INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO AMAZONAS – IFAM.

CAMPUS MANAUS DISTRITO INDUSTRIAL

CURSO DE BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

WENDEL DA COSTA PRADO

DESENVOLVIMENTO DE UM SISTEMA DE INSPEÇÃO DE COMPONENTES UTILIZANDO TÉCNICAS DE VISÃO COMPUTACIONAL

MANAUS-AM

2022

WENDEL DA COSTA PRADO

DESENVOLVIMENTO DE UM SISTEMA DE INSPEÇÃO DE COMPONENTES UTILIZANDO TÉCNICAS DE VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus - Distrito Industrial, Curso de Bacharelado em Engenharia de Controle e Automação, como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Ailton Gonçalves Reis

MANAUS-AM

2022

Dados Internacionais de Catalogação na Publicação (CIP)

P896d Prado, Wendel da Costa.

Desenvolvimento de um sistema de inspeção de componentes utilizando técnicas de visão computacional. / Wendel da Costa Prado. – Manaus, 2022.
63f. : il. color

TCC (Graduação em engenharia de controle e automação) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, 2022.
Orientador: Prof. Dr. Ailton Gonçalves Reis

1. Visão computacional. 2. Processamento de imagem. 3. Sistema de inspeção automático. I. Reis, Ailton Gonçalves (orient.) II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 629.8

Elabora por Fcª. Amélia Frota, registro n.858 (CRB11)

POR

WENDEL DA COSTA PRADO

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 29 de Abril de 2022 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Controle e Automação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

(Prof.Dr. Ailton Gonçalves Reis)

Orientador

(Prof.Msc. Alberto Luis Fernandes Queiroga)

Avaliador

(Prof.Dr. Alysson Jesus dos Santos)

Avaliador

RESUMO

Este trabalho de conclusão de curso (TCC) tem como objetivo apresentar um sistema que através da Visão Computacional, possa identificar a presença de componentes eletrônicos em placas eletrônicas, sem a intervenção humana, e tomar decisões acionando um painel que exibe o erro localizado. Assim, desenvolver um sistema que possa captar dados como, perímetro, área da peça, posição, cor, utilizando somente um sensor, e a partir dessas características o sistema ser capaz de mostrar se está faltando um componente específico ou se está apresentando algum tipo de falha, como: ausência de um terminal ou deslocamento.

PALAVRAS CHAVES: Visão computacional; Processamento de Imagens; Sistema de Inspeção Automático.

ABSTRACT

This course conclusion paper (CCP) aims to present a system that, through Computer Vision, can identify the presence of electronic components on electronic boards, without human intervention, and make decisions by triggering a panel that displays the localized error. Thus, to develop a system that can capture data such as perimeter, part area, position, color, using only one sensor, and from these characteristics the system will be able to show if a specific component is missing or if it is showing some type of failure, such as: absence of a terminal or displacement.

KEYWORDS: Computer Vision; Image Processing; Automatic Inspection System.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE SIGLAS

BR: Brasil;

PT: Português.

HSV: matiz (Hue), saturação, (Saturation), valor (Value)

TRACKBAR: Barra de Rolagem

RGB: Vermelho (Red), verde (Green), Azul (Blue)

OPENCV: *Open Source Computer Vision Library*

SOP*:* Small outline package

LED: Light Emitting Diode

ROI: Region of Interest

OS: Operate System

SMD: Surface Mounted Device.

PCB: Placa de Circuito Impresso.

SUMÁRIO

1 INTRODUÇÃO

A visão computacional é uma forma de inteligência artificial na qual podemos Programar computadores para enxergar o ambiente e torná-los capazes de extrair informações, através de imagens capturadas se torna possível reconhecer e manipular os dados. Assim a visão artificial busca a implementação de sistemas que possam enxergar o mundo através de processos artificiais.

A visão computacional vem evoluindo bastante ao longo do tempo, o setor de controle de qualidade das indústrias é uma das áreas mais beneficiadas desse avanço tecnológico. Dentro das fabricas existem diversos processos que exigem inspeção e essas geralmente são executados por pessoas, mas esse processo pode ter um controle mais preciso e eficaz com o auxilio da visão computacional.

Um sistema de visão industrial pode ser essencial neste processo e tornar-se a principal ferramenta de gestão de qualidade da indústria. Câmeras podem não só inspecionar peças de acordo com um padrão de qualidade pré-determinado, como monitorar cada etapa do processo, acompanhando cada peça durante os estágios de produção.

Este Trabalho de Conclusão de Curso (TCC) delimitou-se em colher informações sobre de que forma a aplicação de um sistema de visão computacional pode auxiliar a detecção de componentes eletrônicos inseridos de forma não conforme. Resultando em melhor controle de qualidade no setor produtivo das indústrias.

Produções em larga escala fazem uso de linhas produtivas, que utilizam de sistemas automatizados para lidar com produtos que precisam de confiabilidade e complexidade. Inspeções visuais fazem parte dos processos de controle de qualidade, porém agregam custo e tempo à produção, além de serem afetados pela experiência e fatiga do inspetor responsável.

Assim encontrar defeitos em placas eletrônicas é uma das grandes preocupações das montadoras, por isso existe uma busca constante para evitar que esses erros aconteçam, mas ainda podem acontecer. Por isso, um método de verificação eficiente poderia identificá-los no inicio evitando que estes sejam replicados.

Nesse contexto apresenta-se a seguinte pergunta problema: “O uso de um sistema de visão computacional pode ser usado como ferramenta para mitigar e/ou solucionar problemas de controle de qualidade dentro do processo produtivo?”.

Nossa teoria é que a dificuldade para detectar componentes eletrônicos inseridos de forma errada. Pode ser resolvido coma aplicação de um sistema de visão computacional.

Assim, o objetivo principal desse trabalho é: “Utilizar um sistema de visão computacional inserido no processo produtivo de forma que seja capaz de fazer o controle de qualidade de circuitos eletrônicos, perceber mudanças de componentes, identificar componentes e localizá-los na placa.”

Desse objetivo geral decorrem quatro outros específicos, quais sejam:

a) Adquirir por meio de leitura de artigos e livros os conhecimentos das bibliotecas de visão computacional, assim como da linguagem *python*, para tornar possível desenvolver um algoritmo de extração de características físicas.

b) Desenvolver um algoritmo que seja capaz de ser suportado por um computador de baixo custo.

c) Desenvolver um sistema para detecção e localização de falhas em montagem de *PCB’s;*

d) Desenvolver um algoritmo para pré-processar as imagens de entrada e segmentar, detectar e reconhecer um determinado objeto, sendo este de cores e formas variadas.

A metodologia respeita as características da pesquisa aplicada e de campo, onde a coleta de dados foi realizada dentro da indústria, assim como alguns experimentos com Hardware. Assim então foi possível aplicar os conhecimentos de visão computacional para o desenvolvimento desse sistema.

Os referenciais teóricos seguem as ideias de autores que tratam da temática de visão computacional plicada, Tais como: Budiharto (2014) e Barelli (2018) respeitando suas ideias para aplica-las no ambiente industrial.

Os resultados mostram que esse sistema pode ser implantando para ajudar no controle de qualidade dos componentes inseridos em *PCB’s* de modo que apesar

de algumas limitações, consegue perceber de forma efetiva a ausência ou defeitos nos componentes principais.

Esperamos que esse trabalho possa contribuir para o desenvolvimento de pesquisas aplicadas com o uso de visão computacional, assim então, ajudar a desenvolver sistemas cada vez amais robustos e capazes de ajudar no ambiente industrial.

Desta maneira, cabe apresentar como este TCC encontra-se formatado. O primeiro capítulo apresenta essa própria introdução ao trabalho, abordando as várias características do projeto e seus objetivos.

O segundo capítulo mostra o referencial teórico que foi utilizado no desenvolvimento dessa pesquisa, abordando a base para um sistema de inspeção utilizando visão computacional.

O terceiro capítulo preocupa-se em mostrar a metodologia, e como esse trabalho foi elaborado.

O quarto capítulo mostra os principais resultados, e a evolução do sistema a partir de testes genéricos. Por fim, temos as considerações finais que vai relatar os principais resultados dos objetivos propostos e as melhorias futuras.

2. REFERENCIAIS TEÓRICOS

Neste capítulo serão apresentados conceitos dos principais tópicos para o desenvolvimento do trabalho, abordando principalmente conceitos à visão computacional e processamento de imagens.

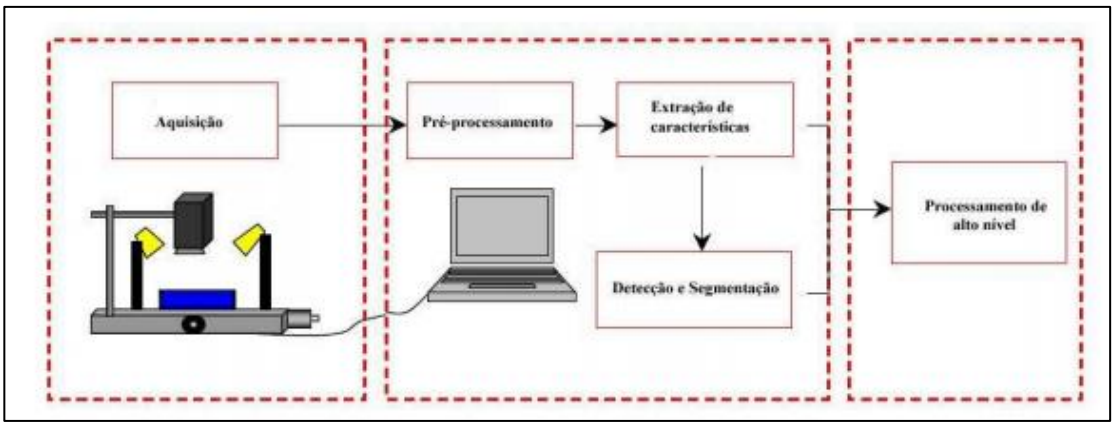
Esta seção abordará principalmente os conceitos relacionados ao processamento de imagens e visão computacional.

2.1 SISTEMAS DE INSPEÇÃO AUTOMÁTICA

Segundo Baldner et al. (2017) os sistemas de visão computacional são utilizados a partir de várias tecnologias, podendo ser utilizado em diferentes áreas, nas indústrias é utilizado principalmente no chão de fabrica, a fim de tornar processos mais rápidos e simples, retirando o fator visual humano. Isso é uma vantagem se considerar que um ser humano está sujeito a ficar sobrecarregado e cansado, ao contrario de um computador que pode ser utilizado continuamente, e analisar com um grande número de detalhes de forma paralela.

A figura 1 mostra a arquitetura simplificada a execução de um sistema de visão aplicado dentro de uma indústria. Neste exemplo, o processo de navegação é seguindo o fluxo da linha, onde existe um dispositivo para aquisição da imagem, que responsável por adquirir uma imagem da peça, essa imagem posteriormente é transmitida para o sistema de processamento, onde Software desenvolvido para análise e identificação de defeitos na peça. Se então esta estiver com alguma não conformidade, o sistema de processamento usa sinais digitais para controlar o sistema. Assim pode ser aplicado algum dispositivo atuador para remover as peças que estão com defeito, ou gerar uma notificação para o administrador do sistema.

Figura 1: Sistema de inspeção



Fonte: Damaitre (2018)

2.2 PROCESSAMENTOS DIGITAL DE IMAGENS

O processamento de imagens é uma etapa muito importante em uma aplicação de visão computacional, pois nela estamos trabalhando para que nossa base de dados possa estar adequada para uma analise posterior da nossa aplicação, alguns exemplos são aplicações de filtros para redução de ruídos, balanceamento de cores, giro e corte (SZELISK, 2010).

2.2.1 Imagens digitais

Segundo Pazos (2002) é a representação de uma imagem bidimensional usando números binários, em que cada elemento dessa matriz corresponde a um pixel, cada pixel contem informação correspondente na imagem captada pela câmera, assim estes podem ser codificados de modo a permitir seu armazenamento, transferência, impressão ou reprodução.

A etapa de processamento da imagem é realizada traduzindo os dados contidos na imagem em números binários, que podem ser usados para armazenar e transferir as informações. Cada pixel na imagem possui informações correspondentes, que podem ser codificadas para permitir que os dados sejam transferidos ou reproduzidos (PAZOS, 2002).

Figura 2: Imagem digital.

Fonte: Autoria Própria (2020)

2.2.2 Histograma

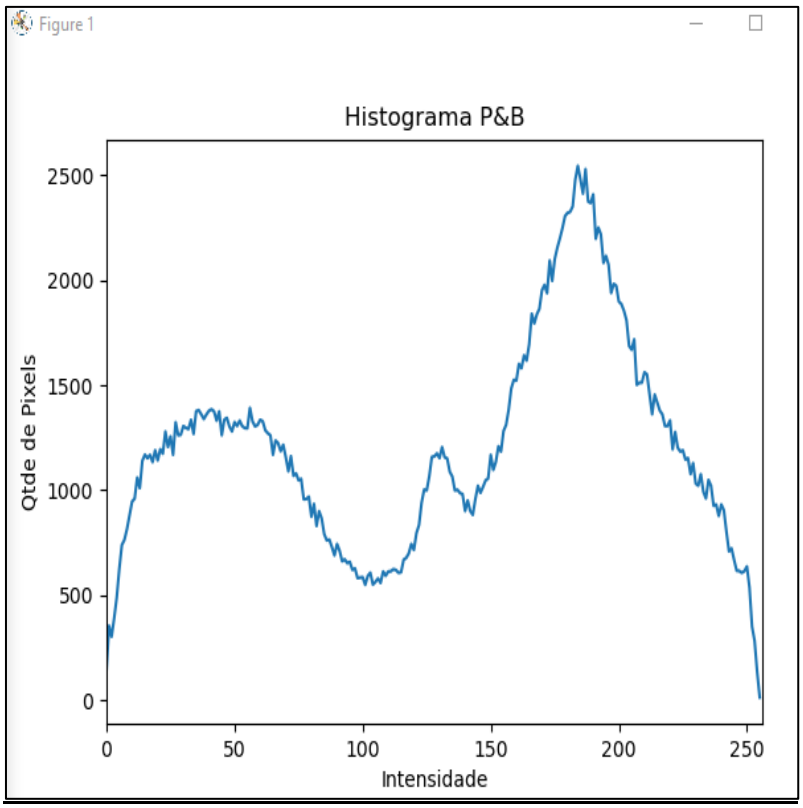
Um histograma é uma representação gráfica das cores que estão distribuídas em uma imagem, neste gráfico resultante é possível verificar quais são as cores mais presentes na imagem. Assim o histograma vai representar um gráfico que mostra a relação de intensidade e quantidade de pixels presente, ou seja, histograma em uma imagem digital pode medir para cada tom de cor quantos pixels ocorrem naquele tom. Na imagem, sendo a intensidade variando de 0 a 255 (BALDNER, 2017).

