



**COORDENADORIA DE CIÊNCIA DE DADOS**

**LUIS FELIPE BEU DE ABREU**

**APLICAÇÃO DA VISÃO COMPUTACIONAL PARA INSPEÇÃO DE  
QUALIDADE EM UMA INDÚSTRIA GRÁFICA**

**SOROCABA, SP**

**2024**



## **ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS**

**LUIS FELIPE BEU DE ABREU**

### **APLICAÇÃO DA VISÃO COMPUTACIONAL PARA INSPEÇÃO DE QUALIDADE EM UMA INDÚSTRIA GRÁFICA**

**SOROCABA, SP  
2024**



**LUIS FELIPE BEU DE ABREU**

**APLICAÇÃO DA VISÃO COMPUTACIONAL PARA INSPEÇÃO DE  
QUALIDADE EM UMA INDÚSTRIA GRÁFICA**

Trabalho de conclusão de curso  
apresentado ao Centro Universitário  
Facens como exigência parcial para  
obtenção do certificado de conclusão de  
curso Especialização em Ciência de  
Dados.

Orientador (a): Júlio Mastrodomenico.

**SOROCABA, SP**

**2024**

**LUIS FELIPE BEU DE ABREU**

**APLICAÇÃO DA VISÃO COMPUTACIONAL PARA INSPEÇÃO DE  
QUALIDADE EM UMA INDÚSTRIA GRÁFICA**

Trabalho de conclusão de curso apresentado  
ao Centro Universitário Facens como  
exigência parcial para obtenção do diploma de  
pós-graduação em Ciência de Dados.  
Orientador (a): Júlio Mastrodomenico.

**BANCA EXAMINADORA**

---

Prof. 1  
Afiliação (Centro Universitário Facens)

---

Prof. 2  
Afiliação (Centro Universitário Facens)

---

Prof.<sup>a</sup> 3  
Afiliação (Centro Universitário Facens)

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus por me ajudar à alcançar meus objetivos, à esta instituição de ensino pela oportunidade, à todos os professores, coordenadores, colegas de classe e funcionários em geral que de alguma forma estiveram comigo nesta trajetória.

Aos meus familiares e minha esposa, que depositaram muita confiança e credibilidade em mim e em meus planos.

Aos orientadores pelo incentivo disponibilizado, ao tempo doado para que eu pudesse encontrar a melhor forma de concluir o meu projeto.

E, em especial, à gráfica XYZ por disponibilizar seus materiais e tempo para que eu pudesse utilizar este caso como foco do meu trabalho.

## RESUMO

A impressão gráfica, seja em máquina offset, digital ou flexográfica, requer uma cuidadosa inspeção de qualidade para garantir que o produto final atenda às especificações necessárias exigidas pelo cliente. Em lotes maiores, dificilmente há uma inspeção individual para todos os produtos, em vez disso, é feito o uso da técnica de inspeção NQA, onde apenas uma amostra é inspecionada e a partir de um número aceitável de falhas, o lote é aprovado ou não. Apresenta-se no decorrer do trabalho uma forma automatizada de inspeção com visão computacional, onde cada item é verificado individualmente em busca de falhas que podem não ser encontradas na inspeção feita atualmente. O código foi escrito em *python* utilizando principalmente a biblioteca *opencv* e ao final gera tabelas de valores de possíveis falhas identificadas no tratamento da imagem digitalizada. Conclui-se com base nos testes que, o código conseguiu retornar os dados necessários para classificar um item como aprovado ou rejeitado, com potencial de reduzir o tempo de inspeção de produtos com poucas cores em sua composição.

**Palavras-chave:** Qualidade; Visão Computacional; Automação; Indústria Gráfica.

## **ABSTRACT**

Graphic printing, whether on offset, digital, or flexographic machines, requires careful quality inspection to ensure that the final product meets the necessary specifications demanded by the client. In larger batches, individual inspection of all products is rarely performed. Instead, the AQL inspection technique is used, where only a sample is inspected, and based on an acceptable number of defects, the batch is either approved or rejected. Throughout the work, an automated inspection method using computer vision is presented, where each item is individually checked for defects that may not be detected in the current inspection process. The code was written in Python, primarily using the OpenCV library, and in the end, it generates tables of values for possible defects identified in the processing of the digitized image. Based on the tests, it is concluded that the code successfully returned the necessary data to classify an item as approved or rejected, with the potential to reduce inspection time for products with few colors in their composition.

**Keywords:** Quality; Computer Vision; Automation; Printing Industry.

## LISTA DE ILUSTRAÇÕES

FIGURA 1. Exemplo de rótulo original; Rótulo “bom”; Rótulo “ruim”. .....	17
FIGURA 2. Contornos identificados nos rótulos impressos a partir do contorno da imagem original. ....	18
FIGURA 3. Exemplo de rótulo original com cor para demonstração do código; Rótulo “bom”; Rótulo “ruim”. ....	19
FIGURA 4. Histogramas de cada canal de cor, para cada uma das três imagens mostradas anteriormente. ....	20
FIGURA 5. Diferença entre a imagem original e digitalizada; Diferença com <i>blur</i> e diferença com <i>threshold</i> . ....	21
FIGURA 6. Imagem original com contornos identificados; Diferença com <i>threshold</i> aplicado; Diferença filtrada. ....	21
FIGURA 7. Imagem original; Imagem digitalizada; Diferença SSIM. ....	22
FIGURA 8. Diferença SSIM; Top 5 contornos detectados; Defeitos detectados. ....	23
Figura 9. Máscara criada inicialmente; Máscara cortada para os quatro lados. ....	24
FIGURA 10. Bounding box dos contornos de desalinhamento identificados. ....	24
FIGURA 11. Diferença entre dois centroides de contornos de canais diferentes. ....	25
FIGURA 12. Distribuição da diferença entre áreas de cada par de cores. ....	25
FIGURA 13. Histogramas dos canais RGB comparando a imagem escaneada. ....	27



## **LISTA DE TABELAS**

TABELA 1. Resultados da análise de cor. ....	27
TABELA 2. Resultados da análise de detecção de manchas. ....	27
TABELA 3. Resultados da análise de desalinhamento. ....	28
TABELA 4. Resultados da análise de desalinhamento de cores. ....	28
TABELA 5. Resultados da análise de desalinhamento de cores versão final. ....	29

