

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



GENTIL AUER NETO

**UM ESTUDO SOBRE DETECÇÃO E CLASSIFICAÇÃO DE
DEFEITOS EM TIRAS DE AÇO VIA PROCESSAMENTO
DIGITAL DE IMAGENS**

**VITÓRIA – ES
DEZEMBRO/2014**

GENTIL AUER NETO

**UM ESTUDO SOBRE DETECÇÃO E CLASSIFICAÇÃO DE DEFEITOS
EM TIRAS DE AÇO VIA PROCESSAMENTO DIGITAL DE IMAGENS**

Parte manuscrita do Projeto de Graduação do aluno **Gentil Auer Neto**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Evandro Otoni Teatini
Salles

Coorientador: Eng. Philipe Rangel Demuth

VITÓRIA – ES
DEZEMBRO/2014

GENTIL AUER NETO

UM ESTUDO SOBRE DETECÇÃO E CLASSIFICAÇÃO DE DEFEITOS EM TIRAS DE AÇO VIA PROCESSAMENTO DIGITAL DE IMAGENS

Parte manuscrita do Projeto de Graduação do aluno **Gentil Auer Neto**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

COMISSÃO EXAMINADORA:

Prof. Dr. Evandro Otoni Teatini Salles
Universidade Federal do Espírito Santo - UFES
Orientador

Eng. Philipe Rangel Demuth
Universidade Federal do Espírito Santo - UFES
Coorientador

Prof. Dr. André Ferreira
Universidade Federal do Espírito Santo - UFES
Examinador

Prof. Dr. Patrick Marques Ciarelli
Universidade Federal do Espírito Santo - UFES
Examinador

À Sarah.
À Maria Ângela, Paulo e Lorena

Agradeço primeiramente a Deus, pelo sustento e iluminação. Aos meus familiares, em especial meus pais Paulo e Maria Ângela, e minha irmã Lorena, por todo suporte. À Sarah, companheira incrível, pelo amor, carinho, e fonte de incentivo. Ao meu orientador Prof. Dr. Evandro Ottoni Teatini Salles e ao meu coorientador Eng. Philipe Rangel Demuth pelo conhecimento compartilhado e pela excelência como profissionais e pessoas.

RESUMO

O Brasil abriga diversas siderúrgicas de grande porte, possuindo o maior parque industrial de aço da América do Sul, além de ter concluído o ano de 2013 com o nono lugar em produção de aço no mundo (INSTITUTO AÇO BRASIL, 2014). Nesse contexto, existem diversos produtos diferentes, dentre os quais estão as tiras de aço laminadas a quente. A qualidade final dessas tiras de aço é avaliada tanto em laboratórios, realizando-se testes mecânicos, quanto visualmente por especialistas ou por sistemas computacionais.

Este trabalho consiste no desenvolvimento de técnicas para detecção automática de defeitos em tiras de aço laminadas a quente, por meio de processamento digital de imagens. A primeira etapa foi a seleção das imagens a serem utilizadas para testes e validação. Foram escolhidos três defeitos: Arranhão Claro, Carepa e Esfoliação. As imagens foram separadas em dois grupos: o primeiro com exemplos de defeitos isolados para estudo de características e utilização como *templates* para testes de correlação; o segundo contendo imagens completas. Foram testados métodos estatísticos para análise das características das imagens, o que resultou em um algoritmo capaz de detectar múltiplos defeitos em cada imagem das tiras, além de contar com um ajuste de sensibilidade de detecção. Posteriormente, foi criado um classificador baseado em árvore de decisão, resultando em um índice de acerto entre 82% e 94% na classificação dos defeitos testados.

Palavras-chave: Processamento de imagens, laminação, detecção de falhas, classificação de imagens, árvore de decisão.