Figura 3: Imagem em escala de cinza.

Fonte: Autoria Própria (2021)

O histograma gerado nos permite ter uma visão geral de como a intensidade esta variando, sua visualização pode ser vista seja em escala de cores, escala de preto e branco, sendo na primeira, o histograma fica dividido em três componentes correspondentes para cada cor primaria, *RGB.* Na imagem na figura abaixo é gerado o histograma da figura 3, através de um código em *python* utilizando a função *cv2.calcHist :*

Gráfico 1: Histograma da figura 3



Fonte: Autoria própria (2020)

2.2.3 Modelos de cores

De acordo com Baldner et al.(2017) O desenvolvimento de modelos de cores foi realizado devido a importância de registrar numericamente uma cor especifica, assim ela poderia ser descrita numericamente. Dessa maneira, um sistema para solucionar e descrever, os aspectos perceptivos da visão humanadas cores deve apresentar três características básicas: matiz, luminosidade e croma.

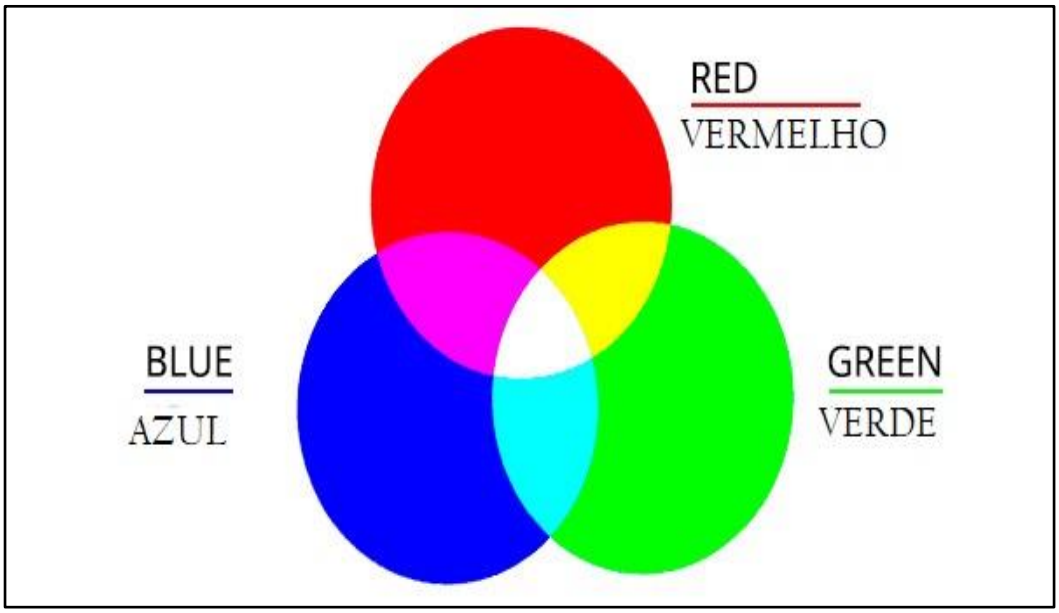
Um espaço de cor é uma maneira quantitativa de representar uma cor, de modo que as cores não possam ser definidas de acordo com um referencial humano afim de não gerar respostas interpretativas, assim existem diversos espaços de cores e todos eles buscam padronizar um modelo matemático que formaliza uma cor através de uma sequência numérica específica (FORSYTH; PONCE, 2011).

2.2.4 Espaços de cores rgb

De acordo com Zanotta et al. (2019) conceitua como um modelo de representação de cores que faz uso das cores primárias, onde uma cor é formada pela combinação de várias cores primárias como: vermelho, verde e azul. Assim essas correspondem ser combinadas de diversas maneiras para gerar um leque extensivo de cores variadas.

O uso do sistema de cores RGB *(Red, Green, Blue)* é comumente utilizado em equipamentos eletrônicos como, televisores e monitores. Esse modelo utiliza um modelo matemático que representa as cores de forma mais semelhante à visão humana.

Figura 4: Espaço de cor RGB



Fonte: Queiroz (2018)

2.2.5 Espaços de cores hsv

É o resultado de uma transformação não linear do modelo RGB, sendo este novo sistema de cores definido através de três parâmetros: matiz (*Hue),* saturação*, (Saturation),* valor *(Value). E*sses parâmetros resultam em um sistema denominado

sistema de cores HSV, o qual é mais intuitivo que o sistema RGB, uma vez que por meio dele se torna mais fácil a escolha de uma determinada cor (BARRELI, 2018).

Para Zanotta et al.(2019),os três principais parâmetros do sistema HSV são:

> **Matiz (tonalidade):** examina a principal cor presente no leque de cores do domínio *HSV.*
>
> **Saturação (pureza):** também chamada de pureza da cor. Quanto menor for este valor, mais misturada a cor será com as outras, puxando para tons mais acinzentados. Cores saturadas são conhecidas popularmente como cores viva, enquanto cores pouco saturadas são chamadas de tons pastéis.
>
> **Intensidade (luminosidade):** Define o brilho da cor. Quanto menor for o brilho, mais escura será a cor. Essa componente pode ser pensada como a quantidade de luz iluminando o objeto colorido. (ZANOTTA et al.2019,p.55).

A variação do matiz pode nos dar uma informação importante, já que representa a tonalidade da cor ou podemos dizer se a cor está genuína ou tem alguma alteração. Na figura 5 é mostrada uma imagem de frutas, que não tem qualquer alteração em sua composição.

Figura 5: Imagem original

Fonte: Barelli (2018).

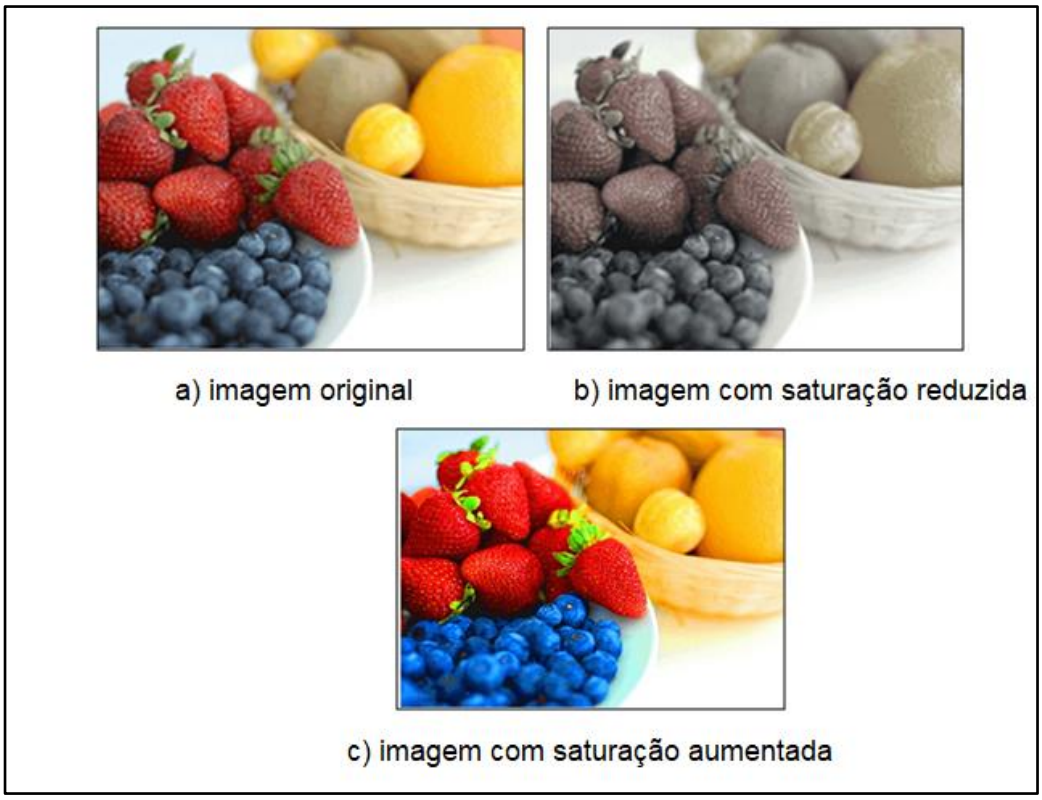
Porém, a figura 6 tem sua componente de matiz alterada, no caso seu matiz teve uma redução de 80% em relação à imagem original, o que resulta em uma imagem com variações quando comparadas a sua imagem base.

Figura 6: Imagem com alteração de matiz

Fonte: Barelli (2018).

Saturação que pode ser dita como a intensidade da cor, e está diretamente proporcional a sua pureza, quanto maior a saturação, mais pura é a cor, e assim essa imagem com uma saturação alta apresentará cores fortes, e quanto menor seu valor aproxima uma imagem à tonalidade cinzenta. A figura 7 exemplifica a variação da saturação de modo aumentado e reduzido.

Figura 7: Variação de saturação em uma imagem



Fonte: Barelli (2018).

2.2.6 Limiarização

Thresholding ou limiarização é um processo que envolve a conversão de certo número de pixels em uma imagem de cor binária. Assim um valor de limiar é definido, para que um intervalo de pixels se torne branco se mantendo na imagem, enquanto os valores abaixo de limiar se tornam escuros (ARTERO; TOMASSELLI).

Na figura 8 é apresentada uma imagem de sementes de café e uma xicara, sobre uma superfície branca, assim foi feito uma binarização da imagem, usando um valor de limiar igual a 100. E é possível visualizar um grande contraste na imagem resultante, pois os pixels com intensidade de cor mais elevadas foram mantidos, e os que possuíam intensidade menor ao limiar foram transformados em pretos, como pode ser vistos na figura 9.

Figura 8: Grãos de café

Fonte: Autoria Própria (2020)

Figura 9: Grãos após limiarização

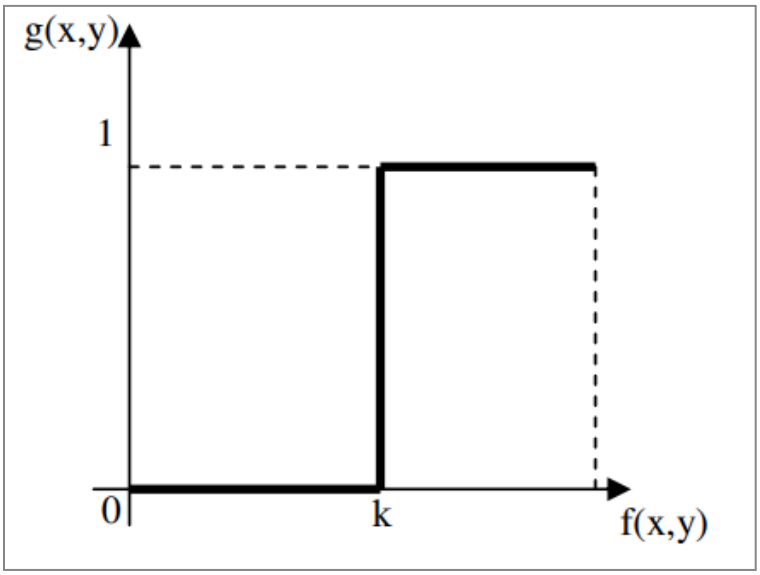
Fonte: Autoria Própria (2020)

Uma maneira de representar o corte utilizado é através da equação 1:

$$g(x,y) = \begin{cases} 1, & se\ f(x,y) \geq T \\ 0, & caso\ contrario \end{cases} \quad (1)$$

Observando a função é possível notar que o corte de binarização e feito de forma brusca, assim os valores acima de T são mapeados para o valor máximo um, e os abaixo do valor T recebemos valor zero.

Figura 10: função de binarização por um valor K



Fonte: Marengoni; Stringhini, (2009).

2.2.7 Operações morfológicas

Nesse item apresentaremos conceitos das principais operações morfológicas utilizadas nesse trabalho, começando pela própria definição da secção.

Em estudo feito por Barelli (2019) a respeito do processamento de imagens as operações morfológicas são comumente usadas para modificar a estrutura ou formato do conteúdo da imagem. Essas operações podem ser aplicadas em diversos tipos de imagens, e geralmente são aplicadas quando a imagem está em escala de cinza, é bastante comum o seu uso antes de uma extração de bordas.

O uso dessas operações pode ajudar a realçar determinadas características em imagens, a figura 11 mostra uma imagem de uma placa de carro, seu formato original e sua forma binarizada. É possível observar que sua binarização não foi ideal e apresentas algumas falhas em alguns pixels.

Figura 11: representação binaria não ideal

Fonte: Barelli (2018)

Dependendo do sistema utilizado essas não continuidades entre os pixels poderiam gerar erros, e o não reconhecimento de algum caractere. Assim a figura 11 foi sujeita ao um tratamento morfológico com o objetivo de retirar essas não continuidades entre os caracteres, assim é possível notar que após o tratamento os caracteres estão mais nítidos e com menos imperfeiçoes assim podem ser considerados como regiões isoladas.

Figura 12: Representação binária após tratamento

Fonte: Barelli (2018)

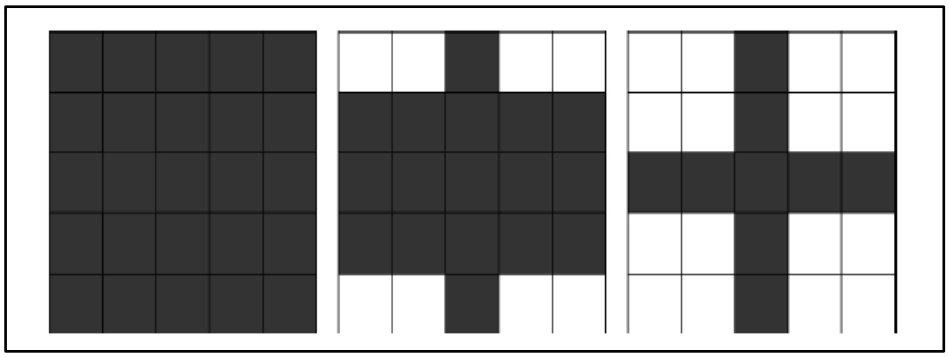
2.2.8 Elemento estruturante

É um elemento que pode ser descrito como uma imagem binária, mas de formato menor que a original, geralmente é utilizado dentro de uma matriz quadrada e é muito comum quando usado em operações morfológicas (BARELLI, 2019).

Os elementos serão como máscaras nas quais aplicamos as operações lógicas. Portanto, este elemento é uma matriz com valores num intervalo específico, que vai atuar na vizinhança do pixel para tomar uma decisão (HEISE; SALUSTIANO, 2020).

A figura 13 mostra os três tipos de matrizes de elementos estruturantes mais utilizados. Na seguinte ordem, retangular, elipse e cruz.

Figura 13: Tipos de elementos estruturantes



Fonte: Barelli (2018)

2.2.9 Erosão e dilatação

Erosão e dilatação são duas principais operações básicas quando estamos trabalhando com o processamento de imagens digitais, e esses operadores morfológicos são usados como base e referência para a criação de funções mais robustas (HEISE; SALUSTIANO, 2020).

De acordo com GONZALES e WOODS para trabalhar com essas operações é preciso conhecer dois operandos, uma imagem binária e um elemento chamado elemento estruturante, quando aplicado o operador morfológico, o resultado será uma imagem binaria modificada.

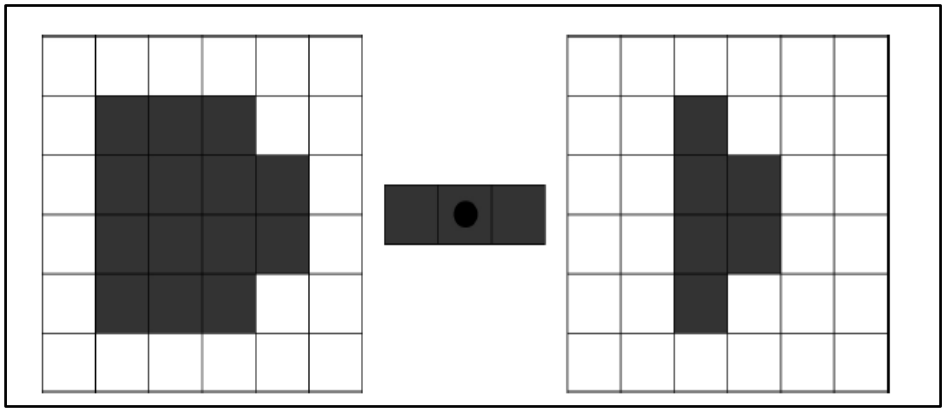
2.2.10 Operação de erosão

A operação de erosão é pode ser resumida como uma corrosão da imagem, onde há uma perca de pixels nas áreas positivas, ou seja, esta operação remove detalhes finos (BARELLI, 2019).

A figura 14 apresenta uma imagem binária, seu elemento estruturante no centro, e o resultando da operação de erosão à direita. O resultando apresentado foi gerando através de um elemento estruturante retangular.

Assim o elemento estruturante vai percorrer toda a imagem pixel a pixel de modo a sobrepor o pixel central do elemento estruturante com cada pixel da imagem. Se o exato formato do elemento estruturante não der positivo com os elementos ao redor do pixel selecionado, este pixel selecionado vai ser retirado da imagem resultante.

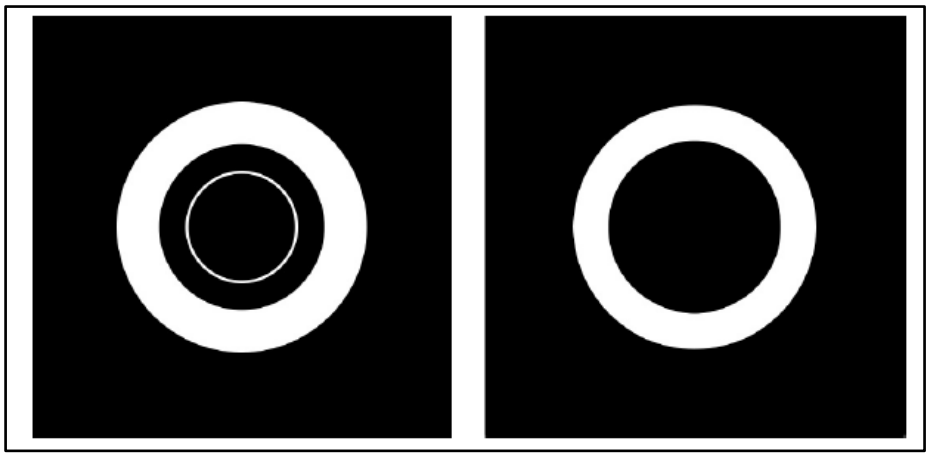
Figura 14: Aplicação do elemento estruturante



Fonte: Barelli (2018)

O resultado de operação de erosão pode ser visualizado na figura 15, este foi gerado utilizando a função *erode* da *Opencv*.

Figura 15: imagem após o processo de erosão



Fonte: Barelli (2018)

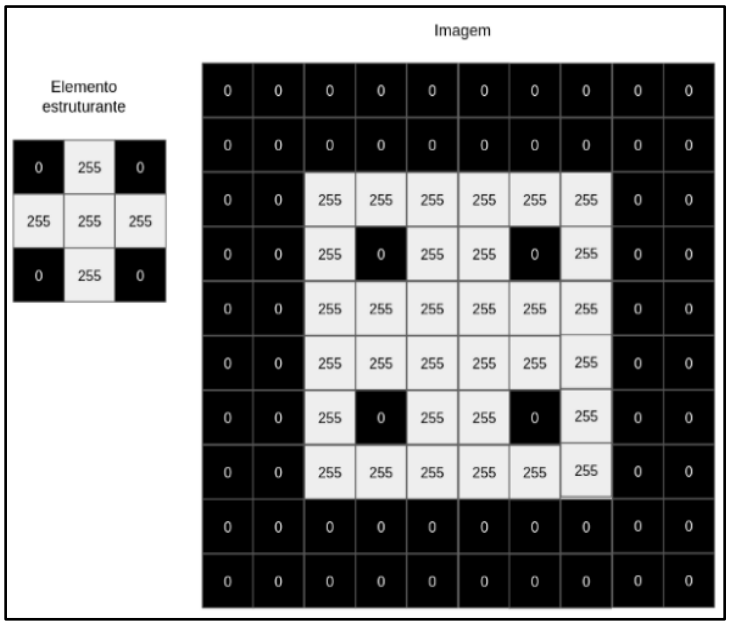
2.2.11 Operação de dilatação

A operação de dilatação segue o sentido contrario da operação de erosão, ou seja, a imagem resultante tende a ser maior que a original, pois acrescentará pixels positivos a imagem final (BARELLI, 2019).

A dilatação de imagens é uma operação que utiliza um operador OU, ou seja, se um dos parâmetros do elemento estruturante for positivo, logo o elemento resultante será positivo (HEISE; SALUSTIANO, 2020).

A figura 16 mostra um elemento estruturante retangular e ao seu lado a imagem onde este será aplicado.

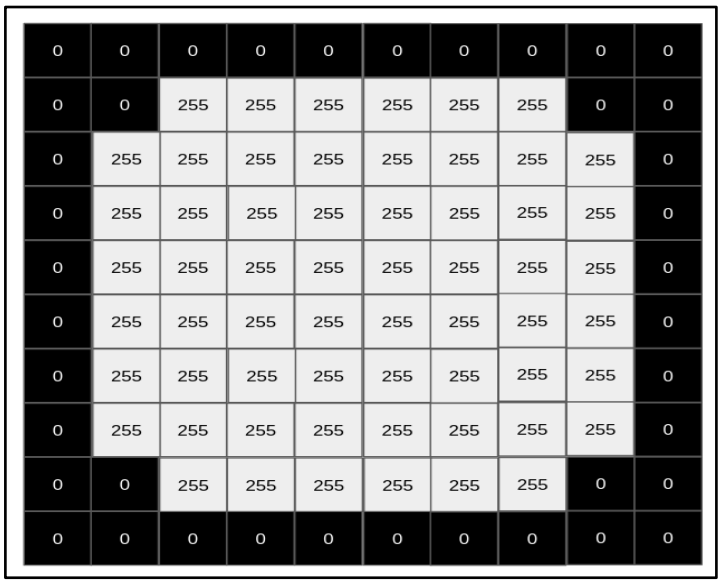
Figura 16: Elemento estruturante e imagem alvo.



Fonte: Heise; Salustiano (2020).

Como é possível perceber a parte de pixels positivos da imagem (com valor 255) foi aumentada isso porque elemento estruturante de dilatação foi efetuado.

Figura 17: Imagem após a operação de dilatação



Fonte: Heise; Salustiano (2020).

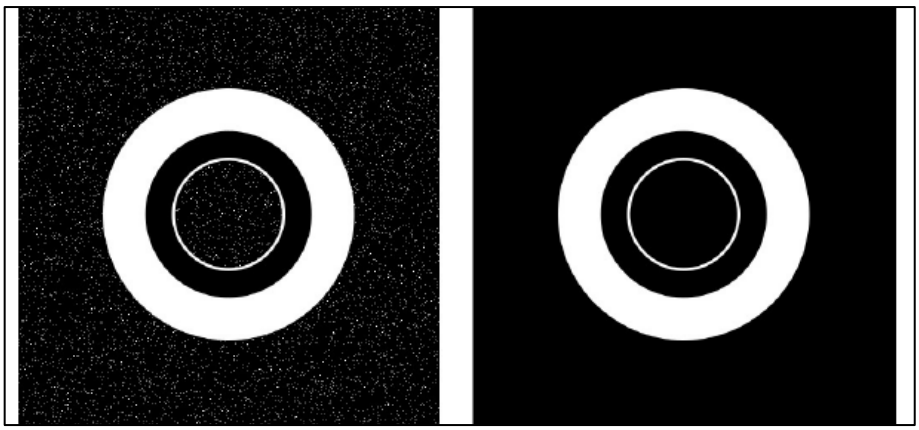
2.2.12 Operação de abertura

Podemos definir a operação de abertura como uma operação de erosão, mas seguida de uma operação de dilatação, esse tipo de tratamento é bastante comum para o tratamento de ruído em imagens (BARELLI, 2019).

O *OpenCV* disponibiliza uma função nativa que possibilita a execução de uma operação de abertura de modo fácil e simples, esta função é chamada *morphologyEX*. Para sua aplicação é preciso de três parâmetros: A imagem desejada, tipo de operação e o elemento estruturante.

A figura 18 exemplifica o que é esperado após a aplicação de uma operação de abertura, removendo o ruído da imagem.

Figura 18: Remoção do ruído após operação de abertura

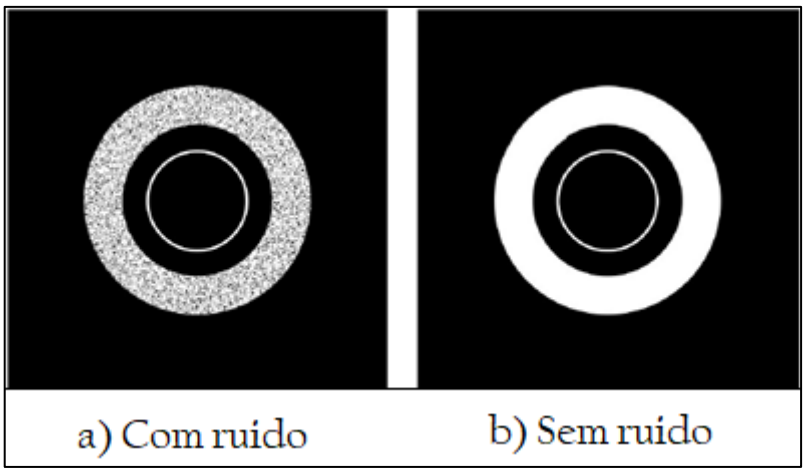


Fonte: Barelli (2018)

2.2.13 Operação de fechamento

A operação de fechamento segue a sequência inversa da operação de abertura, ou seja, é efetuada primeira a operação de dilatação e em seguida a de erosão. Esta operação é muito comum para tratamento de imagens com pequenas falhas (BARELLI, 2018).

Figura 19: Remoção de falhas após a operação de fechamento



Fonte: Barelli (2018)

É possível observar que na figura 19 que após o tratamento a imagem resultante passa a apresentar maior nitidez e sem ruído.

2.3 VISÃO COMPUTACIONAL

Para Barelli (2018), a visão computacional enquanto Ciência, pode ser entendida como aquela que estuda e desenvolve tecnologias que permitem que máquinas sejam capazes de extrair dados do ambiente ao redor, através de imagens capturadas por sensores de aquisição.

A visão computacional é a ciência que busca emular a visão humana no computador, ou seja, a partir da coleta imagens de uma câmera ou outro sistema de aquisição, o computador possa analisar os dados dessas imagens e obter características que possam auxiliá-lo a tomar uma decisão. Isso é tentar reproduzir da melhor maneira possível à visão humana (BUDIHARTO; GROUP, 2014).

Consoante a Feltrin (2022) a visão computacional é uma tecnologia que permite que um computador veja o mundo ao seu redor através de uma imagem, buscando colher informações como a visão humana. Ele usa as informações coletadas por vários dispositivos eletrônicos para fornecer um melhor entendimento do que está acontecendo no ambiente. Sendo possível o reconhecimento de formas e padrões sobre as imagens.

Segundo Marengoni e Stringhini (2009) o objetivo da visão computacional é aproximar-se da realidade da visão humana. Assim seus primeiros estudos começaram por volta dos anos cinquenta, ainda que não tivessem o poder de processamento de imagens dos dias de hoje, foi a primeira tecnologia a ser combinada com a inteligência artificial.

Inicialmente, acreditava-se que as máquinas teriam um melhor senso de visão do que os humanos. No entanto, devido aos avanços da tecnologia, o grau de complexidade envolvido no desenvolvimento de tal sistema foi muito maior.

Muitos estudiosos tentam entender o funcionamento do córtex visual do cérebro, onde são processadas as imagens captadas pelo olho. Esta é parte de grande complexidade para o cérebro humano. Eles analisam o funcionamento para por em prática novas ideias com o objetivo de evoluir gradativamente a visão computacional até o ponto em que não será possível diferenciar a visão humana com a visão da máquina, de tão perfeita que será sua imagem (MILANO; HONORATO, 2010).

Na visão computacional, o objetivo do sistema é descrever o mundo em que vivemos reconstruindo propriedades como forma, tamanho e cores. Embora humanos e animais possam fazer isso, os algoritmos são muito propensos a erros. Assim, várias novas tecnologias surgiram na área de visão computacional, com o intuito de resolver problemas cada vez mais específicos e sendo aplicados em vários campos do conhecimento, como: astronomia, medicina e multimídia (SZELISK, 2010).

Na contemporaneidade muitas tecnologias novas na área de visão computacional e processamento de imagens estão surgindo, sendo utilizadas em importantes áreas da ciência, como: aeronáutica, astronomia, medicina, multimídia, entretenimento, robótica, sistemas produtivos, entre muitas outras (RUDEK, COELHO, CANCIGLIERI; 2001).

Por meio das imagens e vídeos que coleta, a visão computacional pode fornecer um suprimento infinito de informações, o que possibilita que as máquinas executem tarefas mais inteligentes (SZELISK, 2010).

Um sistema de visão é composto por vários componentes, como a câmera, o mecanismo de processamento e o sistema de comunicação. Esses componentes são necessários para que um sistema de visão de máquina funcione corretamente e produza resultados consistentes.

Jähne (2000) definiu uma estrutura geral dos principais componentes de um sistema de visão computacional, tais como:

Câmera: A aquisição de uma imagem é o primeiro passo na sequência de processamento do sistema de visão de máquina. Sem uma imagem, o sistema não pode executar a função pretendida. Para obter uma representação precisa de uma cena, o hardware da câmera pega as informações de luz da cena e as converte em dados digitais.

Iluminação: O sistema de iluminação também é considerado o componente mais importante de um sistema de visão de máquina. Deve fornecer iluminação uniforme em todos os objetos na cena

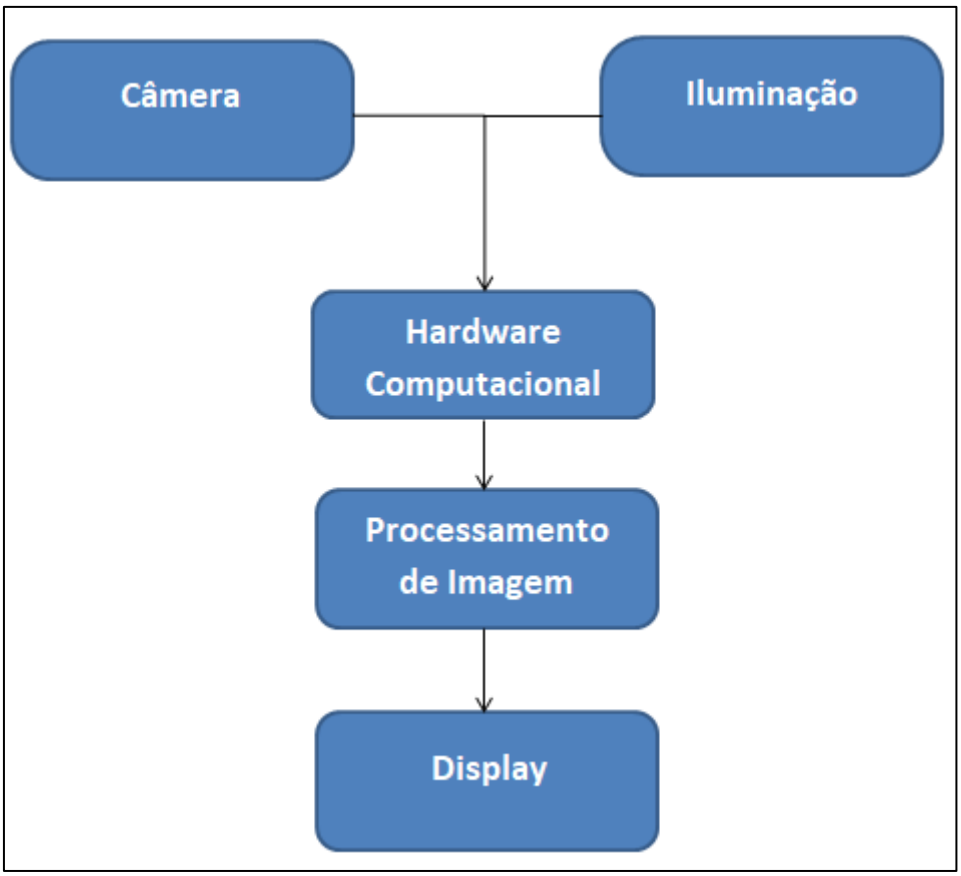
Hardware computacional: Os vários módulos de software de um sistema de visão de máquina são normalmente construídos em hardware. Eles podem ser feitos facilmente em sistemas embarcados ou com computadores dedicados. Normalmente, as câmeras são equipadas com hardware integrado para minimizar as restrições de espaço.

Processamento de imagem: O software que roda em um computador pega a imagem capturada e aplica o mesmo algoritmo a ela. O algoritmo de extração de conhecimento resultante é responsável por extrair os detalhes das imagens.

Display: Monitores ou outros dispositivos de saída de vídeo para exibição das informações geradas pelo software de processamento de imagens.

Esses componentes principais indicados por Jähne (2000) podem ser constatados na figura 20.

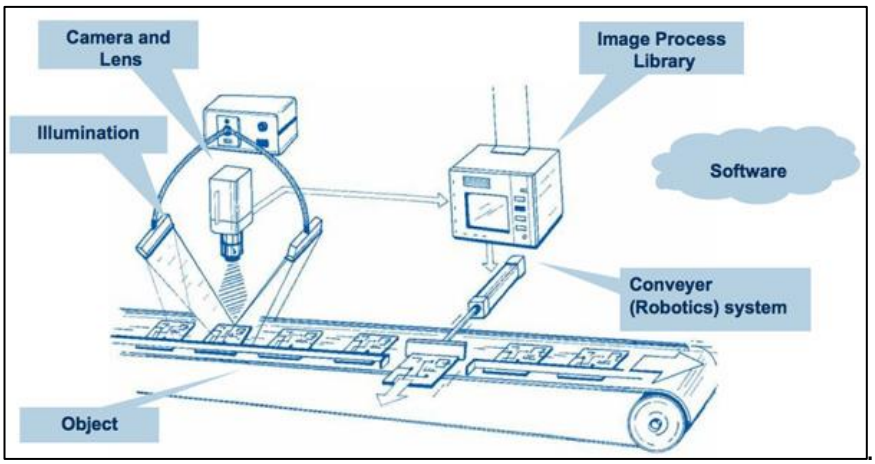
Figura 20: Componentes de um sistema de visão computacional.



Fonte: Autoria Própria (2021)

Com base no que Jähne (2000) definiu sobre os componentes de um sistema de visão computacional, pode-se observar na Figura 21 um exemplo de aplicação destes componentes, em um processo de reconhecimento e interpretação de rótulos de latas de refrigerante que passam em uma esteira rolante.

Figura 21: Sistema de visão artificial na indústria.



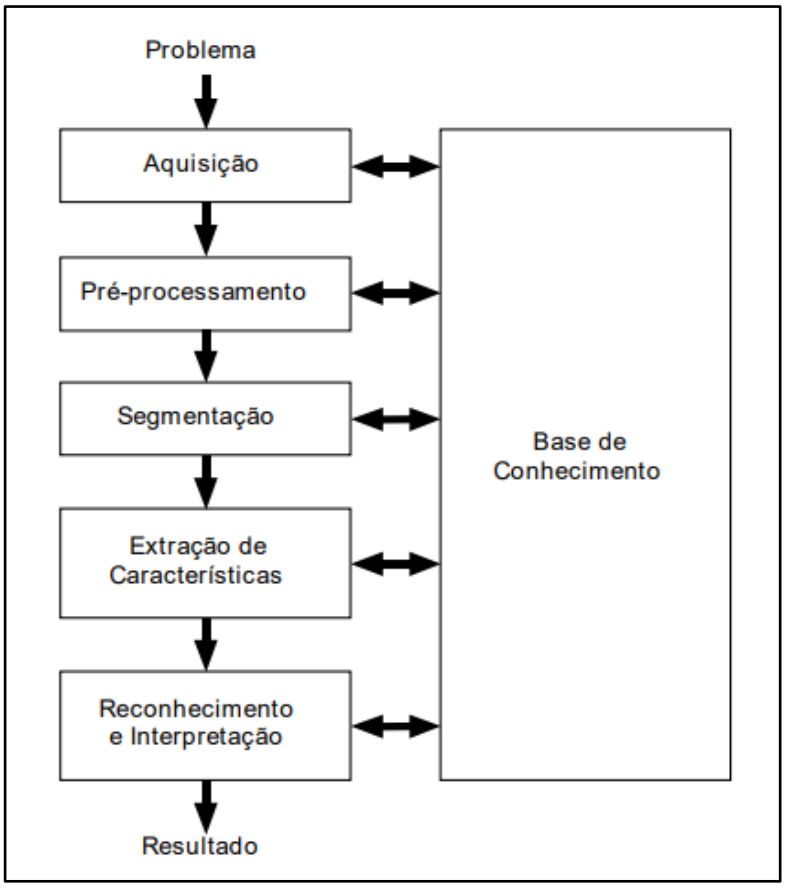
Fonte: Damaitre (2018)

2.3.1 Principais etapas de um sistema de visão computacional

O objetivo da visão computacional é reproduzir a visão humana que os humanos através de um computador, analisando os dados coletados pelas câmeras. Em seguida, realiza o processamento e a interpretação da imagem como saída.

Iremos definir um sistema de visão computacional (SVC) como um sistema computadorizado que possibilite adquirir, processar e interpretar imagens. A figura 22, a partir de um diagrama de blocos exemplifica de um modo simples as principais etapas de um sistema de visão computacional, que serão estudadas e aplicadas nesta pesquisa.

Figura 22: Etapas de um sistema de visão computacional



Fonte: Marques Filho; Vieira Neto (1999).

2.3.2 Aquisições de imagens

A aquisição de imagens é a primeira etapa de um fluxo de sistema baseado em visão artificial, nela são utilizados sensores capazes de captar informações do ambiente e armazená-las em formato digital, ou seja, em um formato manipulável como uma matriz (GONZALES; WOODS, 2009).

Nesta etapa é de fundamental importância se preocupar com algumas variáveis, como: a escolha do sensor a serem utilizadas, as condições de iluminação do local onde o sistema vai atuar a velocidade com que as imagens vão ser capturadas (MARQUES FILHO; VIEIRA NETO, 1999).

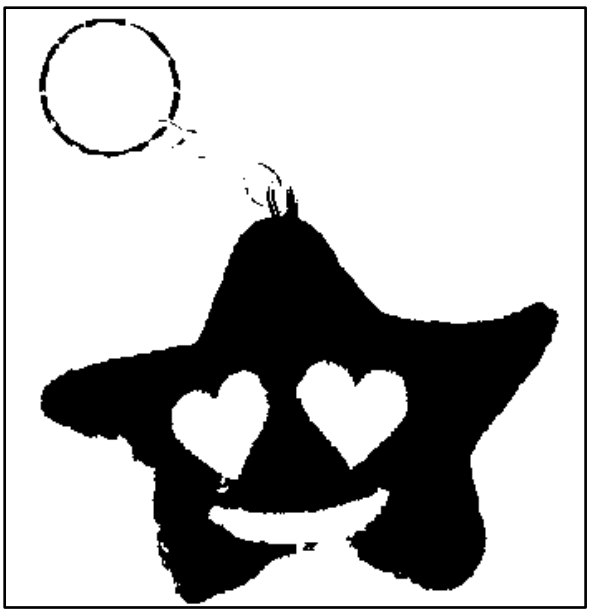
2.3.3 Pré-processamento

No estudo e aplicação das técnicas de visão computacional, muitas vezes podem precisar de uma fase de pré-processamento. Essa fase é essencial para facilitar a extração de características posteriormente, ou seja, tratam de conversões de formato e tamanho e aplicações de filtros para remoção dos ruídos que podem surgir no processo de aquisição (MARENGONI; STRINGHINI, 2009).

Para Barelli (2018), ele é responsável por destacar regiões de interesses nas imagens, isso significa usar técnicas para realças informações importantes, como: bordas e formas geométricas, eliminando o ruído e visando facilitar a extração de características desejadas.

Esse pré-processamento é muito importante, pois na etapa de aquisição de imagens pode apresentar ruídos e imperfeições, como por exemplo: Presença de ruído nos pixels, contraste e brilho inadequado. Assim a etapa de pré-processamento é melhorar a condição da imagem para as etapas futuras (MARQUES FILHO; VIEIRA NETO, 1999).

Figura 23: Imagem após o processo de binarização.



Fonte: Autoria Própria (2020).

2.3.4 Segmentação

A segmentação é responsável por dividir a imagem em unidades, ou seja, destacar as regiões de interesse desejadas. Técnicas de processamento de imagens como a detecção de bordas, afinamento, operadora morfológica e detecção de regiões, podem ser utilizadas nesta etapa. Algoritmos específicos para identificação de formas (linhas, curvas, e outras formas que podem ser parametrizadas) em imagens, cores ou intensidades também são utilizados para conduzir o processo de extração dos objetos de interesse da imagem (BARELLI, 2018).

De acordo com Bradski e Kaehler (2008) a função *Canny Edge Detector* é uma função amplamente utilizada desenvolvida por John F. Canny em 1986, com o objetivo de detectar as bordas de uma imagem. Também é chamado de detector ideal por muitos. O algoritmo é projetado para atender a três critérios principais:

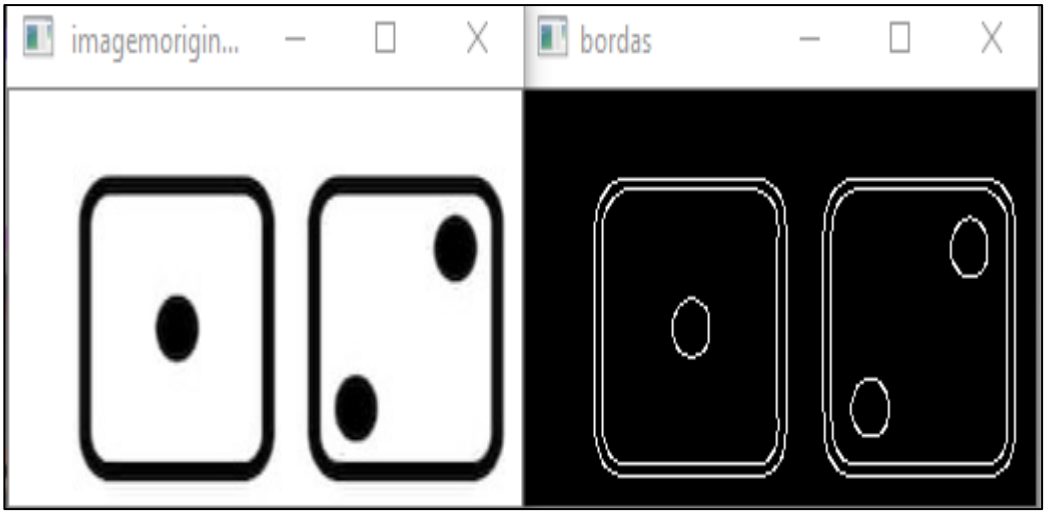
-Baixa taxa de erro: Pode fazer detecções de forma satisfatória, detectando somente as bordas realmente presentes na imagem.

-Boa localização: o pixel da borda detectada e da borda real possuiu um bom limite.

-Resposta mínima: o detector retorna apenas uma resposta.

A imagem 24 abaixo mostra a demonstração da função cv2.canny,onde é possível visualizar as bordas da imagem destacadas.

Figura 24: Aplicação da função *canny*.



Fonte: Autoria Própria (2020).

2.3.5 Processamento

Esta última etapa tem o objetivo de identificar os objetos na imagem usando suas características, fazendo uso de procedimentos algoritmos. Isto é, obter informações que possibilitem classificar ou identificar um objeto (MARQUES FILHO; VIEIRA NETO, 1999).

Uma vez que o controlador principal recebe a imagem pré-processada, isto é, sem ruído, binária e/ou com as bordas detectadas caso seja necessário, ele deve processar a imagem de uma maneira adequada. As técnicas para processar imagens envolvem procedimentos destinados à determinação de certas características do objeto captado, necessárias para a determinação de uma ação por parte do controlador, sendo eventualmente necessário um reconhecimento da figura (PAZOS, 2002).

2.4 TRANSFORMADA DE HOUGH

Em 1962, Paul Hough desenvolveu um método que pode detectar e analisar características em imagens binárias, como linhas e círculos. Tornou-se uma ferramenta amplamente utilizada em visão artificial na última década (DUDA; HART, 1972).

A transformada de Hough é um componente fundamental da visão artificial que pode ser usada para reconhecer várias características e formas de objetos em uma imagem. (FISHER et al., 2003).

2.5 OPENCV BIBLIOTECA

OpenCV é uma biblioteca de visão computacional que está disponível no *SourceForge*. Ele é escrito em C e C++ e pode ser executado em Windows, Mac *OS* X e Linux. Também atua no desenvolvimento de interfaces para vários idiomas.

O *OpenCV* foi desenvolvido para usar o máximo de poder computacional, por isso é escrito em C otimizado e pode tirar proveito de processadores multicore.

Um dos objetivos dessa biblioteca é fornecer uma infraestrutura de visão computacional simples de usar que ajuda as pessoas a criar aplicativos de visão bastante sofisticados rapidamente. O *OpenCV* biblioteca contém mais de 500 funções que abrangem muitas áreas de visão, incluindo fábrica inspeção de produto, imagens médicas, segurança, interface do usuário, calibração de câmera, estéreo, visão e robótica(BRADSKI; KAEHLER,2008).

Figura 25: *OpenCV* logo

Fonte: wikipedia.org

OpenCV é uma biblioteca livre de visão computacional que foi criada visando ajudar a alavancar os estudos e desenvolvimento de visão computacional e inteligência artificial, com uma grande variedade de funções integradas.

2.6 *PYTHON*

Python é uma linguagem de programação de alto nível, desenvolvida por Van Rossum em 1991, sua existência se deve a uma ideia de facilitar a vida dos desenvolvedores, de maneira que sua sintaxe fosse mais simples do que outras linguagens mais conhecidas (SILVA, 2019).

O uso da linguagem *python* facilita a vida dos programadores devida sua sintaxe clara e sucinta, que além de ser usada no desenvolvimento de sistemas, pode ser usada como script em vários softwares, possibilitando a automatização de tarefas, o *python* pode também ser integrado a outras linguagens de programação (BORGES,2014).

3. MATERIAIS E MÉTODOS

Este capítulo tem como objetivo apresentar os métodos aplicados para o desenvolvimento desse trabalho, explicando como aplicação destes ajuda a resolver as problemáticas propostas na introdução deste trabalho.

3.1 METODOLOGIA

O termo metodologia significa um caminho para realização de algo é parte do processo de investigação que possibilita a sistematização dos métodos e das técnicas necessárias para alcançar determinado fim, partindo do problema a ser resolvido até sua solução.

3.1.2 Quanto à abordagem

A pesquisa quantitativa, muitas vezes referida como pensamento lógico positivista, concentra-se nos princípios da lógica. Raciocínio e da experiência humana. (GERHARDT, 2009).

Para o desenvolvimento desse sistema será preciso levantar dados de como os componentes são vistos pela câmera de acordo com o brilho e a cor da luz que os atingem, o número de pixels necessários para detectar um objeto, realização de inúmeros testes para verificar as varáveis citadas acima.

3.1.3 Quanto à natureza

A pesquisa aplicada- objetiva produzir conhecimento que possa ser utilizado em uma aplicação pratica, dirigidos à solução de problemas específicos (GERHARDT, 2009). Como será preciso necessário uma pesquisa aprofundada para fazer com que o sistema de visão computacional possa realmente fazer a detecção e distinção de componentes eletrônicos dentro do processo produtivo. Então serão aplicados conhecimentos de visão artificial e processamento de imagens. Estudando técnicas que possam ser utilizadas da melhor maneira para realçar as bordas e remover ruídos das imagens, a fim de facilitar a localização no espaço do objeto de interesse, para identificá-lo e classificá-lo e efetuada a tomada de decisão.

3.1.4 Quanto aos objetivos

A pesquisa explicativa está focada em identificar os fatores que contribuem para a ocorrência de determinados fenômenos. (GERHARDT, 2009). Como a resolução da câmera, que será utilizada para fazer as aquisições de imagem, ou até mesmo a iluminação que será usada no ambiente de trabalho do sensor, podendo gerar assim variações na resposta, assim também como os variados tipos de componentes eletrônicos que possam ser detectados, tendo em vista que os componentes estão cada vez menores no mercado, e a busca por uma maneira de mostrar esses resultados da inspeção através de uma interface de usuário.

3.2 MATERIAIS

Apresentam os materiais utilizados no desenvolvimento desse trabalho, e suas principais especificações.

3.2.1 Hardware

Todo componente físico do equipamento essencial para o funcionamento do sistema, nesse caso um computador para o processamento e uma câmera para aquisição de imagens.

3.2.2 Computador

O computador será de fundamental importância para a totalidade do sistema, pois todas as operações passarão por ele, principalmente as operações de processamento de imagem e visualização dos dados. Assim as especificações do hardware utilizado foram: Notebook Acer Aspire 5,processador Intel Core I5-7200U,Hd 1000 GB, 8 gigas de memoria RAM e placa dedicada de vídeo NvideaGforce 940MX, utilizando o sistema operacional Windows 10.

3.2.3 Webcam

Para desenvolver este trabalho foi preciso estudar as especificações de câmera no mercado para utilizar o melhor equipamento possível, os pontos principais da pesquisa foram: preço, resolução e tamanho.

O sistema precisa de uma câmera com boa resolução que pudesse capturar facilmente as regiões de interesse, e que pudesse ser compacta para sua aplicação em um ambiente industrial não pudesse ser limitada fisicamente.

Neste trabalho foi utilizada a câmara OMRON, que possui alta resolução e é compacta.

Figura 26: Câmera OMRON

Fonte: Autoria Própria (2021).

3.3 EXPERIMENTAÇÃO

Para o desenvolvimento dessa pesquisa, foram analisados diversas técnicas de visão computacional e processamento de imagens visando estabelecer qual dessas desempenharia um melhor rendimento no processamento e reconhecimento dos padrões desejados, assim pequenos testes foram realizados visando detectar formas geométricas básicas e enquanto as técnicas iam desempenhando um bom rendimento a complexidade das imagens poderia aumentar.

3.4BIBLIOTECA

A biblioteca utilizada foi a *openCV*, uma biblioteca livre para desenvolvimento, o principal atrativo é que ela tem um grande acervo de funções e um imenso acervo de documentação online, facilitando sua utilização.

Para definir a biblioteca que será usada no projeto, foram analisados os seguintes fatores:

- Afinidade com a linguagem *python*
- Grande variedade de exemplos
- Possibilidade de aplicação em dispositivos embarcados

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste capítulo será apresentado como foram realizados os experimentos iniciais, a fim de gerar uma base dos conhecimentos adquiridos e aplicá-los para resolver a problemática desse trabalho.

4.1 TESTES REALIZADOS

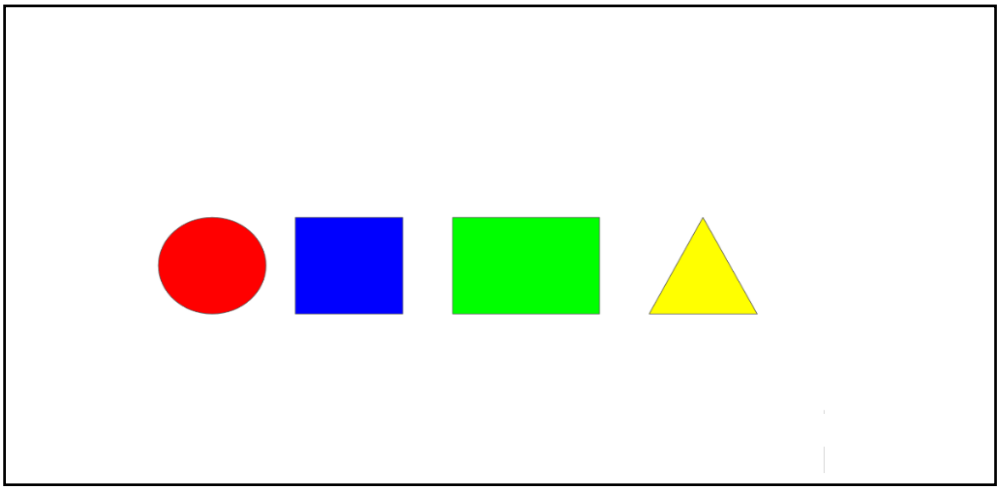
Os testes realizados nessa pesquisa foram divididos em dois tipos, o primeiro com imagens obtidas previamente de maneira a não utilizar uma câmera para aquisição de imagens, sendo essas imagens contendo as figuras geométricas mais populares a fim de testes iniciais com os algoritmos.

O primeiro passo importante é que o algoritmo o possa reconhecer esse diferente geometrias simples com boa eficiência para que ao decorrer do trabalho possa ser incrementada mais complexidade as imagens.

4.2 EXPERIMENTO

O primeiro teste foi realizado visando encontrar os contornos das figuras geométricas, os contornos são como curvas que unem todos os pontos contínuos ao longo de uma fronteira, sendo adequado para a detecção de formas, para isso foi utilizada algumas funções. A figura 27 será utilizada para realizar esse experimento.

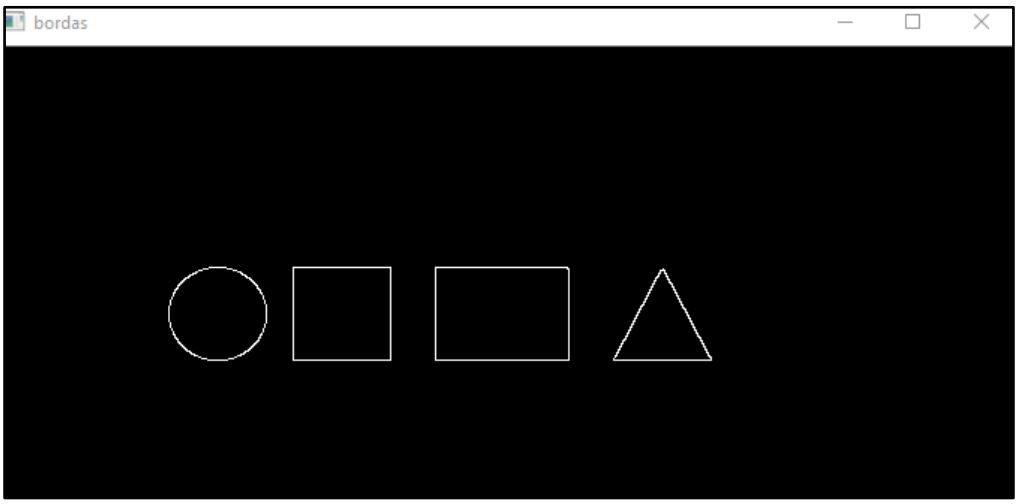
Figura 27: Formas geométricas básicas



Fonte: Autoria Própria (2021).

Utilizando a função *CV2.findContours*: essa função tem como parâmetros principais: imagem de entrada, modo de recuperação de contornos, metodo de aproximação do contorno.

Figura 28: Aplicação da função para encontrar os contornos.

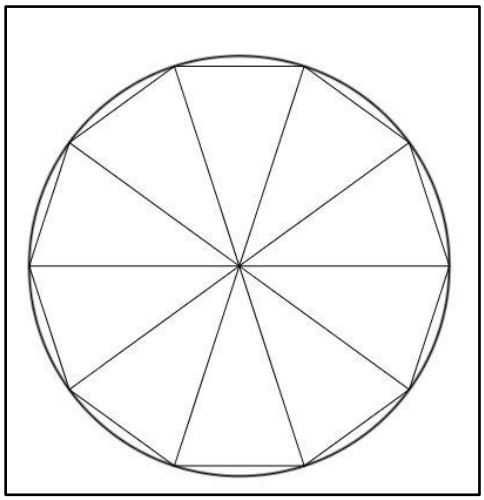


Fonte: Autoria Própria (2021).

De acordo com o *OpenCV* , o *cv2.approxPolyDP* aproxima curvas ou uma curva poligonal com uma precisão especificada. Isto é, nos diz que aproxima uma forma de contorno de outra com menos vértices, dependendo da precisão que especificarmos.

A explicação do *OpenCV* para esta função afirma que se a aplicarmos a um círculo, ela representará uma forma poligonal com muitos lados e vértices, e isso nos ajuda a validar a precisão da função para demais formas geometricas. Quando aplicada em um círculo obtemos o seguinte resultado parecido com a figura abaixo:

Figura 29: círculo representado com vários vértices



Fonte: Dante (2013)

Assim, para utilizar essa função devemos respeitar os seguintes argumentos:

Curva: Corresponde ao contorno que estamos analisando naquele momento.

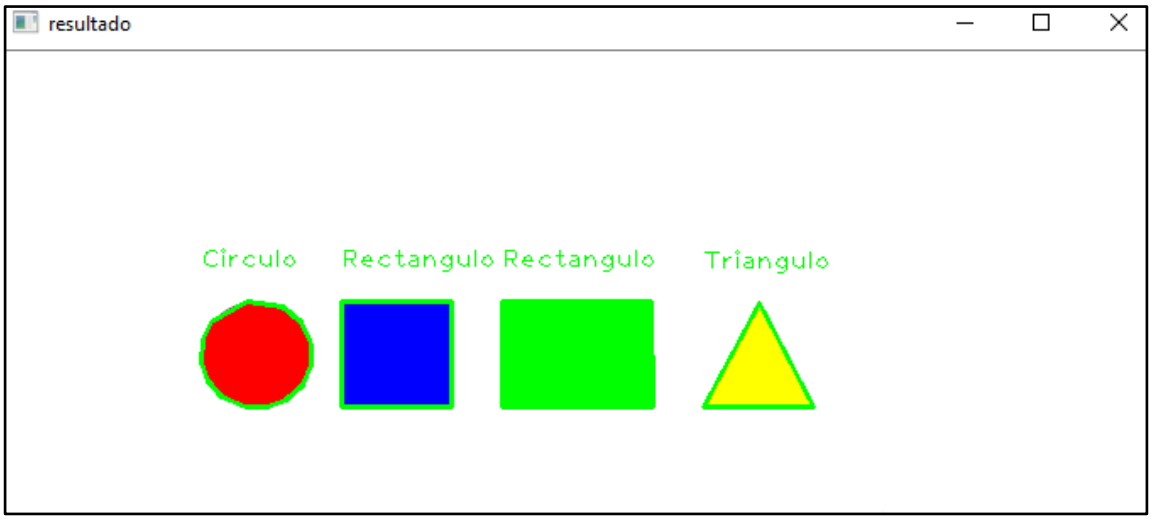
Epsilon: A precisão da função é especificada pela função arcLength, que então calculará o perímetro da curva ou o comprimento da curva.

Como argumentos para a função arcLength, a função analisará o contorno da curva seguido de verdadeiro ou falso para indicar que ela está fechada.

Fechado: verdadeiro quando a curva aproximada está fechada, ou seja, o primeiro e o último vértice estão conectados. Caso contrário, esta aberta.

As informações que a função reuniu são usadas para determinar a presença de vários objetos na curva. Por exemplo, se obtivermos três a curva será representada por uma forma triangular, quatro quadrangular e assim por diante.

Figura 30: Detecção de formas geométricas simples.



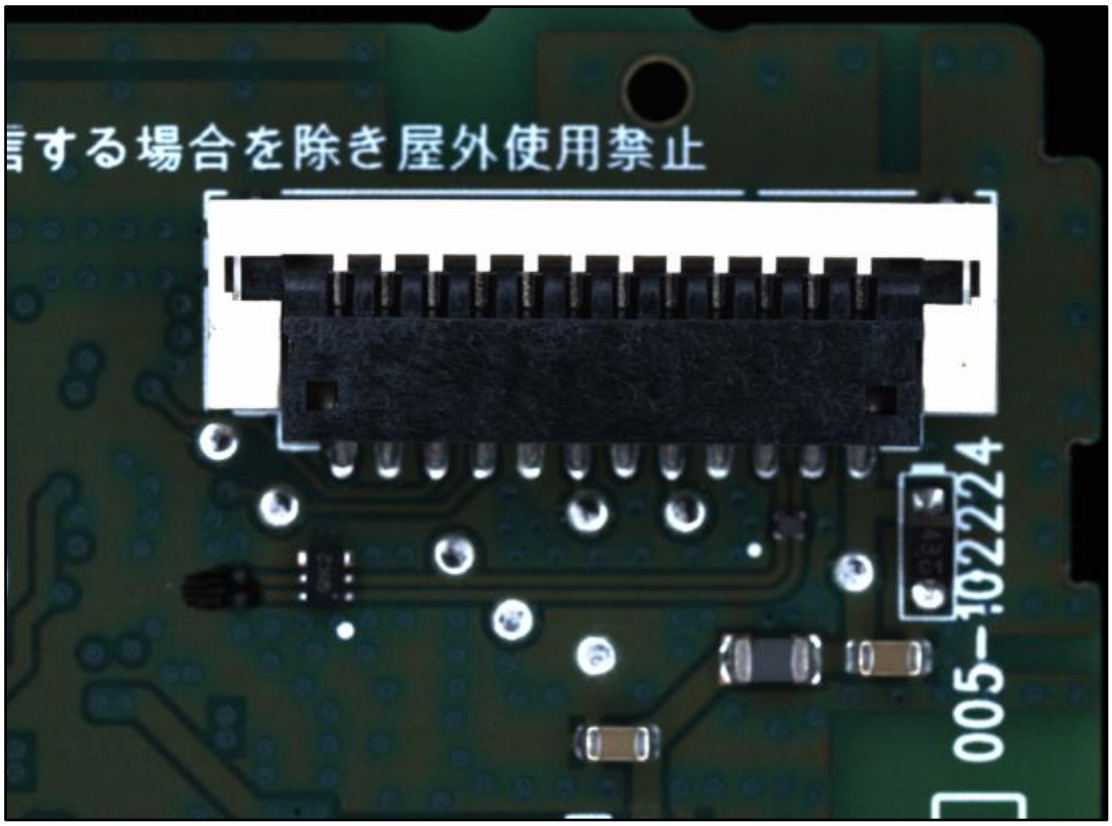
Fonte: Autoria Própria (2021).

A partir da detecção de formas geometricas básicas, possibilitou a alicação do algoritmo em figuras realistas, ou seja , de regiões de interesse para essa pesquisa.

Assim neste trabalho, procura-se fazer a detecção da presença de componentes *SMD's* presentes na placa, cada componente tem uma característica diferente, e vão ser tratados de acordo ccom sua caracteristica.

A figura 31 foi gerada através de uma câmera de alta resolução, é uma placa de circuito eletrônico responsável pelo WIFI de SMART TVS.

Figura 31: Placa de circuito WIFI

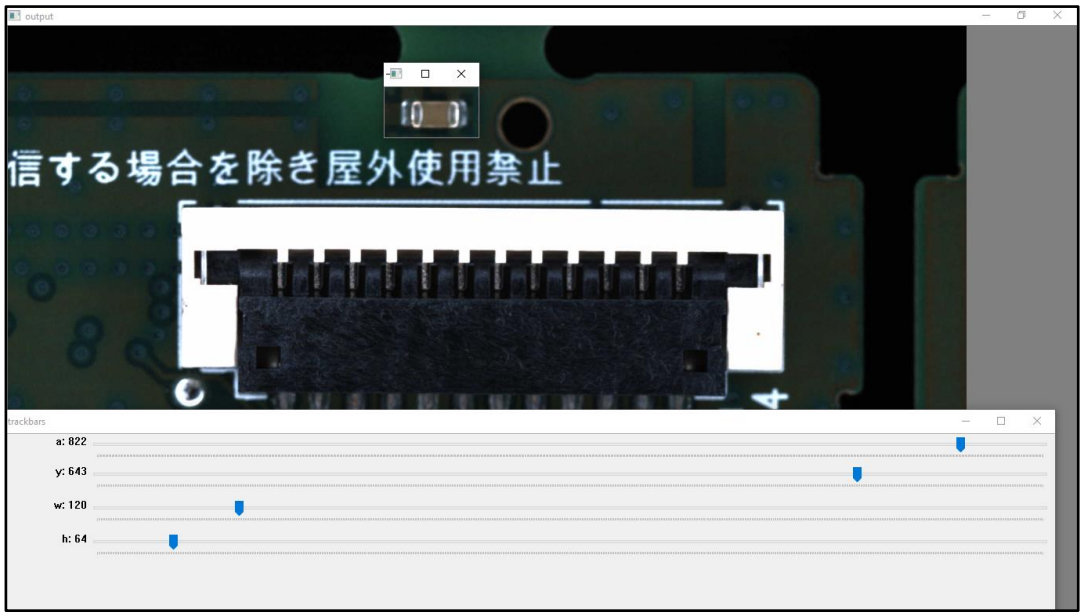


Fonte : Autoria Própria (2021).

Assim, foi desenvolvido um algoritmo que possa facilitar a extração das coordenadas dos componentes *SMD's* presentes na placa, de maniera a evitar procurar tais coordenadas usando tentativa e erro, o que em uma placa com maior número de componenestes seriam inviavel.

A figura 32 apresenta uma janela onde apresenta varias trackbars, que são barras que podem ser variadas de acordo com o usuario, duas delas são responsaveis pelo deslocamento no eixo X, e duas no eixo Y, fazendo uma combinação dessas trackbars é possivel isolar um componente epecifico e guardar suas coordenadas.

Figura 32: Criação de *trackbars*.



Fonte : Autoria Própria (2021).

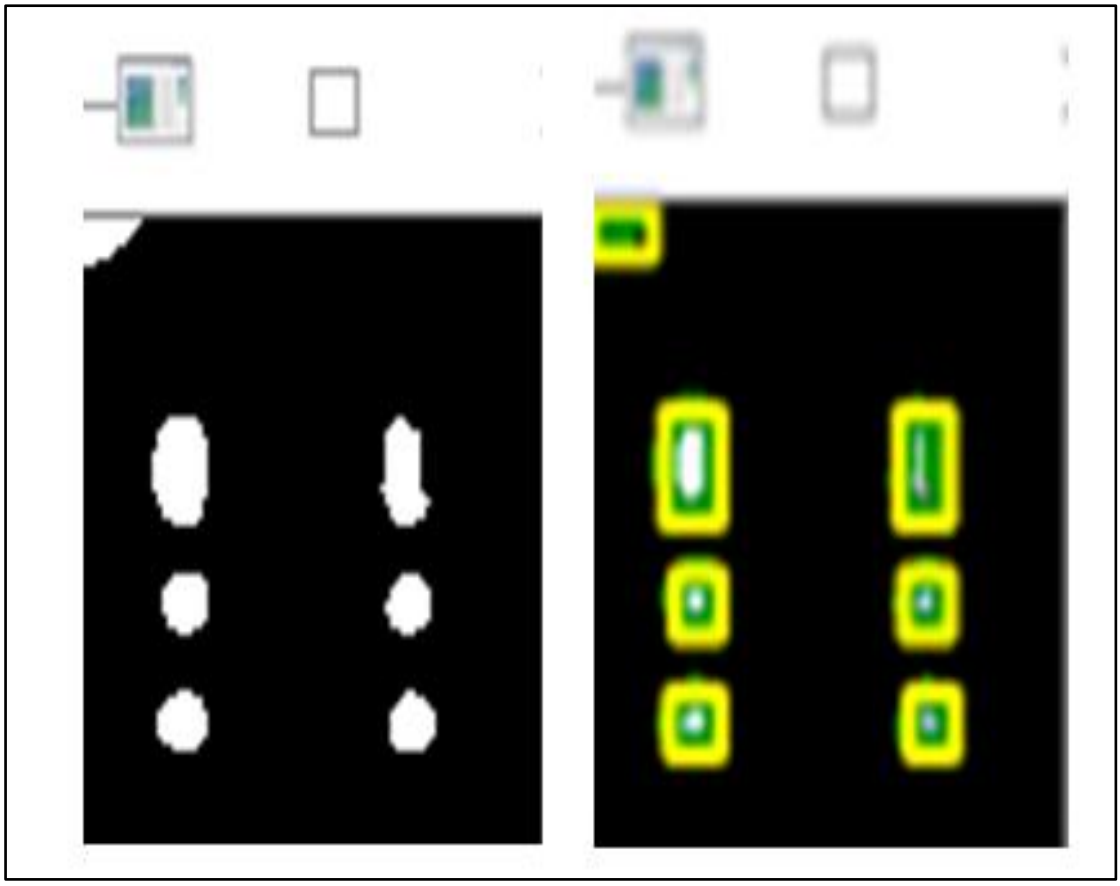
A primeira análise foi realizada em um componente *Small outline package (SOP)*. Para esse componente foi utilizada dois parametros de detecção. O primeiro é para verificar a quantidade exata de terminais no componente.