## LISTA DE SIGLAS

AQL	<i>Acceptable Quality Level</i>
BOPP	<i>Biaxially Oriented Polypropylene</i>
FAST	<i>Features from Accelerated Segment Test</i>
ISO	<i>International Organization for Standardization</i>
MSE	<i>Mean Squared Error</i>
NQA	Nível de Qualidade Aceitável
ORB	<i>Oriented FAST and Rotated BRIEF</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SNR	<i>Signal-to-Noise Ratio</i>
SSIM	<i>Structural Similarity Index</i>
SVG	<i>Scalable Vector Graphics</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>12</b>
<b>2.1</b>	<b>INSPEÇÃO DE AMOSTRAGEM .....</b>	<b>12</b>
<b>2.2</b>	<b>COMPARAÇÃO ENTRE IMAGENS.....</b>	<b>12</b>
2.2.1	MSE ( <i>Mean Squared Error</i> ) .....	12
2.2.2	SSIM ( <i>Structural Similarity Index Measure</i> ) .....	13
2.2.3	Comparação de histogramas.....	13
<b>2.3</b>	<b>FEATURE MATCHING.....</b>	<b>14</b>
2.3.1	SIFT ( <i>Scale-Invariant Feature Transform</i> ) .....	14
2.3.2	ORB ( <i>Oriented FAST and Rotated BRIEF</i> ).....	15
<b>2.4</b>	<b>OUTRAS TÉCNICAS FUNDAMENTAIS.....</b>	<b>15</b>
<b>2.5</b>	<b>PRODUTOS JÁ UTILIZADOS NO MERCADO.....</b>	<b>16</b>
<b>3</b>	<b>METODOLOGIA CIENTÍFICA .....</b>	<b>17</b>
<b>3.1</b>	<b>IDENTIFICAÇÃO DOS CONTORNOS.....</b>	<b>17</b>
<b>3.2</b>	<b>ANÁLISE DE COR .....</b>	<b>18</b>
<b>3.3</b>	<b>DETECÇÃO DE MANCHAS.....</b>	<b>20</b>
<b>3.4</b>	<b>DETECÇÃO DE DESALINHAMENTO .....</b>	<b>23</b>
<b>3.5</b>	<b>DETECÇÃO DE DESALINHAMENTO DE CORES .....</b>	<b>25</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES.....</b>	<b>26</b>
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>30</b>
	<b>REFERÊNCIAS.....</b>	<b>31</b>
	<b>ANEXO A - FLUXO DE INSPEÇÃO SMARTPQ.....</b>	<b>33</b>

## 1 INTRODUÇÃO

Em um ambiente competitivo, onde a redução de custos e a satisfação das demandas dos clientes são imperativos para a sustentabilidade organizacional, o departamento de qualidade assume um papel central na obtenção desses resultados, impactando diretamente o aumento da lucratividade. A gestão da qualidade consiste em uma abordagem sistemática e contínua, orientada à manutenção e melhoria de elevados padrões, com vistas a otimizar o desempenho de indivíduos, processos, produtos e o ambiente organizacional como um todo. Para tanto, essa gestão emprega um conjunto de filosofias, metodologias e tecnologias rigorosamente selecionadas e implementadas de forma alinhada às necessidades e estratégias da organização.

A utilização da tecnologia, principalmente a automação de processos, reduz a dependência de trabalho manual em atividades repetitivas, o que não apenas minimiza os custos operacionais relacionados à mão de obra, mas também diminui o tempo de execução e melhora a consistência dos resultados. A integração de IA nos fluxos de trabalho permite uma análise preditiva de falhas, identificando oportunidades de manutenção preventiva e evitando paradas não programadas, o que contribui para a redução de custos associados à indisponibilidade de equipamentos.

Neste estudo de caso, será apresentada uma alternativa ao método tradicional de inspeção de qualidade amplamente empregado na atualidade, utilizando tecnologias avançadas, com ênfase na aplicação de visão computacional no contexto do ambiente produtivo de uma indústria gráfica.

Para este estudo, vamos utilizar alguns rótulos de clientes da empresa XYZ, que é uma indústria especializada na produção de rótulos utilizando tanto impressoras digitais quanto impressoras flexográficas. Sua linha de produção abrange a impressão de rótulos adesivos em diferentes substratos, como BOPP branco, metalizado e papel adesivo. A presença de ambos os tipos de tecnologia — digital e flexográfica — em seu parque fabril permite à empresa atender a demandas de diferentes volumes, desde pequenas até grandes tiragens. Além disso, a XYZ oferece flexibilidade em seus processos, realizando a impressão tanto em bobinas quanto em folhas, conforme as especificações e necessidades de seus clientes.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 INSPEÇÃO DE AMOSTRAGEM

A inspeção de qualidade é um processo sistemático que visa garantir que produtos ou serviços atendam aos padrões normativos e técnicos estabelecidos. Este processo, realizado por equipes especializadas, pode envolver verificações visuais, instrumentos de medição ou testes de desempenho, dependendo da natureza do item em análise. Devido aos altos custos de inspecionar 100% dos produtos, as empresas frequentemente adotam a inspeção por amostragem, uma estratégia eficaz para avaliar a conformidade de lotes inteiros com base em uma amostra representativa, permitindo decisões sobre aceitação ou rejeição do lote com base nas características observadas (Montgomery, 2009).

A amostragem pode ser realizada por atributos, que classifica itens como "conformes" ou "não conformes", ou por variáveis, que mede características quantitativas como peso ou dimensões. Parâmetros como o nível de qualidade aceitável (NQA) e a taxa de aceitação são essenciais para definir limites de defeitos permitidos e garantir a eficácia da inspeção. A adoção de normas como a ISO 2859-1 padroniza esses processos, balanceando os custos da inspeção com a manutenção de um nível de qualidade adequado para o consumidor final.

### 2.2 COMPARAÇÃO ENTRE IMAGENS

Existem diversas formas de comparar duas imagens usando *python*, neste trabalho usamos mais especificamente a biblioteca *opencv*, que contém uma série de funcionalidades capazes de resolver grande parte dos problemas de visão computacional. A seguir veremos algumas das opções mais comuns de comparação disponíveis em pesquisas.

#### 2.2.1 MSE (*Mean Squared Error*)

O MSE ou Erro Quadrático Médio é uma métrica estatística usada para medir a diferença entre duas imagens, comparando pixel a pixel e calculando a média dos

quadrados das diferenças entre os valores de intensidade correspondentes. Embora forneça uma medida quantitativa do erro, com valores mais baixos indicando maior similaridade, o MSE não considera aspectos perceptivos da imagem, como a estrutura e a relação espacial dos pixels, tornando-o sensível a pequenas variações que podem não ser perceptíveis ao olho humano. Portanto, essa técnica pode não ser vantajosa quando se comparam imagens digitais e digitalizadas, devido às suas diferenças em nível de pixel (Wang & Bovik, 2009).

### 2.2.2 SSIM (*Structural Similarity Index Measure*)

O SSIM é uma métrica amplamente utilizada para avaliar a similaridade entre duas imagens, com foco na percepção de qualidade visual. Diferente de métodos tradicionais como o MSE (*Mean Squared Error*) ou o PSNR (*Peak Signal-to-Noise Ratio*), que medem a diferença ponto a ponto entre os valores de pixel, o SSIM busca modelar a maneira como os humanos percebem a qualidade de uma imagem, levando em consideração aspectos como luminância, contraste e estrutura (Wang et al., 2004). Para este estudo, esta técnica consegue obter resultados melhores que utilizando a técnica MSE.

O SSIM é calculado localmente, em pequenas janelas que percorrem a imagem, o que permite uma avaliação mais robusta de variações em regiões específicas. O índice é dado por três componentes: (1) Luminância, que avalia a diferença de brilho entre as imagens, (2) Contraste, que mede as variações de intensidade, e (3) Estrutura, que compara as correlações entre padrões de pixels nas imagens. A métrica resultante varia entre -1 e 1, onde 1 indica que as duas imagens são idênticas, e valores próximos de 0 ou negativos indicam grande dissimilaridade (Wang, Bovik, Sheikh, & Simoncelli, 2004).

### 2.2.3 Comparação de histogramas

A comparação de histogramas é uma técnica amplamente utilizada para medir a similaridade entre imagens, especificamente no que diz respeito à distribuição das cores. Um histograma de uma imagem representa a distribuição de intensidade de cores para cada canal de cor (vermelho, verde e azul no caso de imagens RGB). Cada

canal gera um histograma que mostra a frequência de cada valor de intensidade, o que permite uma análise detalhada da composição de cores da imagem. Ao comparar dois histogramas, é possível determinar o quão semelhantes as imagens são em termos de cor, independentemente de pequenas variações estruturais (Gonzalez & Woods, 2008).