LISTA DE FIGURAS

Figura 1 – Cadeira de laminação	14
Figura 2 – Sistema de aquisição de imagens	15
Figura 3 – Exemplos de imagens de tiras laminadas a quente	16
Figura 4 – Exemplos de produtos/serviços em que utiliza-se processamento de imagens.....	20
Figura 5 – Exemplo de grid de uma imagem digital, estruturada em Linhas e Colunas	21
Figura 6 – (a)Vizinhança-4 de um pixel (b)Vizinhança-8 de um pixel.....	22
Figura 7 – Gráfico com ocorrências de níveis de cinza e <i>threshold</i> para limiarização	24
Figura 8 – (a) Regiões de intensidade constante separadas por uma borda vertical (b) Perfil horizontal da região de transição, juntamente com o sinal de resposta da primeira e da segunda derivadas.....	25
Figura 9 – Função de densidade de probabilidade com forma de ‘sino’	26
Figura 10 – Histograma de dados distribuídos desregularmente.....	27
Figura 11 – Árvore de decisão.....	30
Figura 12 – Diferentes defeitos encontrados em tiras de aço	31
Figura 13 – Tiras de aço com iluminação descontínua.....	32
Figura 14 –Tira de aço com iluminação uniforme.....	33
Figura 15 – Imagens de borda lateral	33
Figura 16 – Imagem contendo defeito tipo ARC.....	34
Figura 17 – Imagem contendo defeito tipo CA	34
Figura 18 – Imagem contendo defeito tipo ESF	35
Figura 19 – Defeito tipo ARC encontrado via <i>template matching</i>	37
Figura 20 – Defeito tipo CA encontrado via <i>template matching</i>	37
Figura 21 – Defeito tipo <i>ESF</i> encontrado via <i>template matching</i>	38
Figura 22 – Falsos positivos encontrados para defeito tipo ARC	39
Figura 23 – Falsos positivos encontrados para defeito tipo CA	39
Figura 24 – Falsos positivos encontrados para defeito tipo ESF.....	40
Figura 25 – Resultado correto na busca por defeitos ARC utilizando imagem contendo CA	41
Figura 26 – Erro de classificação.....	41
Figura 27 – Mapeamento dos valores de correlação entre templates <i>ARC</i> e imagens <i>ARC</i> ...	42
Figura 28 – Mapeamento dos valores de correlação entre templates <i>CA</i> e imagens <i>ESF</i>	43
Figura 29 – Resultado para o par <i>template CA_10 - imagem ESF_24</i>	43
Figura 30 – Resultado para o par <i>template CA_24 - imagem ESF_27</i>	44

Figura 31 – Defeito tipo ARC	45
Figura 32 – Descontinuidade no nível de cinza entre colunas de imagem com defeito tipo ARC	45
Figura 33 – Visão 3D de imagem contendo defeito tipo ARC.....	46
Figura 34 – Média do nível de cinza por coluna	47
Figura 35 – (a) Imagem com defeitos tipo ARC. (b) Gráfico com valores de média de nível de cinza por coluna	48
Figura 36 – (a) Imagem com defeitos tipo CA (b) Gráfico com valores de média de nível de cinza por coluna.....	49
Figura 37 - Imagem com defeito tipo ESF (b) Gráfico com valores de média de nível de cinza por coluna.....	49
Figura 38 – (a) Imagem com iluminação uniforme (b) Gráfico com máximo local igual ao máximo global	50
Figura 39 – (a) Imagem com iluminação descontínua (b) Gráfico com máximo local mascarado pelo máximo global	51
Figura 40 – (a) Imagem com defeitos tipo ARC (b) Gráfico da derivada da média do nível de cinza por coluna	52
Figura 41 – Defeito tipo ARC localizado automaticamente.....	53
Figura 42 – Defeito tipo CA localizado automaticamente	53
Figura 43 – Defeito tipo ESF localizado automaticamente	54
Figura 44 – (a) Apenas um defeito é localizado automaticamente na imagem. (b) Gráfico exibindo vários picos de derivada de média do nível de cinza.....	54
Figura 45 – Histograma e função de densidade de probabilidade para imagens contendo defeito tipo ARC.....	55
Figura 46 – Histograma e função de densidade de probabilidade para imagens contendo defeitos tipo CA.....	56
Figura 47 – Histograma e função de densidade de probabilidade para imagens contendo defeitos tipo ESF.....	57
Figura 48 – Detecção de múltiplos defeitos tipo ARC	60
Figura 49 – Detecção de múltiplos defeitos tipo CA.....	60
Figura 50 – Detecção de múltiplos defeitos tipo ESF	61
Figura 51 – Comparativo de sensibilidade do algoritmo para defeitos tipo ARC.....	61
Figura 52 – Comparativo de sensibilidade do algoritmo para defeitos tipo CA	62
Figura 53 –Comparativo de sensibilidade do algoritmo para defeitos tipo ESF	62

