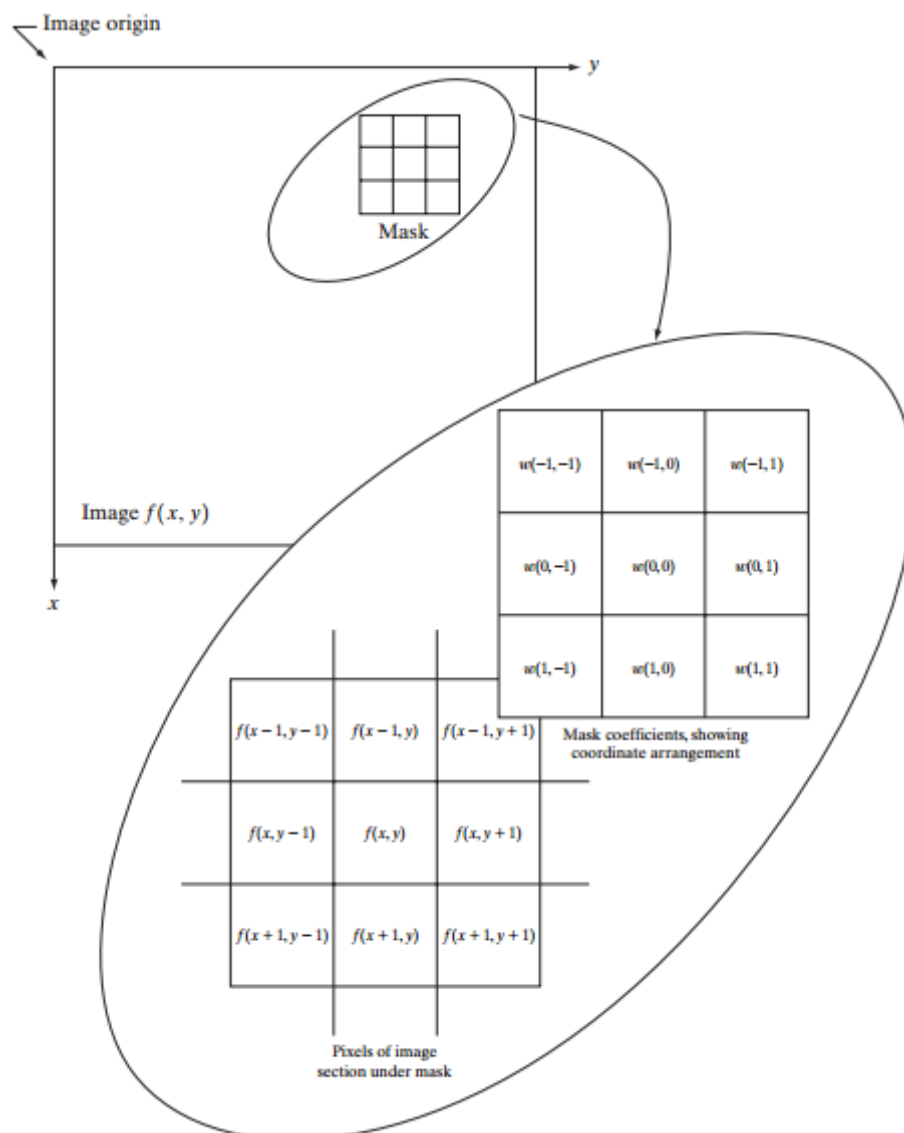


Figura 8 – Exemplo de máscara 3 x 3 sobre uma imagem

Fonte: [GONZALEZ, 2002]

As categorias de filtros mais comuns segundo Pedrini (2008) são os do tipo passa-baixa e passa-alta. Filtros passa-baixa atenuam as altas frequências de uma imagem, que estão relacionadas com os detalhes da imagem, gerando o efeito de suavização que tem como resultado a diminuição de ruídos.

Ao contrário dos filtros passa-baixa, os passa-alta realçam as altas frequências, tornando as imagens mais nítidas e as transições entre regiões mais visíveis, portanto, é uma operação para detecção de bordas. O efeito indesejável das operações dos filtros passa-alta é destacar os ruídos presentes nas imagens [PEDRINI, 2008].

Nos dois próximos capítulos será explicado com mais detalhes sobre o funcionamento dos filtros passa-baixa e passa-alta.

### **2.5.1 Filtros de Suavização (Passa-Baixa)**

Filtros de suavização são usados para borrar uma imagem reduzindo seus ruídos. A técnica é utilizada na etapa de pré-processamento para remover pequenos detalhes antes da extração de objetos maiores, preenchimento de alguns espaços em branco entre linhas ou curvas. Normalmente o resultado de uma operação de suavização é a média dos *pixels* vizinhos da máscara. A ideia geral é substituir cada *pixel* pela média dos níveis de cinza da vizinhança definidas pela máscara, que resulta na redução dos realces na transição dos níveis de cinza [GONZALEZ, 2002], [PEDRINI, 2008].

Ruídos são conhecidos pela abrupta transição nos níveis de cinza, desta forma a maior utilização destes filtros é para a redução de ruídos. Entretanto, podem afetar na detecção de bordas que são característica desejáveis no processamento de imagens [GONZALEZ, 2002].

### **2.5.2 Filtros de Realce (Passa-Alta)**

O principal objetivo dos filtros passa-alta é realçar detalhes finos em uma imagem aumentando assim detalhes que foram denegridos tanto por uma ocorrência de erro no momento da aquisição da imagem como por um efeito natural. O operador, em um determinado ponto, atua no nível de descontinuidade de intensidade da

imagem, desse modo realçando bordas, linhas e outras descontinuidades (como ruídos), e atenuando áreas com intensidades de variação mais suave [GONZALEZ, 2002].

## 2.6 Subtração de imagens

A diferença entre duas imagens  $f(x, y)$  e  $h(x, y)$  é expressa como:

$g(x, y) = f(x, y) - h(x, y)$ , e é obtida computando as diferenças entre os pares correspondentes de *pixels* das imagens  $f$  e  $h$ . A real utilidade da subtração de imagens é de aprimorar e ressaltar a diferença entre duas imagens [GONZALEZ, 2002].

Para analisar imagens a questão da iluminação é um fator importante, pois pode auxiliar na discriminação de características para facilitar na contagem, encontrar medidas e identificar objetos. Sem dúvida a questão da iluminação no momento de transformar uma imagem em tons de cinza para uma imagem binária seria um método simples e rápido [RUSS, 1995].

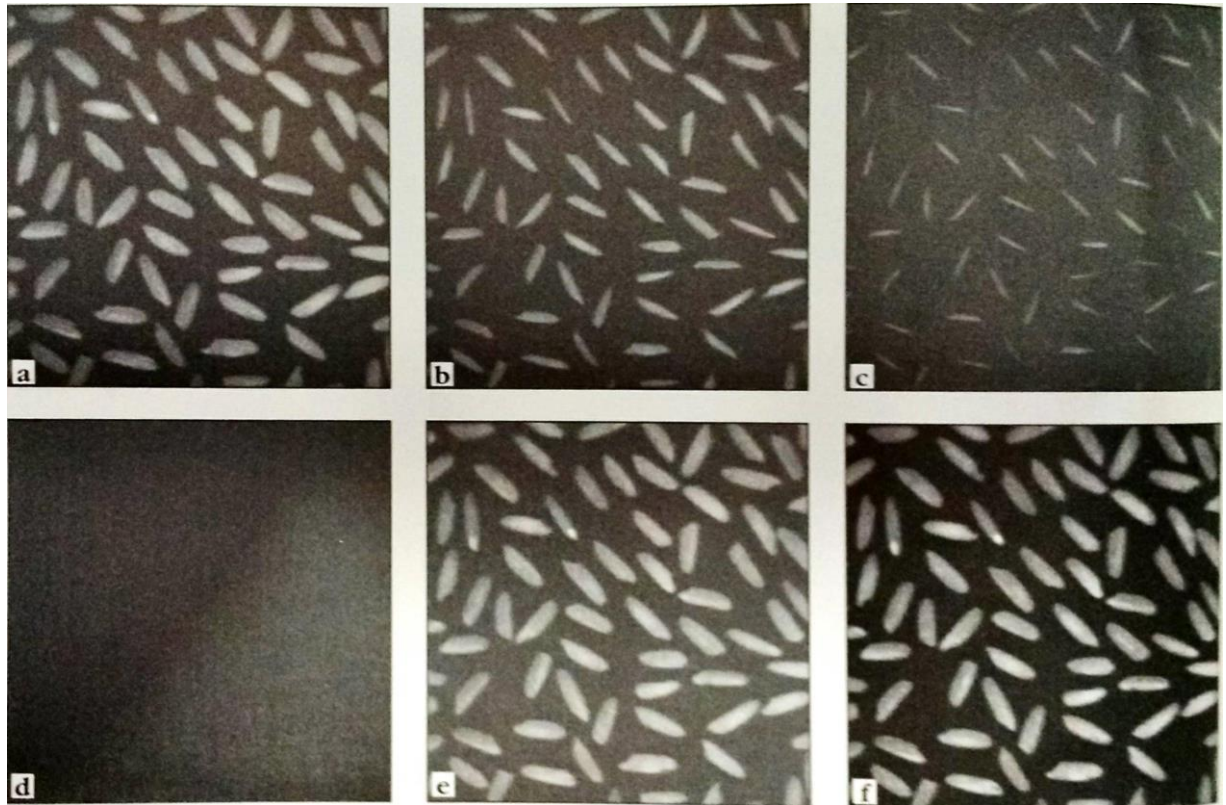
Muitas vezes, devido a iluminação de um ambiente no momento da aquisição da imagem, pode causar uma iluminação irregular da imagem. Portanto a subtração da imagem original por uma com o fundo corrigido pode auxiliar na uniformização da iluminação, balanceando a imagem e auxiliando na etapa de limiarização [RUSS, 1995].

Quando a variação de fundo é mais irregular do que funções simples que poderiam auxiliar na solução do problema, outras técnicas podem ser utilizadas, principalmente quando as características desejáveis da imagem são menores do que as diferenças do fundo.

Uma técnica proposta é a de utilizar filtros de suavização para borrar a imagem diminuindo as características de realce da imagem. Tal diminuição ocorre fundindo os objetos com fundo, ou seja, estendendo o fundo para toda a imagem para posteriormente subtrair a imagem processada com a original [RUSS, 1995].

Conforme exemplo da figura 9, pode ser observado o procedimento sendo aplicado em uma imagem com grãos de arroz com um fundo escuro e desigual. As aplicações de filtros de suavização acabam fundindo os grãos, ou seja, diminuindo os mesmos e estendendo o fundo da imagem. Depois de algumas repetições os grãos de arroz desaparecem. A técnica tende a remover os picos do histograma e com a

subtração da imagem original com a imagem processada tem como resultado uma imagem mais balanceada que claramente define as características que permitem a separação do fundo com a técnica de limiarização [RUSS, 1995].



**Figura 9 – Exemplo de técnica de subtração de imagem**

Fonte: [RUSS, 1995]

- a) Imagem de grãos de arroz com iluminação não uniforme;
- b) Suavização da imagem para fundir itens ao fundo;
- c) Repetição da suavização;
- d) Depois de algumas repetições apenas permanece o fundo;
- e) Resultado da subtração da imagem *d* por *a*;
- f) Imagem balanceada com o contraste expandido.

Entretanto a habilidade de balancear variações de fundo através da subtração de imagens tem um custo moderado de processamento e por final a imagem resultante não tem mais iluminação remanescente para demonstrar detalhes nas áreas escuras o que pode vir a interferir em análise e processamento futuro dependendo da aplicação [RUSS, 1995].

## 2.7 Limiarização (*Thresholding*)

Um método simples para a segmentação de imagem é o de limiarização, também conhecido como *thresholding*, que transforma uma imagem em tons de cinza numa imagem binária. A forma mais simples de *thresholding* faz-se através da partição do histograma da imagem pelo uso de único do parâmetro de *threshold*, sendo conhecida como a variável  $T$ .

A segmentação da imagem é feita da seguinte maneira, separando a imagem *pixel a pixel* o objeto ou fundo (*background*), se o valor de intensidade do nível de cinza é maior ou menor que o valor da variável  $T$ . Este método pode ser definido por: onde  $f(x, y)$  representa a função original,  $g(x, y)$  a função binária e  $T$  o valor de *threshold* [GONZALEZ, 2002].

O valor do *threshold* deve ser determinado dependendo da imagem em questão, existindo várias possibilidades para atribuir valores a esta variável. A forma mais simples é um valor estático: cada valor de cinza menor ou igual que um determinado valor  $T$  fica com o valor zero na imagem final e qualquer outro valor fica como 1.

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

Existem ainda técnicas de *threshold* que utilizam vários valores  $T$ , por exemplo, subdividindo a imagem em vários segmentos. Os histogramas destas imagens são multimodais, com picos acentuados na sua distribuição, sendo estes escolhidos como valores de *threshold* na segmentação de imagem.