No processo de comparação de histogramas, um método comum é o uso de métricas como a correlação, a distância de Bhattacharyya ou a interseção de histogramas. A correlação mede a relação linear entre os histogramas das imagens, enquanto a interseção quantifica o grau de sobreposição entre eles. A distância de Bhattacharyya, por sua vez, mede a semelhança entre duas distribuições de probabilidade e é utilizada para identificar a proximidade entre os histogramas. Essas métricas retornam um valor numérico que indica a semelhança das imagens, sendo que valores mais próximos de 1 indicam maior similaridade (Zitová & Flusser, 2003).

## 2.3 FEATURE MATCHING

O feature matching é uma técnica fundamental em visão computacional que visa identificar correspondências entre pontos de interesse em diferentes imagens. Dois dos algoritmos mais utilizados para essa tarefa são o SIFT (*Scale-Invariant Feature Transform*) e o ORB (*Oriented FAST and Rotated BRIEF*), cada um com características distintas e aplicações em diferentes contextos.

### 2.3.1 SIFT (*Scale-Invariant Feature Transform*)

O SIFT, introduzido por David Lowe em 2004, é um dos métodos mais robustos para detecção e descrição de características locais. Ele opera em quatro etapas principais: detecção de pontos-chave, refinamento da localização, atribuição de orientação e descrição das características. A detecção de pontos-chave ocorre através da aplicação de uma série de convoluções Gaussianas na imagem para detectar extremos na escala de pirâmides, o que permite identificar características invariantes a escala e rotação (Lowe, 2004). O algoritmo também é robusto a variações de iluminação e transformações não lineares, o que o torna especialmente útil em cenários de correspondência de imagens com alterações significativas. Cada

ponto de interesse é descrito por um vetor de 128 dimensões, representando a distribuição de gradientes locais ao redor do ponto chave.

No entanto, apesar de sua eficácia, o SIFT é computacionalmente caro. O processo de detecção de extremos em múltiplas escalas e o cálculo das descrições de 128 dimensões demandam um tempo considerável de processamento, o que pode ser proibitivo em sistemas que exigem alta performance em tempo real.

### 2.3.2 ORB (*Oriented FAST and Rotated BRIEF*)

Como alternativa ao SIFT, o ORB foi desenvolvido como um método mais eficiente em termos de tempo de execução, especialmente para sistemas embarcados e de tempo real. O ORB combina o detector FAST (*Features from Accelerated Segment Test*) com o descritor BRIEF (*Binary Robust Independent Elementary Features*), resultando em um algoritmo rápido e eficaz. O FAST é utilizado para detectar os pontos-chave, sendo um detector simples que compara intensidades de pixels ao redor de um ponto, enquanto o BRIEF cria descritores binários baseados em testes de intensidade de pares de pixels (Rublee et al., 2011). Para garantir invariância a rotação, o ORB introduz uma etapa de atribuição de orientação aos pontos-chave, utilizando o gradiente da intensidade local.

O ORB é significativamente mais rápido que o SIFT, com desempenho adequado para muitas aplicações em tempo real, como navegação autônoma e rastreamento de objetos. Entretanto, sua precisão pode ser inferior em cenários onde mudanças severas de iluminação ou transformações geométricas complexas estão presentes, uma vez que seus descritores binários são menos robustos a essas variações quando comparados ao vetor de gradientes do SIFT.

## 2.4 OUTRAS TÉCNICAS FUNDAMENTAIS

A manipulação de imagens envolve diversas técnicas utilizadas para processar e analisar características visuais de uma imagem, com o objetivo de extrair informações ou melhorar sua qualidade. Entre as técnicas mais comuns está o thresholding, ou limiarização, que consiste em converter uma imagem em escala de cinza em uma imagem binária, separando os pixels em duas categorias, conforme um



valor de limiar. Essa técnica é amplamente usada para segmentar objetos de interesse em imagens, destacando áreas de alta intensidade ou contrastes relevantes. Um dos métodos mais utilizados é o *Thresholding* de Otsu, que ajusta automaticamente o valor de limiar para maximizar a separação entre as classes de pixels (Gonzalez & Woods, 2008).

Outra técnica fundamental é a detecção de contornos, que visa identificar as bordas de objetos dentro de uma imagem. A detecção de contornos é frequentemente aplicada após a binarização ou suavização da imagem, sendo o algoritmo de Canny um dos mais eficazes, pois utiliza um processo de derivada para identificar gradientes de intensidade que correspondem às bordas. Os contornos extraídos podem ser utilizados para reconhecimento de formas, detecção de objetos e análise de movimento (Szeliski, 2010).

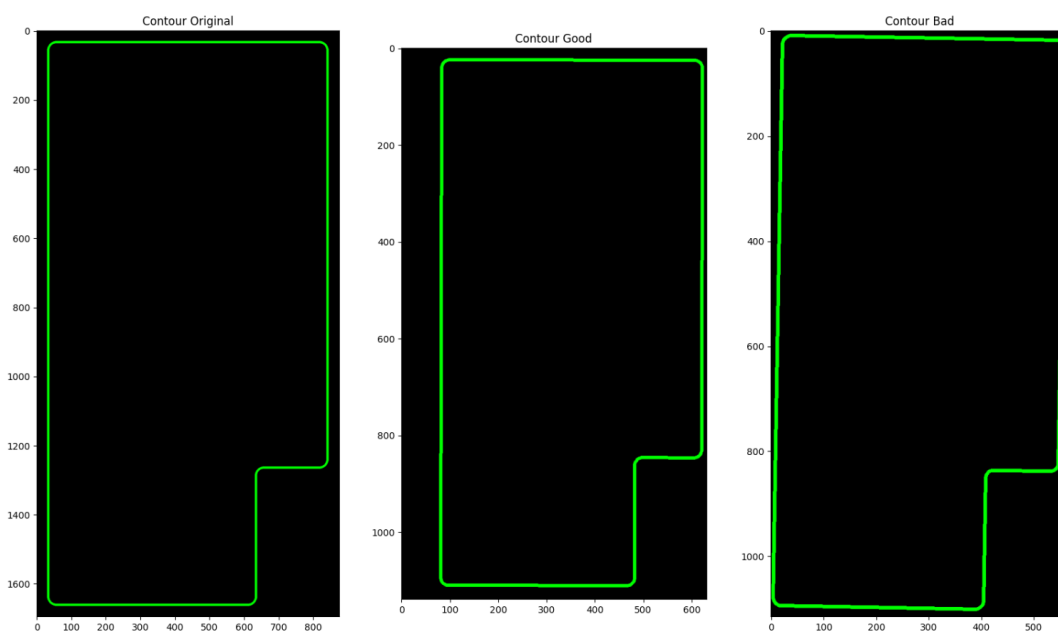
## 2.5 PRODUTOS JÁ UTILIZADOS NO MERCADO

Além dos resultados encontrados acima, foram identificados também diversos produtos de empresas que atuam com o mesmo propósito deste trabalho. Um deles é o produto da empresa Lake Image, onde foi desenvolvido um sistema de detecção de falhas de impressão em tempo real tanto de defeitos aleatórios quanto de defeitos recorrentes, podendo agir até preventivamente, ao perceber que uma falha esteja chegando à níveis que ultrapassem a qualidade desejada. O fluxo pode ser encontrado no ANEXO A.

Fonte: Autor.