Figura 54 – (a)Defeito ARC (b)Defeito ESF	63
Figura 55 – (a)Imagem contendo defeito CA (b)Histograma da imagem com defeito CA (c)Resultado da binarização pelo método de Otsu	64
Figura 56 – (a)Defeitos tipo CA (b)Defeitos tipo ESF	65
Figura 57 – (a)Defeitos CA binarizados (b)Defeitos ESF binarizados	65
Figura 58 – Carepa binarizada e detectada para cálculo de atributos.....	66
Figura 59 – Esfoliações binarizadas e detectadas para cálculo de atributos.....	66
Figura 60 – Árvore de decisão utilizada para classificação dos defeitos	67
Figura 61 – Resultado do classificador para imagem contendo ARC	68
Figura 62 – Resultado do classificador para imagem contendo CA.....	68
Figura 63 – Resultado do classificador para imagem contendo ESF	69

LISTA DE TABELAS

Tabela 1 – Valores de média dos dados para todas as imagens utilizadas durante os testes..	58
Tabela 2 – Resultados de classificação dos defeitos.	69

LISTA DE ABREVIATURAS E SIGLAS

ARC	Arranhão Claro
CA	Carepa Primária
ESF	Esfoliação
UFES	Universidade Federal do Espírito Santo
LCEE	Laboratório de Computação da Engenharia Elétrica
MATLAB	MATrix LABoratory
ROI	Region of interest

LISTA DE SÍMBOLOS

$f(x,y)$	Função bidimensional que representa uma imagem digital
α	Parâmetro utilizado no cálculo de intervalo de confiança
$c(s,t)$	Correlação entre funções de variáveis discretas
$diff(x)$	Derivada de primeira ordem de x
T	Limiar escolhido para binarização
μ	Média
σ	Desvio-padrão
ρ	Função de densidade de probabilidade
γ	Coeficiente de correlação

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Processo de laminação.....	14
1.2	Motivação	17
1.3	Objetivo geral e objetivos específicos	18
1.4	Estrutura do trabalho	18
2	ASPECTOS TEÓRICOS.....	20
2.1	Introdução ao Processamento Digital de Imagens.....	20
2.2	Representação de uma imagem digital	21
2.3	Amostragem e quantização.....	22
2.4	Vizinhança entre pixels	22
2.5	Etapas do Processamento de Imagens	23
2.6	Segmentação.....	23
2.7	Binarização	24
2.8	Filtro de realce: derivada	25
2.9	Conceitos de Estatística.....	26
2.10	Métodos de classificação	28
2.10.1	Casamento por correlação (<i>template matching</i>).....	28
2.10.2	Árvores de decisão	29
3	IMAGENS SELECIONADAS PARA ESTUDO.....	31
3.1	Avaliação de características para seleção das imagens	31
3.2	Iluminação	32
3.3	Posicionamento.....	33
3.4	Marcação de defeitos para criação de base de dados	34
4	TESTES E RESULTADOS	36
4.1	Testes de correlação	36
4.1.1	Resultados positivos – Identificação e classificação correta.....	36
4.1.2	Primeiro problema: falsos positivos	38
4.1.3	Problema crítico: erros de classificação	40
4.1.4	Conclusão dos testes de correlação	42
4.2	Média do nível de cinza por coluna.....	44

4.3 Derivada da média do nível de cinza por coluna.....	51
4.4 Algoritmo criado para detecção de múltiplos defeitos	55
4.4.1 Distribuição das amostras.....	55
4.5 Classificação.....	62
4.5.1 Classificação entre defeitos claros e escuros.....	63
4.5.2 Binarização para definir ROI	64
4.5.3 Obtenção de atributos para montar classificador	65
4.5.4 Classificador – Árvore de decisão.....	67
5 CONCLUSÕES E PROJETOS FUTUROS.....	70
5.1 Testes de correlação	70
5.2 Testes estatísticos	70
5.3 Classificador baseado em árvore de decisão	71
5.4 Projetos futuros.....	71
REFERÊNCIAS BIBLIOGRÁFICAS	72
APÊNDICE A – CÓDIGOS.....	74