Figura 33: Componente *SOP*

Fonte :Autoria Própria (2021)

Com o uso da função cv2.approxPolyDP *f*oi possivel estrarir uma forma geometrica que se assemelha a uma elipse, assim retorno da função cv2.approxPolyDP seria um número maior que dez resultando em uma forma circular variada, assim então caso detectada esse tipo de formato nos 6 terminais do componente, resultaria em um resultado positivo do componente.

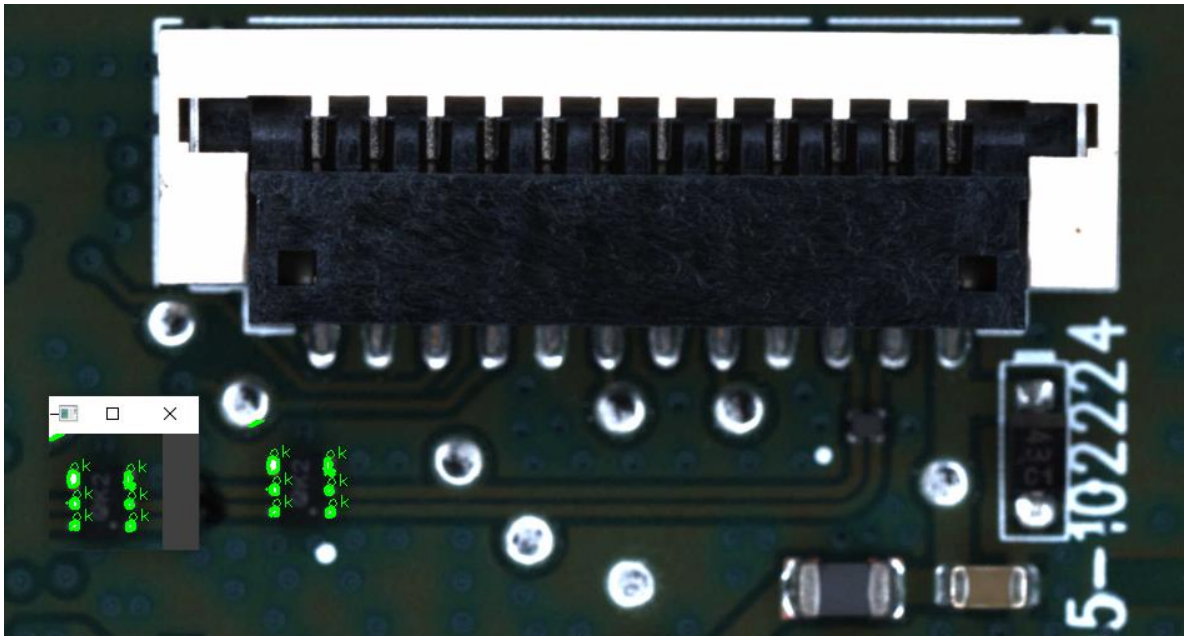
Figura 34: detecção dos terminais do componente *SOP*.



Fonte : Autoria Própria (2021)

Assim então, a resposta da análise do componenete *SOP* seria como mostrada na figura 35:

Figura 35: detecção dos terminais do componente *SOP*

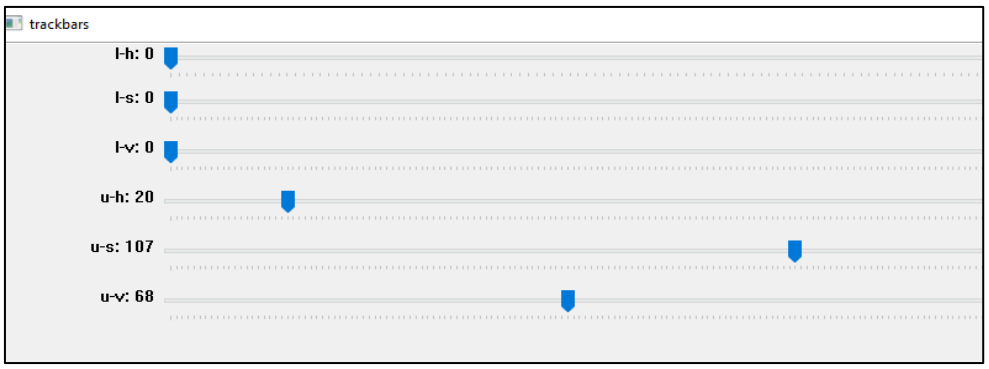


Fonte: Autoria Própria (2021)

O segundo parâmetro é analisar a presença do corpo do componente, para isso foi feito uma analise através da cor do componente.

Para isso foi feita uma conversão na imagem, passando-a para o sistema de cores *HSV,* assim foi criada um *trackbars* de modo análogo ao da figura 32, porém usado para a variação dos parâmetros de cores no sistema *HSV.*

Figura 36: *Trackbars* para limites dos parâmetros *HSV*

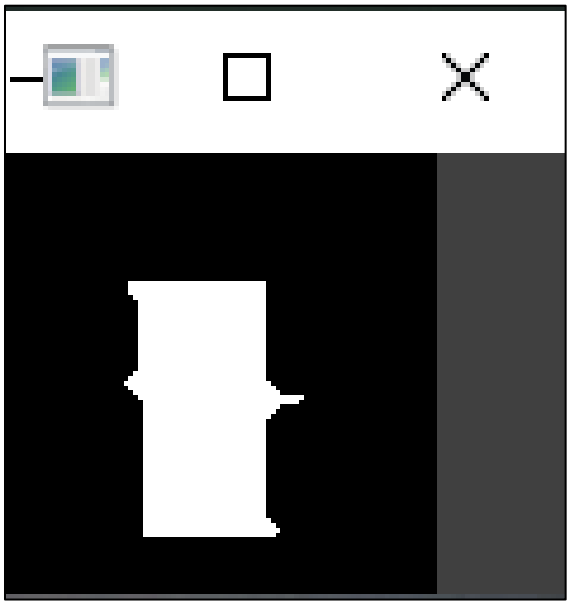


Fonte : Autoria Própria (2021)

Os *trackbarks* criados mostram a variação dos seguintes parâmetros: Os valores mais baixos do sistema de cor *HSV*, ou seja, o limite inferior, e os valores do limite superior. Tal combinação gera um intervalo de pequenas varrições da cor preta, a fim de detectar qualquer pequena mudança na tonalidade da cor que possa ser gerada devido a uma variação na cor do componente, ou uma variação na luz ambiente.

A aplicação de uma máscara vai filtrar qualquer outro ruído ou cor indesejada da *ROI* permitindo que o algoritmo possa extrair as caraterísticas desejadas de forma satisfatória.

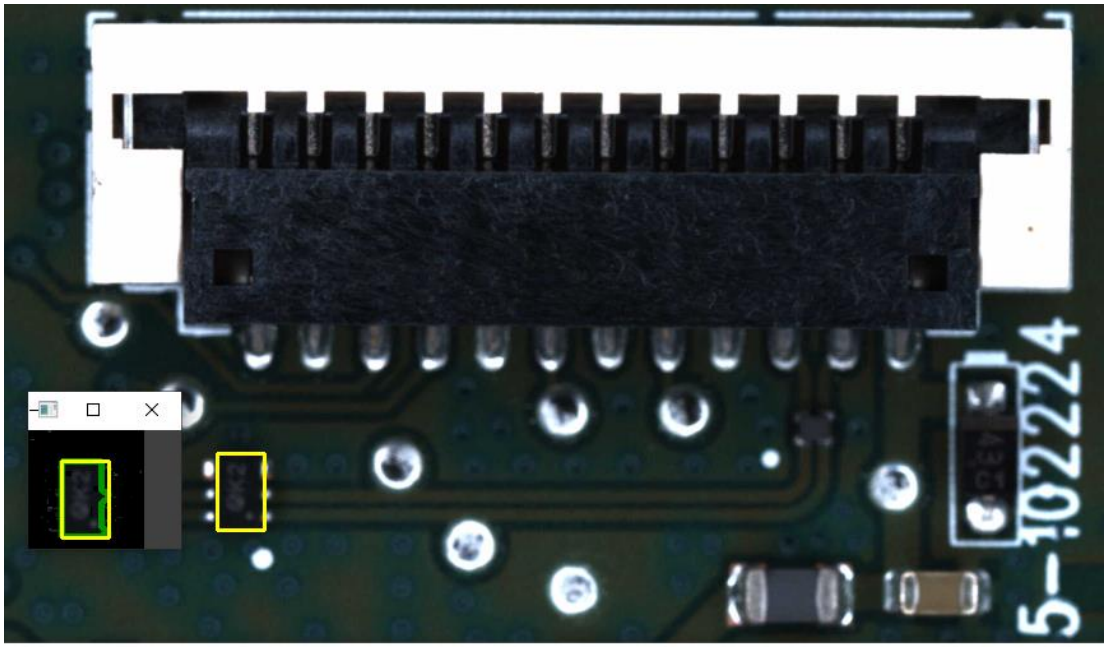
Figura 37: Aplicação da máscara no Componente *SOP*.



Fonte : Autoria Própria (2021)

Com o resultado da máscara, é possível realizar a detecção da presença do componente *SOP*, pois como o algoritmo faz uma análise focalizada nas coordenadas exatas do componente, podemos ter como resposta a sua presença ou ausência.

Figura 38: Detecção do componente *SOP.*



Fonte : Autoria própria (2021)

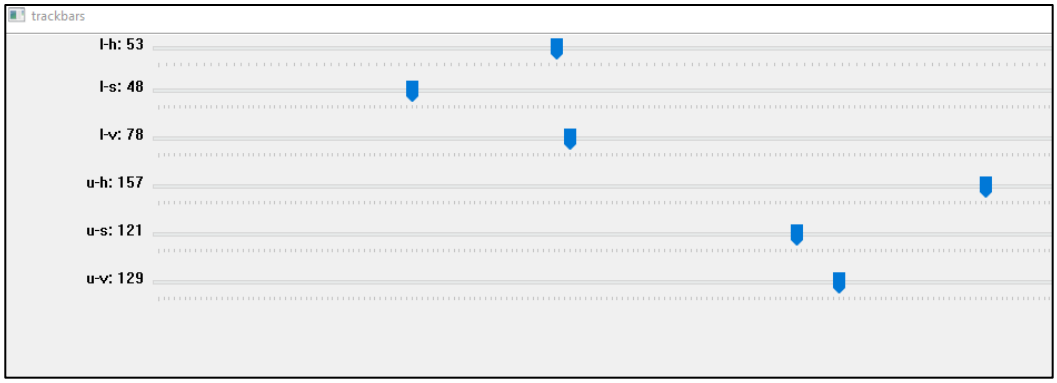
Análise do componente Capacitor *SMD*, o capacitor *SMD* apresenta um formato que facilita bastante o algoritmo devido ao sua forma geométrica simples e uma cor que faz bastante contraste com o fundo da *PCB*.

Figura 39: Capacitor *SMD*.

Fonte : Autoria própria (2021)

Assim como na extração das características do componente *SOP*, o mesmo principio para aplicação de uma mascara foi utilizado no capacitor.

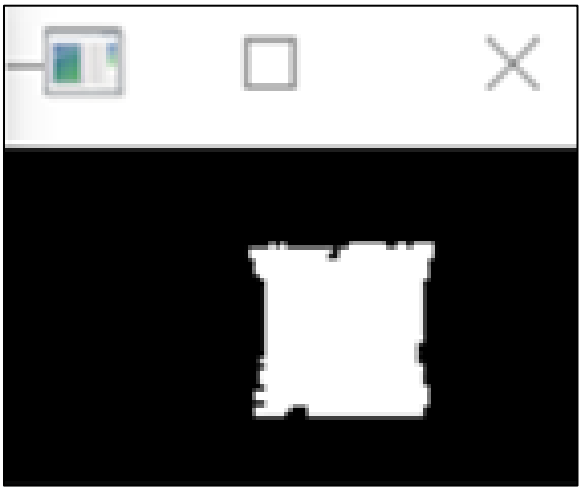
Figura 40: *Trackbars* para os limites dos parâmetros *HSV do capacitor.*



Fonte : Autoria própria (2021)

Fazendo uma segmentação através do sistema de cores *HSV,* foi possível obter uma boa segmentação, pois o capacitor possui uma forma geométrica bastante precisa e de cores que se destacam perante o fundo da *PCB.*

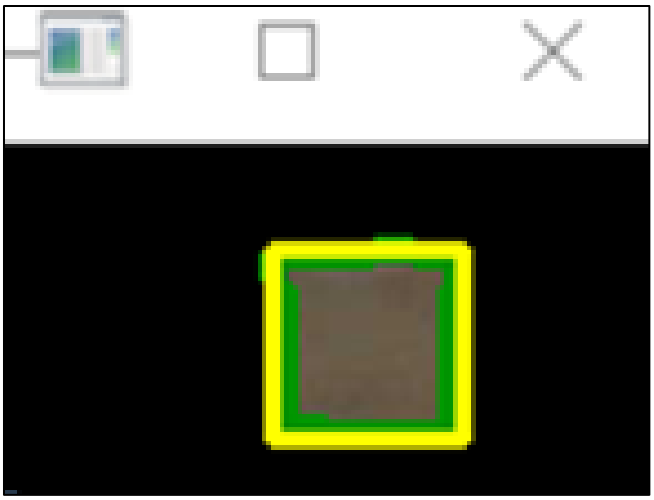
Figura 41: resultado da aplicação da máscara.



Fonte : Autoria própria (2021)

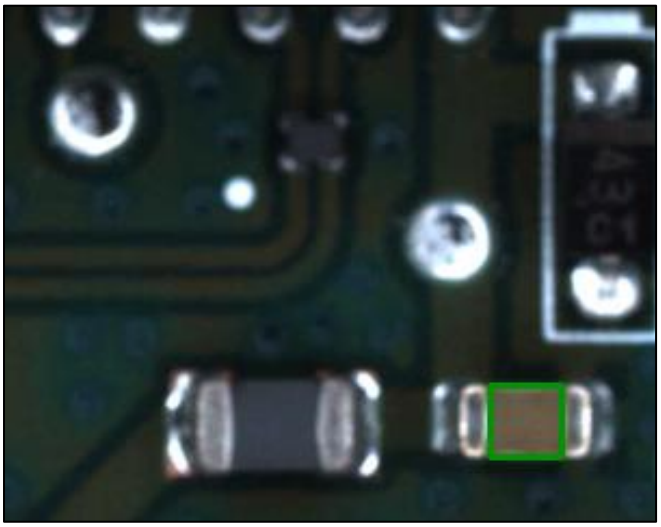
Agora com a máscara aplicada é possível fazer a detecção do capacitor utilizando novamente a função *cv2.approxPolyDP,* porém com uma busca de forma geometrica de 4 cantos, pois capacitor tem um formato retangular.