## 2.8 Operadores Morfológicos

A utilização de operações morfológicas em imagens está relacionada a estrutura dos objetos dentro de uma imagem. Após o pré-processamento e a discriminação inicial dos objetos, as operações morfológicas trabalham para clarear a estrutura dos objetos. Isto é feito pela simplificação dos *pixels* nos limites de cada



objeto, ou seja, na extração, afinamento e preenchimento das bordas como também preenchimento de frestas e fissuras, conseqüentemente facilitando as etapas de medidas e reconhecimento dos objetos que serão necessárias para as operações de classificação.

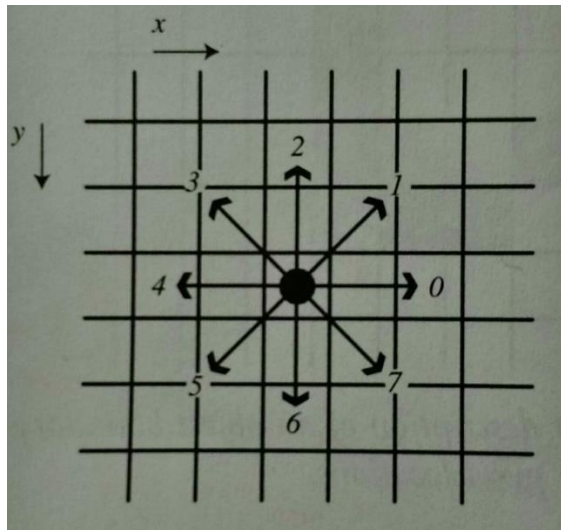
Portanto, o processo acima descrito utiliza um elemento estruturado, ou também conhecido como máscara morfológica. Tal elemento, dependendo da aplicação, é geralmente formado com as dimensões de um quadrado tipo 3 x 3, 5 x 5 e se move através da imagem de entrada comparando os *pixels* e alterando a imagem de saída de acordo com o especificado no elemento estruturado. As operações morfológicas mais fundamentais e citadas são erosão e dilatação [BAXES, 1994] [PEDRINI, 2008].

A operação de erosão uniformemente reduz o tamanho dos objetos em relação ao seu fundo, já a dilatação faz o inverso da erosão, expandindo o tamanho dos objetos. Tais operações são usadas para eliminar objetos indesejáveis como o ruído e bordas irregulares, portanto deixando a imagem mais adequada para as demais etapas de processamento de imagens [BAXES, 1994].

## 2.9 Representação e Descrição dos Objetos

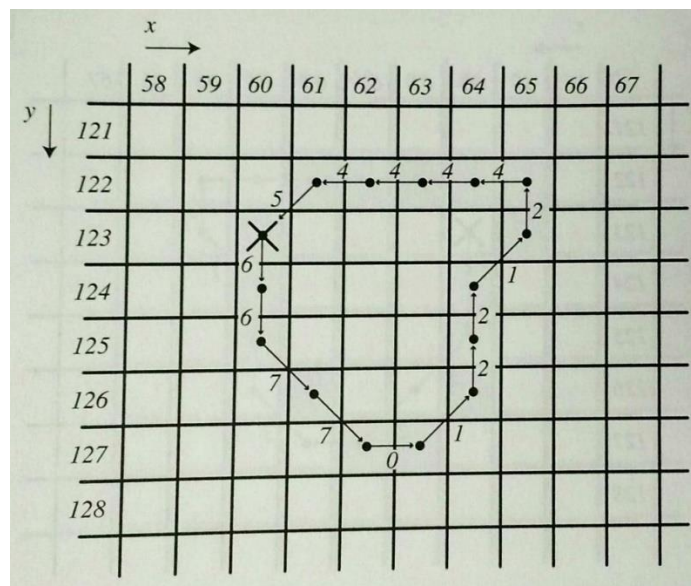
Nesta etapa do processamento de imagens é onde se seleciona um conjunto de características extraídas do objeto de interesse para o propósito de classificação, ou seja, após a imagem ter sido segmentada em diferentes objetos os mesmos agora contém um formato e conseqüentemente uma medida. Tais objetos contém características externas (bordas) e internas (pixels que compõem o objeto) [PEDRINI, 2008].

Para definir o contorno de um objeto inicialmente é escolhido um ponto da borda e o passo seguinte é encontrar o próximo pixel do contorno seguindo uma direção definida conforme a figura 10 até que a cadeia de *pixels* forme um vetor de pontos interligados conhecida como código de cadeia. Sendo então necessário seguir a seqüência do código de cadeia para recriar a borda e obter assim a medida do perímetro do objeto conforme figura 11 [BAXES, 1994].



**Figura 10 – Direções do código de cadeia**

Fonte: [BAXES, 1994]



**Figura 11 – Lista do código de cadeia do contorno**

Fonte: [BAXES, 1994]

As medidas mais comuns dos objetos são as que descrevem seu formato e são as dimensões físicas que caracterizam a aparência de um objeto. Há uma lista de diferentes medidas possíveis de serem abstraídas, portanto dependendo da aplicação apenas algumas são necessárias. Objetos podem as vezes conter formas regulares, tais como quadrada, retangular, circular ou elipsoidal, mas as vezes pode ter um

formato estranho, porém, são importantes para a aplicação desenvolvida [BAXES, 1994].

Abaixo seguem alguns exemplos de medidas conforme [BAXES, 1994]:

- a) Perímetro: distância em *pixels* ao redor de um objeto, o resultado é a medida do comprimento do limite de um objeto;
- b) Área: área de *pixels* do interior do objeto. É computada pelo número total de *pixels* dentro do objeto, incluindo a borda. O resultado é o tamanho do objeto;
- c) Centroide: é o *pixel* do objeto que está balanceada acima, abaixo, esquerda e a direita. Para objetos simples como um quadrado ou um círculo a centroide é o *pixel* central;
- d) Caixa delimitadora: mais conhecida como *bounding box* é um quadrado ou retângulo que cerca todo o objeto sendo sua dimensão o eixo maior multiplicado pelo eixo menor.

### 3 TRABALHOS RELACIONADOS

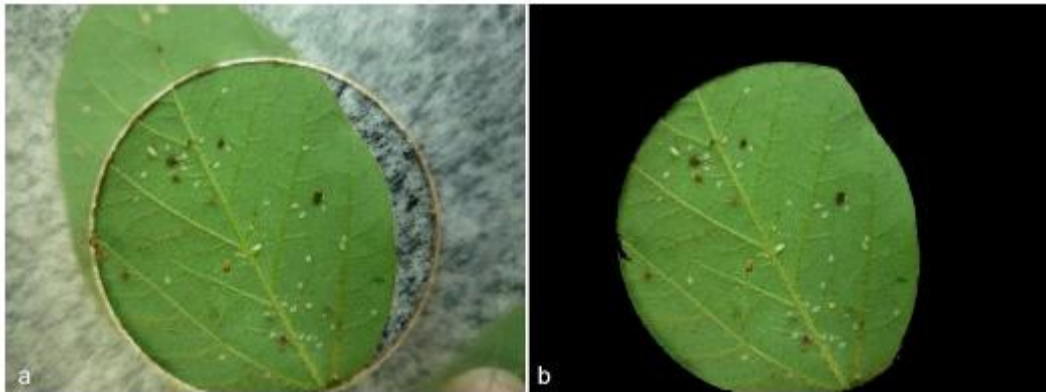
Conforme pesquisa realizada sobre o assunto no Portal Capes, que é um portal de internet de caráter educacional para fins de pesquisa, foram encontrados sete trabalhos similares utilizando técnicas de processamento de imagens para contagem automatizada de insetos com a utilização de armadilhas. A pesquisa foi realizada com as seguintes palavras: *image processing*, *insect count*, *trap*. Dentre os pesquisados, consideraram-se os de maior relevância os seguintes:

#### 3.1 Método automatizado para contagem de mosca branca em folhas de soja utilizando Processamento de Imagens Digitais.

Segundo Barbedo [2012] a mosca branca é um inseto que causa bastante prejuízo principalmente no cultivo da soja, pois se alimentam diretamente das folhas e transmitem vários tipos de doenças. Portanto o controle de infestação ajuda na detecção e na posterior aplicação de estratégias para eliminação dos focos. E devido ao tamanho das lavouras fica muito difícil monitorar tais insetos, então surgiu a ideia de utilizar processamento de imagens para contagem de insetos.

A técnica proposta por Barbedo [2012] é definida por duas estratégias principais: 1) a modificação da imagem original de acordo com suas cores para realçar características desejáveis dos objetos; 2) algumas operações matemáticas morfológicas para delimitar os objetos. Com a utilização deste método é possível também identificar e contar os insetos em suas diferentes etapas do ciclo de vida bem como algumas lesões presentes na folha.

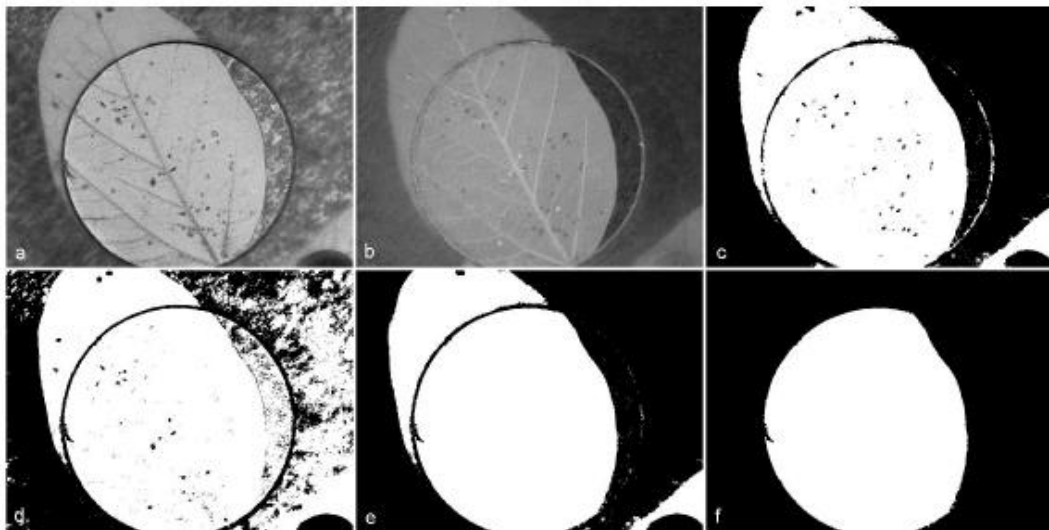
Foram usadas diversas fotos para validação do trabalho, cada uma contendo diversos estágios do ciclo de vida da mosca branca. As fotos foram tomadas com uma câmera com um tubo de 10 cm de diâmetro acoplada para limitar a área. As imagens foram tiradas no formato RGB (*Red-Green-Blue*), conforme pode ser visto na figura 12.



**Figura 12 – Imagem típica contendo moscas e a máscara**

Fonte: [BARBEDO, 2014]

Como pode ser visto na figura 12, o tubo plástico isolou a área de interesse da imagem. Foram aplicadas então conversões dos modelos de cores RGB para CMYK (*cyan, magenta, yellow e key*) com a separação de canais específicos para posterior binarização através da aplicação de limiares para criar máscaras e posteriormente operação bit-a-bit AND, lógica pode ser visualizada na figura 13.



**Figura 13 – Etapas de segmentação**

Fonte: [BARBEDO, 2014]

- a) Representação do canal C do modelo de cor CMYK;
- b) Representação do canal Y do modelo de cor CMYK;

- c) Máscara binária do canal C;
- d) Máscara binária do canal Y;
- e) Combinação (AND) das duas imagens;
- f) Máscara final definindo a área de interesse.

Barbedo [2014] aplicou as seguintes técnicas de conversão de modelos de cor e *threshold* para detectar os diferentes objetos na imagem;

- Mosca branca (adulta): RGB – Lab – XYZ (canal Z) e *pixels* com *threshold* acima de 242 são transformados em brancos.
- Mosca branca (jovem): RGB – XYZ – XYZ - CMYK (canal C) e *pixels* com *threshold* abaixo de 13 são transformados em brancos.
- Esqueletos vazios: RGB – CMYK – XYZ (canal Z) e *pixels* com *threshold* abaixo de 128 são transformados em brancos.
- Lesões e mofo: RGB – Lab – CMYK (canal M) e *pixels* com *threshold* acima de 128 são transformados em brancos.

Uma observação de Barbedo [2014] é sobre a diferença do tamanho da mosca branca em outros estágios do ciclo de vida, porém as procuras das mais jovens se são maiores que o tamanho normal, acabam sendo descartadas.

Na conclusão de Barbedo [2014], é demonstrado que o processo automatizado, apesar de ter atingido 93% de acerto, é muito mais rápido do que o processo manual de contagem.