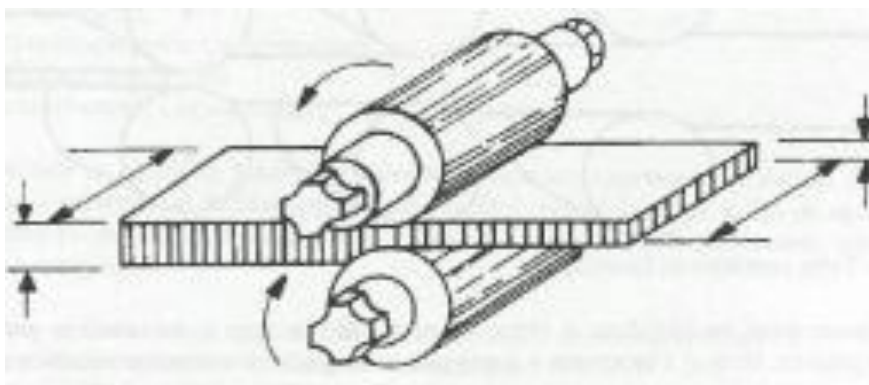
1 INTRODUÇÃO

1.1 Processo de laminação

A laminação é um processo de conformação mecânica no qual o material é forçado a passar entre dois cilindros, girando em sentidos opostos, com praticamente a mesma velocidade superficial e separados entre si de uma distância menor que o valor da espessura inicial do material a ser deformado (MACHADO, 2006).

A Figura 1 mostra uma cadeia de laminação.

Figura 1 – Cadeira de laminação



Fonte: MACHADO, 2006.

A laminação tem alguns objetivos principais: produzir um produto final com a melhor qualidade possível, de forma a adequar-se aos critérios exigidos pelo cliente, tendo espessura e comprimento específicos, além de propriedades mecânicas e condições superficiais adequadas. Como em toda aplicação de engenharia, aqui também se busca uma alta taxa de produção com baixo custo.

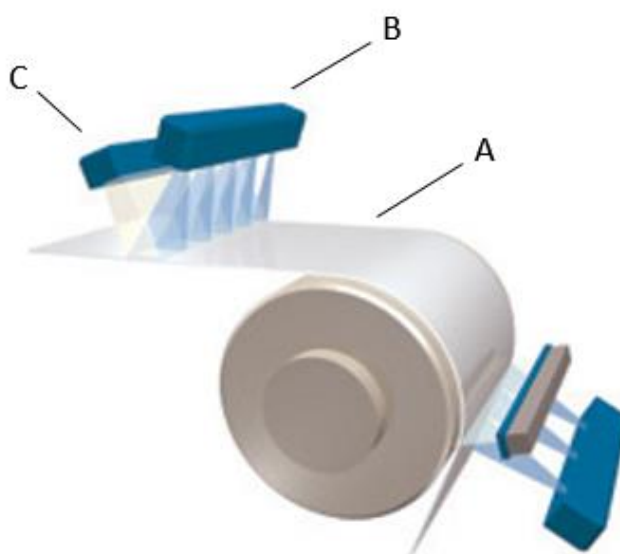
No processo de laminação de tiras a quente as principais etapas são: preparação do material inicial para laminação, aquecimento do material inicial, laminação a quente, acabamento e tratamento térmico (caso o produto já esteja finalizado), laminação a frio (caso seja necessário), tratamento térmico e por fim o acabamento e revestimento (MACHADO, 2006).

Uma tarefa essencial durante o processo de laminação é o controle de qualidade do produto. Devem ser efetuadas inspeções em busca de possíveis defeitos, por exemplo os superficiais

(arranhões, carepa, esfoliações, trincas, vazios internos, etc). Isto é importante pois produtos com excesso de defeitos são rejeitados, o que gera desperdício de tempo e energia.

O objeto de pesquisa deste trabalho é a detecção e análise de defeitos superficiais em tiras laminadas a quente. A Figura 2 mostra uma maneira de obter imagens das tiras de aço no fim do processo de laminação.

Figura 2 – Sistema de aquisição de imagens



Fonte: PARSYTEC, 2009 (modificado pelo autor).