Figura 42: deteção do capacitor.



Fonte : Autoria própria (2021)

Figura 43: Resultado final com capacitor detectado.



Fonte : Autoria própria (2021)

Com os exemplos mostrados foi possível visualizar a detecção dos componentes, porém é preciso que o algoritmo indique a ausência destes quando preciso. Apesar de cada componente ter um formato e cor padrão, um único parâmetro foi utilizado para detectar sua ausência, que foi através da área de detecção mínima.

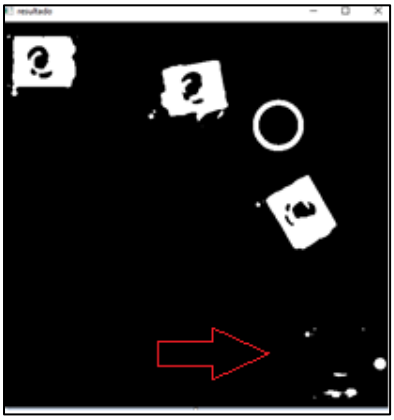
Para a detecção do *LED* positiva deve gerar uma área de detecção mínima do seu tamanho, quando isto não acontece é ativada uma condição para notificar sua ausência. Na figura 44, temos uma imagem que mostra um frame de placa com alguns *LEDs*, dois deles estão conformes, um faltando e um deslocado.

Figura 44: placa de iluminação de câmera.

Fonte : Autoria própria (2021)

Como mostra na figura 45, a ausência do último *LED* gera a detecção somente de fragmentos e ruídos, sendo este não suficiente para gerar um resultado de presença positivo.

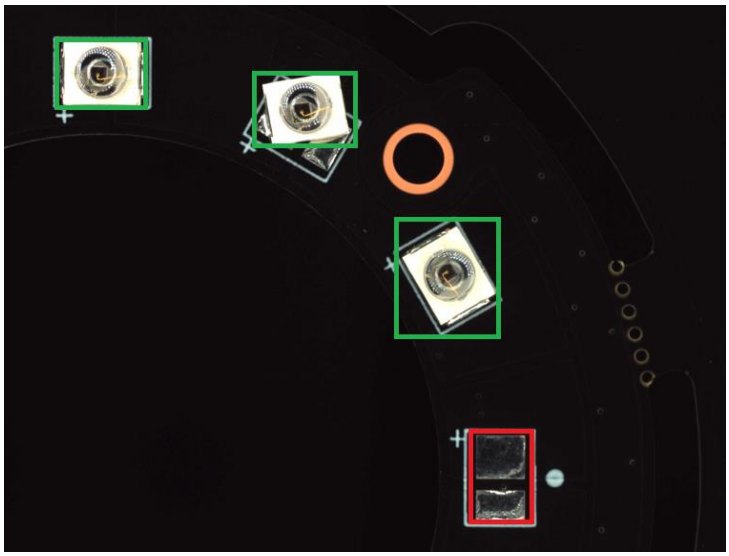
Figura 45: resultado da máscara na placa de iluminação.



Fonte : Autoria própria (2021)

Entretanto, apesar de ter detectado de forma satisfatória o componente ausente, se efetuada a analise do funcionamento da placa, o segundo *LED* não estaria conforme, pois alguns dos seus terminais estão sem contato com a solda o que resultaria em um mau funcionamento, mas ainda assim foi detectado como coo bom. Assim é preciso refinar o algoritmo para detectar com precisão as variações de ângulos padrões dos componentes em cada uma de suas posições.

Figura 46: resultado final, mostrando a detecção da ausência do *LED*.



Fonte : Autoria própria (2021)

Algumas detecções podem tender a dar erros falsos devido à máscara ficar um pouco poluída, o que pode ser resultado de ruídos nas imagens coletadas e variações na luminosidade do ambiente.

Assim uma possível maneira de melhorar esse sistema seria usando o que há de mais recente em técnicas de detecção de imagens, o que chamamos *de Deep Learning*, que significa aprendizado profundo, é um tipo de IA que visa fazer os computadores pensar e raciocinarem como humanos. Ele usa modelos hierárquicos complexos para simular como os humanos aprendem novas informações. Porém essa tecnologia não será utilizada neste trabalho.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento desse sistema de detecção de componentes eletrônicos mostrou como a visão computacional está mais aplicada, sendo usada para auxiliar a resolver problemas que antes pareciam ser resolvidos somente através da visão humana.

Com este estudo, é possível inferir que com o uso de algoritmos de processamento de imagem podemos refinar uma imagem com um grande número de detalhes para uma imagem que contém as informações necessárias para resolver determinado problema, nesse caso, a detecção de componentes eletrônicos. Assim, foi possível analisar alguns tipos de componentes eletrônicos e entender como a máquina ira enxerga-los para dizer se estão presentes ou não.

Sendo assim, entende-se que os objetivos propostos foram alcançados, uma vez que o os distintos componentes simulados puderam ser detectados de maneira satisfatória, obtendo assim um resultado positivo.

Contudo, devido à diversidade de detalhes das imagens que exigem uma grande resolução a aquisição destas imagens foi feita utilizando uma máquina *AOI* (*Automatic Optical Inspection*) que tem grande poder de captura. Ainda assim pode-se inferir que várias melhorias podem ser feitas em trabalhos futuros:

- Montar um sistema de aquisição próprio, que possa capturar as imagens em alta definição.
- Melhorias no algoritmo para aperfeiçoamento das análises para evitar falsos positivos como aconteceram na figura 46.
- Implementar uma tecnologia mais recente como *Deep Learning*.

REFERÊNCIAS

ARTERO, ALMIR OLIVETTE; TOMMASELLI, ANTONIO MARIA GARCIA. **LIMIARIZAÇÃO AUTOMÁTICA DE IMAGENS DIGITAIS**. Bulletin of Geodetic Sciences, [S.l.], nov. 2018. ISSN 1982-2170. Available at: <https://revistas.ufpr.br/bcg/article/view/63040>.Acesso em: 20 set. 2021.

BALDNER, Felipe; COSTA, Pedro. M; GOMEZ, Juliana; LETA, Fabiana. **Metrologia por imagem.** Rio de Janeiro: Elsevier, 2017.

BARELLI, F. **Introdução à visão computacional: Uma abordagem prática com Python e OpenCV** . Casa do código, 2018.

BORGES, Luiz Eduardo**. Python para Desenvolvedores: Aborda Python 3.3.** São Paulo**:**Novatec Editora; 1ª edição (3 outubro 2014)

BRADSKI, Gary; KAEHLER, Adrian. ***Learning OpenCV*.** Editora: O'Reilly Media, Setembro 2008:Primeira edição

BUDIHARTO, W.; GROUP, S. P. ***Modern Robotics with OpenCV***. Science Publishing Group, 2014. ISBN 9781940366128. Disponível em: <http://books.google.com.br/books?id=mbFUAwAAQBAJ>. Acesso em: 21 out. 2021.

DAMAITRE, Eugene. ***Machine Vision System*.** 2018**.** Disponível: https://www.roboticsbusinessreview.com/events/sensors-lead-big-data-dont-expect shortcuts-says-sensors-expo-host/attachment/machine_vision_system/Acesso em: 25. Nov 2021.

DANTE, Luiz Roberto. Matemática: contexto & aplicações. 2. ed. São Paulo: Ática, 2013.

DUDA, Richard O; HART, Peter E. ***Use of the hough transformation to detect line sand curves in pictures*.**Comunicações do ACM, 1971.

FELTRIN, Fernando. **Visão Computacional em Python**. Uniorg 2020.

GERHARDT, Engel; SILVEIRA, Denise. **Métodos de Pesquisa.** Rio Grande do SUL, UFRGS, 2009.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais.** São Paulo: Editor Edgard Blücher Ltda., 2000.

JÄHNE, Bernd. ***Digital Image Processing***. Berlin Heidelberg: Editora Springer-Verlag, 2002.

MARENGONI, Maurício; STRINGHINI, Stringhini. **Tutorial: Introdução à Visão Computacional usando OpenCV**. Revista de Informática Teórica e Aplicada, Porto Alegre, RS, v. 16, n. 1, pág. 125-160, mar. 2010. Disponível em: <https://www.seer.ufrgs.br/rita/article/view/rita_v16_n1_p125/7289 >. Data de acesso: 06 fev. 2021.

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

MILANO, Danilo; HONORATO, Luciano B. **Visão computacional**. Disponível:<http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNICAMP_visa oComputacional.pdf>. Unicamp, 2010. Acesso em: 18 dec. 2020.

MURAROLLI, Priscila**. Inovações Tecnológicas Nas Perspectivas Computacionais.** São Paulo: Editora Biblioteca24horas, 2015

PAZOS, Fernando. **Automação de sistemas & robótica**. Rio de Janeiro: Axcel Books do Brasil Editora, 2002.

RUDEK, Marcelo; SANTOS, COELHO, Leandro; CANCIGLIERI JÚNIOR, Osiris. **Visão computacional aplicada a sistemas produtivos: fundamentos e estudo de caso. In: XXI Encontro Nacional de Engenharia de Produção, Salvador.** 2001. 1p. Disponível: http://www.las.pucpr.br/rudek/pdf/Enegepxxi.pdf Acesso em: 20 dec. 2020.

SILVA, Tiago. **Python de A Z**. Santa Catarina: Clube dos Autores, 2019.

SZELISKI, Richard. ***Computer Vision: Algorithms and Applications***. Springer Science & Business Media, 30 de set. de 2010.

ZANOTTA, Daniel; FERREIRA, Matheus; ZORTEA, Maciel. **Processamento de imagens de satélite**. São Paulo, Editora: Oficina de Textos, 2019.

ANEXO 1

TERMO DE ACEITE DE ORIENTAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Eu, Docente AILTON GONÇALVES REIS, SIAPE nº 0709656, manifesto, por meio deste, minha participação como orientador do discente WENDEL DA COSTA PRADO matrícula 2015002328, do Curso Superior de BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO, comprometendo-me a acompanhar, analisar e orientar o(a) referido(a) discente nas etapas necessárias ao desenvolvimento do Trabalho de Conclusão de Curso Proposto.

Manaus – AM, 05 de Março de 2022.

Ailton Gonçalves Reis

Prof. Dr. Ailton Gonçalves Reis - Orientador

Ciente em, 26 de Abril de 2022.

Luiz Henrique Portela de Abreu
Coordenador do curso de ENG de controle e Automação - port Nº 479 - 27/10/2020

Coordenador do Curso de Engenharia de Controle de Automação

ANEXO 2

TERMO DE COMPROMISSO DO ORIENTADO SOBRE AS NORMAS/REGULAMENTOS DO TRABALHO DE CONCLUSÃO DE CURSO

Eu, Wendel da Costa Prado, estudante regularmente matriculado no Curso Superior de Engenharia de Controle e Automação, estou ciente e concordo com as normas/regulamentos instituídos para o desenvolvimento do meu Trabalho de Conclusão de Curso. Outrossim declaro seguir tal regimento. Por estar plenamente de acordo firmo o presente.

Manaus – AM, 22 de Abril de 2022.

Wendel da Costa Prado

Assinatura do estudante

ANEXO 3

TERMO DE COMPROMISSO DE ORIGINALIDADE DO TRABALHO DE CONCLUSÃO DE CURSO

O presente termo é documento integrante de todo Trabalho de Conclusão de Curso (TCC) a ser submetido à avaliação do IFAM – Campus Manaus Distrito Industrial como requisito obrigatório à obtenção do título de Engenheiro de Controle e Automação.

Eu, Wendel da Costa Prado, CPF: 00864344201, registro de Identidade 22813330, na qualidade de estudante do Curso Superior de Engenharia de Controle e Automação do IFAM – Campus Manaus Distrito Industrial, declaro que o Trabalho de Conclusão de Curso apresentado em anexo, cujo título é Desenvolvimento de um Sistema de Inspeção de Componentes Utilizando Técnicas de Visão Computacional , encontra-se plenamente em conformidade com os critérios técnicos, acadêmicos e científicos de originalidade.

Nesse sentido, declaro para os devidos fins, que:

a) O referido TCC foi elaborado com minhas próprias palavras, idéias, opiniões e juízos de valor, não consistindo, portanto PLÁGIO, por não reproduzir, como se fossem meus, pensamentos, idéias e palavras de outra pessoa;

b) as citações diretas de trabalhos de outras pessoas, publicados ou não, apresentadas em meu TCC, estão sempre claramente identificadas entre aspas e com a completa referência bibliográfica de sua fonte, de acordo com as normas estabelecidas pela Associação Brasileira de Normas Técnicas - ABNT;

c) todas as séries de pequenas citações de diversas fontes diferentes foram identificadas como tais, bem como às longas citações de uma única fonte foram incorporadas suas respectivas referências, pois fui devidamente informado(a) e orientado(a) a respeito do fato de que, caso contrário, as mesmas constituiriam plágio;

d) todos os resumos e/ou sumários de idéias e julgamentos de outras pessoas estão acompanhados da indicação de suas fontes em seu texto e as mesmas constam das referências do TCC, pois fui devidamente informado(a) e orientado(a) a respeito do fato de que a inobservância destas regras poderia acarretar alegação de fraude.

O(a) docente responsável pela orientação do meu trabalho de conclusão de curso (TCC) apresentou-me a presente declaração, requerendo o meu compromisso de não praticar quaisquer atos que pudessem ser entendidos como plágio na elaboração do meu TCC, razão pela qual declaro ter lido e entendido todo o seu conteúdo e submeto o documento em anexo para apreciação do IFAM - Campus Manaus Distrito Industrial como fruto do meu exclusivo trabalho.

Manaus – AM, 22 de Abril de 2022.

Assinatura do estudante