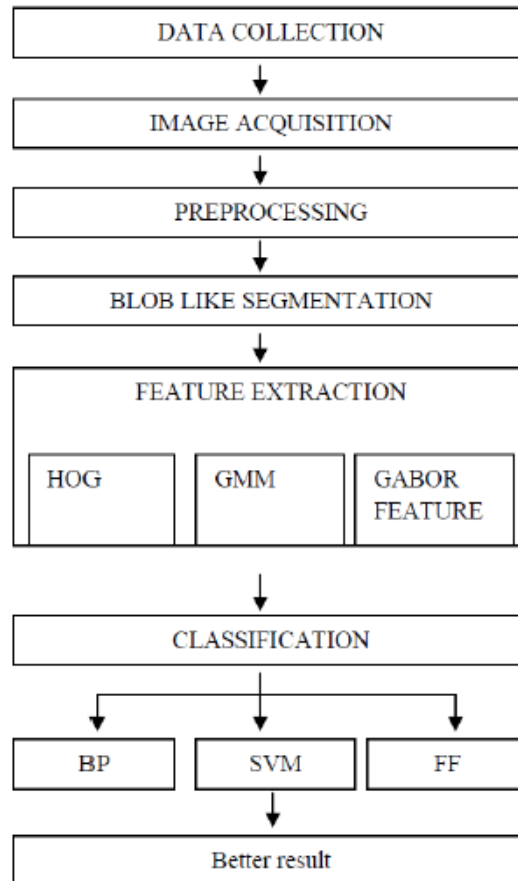
### **3.2 Detecção Precoce de Pragas em Folhas usando *Support Vector Machine*.**

O trabalho desenvolvido por Manoja [2014] também se refere a pesquisas para auxiliar na detecção de infestações no cultivo de variados produtos agrícolas na Índia, seu país de origem. O autor cita que contagem manual demanda muito tempo e que a detecção precoce do bioagressor é fundamental para o gerenciamento de culturas.

Manoja [2014] citou diversos trabalhos semelhantes e destacou que alguns autores utilizam típicas técnicas de processamento de imagens tais como: diferenciação por tamanho, morfologia matemática para reconstrução de contornos e também a utilização dos modelos de cores. Destacou o trabalho de Huddar [2012] que

propôs as etapas de conversão de cor, segmentação, redução de ruídos e contagem dos insetos e alcançando uma precisão de 96%.

A proposta de Manoja [2014] é mostrada na figura 14 que consiste na realização de etapas até o resultado final. Foi utilizado o *software* MATLAB.



**Figura 14 – Fluxo de etapas para detecção de insetos**

Fonte: [MANOJA, 2014]

Na etapa de pré-processamento são removidos do fundo ruídos de baixa frequência, normalizando a intensidade das partículas da imagem e melhorando a aparência visual. Manoja [2014] destaca a importância desta etapa para que não haja perda de qualidade. Seguidamente a imagem RGB é convertida em uma imagem em tons de cinza para realçar os insetos na folha. Próxima técnica aplicada são filtros de média e mediana com máscara 3 x 3 e o tipo de filtro que tiver melhor resultado permanece para a próxima etapa e o resultado pode ser visualizado na figura 15.



**Figura 15 – Imagens processadas**

Fonte: [MANOJA, 2014]

Esquerda: imagem original;

Centro: imagem convertida para tons de cinza;

Direita: imagem após a aplicação do filtro da média.

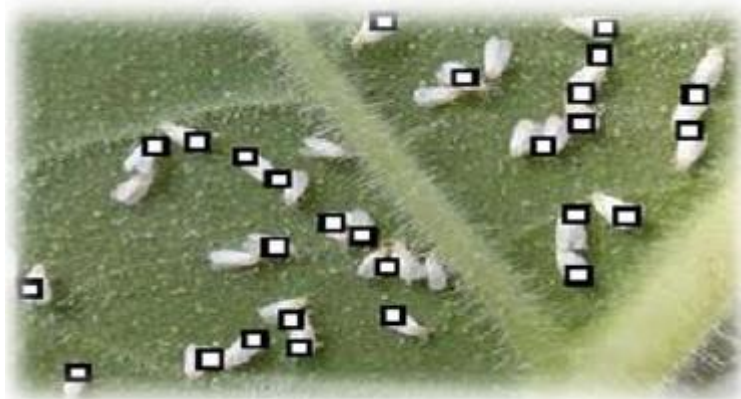
A etapa seguinte é a de segmentação que é geralmente definida como um processo para separar a imagem entre grupos ou regiões diferentes. É utilizado *Blob* para detecção das regiões. *Blob* auxilia na obtenção de regiões de interesse para aplicação nas próximas etapas de processamento. A técnica é aplicada para detectar a presença dos mesmos objetos em uma imagem.

Seguindo o fluxo do trabalho de Manoja [2014], a próxima etapa é de extração de características. Nesta etapa usualmente trabalha-se com cor, formato e textura e este passo inclui três etapas:

- 1) HOG (*Histogram Oriented Gradient*): é o descritor utilizado com propósito de detectar o objeto através da distribuição do raio da cor na imagem.
- 2) GMM (*Gaussian Mixture Model*): utilizado para análise da textura que no trabalho refere-se à detecção das formas dos insetos presentes na imagem.
- 3) *Gabor Feature*: calcula a relação entre grupos de dois *pixels* para determinar a orientação do inseto na imagem.

Manoja [2014], na última etapa do seu trabalho que se refere a classificação de imagens, ele aplica três tipos de classificadores: *Back Propagation*, *Feed Forward* e SVM. E como conclusão pode verificar que SVM trouxe melhores resultados. SVM é um classificador não linear e utilizado para reconhecimento de padrões que permitiu distinguir entre insetos e a folhas, conforme podemos observar na figura 16.





**Figura 16 – Imagem de saída após classificador SVM**

Fonte: [MANOJA, 2014]

### **3.3 Novas Técnicas para Contagem de Mosquitos Adultos: Usando *ImageJ Software* para Estimar o Número de Mosquitos Adultos em uma Armadilha.**

O trabalho de Kesavaraju [2012] também tem como objetivo auxiliar no controle e vigilância de insetos através da contagem realizada por técnicas de processamento de imagens em armadilhas com o intuito de diminuir o tempo entre a coleta e contagem. O modelo da armadilha utilizada é a de *CO<sub>2</sub>-Baited* e foram espalhadas 51 delas em regiões de Salt Lake City, USA. Houve a coleta de uma quantidade média de 1541 insetos por armadilha.

Para o desenvolvimento da pesquisa foi utilizado o software de código aberto ImageJ, que tem seu desenvolvimento baseado em Java.

Após a coleta, as redes que capturam os insetos foram esvaziadas transferindo-os para uma bandeja untada com amaciante de roupas para reduzir a eletricidade estática, prevenindo assim a aglomeração de insetos. Alguns insetos que não interessavam na pesquisa foram manualmente removidos. As bandejas foram colocadas sob iluminação de lâmpadas fluorescentes e fotografadas com uma máquina fotográfica fixada em uma base. Posteriormente os insetos foram contados manualmente para comparar com a contagem automática.

ImageJ permite realizar diferentes ajustes das variáveis que permite executar a calibração e adaptação das técnicas para encontrar o objeto de interesse. Kesavaraju [2012] realizou diferentes calibrações do software até identificar a menor variação entre a contagem manual e contagem via processamento de imagem para

estimar o número de mosquitos adultos. Conforme pode ser analisado na tabela 1, a melhor calibração encontrada foi a número 14 da tabela (destacada em negrito).

**Tabela 1 – Diferença entre contagem manual e contagem pelo ImageJ**

Num	Limiar	Tamanho	Contagem ImageJ	Variação entre manual e ImageJ ± Correlação de erro		
1	0,43	43–Infinito	1.683.22	235.86	±	27.87
2	0,43	45–Infinito	1.652.55	229.35	±	26.57
3	0,43	50–Infinito	1.583.00	219.76	±	27.81
4	0,43	40–Infinito	1.731.29	253.39	±	30.82
5	0,43	30–Infinito	1.931.71	393.52	±	46.57
6	0,43	20–Infinito	2.261.02	719.31	±	72.34
7	0,43	55–Infinito	1.522.12	220.37	±	31.01
8	0,45	43–Infinito	1.583.00	213.88	±	27.81
9	0,5	43–Infinito	1.679.75	191.62	±	27.77
10	0,55	43–Infinito	1.600.78	229.86	±	25.48
11	0,5	100–Infinito	1.476.94	363.92	±	53.16
12	0,43	100–Infinito	1.183.51	393.45	±	77.06
13	0,5	45–Infinito	1.160.92	182.98	±	24.01
<b>14</b>	<b>0,5</b>	<b>50–Infinito</b>	<b>1.575.67</b>	<b>174.11</b>	<b>±</b>	<b>21.80</b>
15	0,5	53–Infinito	1.519.35	175.88	±	22.27
16	0,5	60–Infinito	1.488.73	193.33	±	25.34

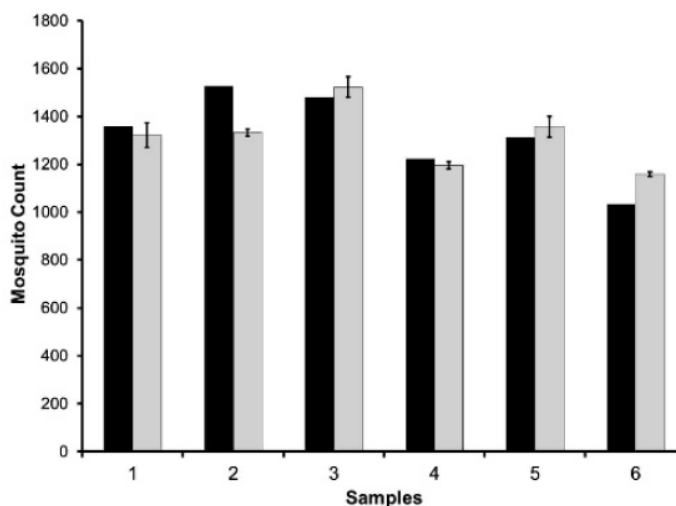
Fonte: [KESAVARAJU, 2012]

As técnicas de processamento de imagens utilizadas por Kesavaraju [2012] dão início com a transformação da imagem para oito *bits* em preto e branco (binária) por meio de um limiar que possibilitou a diferenciação dos insetos e do fundo. Em seguida utilizou-se uma técnica de análise de partículas e dentre as opções de configuração utilizou-se as mais importantes, tais como:

- Tamanho: que determina área mínima e máxima (0-infinito);
- Circularidade (0.00 – 1.00): que determina o formato da área.

O cálculo de correlação para a ideal calibração foi avaliado após comparar as mesmas bandejas, porém, misturando os insetos e fotografando novamente (5 vezes) obtendo assim um teste de regressão. Dentre os dezesseis testes realizados, o que

menor diferença teve foi a número quatorze, que obteve noventa e três por cento de assertividade. O teste de regressão foi concretizado com testes utilizando-se seis armadilhas para amostragem. Pode-se observar na figura 17 um gráfico comparativo. As barras na cor preta é a primeira imagem e as barras cinza são a média das cinco imagens subsequentes.



**Figura 17 – Comparação da imagem 1 com as 5 subsequentes**

Fonte: [KESAVARAJU, 2012]

A conclusão do trabalho de Kesavaraju [2012] é que o protocolo para realizar a contagem teve um bom resultado, porém se a média de insetos por imagem aumentar ou diminuir deve-se alterar as configurações. Variações podem ocorrer conforme localização das armadilhas, como por exemplo, se for à cidade o número de insetos por armadilha seria menor que a média utilizada no desenvolvimento do trabalho e na realização dos testes se comparado com o interior de uma cidade onde a incidência de insetos é consideravelmente maior. E quanto ao objetivo do tempo, os autores também obtiveram um bom resultado, pois as armadilhas foram levadas ao laboratório para processamento e retornadas a campo em média de quatro horas.

### **3.4 Considerações sobre os trabalhos relacionados**

Os trabalhos estudados na sua introdução mencionam a utilização de contagem automatizada de insetos como uma forma de auxiliar no controle de infestação, que é uma prática utilizada em todo o mundo e em diversas culturas, com o objetivo de acelerar o processo de tratamento das áreas infestadas.

Os trabalhos foram desenvolvidos utilizando ferramentas como o MATLAB e ImageJ para realizar o processamento. As etapas sugeridas nos trabalhos seguem basicamente o fluxo da figura 2 que demonstra a sequência padrão de técnicas de processamento de imagem digital. Na literatura não foi encontrado a utilização da biblioteca OpenCV.

Observa-se também o controle do ambiente no momento da aquisição da imagem no trabalho de Kesavaraju [2012], que há toda uma adequação a priori da aquisição. Já comparado com Barbedo [2014] a imagem é adquirida diretamente a folha da planta onde se encontra os insetos, mas que se deve ter atenção com iluminação pois regiões podem ficar sob efeito sombreado, a inclinação da folha no momento da fotografia e que se for necessário é movida com a mão para corrigir o ângulo.

**Tabela 2 – Comparativo entre os trabalhos relacionados**

<b>Trabalho</b>	<b>Ferramenta Utilizada</b>	<b>Ambiente</b>	<b>Técnica Processamento / Contagem</b>
Barbedo [2002]	-	Capturadas em laboratório	Manipulação das cores para identificação e limiarização
Manoja [2014]	MATLAB	Folhas são manipuladas se necessário	Filtro da média, BLOB GMM, Gabor, HOG, SVM
Kesavaraju [2012]	ImageJ	Relocação dos insetos para bandejas	Limiarização, Análise de Partículas (circularidade e tamanho)

Fonte: do autor

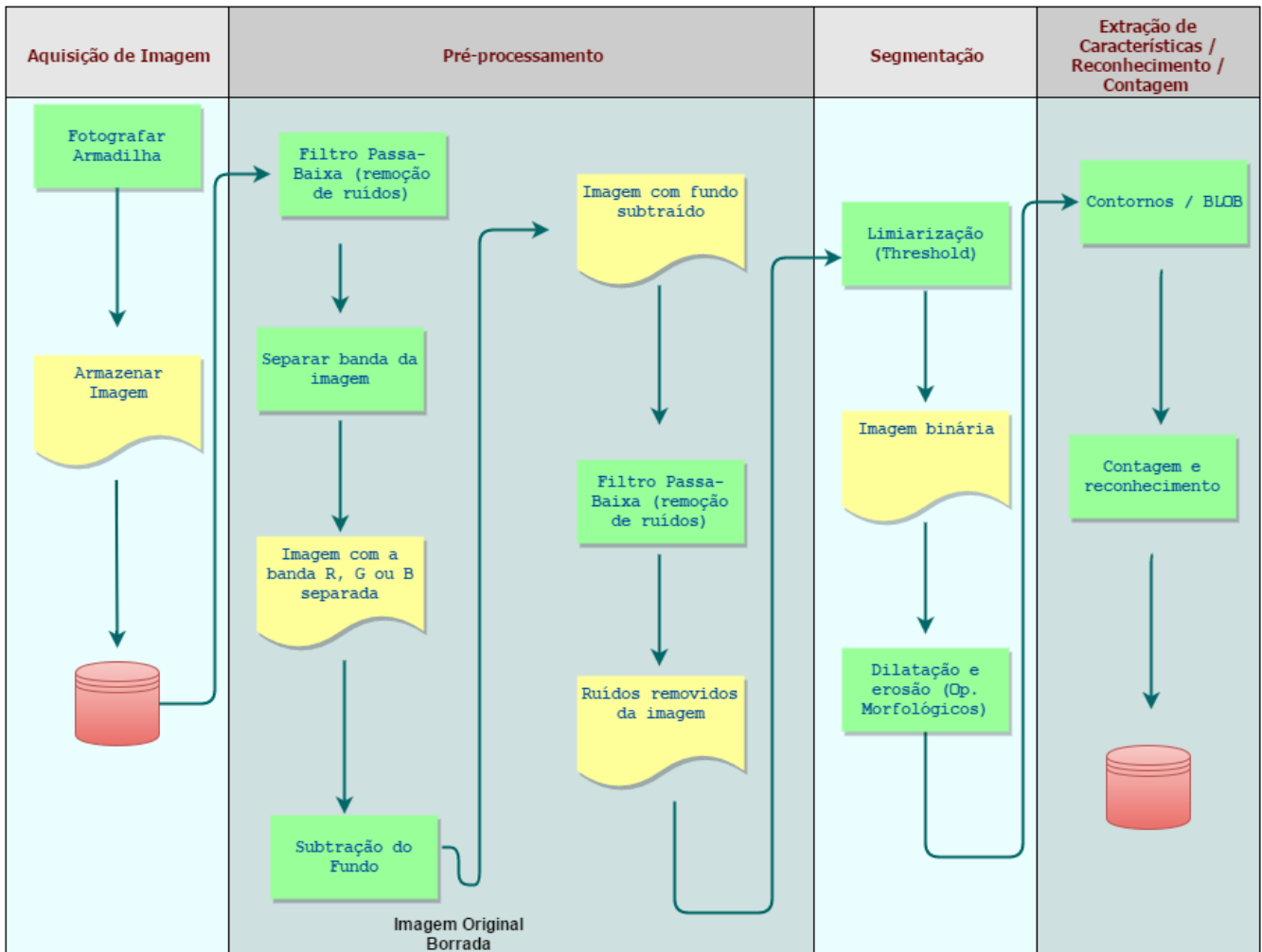
## **4 SOLUÇÃO DESENVOLVIDA**

Após apresentar nas seções anteriores os principais conceitos relacionados aos objetivos deste trabalho como também alguns trabalhos relacionados, esta seção expõe a aplicação destes conceitos ao apresentar a solução desenvolvida para o presente trabalho.

Essa seção inicia com uma visão geral do trabalho, com o objetivo de apresentar as técnicas e os conceitos que serão empregados para o desenvolvimento da ferramenta para contagem automática de insetos. Posteriormente, cada uma das etapas será explicada de maneira detalhada, para uma melhor compreensão do tema.

### **4.1 Visão Geral**

A ideia consiste na realização da contagem automática de insetos por armadilha através de uma imagem imitando a habilidade visual do ser humano de identificar e extrair informações necessárias para a solução do problema. Para isso, tomam-se como base as etapas descritas na Fundamentação Teórica que pode ser visualizado na figura 18.



**Figura 18 – Fluxo de processamento de imagens**

Fonte: do autor

Conforme pode ser observado na figura 18, o processo é iniciado no momento da aquisição da imagem. Esta etapa é muito importante, pois todas as medidas necessárias para se ter uma imagem com qualidade devem ser observadas, como por exemplo a iluminação, ângulo da foto, distância entre a câmera e a armadilha.

Na etapa de pré-processamento o primeiro passo é uma filtragem para suavizar os ruídos e em seguida executar a separação de bandas do modelo de cor RGB transformando a imagem em três bandas separadas de 8 bits cada uma. Após este procedimento ocorre a subtração de fundo que é executada com o objetivo de segmentar a imagem em fundo e objetos que se encontram sobrepostos na armadilha. Entretanto, as imagens são compostas de ruídos que são amenizados com filtros passa-baixa removendo partículas indesejadas para a próxima etapa.

A fase de segmentação da imagem é onde ocorre a divisão da imagem entre a área que realmente interessa. Por exemplo, a imagem é transformada em binária onde a parte escura da imagem é descartada da análise e a parte branca se estabelece como área de interesse, que posteriormente na fase de análise das características, como forma e tamanho da(s) área(s) de interesse, será possível efetuar a contagem da quantidade de insetos presentes na imagem.

Para auxiliar no desenvolvimento, será utilizada a biblioteca de código aberto OpenCV. Esta biblioteca contém diversos algoritmos prontos para utilização com a pretensão de facilitar o desenvolvimento de projetos de visão computacional [BRADSKI, 2008]. A biblioteca foi utilizada de forma integrada ao *Visual Studio* e na linguagem de programação C++.

## **4.2 Etapas de Processamento de Imagens**

Nesta seção, as etapas citadas acima serão descritas e demonstradas para que seja possível uma melhor compreensão da aplicabilidade delas no desenvolvimento do sistema proposto, além da utilização dos conceitos apresentados nas seções anteriores.

### **4.2.1 Aquisição da Imagem**

A fase de aquisição da imagem deve ser feita com cautela para evitar distorções das áreas de interesse. Na figura 19 é demonstrada a imagem original que está sendo utilizada nos testes do corrente trabalho.



**Figura 19 – Imagem original**

Fonte: do autor

#### **4.2.2 Filtragem para Remoção de Ruídos – Primeira etapa**

Nesta etapa é utilizado o filtro conhecido como Filtro Bilateral sobre a imagem original reduzindo ruídos indesejáveis e acentuando as bordas. Nota-se que a escolha da aplicação de tal filtro foi devido a leve suavização aplicada nos insetos, preservando sua originalidade e em contraponto a diminuição dos ruídos do fundo da imagem. A comparação da imagem original e da imagem filtrada utilizando o Filtro Bilateral pode ser vista na figura 20. Para tal resultado, foi usada a função do OpenCV denominada *bilateralFilter(inputImage,bilateral,15,65,25)*, onde os parâmetros significam respectivamente a imagem de entrada, imagem de saída, o diâmetro da vizinhança de *pixels*, 65 o *sigmaColor* que mistura as cores dentro da vizinhança de *pixels* e por último 25 de *sigmaSpace* que influencia no *sigmaColor* quanto à similaridade das cores.



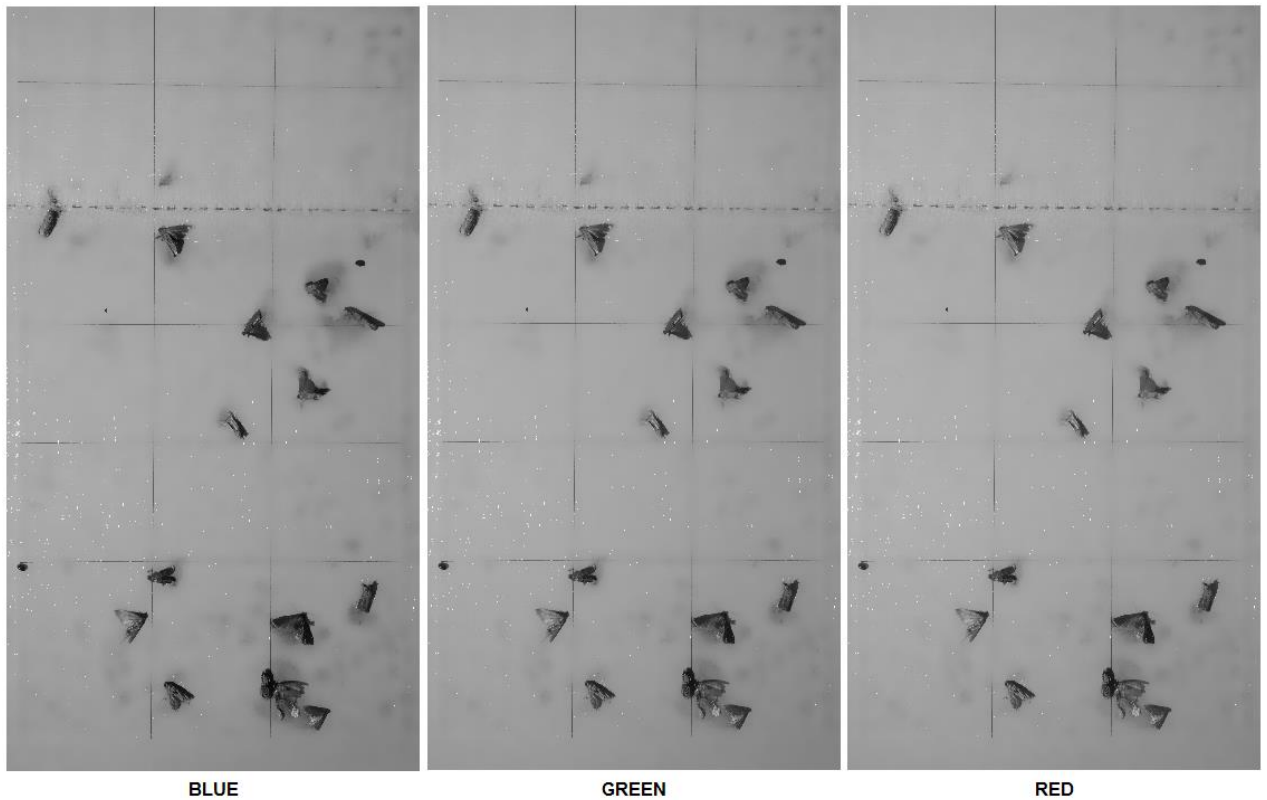


**Figura 20 – Imagem após aplicação do filtro Bilateral (direita) e imagem original (esquerda)**

Fonte: do autor

#### **4.2.3 Separação das bandas**

Na separação de bandas ocorre a divisão de cada canal da imagem realçando o contraste das áreas com a respectiva banda. A separação ocorre da imagem original e colorida para três imagens em tons de cinza com a iluminação realçada diferentemente para cada banda. Por exemplo, se uma imagem de uma flor da cor vermelha fosse separada, a banda *Red* ficaria clara na região onde o vermelho predominasse. Na figura 21 pode-se observar a separação das três bandas da imagem original.

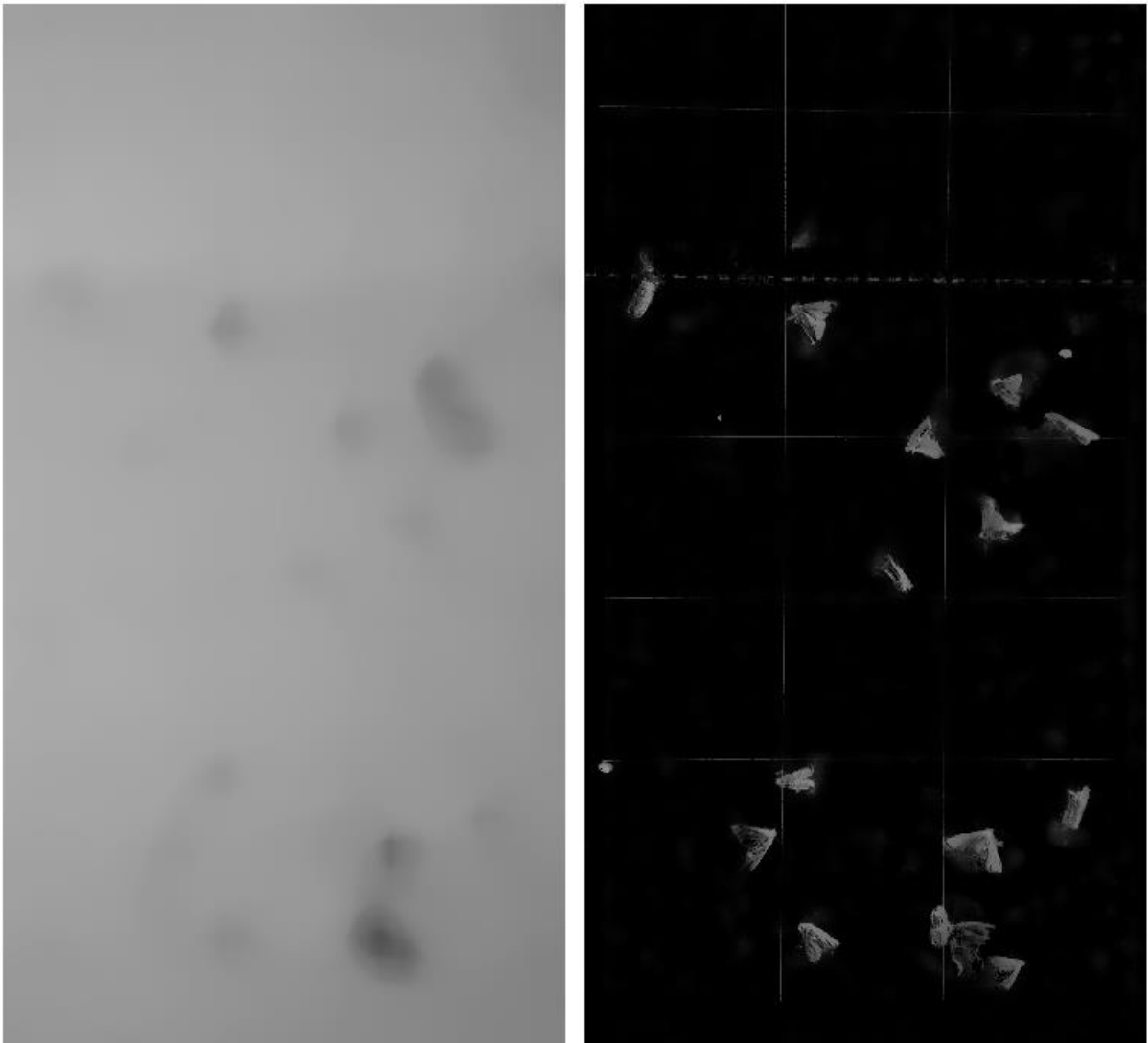


**Figura 21 – Separação das três bandas da imagem original**

Fonte: do autor

#### 4.2.4 Subtração e Correção do Fundo

Para a subtração de fundo, primeiramente a imagem da banda *Blue* é escolhida e processada com uma filtragem para suavizar a imagem. A escolha pela banda *Blue* foi devido ao melhor contraste e também na geração de melhor resultado na etapa seguinte. No filtro a máscara aplicada tem o tamanho consideravelmente grande de vizinhança de pixels, atenuando as baixas frequências, sendo utilizada a função *medianBlur(outputImage,mask,41)*, sendo *outputImage* a imagem de entrada (imagem da banda *Blue*), *mask* a imagem de saída e tamanho 41 de vizinhança de *pixels*. E conseqüentemente ao subtrair a imagem borrada com a imagem original, obtém-se uma imagem com o fundo corrigido. Na figura 22 pode-se visualizar a imagem à esquerda borrada e o resultado da subtração à direita.

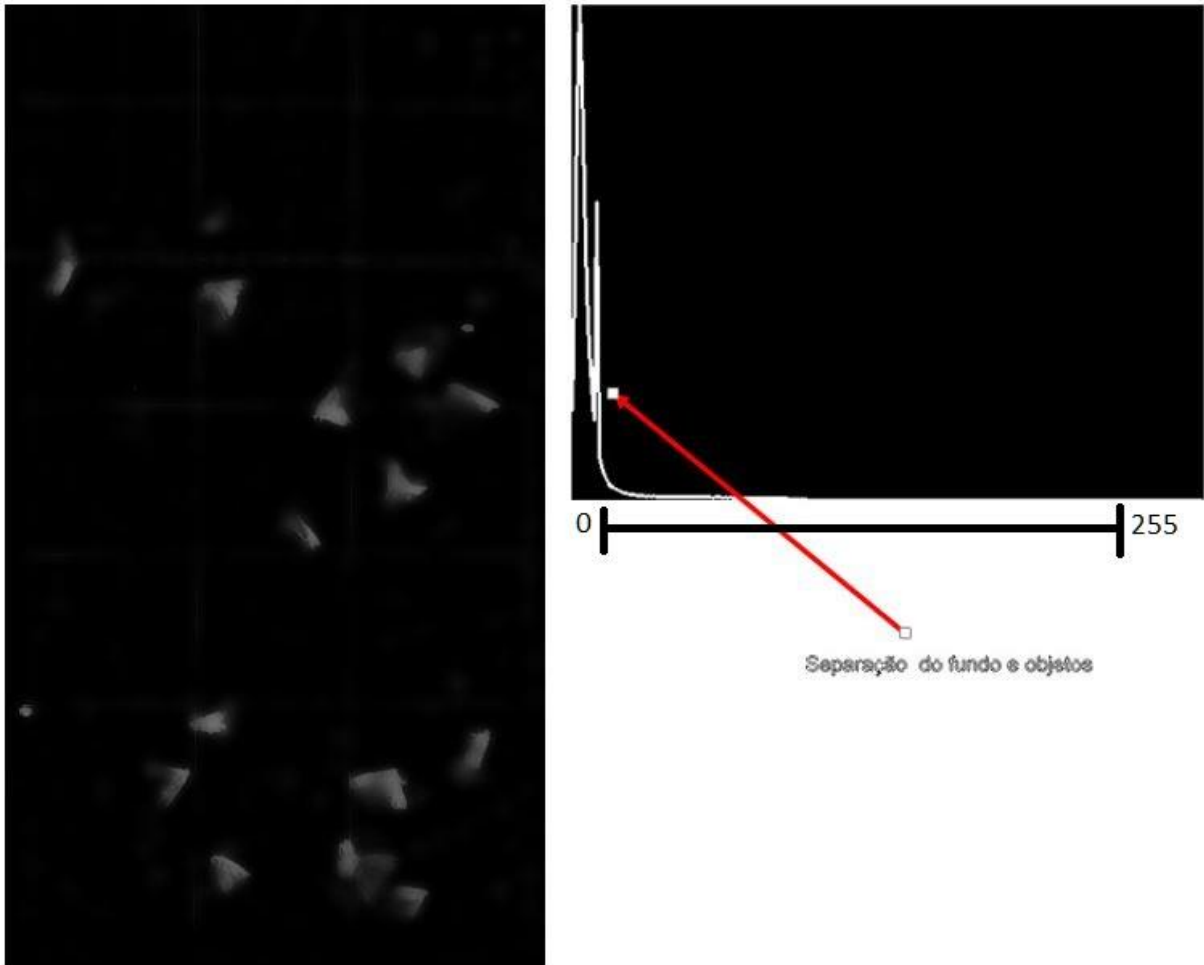


**Figura 22 – Imagem borrada e o resultado da subtração**

Fonte: do autor

#### **4.2.5 Filtragem para Remoção de Ruídos – Segunda etapa**

Após a correção de fundo tem-se uma imagem onde é possível distinguir entre objetos de interesse, no caso os insetos do fumo, e o fundo. Então se aplica um filtro para suavizar os ruídos. Nesta etapa é possível, ao visualizar o histograma, distinguir entre dois picos que correspondem à divisão dos objetos de interesse do fundo. Pode-se notar na figura 23 a imagem à esquerda o histograma mencionado e à direita a imagem após filtragem aplicada.



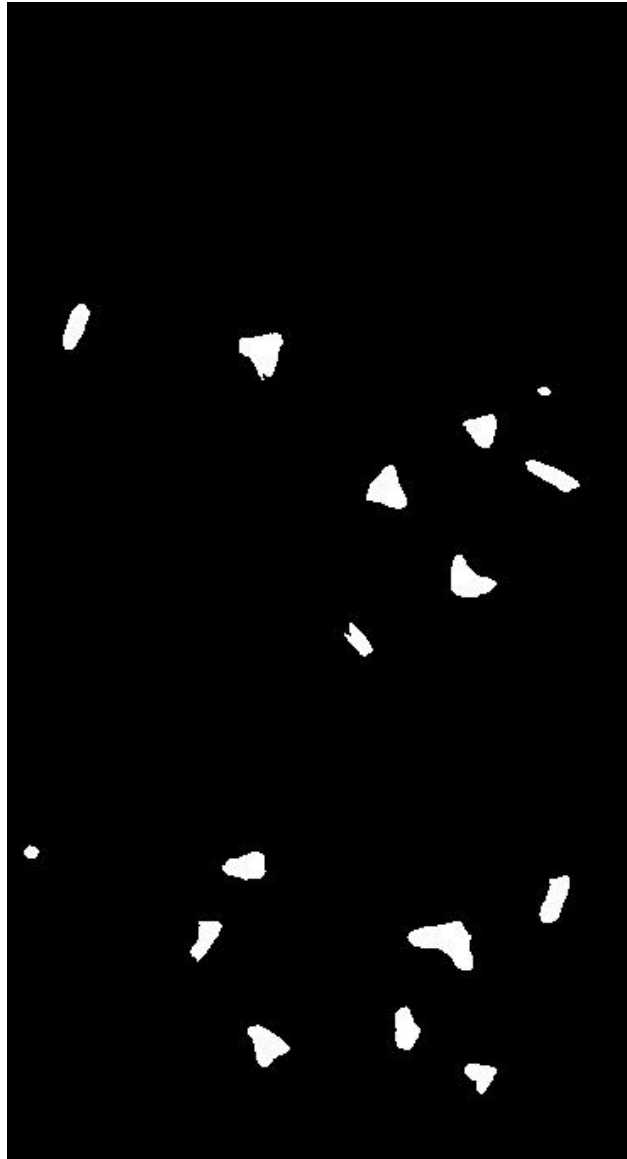
**Figura 23 – Imagem filtrada e histograma**

Fonte: do autor

#### **4.2.6 Segmentação via Limiar**

Com as técnicas de pré-processamento executadas, a imagem tornou-se apta para iniciar a fase de segmentação via limiar, também conhecida como *threshold*.

A escolha do limiar resulta da observação do histograma da figura 20, que conforme testes realizados foi escolhido 42 como parâmetro. Percebe-se a divisão da imagem com a sobressalência dos picos do histograma. Aplica-se então o valor do limiar para transformar a imagem em binária (branca e preta) conforme figura 24.



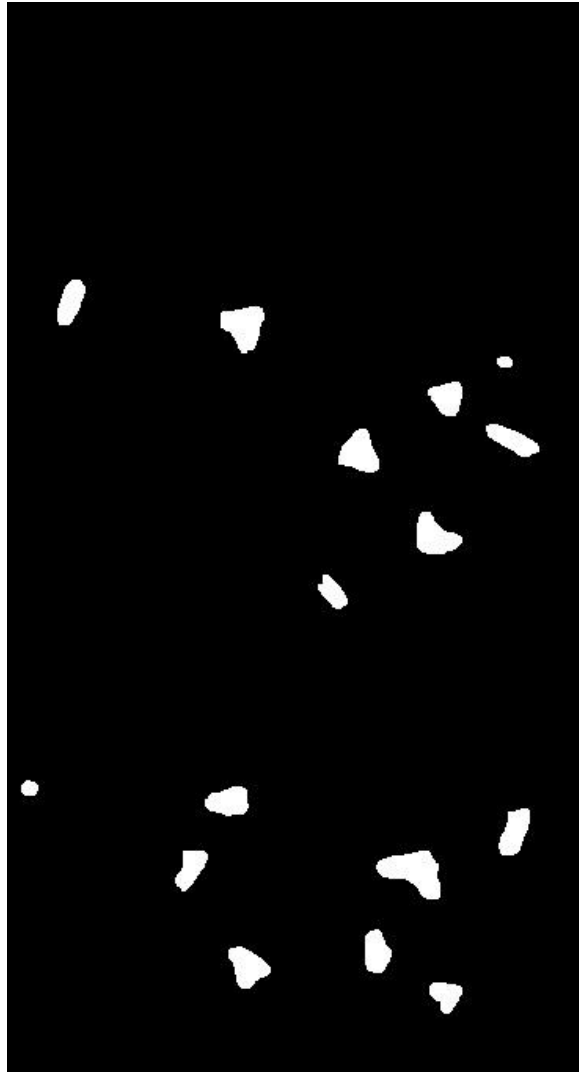
**Figura 24 – Imagem após aplicação do limiar**

Fonte: do autor

#### **4.2.7 Operação Morfológica**

Após a técnica de limiarização obteve-se uma imagem binária onde o fundo da imagem é da cor preta e os objetos na cor branca. A imagem agora possui algumas bordas irregulares e para suavizá-las foi utilizado a operação de dilatação, dada pela função *morphologyEx( imagemEntrada, imagemSaida, MORPH\_DILATE, elemento)* com um elemento estruturado da forma elíptica com tamanho 3 x 3 ( *elemento = getStructuringElement(MORPH\_ELLIPSE, Size(3,3))*), que originou melhores

resultados nos testes visuais. O efeito foi um leve preenchimento das bordas onde havia cantos pontiagudos, conforme pode ser visto na figura 25.



**Figura 25 – Imagem após dilatação**

Fonte: do autor

Em seguida foi aplicada a operação de erosão, dada pela função *morphologyEx(imagemEntrada, imagemSaida, MORPH\_ERODE, elemento)* com um elemento estruturado da forma elíptica de tamanho 2 x 2, (*elemento = getStructuringElement(MORPH\_ELLIPSE, Size(2,2))*), que originou melhores resultados nos testes e como efeito incidiu em uma diminuição dos objetos compensando o aumento gerado pela operação de dilatação, como pode ser percebido na figura 26.



**Figura 26 – Imagem após erosão**

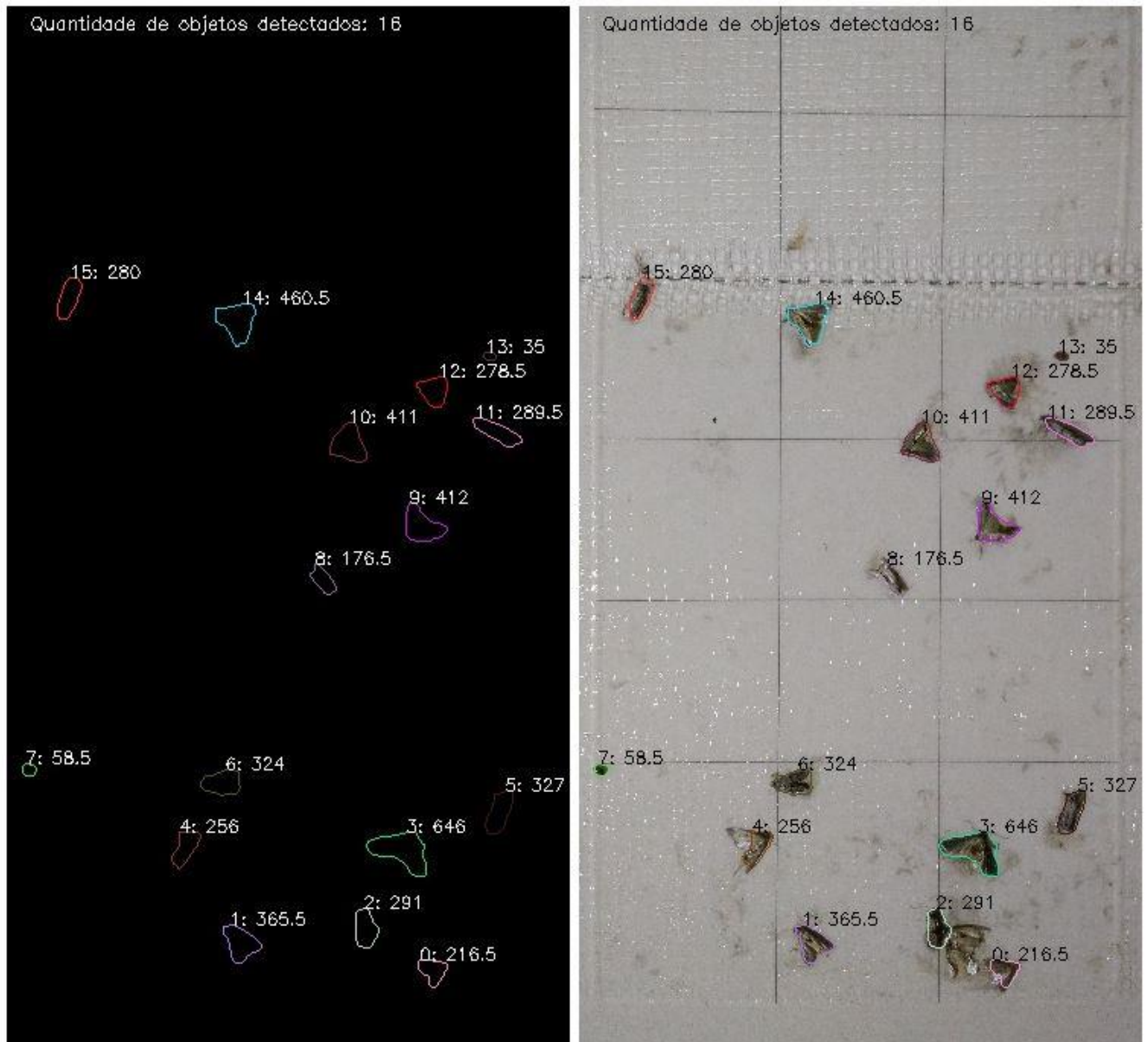
Fonte: do autor

#### **4.2.8 Representação e descrição dos contornos**

Com os objetos já definidos após as operações de dilatação e erosão, tendo agora suas curvas suavizadas dá-se o início ao processo de definição dos contornos dos objetos, a contagem da quantidade de objetos encontrados na imagem e as medidas de cada objeto. Portanto para isso foi utilizado uma função do OpenCV para criar um vetor de contorno da borda externa com o método código de cadeias. A função do OpenCV é a *findContours (imagem, contornos, CV\_RETR\_EXTERNAL, CV\_CHAIN\_APPROX\_SIMPLE)*, sendo o parâmetro imagem o resultado anterior dos operadores morfológicos, contornos é o vetor de vetores de pontos,

*CV\_RETR\_EXTERNAL* define a busca dos pontos externos mais próximos do contorno e *CV\_CHAIN\_APPROX\_SIMPLE* determina método de código de cadeia.

Com os pontos de cada contorno encontrados torna-se possível encontrar medidas relacionadas a cada objeto. Conforme pode ser visualizado na figura 27 obteve-se a identificação de cada objeto e o perímetro de cada contorno.



**Figura 27 – Representação dos contornos e a medida do perímetro**

Fonte: do autor

São registrados também medidas como área, tamanho, área do retângulo, altura, largura e raio conforme exemplificado na tabela 3. Com base nos dados



extraídos de cada objeto é possível dar início ao processo de classificação dos objetos diferenciando entre traça e bicho do fumo.

**Tabela 3 – Dados descritivos das imagens**

<i>id_obj</i>	<i>Perímetro</i>	<i>Área</i>	<i>Tam.</i>	<i>Area_retangulo</i>	<i>Altura</i>	<i>Largura</i>	<i>Raio</i>
0	216,5	62,9	22,0	380,0	19,0	20,0	11,2
1	365,5	81,8	35,0	650,0	25,0	26,0	15,1
2	291,0	70,3	26,0	432,0	27,0	16,0	13,6
3	646,0	119,3	50,0	1.131,0	29,0	39,0	22,6
4	256,0	71,9	27,0	450,0	25,0	18,0	14,1
5	327,0	80,3	37,0	522,0	29,0	18,0	16,4
6	324,0	74,3	32,0	416,0	16,0	26,0	13,7
7	58,5	28,7	12,0	90,0	9,0	10,0	4,9
8	176,5	58,2	23,0	378,0	21,0	18,0	11,5
9	412,0	88,8	28,0	676,0	26,0	26,0	16,2
10	411,0	84,8	41,0	675,0	27,0	25,0	14,8
11	289,5	83,8	50,0	627,0	19,0	33,0	18,2
12	278,5	68,9	31,0	399,0	21,0	19,0	11,8
13	35,0	22,5	8,0	54,0	6,0	9,0	4,4
14	460,5	94,5	41,0	783,0	29,0	27,0	16,3
15	280,0	73,1	35,0	364,0	28,0	13,0	15,2

\* medidas em *pixel*

Fonte: do autor

### 4.3 Validação do Trabalho

Para avaliar o desempenho e os resultados das técnicas de processamento de imagens utilizadas no trabalho, foi mapeada uma área específica dentro dos armazéns onde continham armadilhas para a realização dos testes. As imagens foram adquiridas via *smartphone* e armazenadas no computador para posterior análise. As imagens foram processadas com a utilização da biblioteca OpenCV e o resultado da contagem automática foi comparada com a contagem manual de cada armadilha.

#### 4.3.1 Coleta das imagens

Foram coletadas vinte e quatro imagens de armadilhas, sendo de doze pontos diferentes. Doze em uma semana e as demais na semana seguinte. Isso para agregar um maior número de insetos em uma única armadilha. A coleta aconteceu em janeiro de 2016 logo após o início da comercialização do tabaco e coincidindo com o período do verão, onde a incidência destes insetos é maior.

As imagens foram armazenadas no computador e receberam uma descrição conforme o ponto onde se encontrava e posterior ao hífen está a semana que foi adquirida a imagem (1, na primeira coleta e 2, coletada na semana seguinte). A lista das imagens pode ser vista na tabela 4.

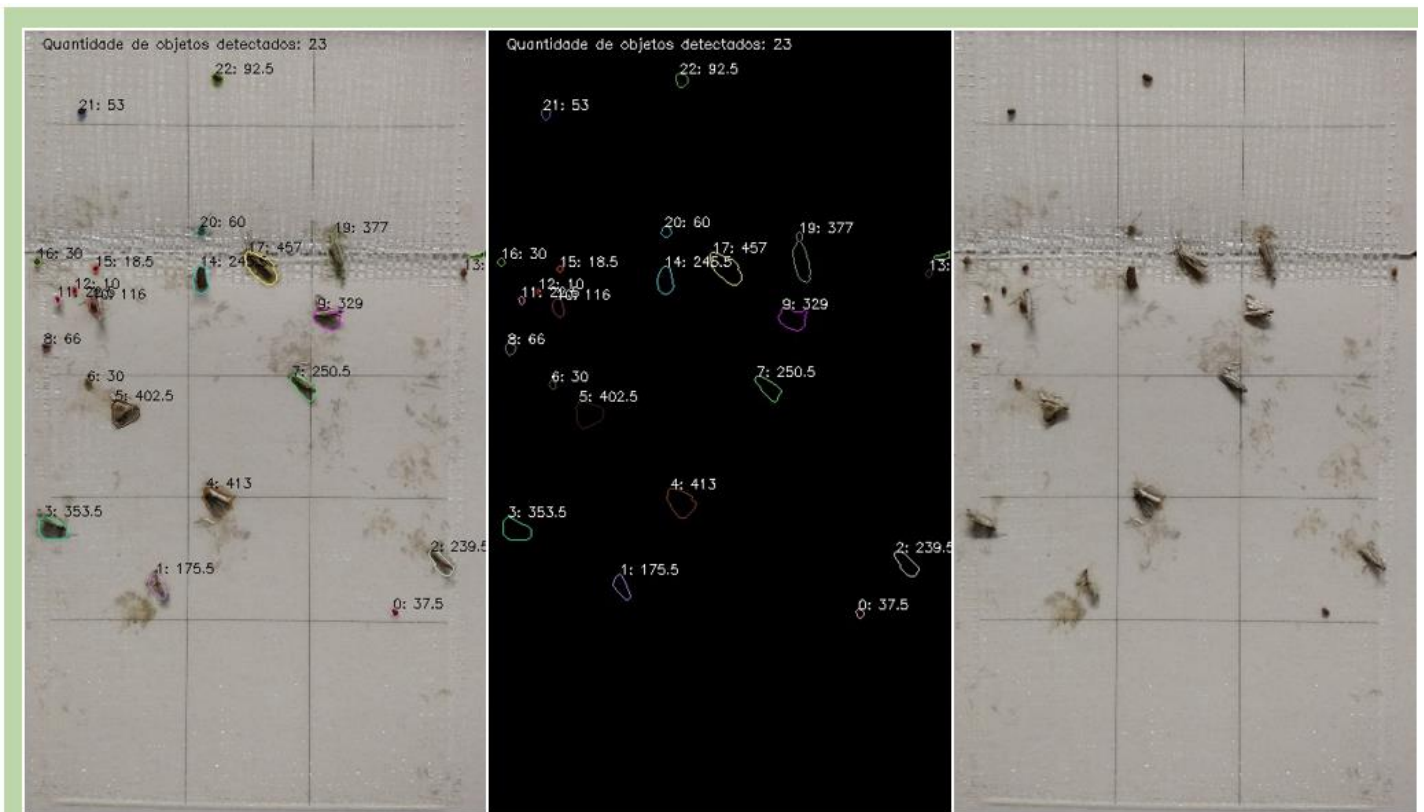
**Tabela 4 – Lista das imagens adquiridas**

Armadilha	Semana	Armadilha	Semana
255-1	1	261-1	1
255-2	2	261-2	2
256-1	1	262-1	1
256-2	2	262-2	2
257-1	1	267-1	1
257-2	2	267-2	2
258-1	1	268-1	1
258-2	2	268-2	2
259-1	1	269-1	1
259-2	2	269-2	2
260-1	1	270-1	1
260-2	2	270-2	2

Fonte: do autor

### 4.3.2 Calibração da aplicação

Após as análises individuais de cada etapa descrita nos capítulos anteriores chegou-se a resultados onde houve um maior equilíbrio de todos os parâmetros aplicados. Em seguida foi montada uma planilha para inserir os dados adquiridos de cada armadilha para uma análise mais aprofundada dos resultados visuais e descritivos (figura 28 e 29). Como pode ser visto na figura 28, tem-se à direita a imagem original, ao centro a imagem final após todo o processo e à esquerda a imagem original sobreposta com os contornos da imagem final para facilitar na apresentação e conferência dos resultados obtidos.



**Figura 28 – Imagens para análise visual**

Fonte: do autor

	A	B	C	D	E	F	G	H	I	J	K
1	Tipo Inseto	Contagem manual	Contagem automática	Variação		Cont				Dados descritivos dos objetos	
2	Traca	3	3	0		6					
3	Bicho	3	3	0							
4	Total	6	6	0							
5											
6	id_obj	perimetro	area	tam	rect_area	height	width	radius	angulo	tipo_inseto	quantidade
7	0	210	57,46	22	308	22	14	11,4	160,62	Traca	1
8	1	61	28,97	8	99	11	9	5,38	162,76	Bicho	1
9	2	308,5	74,18	29	504	18	28	14,36	77,27	Traca	1
10	3	422,5	81,84	30	621	23	27	14,1	120,96	Traca	1
11	4	31	22,14	11	27	3	9	4,25	68,86	Bicho	1
12	5	23,5	18,73	9	16	2	8	3,64	69,33	Bicho	1
13		* medidas em pixel									

**Figura 29 – Dados descritivos dos objetos**

Fonte: do autor

Conforme a obtenção e análise dos dados descritivos e visuais foi identificado que os objetos detectados poderiam ser discriminados entre bicho do fumo e a traça do fumo pelo perímetro do contorno. Conforme distância aplicada na aquisição da imagem (enquadrando a armadilha), bichos do fumo resultaram em perímetros entre 10 e 100 *pixels* e as traças do fumo entre 101 a 600 *pixels*. Porém um dos problemas quanto a traça do fumo é que em algumas armadilhas haviam duas ou até três sobrepostas, conforme pode ser visto na figura 30. Para este caso foi definido que entre 601 a 1000 *pixels* seriam contabilizadas duas traças e maior que 1000, três. Objetos menores que 10 foram classificados como N/A (não aplicável).



**Figura 30 – Traças de fumo sobrepostas**

Fonte: do autor

### 4.3.3 Primeiros resultados obtidos

Com os parâmetros definidos deram-se início a validação do trabalho. Após executar a aplicação para as vinte e quatro imagens obtiveram-se os resultados conforme a figura 31.

Armadilhas	Traça			Bicho			Total		
	Contagem Manual	Contagem Automática	Variação	Contagem Manual	Contagem Automática	Variação	Contagem Manual	Contagem Automática	Variação
	625	566	-9,44%	98	151	54,08%	723	717	-0,83%
<b>255-1</b>	11	11	0	11	12	1	22	23	1
<b>255-2</b>	33	34	1	17	16	-1	50	50	0
<b>256-1</b>	15	13	-2	1	2	1	16	15	-1
<b>256-2</b>	37	40	3	3	6	3	40	46	6
<b>257-1</b>	44	39	-5	7	8	1	51	47	-4
<b>257-2</b>	87	70	-17	8	28	20	95	98	3
<b>258-1</b>	14	14	0	5	5	0	19	19	0
<b>258-2</b>	36	32	-4	7	12	5	43	44	1
<b>259-1</b>	4	4	0	1	1	0	5	5	0
<b>259-2</b>	11	11	0	4	4	0	15	15	0
<b>260-1</b>	6	8	2	3	4	1	9	12	3
<b>260-2</b>	11	11	0	4	4	0	15	15	0
<b>261-1</b>	0	0	0	0	0	0	0	0	0
<b>261-2</b>	0	0	0	1	1	0	1	1	0
<b>262-1</b>	2	2	0	2	2	0	4	4	0
<b>262-2</b>	3	3	0	3	3	0	6	6	0
<b>267-1</b>	4	5	1	1	1	0	5	6	1
<b>267-2</b>	15	14	-1	2	2	0	17	16	-1
<b>268-1</b>	47	41	-6	0	5	5	47	46	-1
<b>268-2</b>	91	75	-16	3	8	5	94	83	-11
<b>269-1</b>	26	25	-1	4	4	0	30	29	-1
<b>269-2</b>	60	50	-10	5	9	4	65	59	-6
<b>270-1</b>	11	11	0	2	2	0	13	13	0
<b>270-2</b>	57	53	-4	4	12	8	61	65	4

**Figura 31 – Resultados obtidos na primeira execução**

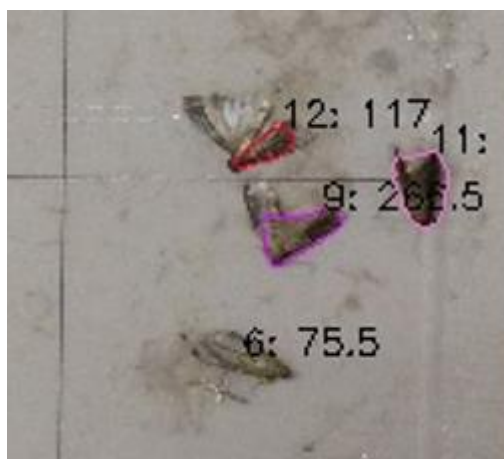
Fonte: do autor

Na figura 31 há uma tabela com a lista de armadilhas ao lado esquerdo, o próximo segmento mostra os resultados da traça do fumo comparando a contagem manual com a automática por armadilha. No segundo segmento está a relação dos bichos do fumo encontrados nas armadilhas e com suas respectivas contagens manual e automática. No último segmento encontra-se a soma das traças e do bicho por armadilha. Em cada segmento contém as variações por unidade e percentual.

Conforme pode ser visto na figura 31, a variação total e final da contagem automática das armadilhas foi de -0,83% resultando em uma diferença de -6, porém quando avaliado separadamente a traça do fumo e bicho do fumo temos uma maior

variação. Na traça do fumo foi obtido -9,44% com diferença de -59. Já para o bicho do fumo a diferença foi de 54,08% com a diferença de 53 insetos a mais na contagem automática.

Devido aos resultados obtidos, principalmente com a diferença entre a classificação do bicho do fumo e da traça, novamente foram analisadas todas as armadilhas uma a uma. Foi verificado que muitas traças foram classificadas como bicho do fumo pelo motivo da iluminação, sendo que no momento da captura da imagem em alguns casos a luz refletiu nas asas destes insetos e diminuiu o perímetro do contorno das traças, como pode ser observado na figura 32.



**Figura 32 – Diminuição do perímetro devido o reflexo da luz**

Fonte: do autor

Na figura 32 constatou-se que o objeto 6 ficou abaixo do perímetro mínimo para ser classificado como traça. Nos objetos 12 e 9 é possível visualizar que parte das asas foram descartadas no pré-processamento pelo reflexo da luz no momento da aquisição da imagem, porém ficaram com o perímetro acima do limite e consequentemente foram classificadas como traças de fumo de acordo com o planejado.

#### **4.3.4 Otimização dos resultados**

De acordo com o problema citado no capítulo anterior, agregou-se outras medidas descritivas para aperfeiçoar a segmentação dos insetos entre traça e bicho

do fumo. Sendo desenvolvido uma versão que explora outra medida descritiva do contorno, não mais apenas o perímetro.

Portanto foi incluído para cada objeto a forma circular utilizando o método de aproximação de polígonos, que diminui a distância entre as curvas do contorno. A função OpenCV é *approxPolyDP (Mat (contornos[i]), contornos\_poly[i], 3, true)*, onde *Mat (contornos[i])* é uma matriz dos contornos, *contornos\_poly[i]* é o vetor destino, 3 e *true* significam o tamanho máximo e se o os pontos do vetor polígono deve ser fechado.

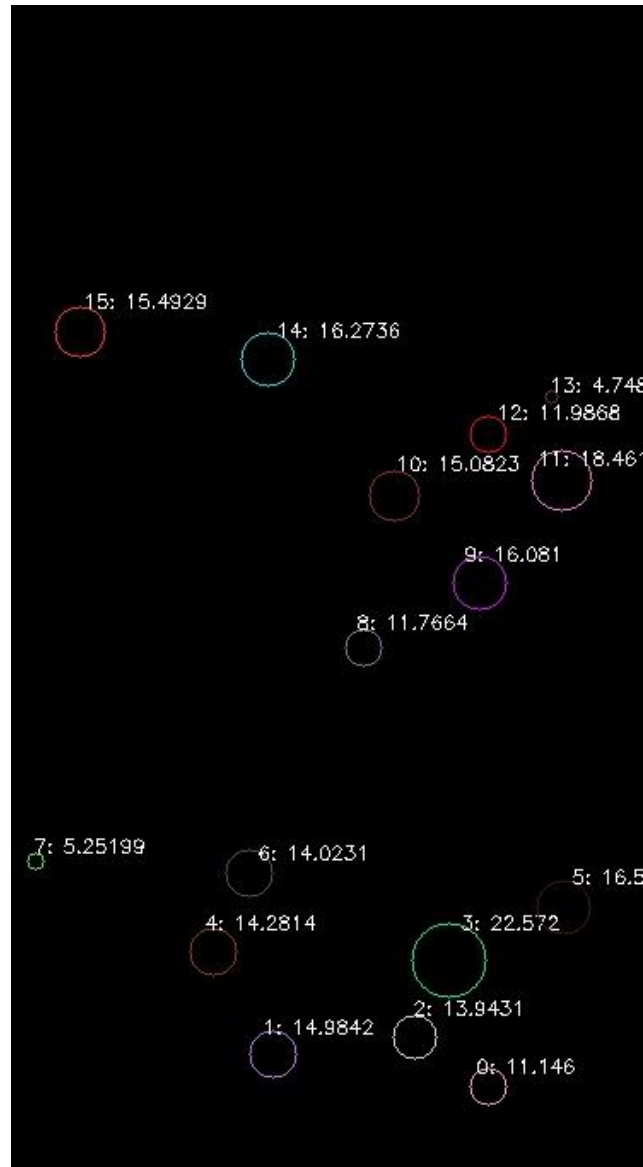
Consequente, encontrando o círculo através do vetor de pontos do polígono com a utilização da função *minEnclosingCircle ((Mat) contornos\_poly[i], centro[i], raio[i])*. Com isso foi possível encontrar o valor do raio de cada círculo e o centróide (ponto de equilíbrio central). Como todo o círculo que tem o raio conhecido, denota-se que o ponto de origem do raio é o centro do objeto. Com essa nova implementação buscou-se encontrar a cor RGB do ponto central de cada objeto.

O motivo de conhecer a cor do centro de cada objeto foi, conforme análise visual, de propor uma divisão entre bicho e traça comparando a cor. A traça do fumo tem uma tonalidade mais clara comparada com a cor do bicho do fumo. Com isto foi possível melhorar o resultado detectando as traças de fumo que ficaram abaixo do limite do perímetro do contorno, definido como menor que 100. Na figura 33 pode ser observado os novos atributos de medida incluídos na aplicação e na figura 34 a definição do círculo e medida de raio para cada objeto.

id_obj	perimetro	area	tam	rect_area	height	width	radius	angulo	R	G	B	R+G+B	tipo_inseto	quantidade	cor
0	231,5	63,36	22	420	20	21	11,15	138,68	71	59	43	173	Traca	1	
1	381,5	82,33	32	702	26	27	14,98	140,91	23	11	2	36	Traca	1	
2	304,5	72,18	29	476	28	17	13,94	173,68	18	16	7	41	Traca	1	
3	674	119,74	47	1240	31	40	22,57	110,41	93	84	70	247	Traca	2	
4	268	74,43	32	494	26	19	14,28	25,14	99	88	67	254	Traca	1	
5	343	82,77	41	600	30	20	16,58	20,53	89	87	80	256	Traca	1	
6	339,5	75,36	29	513	19	27	14,02	80,9	42	40	28	110	Traca	1	
7	62	29,8	11	110	10	11	5,25	88,74	28	18	8	54	Bicho	1	
8	184	61,25	26	418	22	19	11,77	143,68	107	100	86	293	Traca	1	
9	433,5	89,84	30	783	27	29	16,08	118,94	78	71	53	202	Traca	1	
10	421,5	85,84	42	728	28	26	15,08	148,52	81	74	55	210	Traca	1	
11	307,5	85,5	54	646	19	34	18,46	116,99	39	36	21	96	Traca	1	
12	293,5	69,36	32	440	22	20	11,99	132,01	60	59	41	160	Traca	1	
13	40	24,14	10	70	7	10	4,75	91,93	53	37	29	119	Bicho	1	
14	485,5	95,84	46	840	30	28	16,27	167,31	16	10	0	26	Traca	1	
15	290,5	74,18	36	464	29	16	15,49	21,58	108	106	93	307	Traca	1	

**Figura 33 – Novos atributos descritivos de cada objeto**

Fonte: do autor



**Figura 34 – Imagem de objetos com a forma circular**

Fonte: do autor

Para a realização do novo teste, foi definido de forma a otimizar o resultado a inclusão das cores RGB. Sendo que a soma de cada canal (R, G e B) devesse ficar abaixo de 151 para o inseto ser classificado como bicho do fumo. Então ficaram definidas duas condições: a primeira condição é a do perímetro do contorno e a segunda é a cor do ponto central do objeto.



#### 4.3.5 Resultados obtidos após otimização

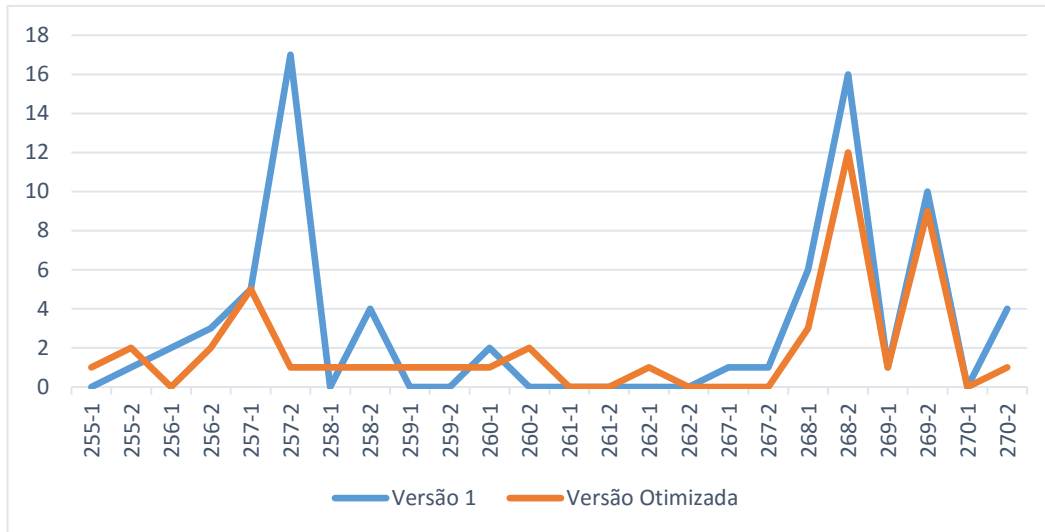
Conforme nova definição das condições citadas na seção anterior, para classificar entre bicho do fumo e traça, foi novamente executada a aplicação para todas as armadilhas e o resultado pode ser visto na figura 34.

Armadilhas	Traça			Bicho			Total		
	Contagem Manual	Contagem Automática	Variação	Contagem Manual	Contagem Automática	Variação	Contagem Manual	Contagem Automática	Variação
	625	608	-2,72%	97	90	-7,22%	722	698	-3,32%
			-17			-7			-24
255-1	11	12	1	11	11	0	22	23	1
255-2	33	35	2	16	12	-4	49	47	-2
256-1	15	15	0	1	0	-1	16	15	-1
256-2	37	39	2	3	3	0	40	42	2
257-1	44	39	-5	7	6	-1	51	45	-6
257-2	87	88	1	8	9	1	95	97	2
258-1	14	15	1	5	4	-1	19	19	0
258-2	36	35	-1	7	7	0	43	42	-1
259-1	4	5	1	1	1	0	5	6	1
259-2	11	12	1	4	3	-1	15	15	0
260-1	6	7	1	3	3	0	9	10	1
260-2	11	13	2	4	3	-1	15	16	1
261-1	0	0	0	0	0	0	0	0	0
261-2	0	0	0	1	1	0	1	1	0
262-1	2	3	1	2	1	-1	4	4	0
262-2	3	3	0	3	3	0	6	6	0
267-1	4	4	0	1	1	0	5	5	0
267-2	15	15	0	2	2	0	17	17	0
268-1	47	44	-3	0	2	2	47	46	-1
268-2	91	79	-12	3	6	3	94	85	-9
269-1	26	25	-1	4	3	-1	30	28	-2
269-2	60	51	-9	5	5	0	65	56	-9
270-1	11	11	0	2	1	-1	13	12	-1
270-2	57	58	1	4	3	-1	61	61	0

**Figura 35 – Novos resultados após otimização**

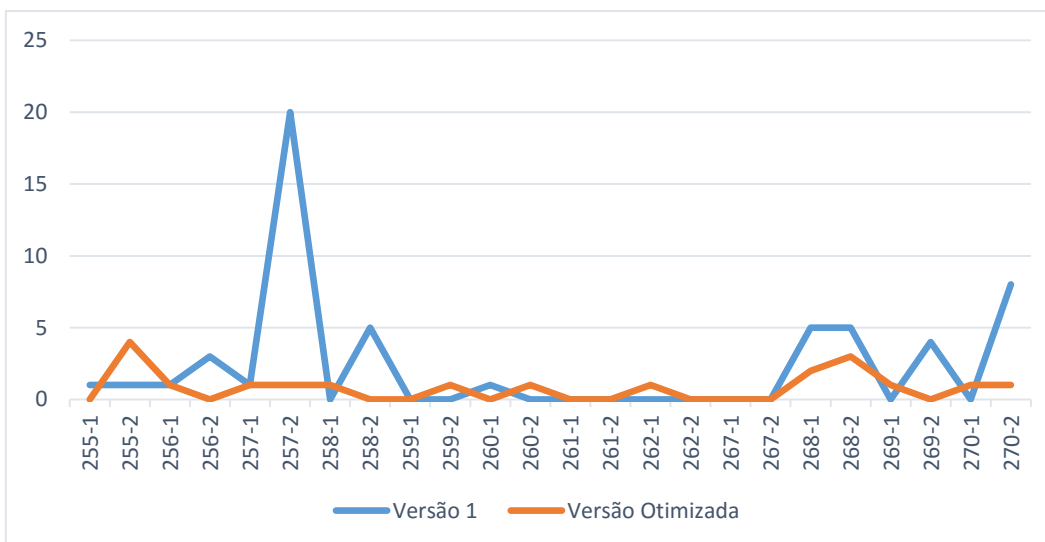
Fonte: do autor

Neste novo resultado, conforme pode ser visto na figura 34, pode-se considerar que a variação total e final da contagem automática das armadilhas teve um leve aumento, chegando a -3,32% resultando em uma diferença de -24, porém quando avaliado separadamente a traça do fumo e bicho do fumo a variação diminui consideravelmente. Na mosca do fumo foi obtido -2,72% com diferença de -17. Já para o bicho do fumo a diferença foi de -7,22% com a diferença de -7 insetos na contagem automática. Pode-se visualizar nas figuras 36 e 37 os dados em forma de gráficos, com as análises das variações por armadilha entre os resultados da versão 1 e da versão otimizada da mosca do fumo e bicho do fumo respectivamente.



**Figura 36 – Análise variação da mosca do fumo**

Fonte: do autor



**Figura 37 – Análise variação do bicho do fumo**

Fonte: do autor

Tais resultados foram avaliados novamente armadilha por armadilha e foi constatado que algumas traças de fumo estavam sobrepostas, porém não atingindo o valor mínimo para ser contabilizado como duas traças (ficando abaixo de 600 *pixels*).

Após a realização da otimização para melhorar a classificação dos insetos, é notável a diminuição do erro na distinção por tipo de inseto, mesmo havendo um considerável aumento na variação total de ambos insetos tendo a aplicação um desempenho expressivo para contagem automatizada de insetos.

## 5 CONCLUSÃO

Com base nos estudos sobre Processamento de Imagens, foi possível desenvolver uma aplicação para o alcance dos objetivos propostos. Para tal, foi utilizado a biblioteca OpenCV para auxiliar no desenvolvimento das etapas de Processamento de Imagens desde o pré-processamento até a análise descritiva dos dados que possibilitou a classificação entre a traça e o bicho do fumo. A proposta inicial era de apenas contabilizar as traças do fumo, porém com o aprofundamento do trabalho foi possível segmentar e classificar entre traça e bicho do fumo.

O processo de contagem de insetos é realizado de forma manual, portanto, podendo causar erros devido a fadiga visual como também por fatores psicológicos dos quais os seres humanos estão à mercê. Entretanto, a aplicação desenvolvida realiza uma contagem padrão e automatizada, seguindo sempre o mesmo procedimento, evitando assim que quaisquer distúrbios influenciem no resultado final.

Outro ponto importante é que os dados são escritos em papéis para depois ser digitada em planilha eletrônica e posteriormente, incluídas em um sistema ERP próprio, portanto, com a aplicação é possível extrair as informações e disponibilizá-las em um arquivo *xml*, que possibilita a importação direta da informação para o sistema.

Quanto aos resultados obtidos pela contagem automática, pode-se concluir que os valores, conforme discutidos com os responsáveis do monitoramento de insetos, são aceitáveis. Apesar de não se ter uma base histórica de erros da contagem manual que permitisse a comparação entre os resultados.

Um dos fatores principais dos erros ocorridos foi a iluminação não uniforme nos ambientes onde foram coletadas as imagens. Este fator tornou-se um obstáculo para a obtenção de um resultado ainda mais próximo da realidade. Portanto uma sugestão, inclusive para trabalhos futuros, é a construção de um protótipo que pode ter o formato de uma caixa e que serviria como um *scanner*. Tal protótipo controlaria o ambiente corrigindo a iluminação, ângulo e distância na coleta das imagens.

Com o desenvolvimento da solução proposta permite-se mais eficiência, agilidade e assertividade no processo de contagem de insetos, com a possibilidade de um tempo de resposta mais rápido na investigação das causas e na decisão das ações que devem ser tomadas para o controle dos focos de insetos.

## REFERÊNCIAS

AZEVEDO, Eduardo; CONCI, Aura; LETA, Fabiana. *Computação Gráfica: Teoria e Prática*. Rio de Janeiro: Elsevier, 2008.

BARBEDO, Jayme Garcia Arnal. *Automatic Method for Counting and Measuring Whiteflies in Soybean Leaves Using Digital Image Processing*. Embrapa Agricultural Informatics, Campinas, SP, Brazil, 2014.

BAXES, Gregory A. *Digital Image Processing: Principals and Applications*. New York: John Wiley & Sons, Inc., 1994.

BEUREN, Daniel. *Avaliar metodologias para a redução do índice de infestação de insetos em depósitos de Produto Acabado*. Trabalho de Graduação II (Curso de Engenharia de Produção) – Universidade de Santa Cruz do Sul - UNISC, Santa Cruz do Sul, 2013.

BRADSKI, Gary. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, California: O'Reilly Media, 2008.

CORESTA: *Cooperation Centre for Scientific Research Relative to Tobacco*. Disponível em: [http://www.coresta.org/In\\_the\\_Press/TR0614-p54-58\\_BeetleBattle.pdf](http://www.coresta.org/In_the_Press/TR0614-p54-58_BeetleBattle.pdf)

ERTÜRK, Sarp. *Digital Image Processing*. University of Kocaeli: National Instruments Corporation, 2003.

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

GOMES, Jonas; VELHO, Luiz. *Computação Gráfica: Imagem*. Rio de Janeiro: IMPA, 2002.

GOMES, Jonas; VELHO, Luiz. *Fundamentos da Computação Gráfica*. Rio de Janeiro: IMPA, 2003.

GONZALEZ, Rafael C; WOODS, Richard E. *Digital Image Processing*. Prentice Hall, 2002.

GONZALEZ, Rafael C. *Processamento de Imagens Digitais*. São Paulo: Edgar Blücher, 2000.

HUDDAR, R; GOWRI, S. *Novel Algorithm for Segmentation and Automatic Identification of Pests on Plants using Image Processing*. Third International Conference Computing Communication & Networking Technologies (ICCCNT), 26-28 July 2012.

MANOJA, M; RAJALAKSHMI, J. *Early Detection of Pests on Leaves Using Support Vector Machine*. International Journal of Electrical and Electronics Research, Vol. 2, Issue 4, pp: (187-194), December 2014.

MARENGONI, Mauricio; DENISE, Stringhini. *Tutorial: Introdução à Visão Computacional usando OpenCV*. Revista de Informática Teórica e Aplicada, v. XVI, p. 1-36, 2009.

KESAVARAJU, Banugopan; DICKSON, Sammie. *New Technique to Count Mosquito Adults: Using ImageJ Software to Estimate Number of Mosquito Adults in a Trap*. Journal of the American Mosquito Control Association, 28(4):330-333, 2012.

KUMAR, Tarun; VERMA, Karun. *A Theory Based on Conversion of RGB image to Gray image*. International Journal of Computer Applications, Volume 7– No.2, September 2010.

PEDRINI, Hélio; SCHWARTZ, William Robson. *Análise de Imagens Digitais: princípios, algoritmos e aplicações*. São Paulo: Thomson Learning, 2008.

RUSS, John C. *The Image Processing Handbook* - 2nd ed. CRC Press, 1995.