Legenda: (A) tira de aço

(B) câmeras

(C) lâmpadas

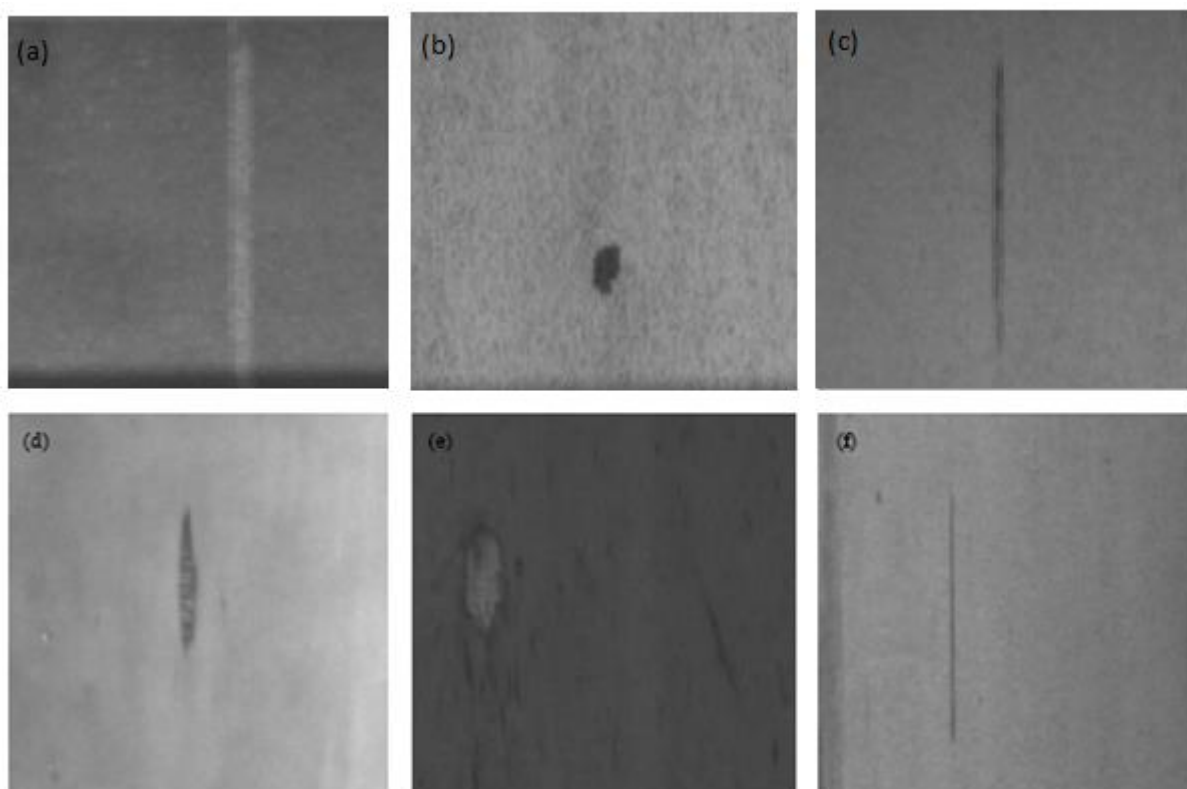
O sistema mostrado na Figura 2 é composto por uma série de câmeras (B) em paralelo, cujo objetivo é capturar imagens de toda a porção horizontal da tira (A) durante sua produção. Lâmpadas (C) podem ser ajustadas para melhorar a uniformidade de iluminação da tira, evitando efeitos luminosos indesejáveis causados pela iluminação do ambiente em que o processo ocorre.

As imagens podem ser enviadas para um sistema computadorizado onde é possível manipulá-las a fim de detectar possíveis defeitos. Uma vez que um defeito é detectado, é interessante classificá-lo quanto ao tipo e ao grau de severidade. Com informações desse tipo é possível tomar providências e adotar medidas para solucionar os defeitos, eliminando suas causas ou

minimizá-los de forma a atender às especificações desejadas. Sistemas como esse utilizam processamento digital de imagens, reduzindo a necessidade de operação humana na avaliação da qualidade do material produzido, já que o próprio sistema avalia as condições atuais da produção, faz uma comparação com requisitos de qualidade e pode exibir avisos ou alertas em casos específicos, otimizando assim o processo de inspeção das tiras.

A Figura 3 exibe exemplos de imagens de algumas tiras de aço laminadas a quente contendo defeitos de superfície. Imagens desse tipo são o objeto de estudo deste trabalho.

Figura 3 – Exemplos de imagens de tiras laminadas a quente



Fonte: Luo, