



UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO

## **Relatório GenericStuff**

Relatório de Projeto Final

Gabriel Santana de Carvalho  
Reinaldo Lima Mendonça Junior



São Cristóvão – Sergipe

2022

# 1

## Introdução

O presente relatório apresenta a solução de software desenvolvida para atender a demanda solicitada pela GenericStuff.

A demanda consiste em um sistema de inspeção de objetos, que captura imagens contendo objetos de duas categorias: objetos com furos, e objetos sem furos. Para cada imagem capturada pelo sistema, a solução deve informar o total de objetos na imagem, e quantos objetos com furos e sem furos a imagem contém. Como parâmetros adicionais os objetos não se tocam nem estão sobrepostos, mas podem estar próximos uns dos outros, além de poderem ter qualquer forma ou tamanho.

É necessário ainda exibir uma aplicação real e adaptações necessárias para aplica-las levando em consideração os conceitos de ética discutidos na disciplina.

### 1.1 Objetivos

#### 1.1.1 Objetivo Geral

- Solução de software que a partir de uma imagem binária de objetos informe quantos objetos a imagem contém, e quantos desses objetos possuem furos ou não.

#### 1.1.2 Objetivos Específicos

- Detectar número total de objetos
- Detectar objetos com furo

# 2

## Desenvolvimento

Inicialmente, antes da elaboração da solução propriamente, foi decidido que a aplicação prática desta seria o uso em uma fábrica alimentícia. O algoritmo seria então utilizado na seção de controle de qualidade, assegurando a integridade dos alimentos fornecidos aos consumidores.

Tomando essa premissa como ponto de partida, a solução foi então dividida em dois subproblemas, o primeiro sendo a detecção e contagem de objetos, e o segundo sendo a distinção entre tais objetos, ou seja, dentre os objetos quais são objetos com furos e quais são objetos inteiriços.

### 2.1 Contagem de Objetos

Para o problema da contagem de objetos inicialmente foi utilizada a mascara a seguir:

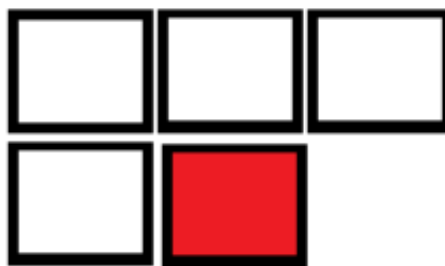


Figura 1 – FONTE: AUTORES 2022.

Para a explicação do processo executado pelo algoritmo são necessárias algumas definições

- Os pixels considerados como 1 são pixels correspondentes a uma imagem, ou seja, correspondem aos pixels pretos, analogamente pixels 0 são correspondentes ao fundo ou a

buracos.

- Todas as imagens no algoritmo passaram por um padding de 1 em todas as laterais.
- A cada iteração será separado globalmente o maior contador encontrado até o momento, seu valor sempre será maior que 2 pois a imagem é binária.

Seguindo as definições anteriores o algoritmo segue a lógica que, se o pixel encontrado no quadrado vermelho é um pixel preto então são avaliados os três casos a seguir:

- Caso nenhum dos vizinhos do pixel vermelho, correspondentes a máscara, sejam maiores do que 1, então o pixel vermelho recebe o valor do contador global e a máscara segue deslizando para a direita.
- Caso algum dos vizinhos seja maior que 1 o vizinho de menor valor será selecionado para ser adicionado no pixel atual, caso o menor valor e o maior valor dos vizinhos seja igual o algoritmo segue normalmente.
- Caso dentre os vizinhos do nó vermelho, o menor e o maior valor sejam diferentes, o algoritmo irá passar novamente por todos os nós da imagem pois será necessário validar se algum componente que tinha um vão no meio foi considerado como dois componentes.

Ao final da execução desse algoritmos irá ser obtida uma matriz em que os diferentes objetos estão marcados por valores maiores que um, a quantidade de elementos maiores que um com valores diferentes será a quantidade de objetos obtida na imagem

## 2.2 Distinção entre Objetos

Com base na matriz de objetos e no número total de objetos na imagem, segue-se então para a etapa de identificar quantos desses objetos são objetos inteiros e quantos são objetos com furos. Foram realizadas duas abordagens em relação a esse propósito, que serão explicadas a seguir.