A técnica aplicada para identificar os contornos nos rótulos impressos combinou SIFT (*Scale-Invariant Feature Transform*) com a função *BFMatcher* da biblioteca OpenCV. O SIFT detecta os principais pontos das três imagens, mesmo com variações de rotação e escala, algo inviável em uma comparação pixel a pixel, que exige dimensões idênticas. A terceira imagem foi propositalmente rotacionada para testar o script, e após a extração dos pontos, o *BFMatcher* sobrepôs esses pontos, reproduzindo o contorno original nas imagens impressas, como ilustrado a seguir.

FIGURA 2. Contornos identificados nos rótulos impressos a partir do contorno da imagem original.



Fonte: autor.

Agora que o contorno das imagens foi identificado, é possível fazer a análise das características principais dentro destes limites.

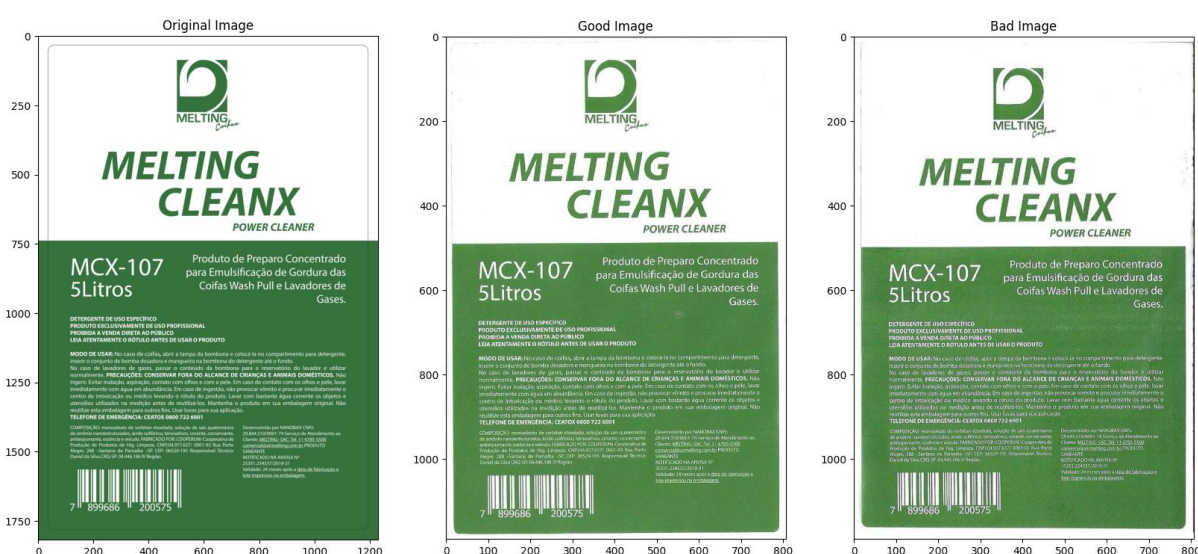
### 3.2 ANÁLISE DE COR

Para a análise e comparação da cor do rótulo, foram escolhidas duas métricas principais que vão verificar se a imagem está classificada como aprovada ou rejeitada:

o valor médio da intensidade da cor (varia de 0 a 255, para medição de tonalidade) e o valor do desvio padrão (para medição de ruído).

O rótulo ilustrado anteriormente não apresenta cores significativas para ilustrar a aplicação do script, portanto, vamos testar com a imagem abaixo. Com ela podemos ver também dois rótulos cuja cor foi inspecionada visualmente e uma foi considerada boa e outra ruim:

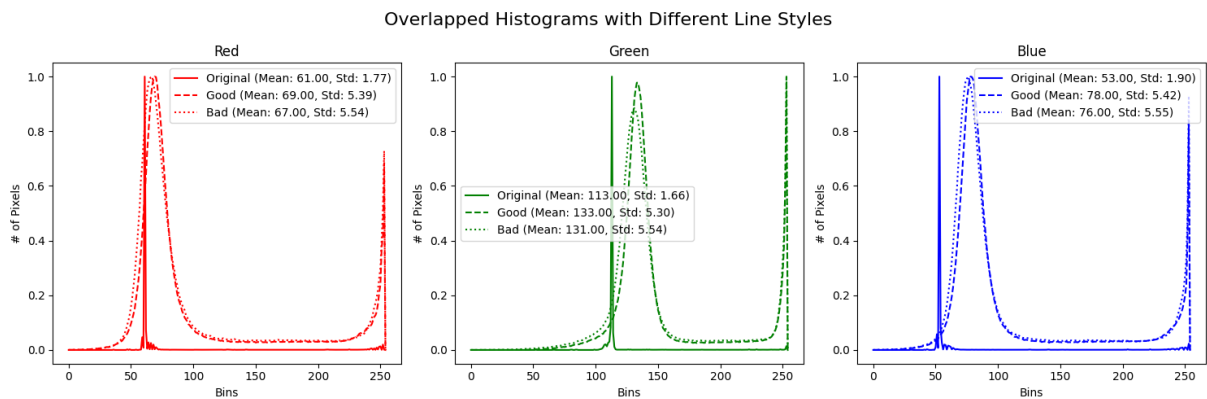
FIGURA 3. Exemplo de rótulo original com cor para demonstração do código; Rótulo “bom”; Rótulo “ruim”.



Fonte: autor.

A metodologia aqui utilizada foi fazer a separação dos três canais da imagem e traçar o histograma de cores para que os picos sejam identificados e a comparação seja com base nestes picos, pois o pico do histograma de cores revela a cor a ser analisada. No caso do exemplo acima, temos apenas um pico, pois há uma única cor predominante, porém se houvessem duas cores, teríamos dois picos e a análise teria que ser feita para cada uma das cores. A exceção para os picos é quando os valores vão para os extremos, ou seja, 0 e 255, uma vez que a análise de cor não será feita para branco e preto.

FIGURA 4. Histogramas de cada canal de cor, para cada uma das três imagens mostradas anteriormente.



Fonte: autor.

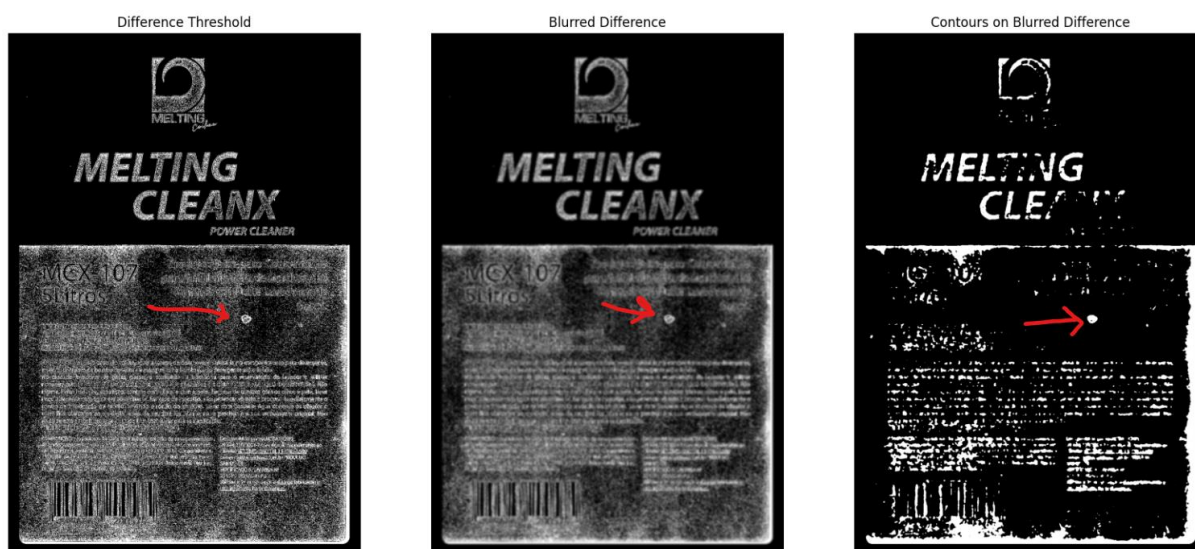
Para cada canal, foi gerado um histograma com valores normalizados, onde se plotaram os três histogramas para os respectivos canais, incluindo médias e desvios padrões do eixo x. Analisando o canal verde, predominante no rótulo, o histograma da imagem ruim está mais próximo do original que o da imagem boa, indicando que a tonalidade média está mais alinhada com o ideal. No entanto, considerando também o desvio padrão, a imagem ruim apresentou maior variância (5.54 contra 5.30), indicando mais ruído e justificando sua classificação como defeituosa.

Há também outras técnicas para identificar o ruído da imagem, como por exemplo SNR, *Blurred Difference*, *Shannon Entropy*, porém como todas chegam num resultado próximo, a opção escolhida se manteve o desvio padrão.

### 3.3 DETECÇÃO DE MANCHAS

O objetivo foi identificar qualquer diferença pontual na imagem, como manchas, amassados, dobras e borrões. A metodologia combinou técnicas eficazes conforme os parâmetros de entrada. Para detectar manchas, a diferença absoluta entre a imagem original e a testada é calculada, seguida por um *blur* e *threshold* para identificar contornos, permitindo a detecção de manchas ou borrões.

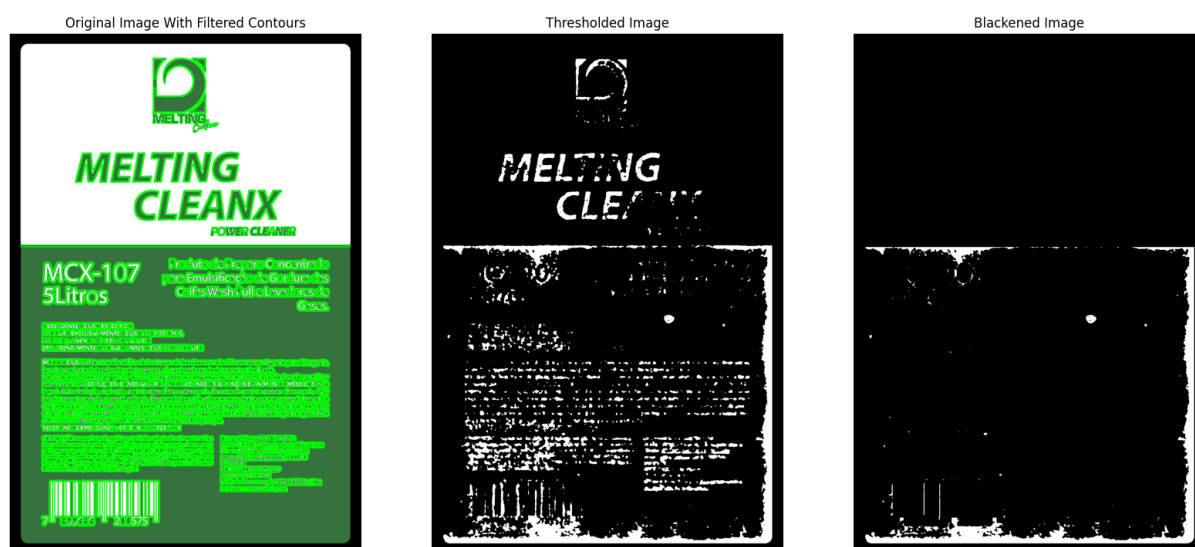
FIGURA 5. Diferença entre a imagem original e digitalizada; Diferença com *blur* e diferença com *threshold*.



Fonte: autor.

É aplicado um *Adaptive Threshold* na imagem original para identificar contornos, seguidos de uma dilatação. A dilatação dos contornos garante que o conteúdo da imagem original dentro dessas áreas seja ocultado na imagem *thresholded*, gerando uma imagem de diferença filtrada, onde os contornos originais são "apagados" para destacar apenas possíveis falhas.

FIGURA 6. Imagem original com contornos identificados; Diferença com *threshold* aplicado; Diferença filtrada.

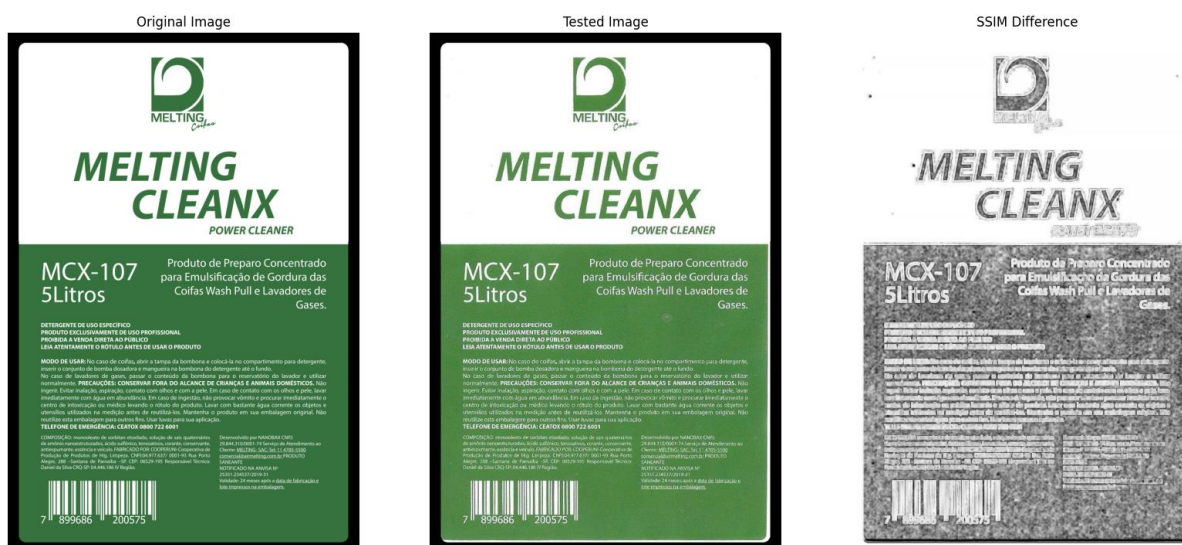


Fonte: autor.



Para pegar somente as manchas de fato, vamos utilizar a técnica SSIM (*Structural Similarity Index*), que considera a similaridade estrutural de duas imagens. Com isso podemos também evidenciar diferenças entre as duas imagens. A intenção é verificar se as mesmas diferenças encontradas por esta técnica, são encontradas na imagem filtrada explicada anteriormente.

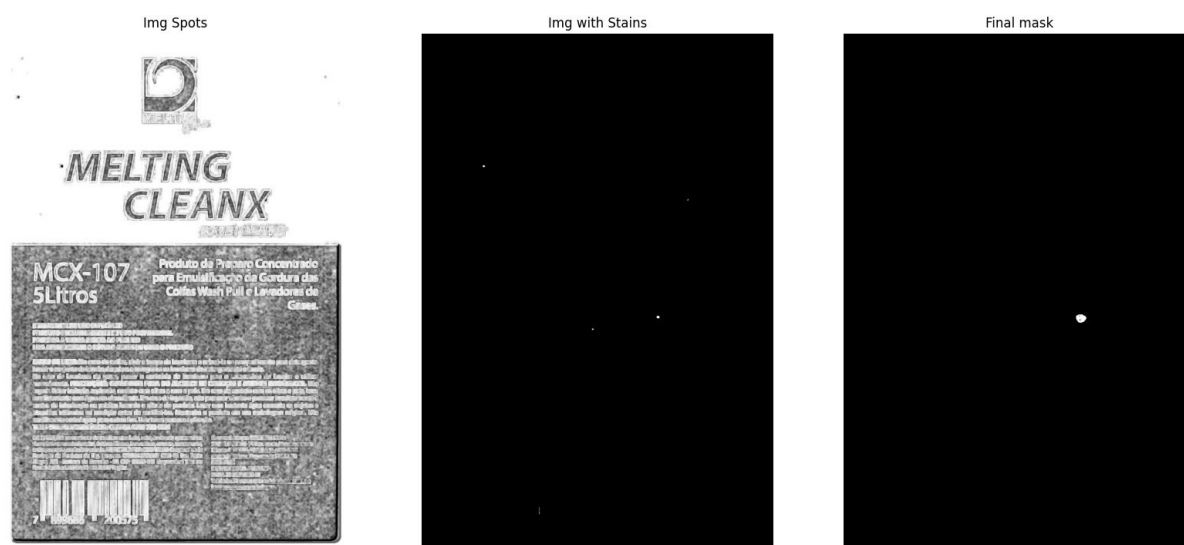
FIGURA 7. Imagem original; Imagem digitalizada; Diferença SSIM.



Fonte: autor.

Para isso, pegamos os top 5 contornos de maior área na imagem gerada pelo SSIM e verificamos se os centroides de seus contornos também estão presentes na imagem filtrada. Caso estejam presentes, os mesmos são expostos e considerados detectados. Conforme ilustra a imagem abaixo:

FIGURA 8. Diferença SSIM; Top 5 contornos detectados; Defeitos detectados.



Fonte: autor.

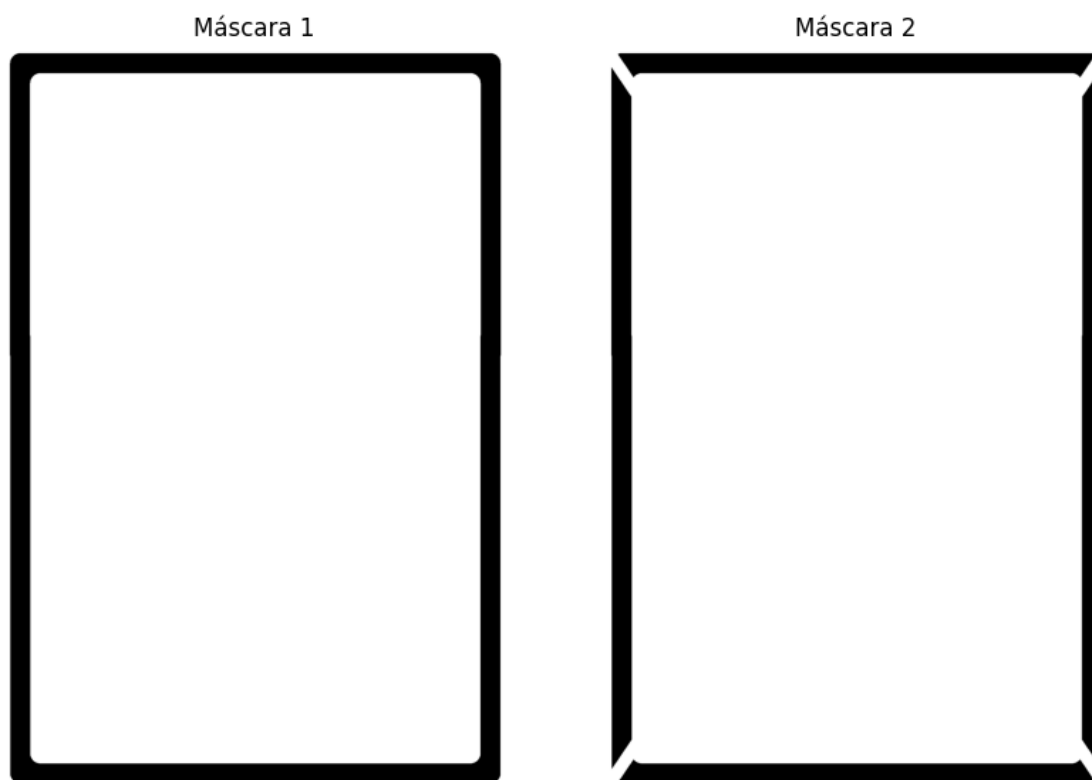
### 3.4 DETECÇÃO DE DESALINHAMENTO

O princípio para a detecção de desalinhamento baseia-se na sobreposição da imagem original à imagem testada. A diferença e a soma entre as imagens revelam áreas de desalinhamento nas bordas. Com isso, é possível calcular a espessura do desalinhamento a partir da análise da diferença entre ambas.

Foi criada uma máscara ao redor das bordas da imagem, com a intenção de pegar somente estas diferenças de alinhamento. Depois esta máscara é separada em quatro partes representando os lados. Para cada lado, foram detectados os contornos que representam as bordas. O valor da espessura, tanto para o desalinhamento vertical quanto para o desalinhamento horizontal pode ser encontrado com a função *Bounding Box*, em que o contorno é encapsulado em um retângulo, sendo possível retornar a largura e altura desse retângulo.

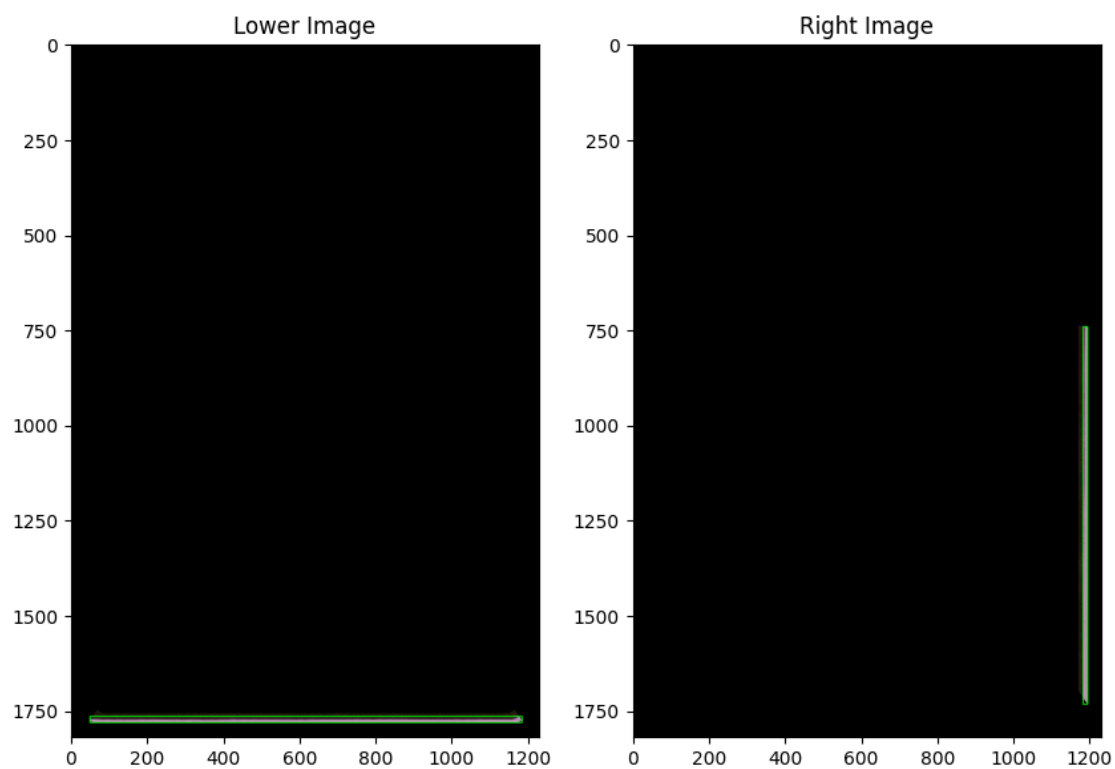


Figura 9. Máscara criada inicialmente; Máscara cortada para os quatro lados.



Fonte: autor.

FIGURA 10. Bounding box dos contornos de desalinhamento identificados.



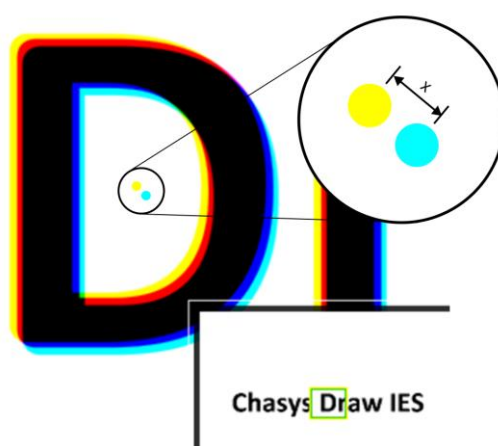
Fonte: autor.

### 3.5 DETECÇÃO DE DESALINHAMENTO DE CORES

Um dos possíveis erros de impressão, é que pode haver um desalinhamento entre a impressão das cores, conforme ilustra a FIGURA 11.

A metodologia escolhida para este caso foi separar a imagem nos três canais de cores e fazer a varredura em cada um dos canais para identificar os contornos. Para cada contorno, o centroide é encontrado e, a partir do encontro dos centroides, é possível calcular a diferença entre os centroides do mesmo contorno para cada canal. Resultando nos valores da diferença entre: vermelho e verde, vermelho e azul, e verde e azul.

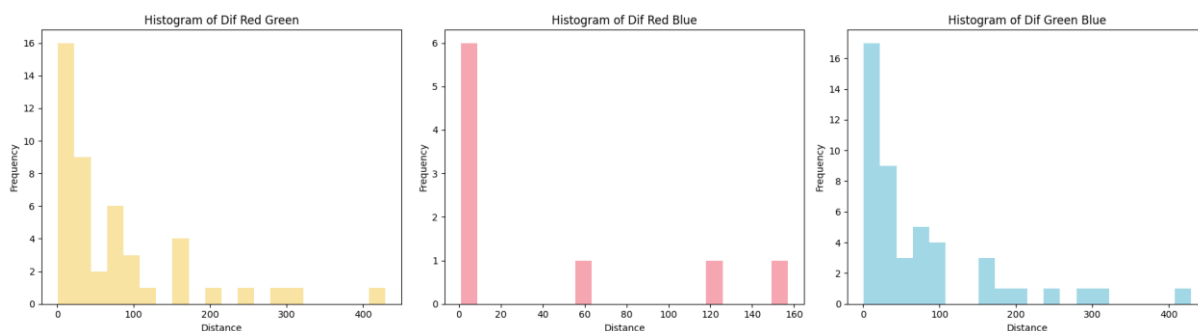
FIGURA 11. Diferença entre dois centroides de contornos de canais diferentes.



Fonte: autor.

Uma visão macro pode ser feita ao plotar o histograma das diferenças entre cada um dos canais, conforme a figura abaixo:

FIGURA 12. Distribuição da diferença entre áreas de cada par de cores.



Fonte: autor.

## 4 RESULTADOS E DISCUSSÕES

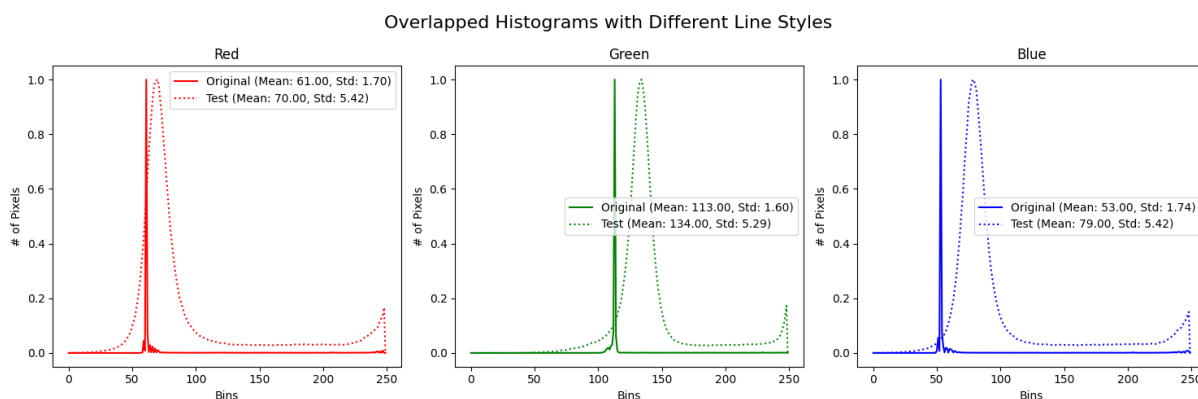
Uma vez que não estamos lidando com um modelo de machine learning onde seria possível treiná-lo, e sim uma combinação de técnicas de visão computacional clássica, os *inputs* de *setup* são inevitáveis. Caso contrário, poderíamos utilizar uma série de rótulos considerados “bons” e outros “ruins” para treinar o modelo e teríamos em uma única solução a identificação de falha de cada um dos pontos que estamos analisando separadamente: cor, manchas, desalinhamento e desalinhamento de cores.

Esta técnica não foi a escolhida pois há poucos rótulos que são produzidos com frequência, a maioria deles são trabalhos enviados esporadicamente e seria impossível fazer o treino e teste destes casos.

Portanto, seguindo desta forma, talvez seja necessário ajustar os parâmetros do script no setup para que ele rode adequadamente antes de cada trabalho enviado ao processo produtivo.

Para a análise dos resultados, a inspeção de cor é feita de forma diferente das outras. Uma vez que existem uma série de condições que justificam a diferença de cor da imagem original para a impressa analisada, se simplesmente compararmos a tonalidade de ambas, mesmo que o rótulo seja considerado “bom”, há uma distância significativa entre elas, conforme mostra a figura abaixo. A influência do substrato, se há brilho ou não, da luminosidade e muitos outros fatores, tornam impossível a validação somente pela comparação bruta da cor dos pixels. A alternativa tomada para este caso foi realizar a impressão de um rótulo “bom” antes de passar o lote para produção, pois uma vez inspecionado visualmente e aprovado, ele pode servir de parâmetro para os próximos. E no plano de inspeção, os próximos rótulos podem ser considerados aprovados caso a média e o desvio padrão estejam dentro de um range previamente definido, como por exemplo: mais ou menos 5% dos valores da imagem de setup aprovada.

FIGURA 13. Histogramas dos canais RGB comparando a imagem escaneada.



Fonte: autor.

Para cada um dos pontos analisados, uma tabela de resultados é gerada, e posteriormente comparada com o plano de inspeção realizado para o produto. A seguir podemos ver como é o resultado gerado para cada item.

Para a inspeção da cor, como vimos anteriormente, utilizamos o indicador da média para tonalidade e desvio padrão para ruído, portanto, temos duas colunas para cada cor. E cada linha representa um pico (ou cor predominante). Ao final, há uma coluna de tipo para identificar qual a imagem está sendo analisada para posterior comparação.

TABELA 1. Resultados da análise de cor.

Id	Mean R	Std Dev R	Mean G	Std Dev G	Mean B	Std Dev B	Type
Peak 1	61	1.704.349	113	1.599.839	53	1.735.274	Original
Peak 1	69	5.396.311	134	5.290.180	79	5.424.758	Good
Peak 1	70	5.418.507	134	5.289.951	79	5.424.332	Test

Fonte: autor.

Diferente da inspeção de cor, os outros itens analisados não necessitam de uma imagem de setup para redefinir os padrões de aprovado e rejeitado. Basta que sejam definidas as tolerâncias de cada uma das medições, e, caso estejam dentro do limite, são consideradas aprovadas.

Para a inspeção de manchas, é gerada uma tabela onde cada linha representa um contorno, e temos as propriedades de área, perímetro, percentual com relação à imagem completa e também as coordenadas do centroide:

TABELA 2. Resultados da análise de detecção de manchas.

Id	Contour	Area	Perimeter	Percent	Centroid
Contour 1	[[[823, 984]], [[822, 985]], [[820, 985...]]	750.0	104.911.687	0.033522	(829, 997)

Fonte: autor.

Para a inspeção de desalinhamento, é gerada uma tabela contendo as medidas do contorno para cada metade analisada. Na maioria dos casos será possível descartar duas linhas desta tabela, pois o contorno será de área insignificante, ou seja, a imagem desalinhada estará sempre deslocada verticalmente ou horizontalmente, logo, uma das metades será descartada. Os valores de *w* e *h* representam respectivamente, os valores de largura e altura do *bounding box*.

TABELA 3. Resultados da análise de desalinhamento.

Local	Area	Perimeter	w	h
Up	1	0.000000	1	1
Bottom	11070	2.379.210.238	1107	10
Left	1	0.000000	1	1
Right	11844	2.047.019.333	12	987

Fonte: autor.

Por fim temos a inspeção de desalinhamento de cores, onde é gerada uma tabela inicial conforme esta abaixo, contendo a listagem dos contornos, a cor do canal deste contorno, as coordenadas do centroide e sua área.

TABELA 4. Resultados da análise de desalinhamento de cores.

Contours	Original Contour	Category	Centroid	Area
Contour 1	[[[721, 1766]], [[721, 1776]], [[732, 1776]], ...]]	Red	(726, 1769)	77.0
Contour 2	[[[500, 1766]], [[500, 1776]], [[510, 1776]], ...]]	Red	(505, 1769)	70.0
Contour 3	[[[1167, 1756]], [[1166, 1757]], [[1166, 1767]]...]]	Red	(1170, 1760)	91.0
Contour 4	[[[1169, 1371]], [[1169, 1382]], [[1180, 1382]]...]]	Red	(1174, 1376)	121.0
Contour 5	[[[1173, 1362]], [[1173, 1374]], [[1183, 1374]]...]]	Red	(1176, 1368)	86.5

Fonte: autor.

A partir da tabela acima é gerada uma outra, onde vemos a distância entre os centroides de cada par de cores e a diferença entre as áreas de cada par de cores.

TABELA 5. Resultados da análise de desalinhamento de cores versão final.

Contours	Original Contour	Category	Dif	Euclidean Distance
Contour 33	[[[447, 416]], [[447, 418]], [[446, 419]], [[4...	Red-Green	81.5	429.004662
Contour 33	[[[447, 416]], [[447, 418]], [[446, 419]], [[4...	Red-Blue	68.5	429.004662
Contour 35	[[[317, 416]], [[316, 417]], [[316, 418]], [[3...	Red-Green	233.0	313.969744
Contour 35	[[[317, 416]], [[316, 417]], [[316, 418]], [[3...	Red-Blue	207.5	313.969744
Contour 39	[[[640, 334]], [[640, 347]], [[641, 348]], [[6...	Red-Green	5.0	294.006803

Fonte: autor.

## 5 CONCLUSÃO

Uma vez que o *software* não consiste em um modelo de machine learning, para que seja possível avaliar a sua acurácia o mesmo deve ser submetido à uma comparação com uma inspeção real, para avaliar se, de fato, as características medidas batem com a inspeção realizada pela equipe da qualidade. Caso as medições estejam dentro de um limite aceitável de erro com relação ao feito pelos inspetores, basta comparar os dados gerados por cada item inspecionado com o plano de inspeção para assim obter um valor de acurácia.

Ao testar o código com rótulos que possuem um maior espectro de cores, foi identificada uma baixa performance no tempo de execução. Além disso, caso o rótulo possua texturas nas cores, elas podem ser entendidas como defeitos e potencialmente classificar as imagens como falso positivas.

Abaixo seguem algumas sugestões para melhorias e próximos passos seguintes à este projeto:

- Validação do script com rótulos de diferentes formatos e mais variação de cores.
- Estudo para melhora de performance do código;
- A possível construção de um *data lake* para armazenamento das imagens coletadas;
- Criação de uma interface ou aplicativo que seja de fácil manuseio para o operador, tanto na alteração de parâmetros quanto para a identificação dos lotes inspecionados;
- Integração com sensores sonoros que permitem avisar a produção de uma falha encontrada;
- Criação de um dashboard para monitorar a produção em tempo real.

## REFERÊNCIAS

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. New Jersey: Prentice Hall, 2008. Disponível em: <https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf>. Acesso em junho de 2024.

ISO 2859-1:1999. **Procedimentos de amostragem para inspeção por atributos**. International Organization for Standardization, 1999. Disponível em: <https://qualidadeonline.wordpress.com/2014/03/10/planos-de-amostragem-e-inspecao-por-atributos/>. Acesso em julho de 2024.

LAKE IMAGE. **Automatic Fast 100% Print Quality Inspection with Comprehensive HD Web Viewing**. Disponível em: <https://lakeimage.com/wp-content/uploads/2023/11/Discovery-SmartPQ-Nov2023.pdf>. Acesso em maio de 2024.

LOWE, D. G. **Distinctive image features from scale-invariant keypoints**. International Journal of Computer Vision, 2004. Disponível em: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>. Acesso em junho de 2024.

MONTGOMERY, D. C. **Introdução ao controle estatístico da qualidade**. Rio de Janeiro: LTC, 2009.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. **ORB: An efficient alternative to SIFT or SURF**. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2011. Disponível em: [https://www.researchgate.net/publication/221111151\\_ORB\\_an\\_efficient\\_alternative\\_to\\_SIFT\\_or\\_SURF](https://www.researchgate.net/publication/221111151_ORB_an_efficient_alternative_to_SIFT_or_SURF). Acesso em julho de 2024.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. London: Springer, 2010. Disponível em: <https://szeliski.org/Book/download>. Acesso em junho de 2024.



ZITOVÁ, B.; FLUSSER, J. **Image registration methods: a survey**. Image and Vision Computing, 2003. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0262885603001379>. Acesso em julho de 2024.

## ANEXO A - FLUXO DE INSPEÇÃO SMARTPQ