Ambas as abordagens consistiam em preencher os furos, e contabilizar novamente quantos objetos foram detectados na imagem. Para cada objeto verifica-se então se o tamanho do objeto mudou entre a primeira detecção e a nova detecção. Caso tenha mudado, o objeto possuía um furo que foi preenchido, logo, é um objeto com furo, caso contrário, é um objeto inteiro.

### 2.2.1 Abordagem A

A primeira abordagem para preencher o furo consistiu em inverter a imagem e em seguida utilizar o método de detecção de objetos, como o método estaria sendo aplicado na imagem

inversa, em vez de detectar os objetos detectaria, portanto, os furos. Uma vez detectados os furos, estes eram preenchidos, substituindo os pontos por 1 na matriz original, tornando assim os objetos inteiriços.

Após o preenchimento dos furos, é realizada uma comparação entre o tamanho dos objetos, entre a primeira detecção, na imagem original, e a segunda detecção, na imagem com furos preenchidos. Caso um objeto tenha mudado de tamanho infere-se que um furo foi preenchido, e portanto, era um objeto com furo. Caso o tamanho não tenha se alterado, era um objeto inteiriço desde o princípio.

Vale mencionar que nessa verificação são validados os objetos diretamente, portanto independente de haver mais de um furo em um objeto ele não será contabilizado como um objeto com furo novamente.

### 2.2.2 Abordagem B

Na segunda abordagem foram utilizados os conceitos de erosão e dilatação, primeiro os objetos são dilatados, de modo a preencher os furos, o que era verificado a cada iteração da dilatação. Nessa abordagem para contabilizar a quantidade de operações de dilatação necessárias para se preencher todos os furos é feito um loop em que a cada dilatação verifica-se se a imagem invertida possui número de objetos maior que um, que seria o fundo, já que a verificação na imagem invertida observa os pontos pretos, enquanto isso acontece é necessário realizar uma nova dilatação na imagem original.

Em seguida, os objetos eram erodidos pela mesma quantidade de iterações de erosões realizadas, obtendo como resultado objetos similares aos originais, mas inteiriços, prosseguindo então para uma nova detecção de objetos na imagem com furos preenchidos e para uma comparação entre o tamanho dos objetos, similar a anterior.

Vale mencionar que foi verificado que as operações de erosão e dilatação poderiam distorcer o objeto original, assim, seria possível que o tamanho do objeto mudasse, mas não por ter um furo que foi preenchido, e sim por mera distorção na erosão e na dilatação.

Ao analisar o caso prático foi percebido que apesar de casos assim existirem, os objetos eram alterados minimamente, então nessa abordagem foi considerada uma margem de erro de 5 pixels ao comparar o tamanho do objeto antes das operações e depois das operações.

### 2.2.3 Possível Aplicação

Como supracitado, uma possível aplicação do algoritmo seria na detecção de furos em produtos alimentícios, para a implementação dessa ideia seria necessário que os produtos fossem colocados em uma esteira com um espaçamento adequado e fossem transmitidos a partir de imagens somente com tons pretos e brancos. Essa aplicação poderia evitar a saída da fábrica de

produtos que possivelmente estejam contaminados devido a avaria na embalagem para a casa dos consumidores.

# 3

## Conclusão

A solução foi implementada, conforme descrito, e alguns testes iniciais foram realizados. Com base na proposta e na solução conclui-se que apesar de alguns problemas de performance na abordagem B, ambas atendem ao solicitado e poderiam ser utilizadas.

No entanto, vale mencionar que a abordagem B além dos problemas de performance ditos considerou uma margem de erro de 5 pixels na etapa de distinção de objetos, o que acaba acarretando também em uma limitação do tamanho do furo, que não poderia ser inferior a 5 pixels, caso que apesar de improvável, pode ocorrer.

Em relação a abordagem A, não é necessária uma margem de erro, e a performance também se apresenta satisfatória, no entanto, é uma abordagem algébrica e que utiliza conceitos de grafos, conteúdo que apesar de relacionado não engloba diretamente os conteúdos ministrados de processamento de imagens.

Vale ainda mencionar que melhorias poderiam ser feitas em ambas as abordagens, desde menos iterações pelas matrizes, ou mesmo operações simultâneas. No entanto, percebeu-se que o nível de complexidade da compreensão e legibilidade do código seria piorado, e decidiu-se priorizar tal fator.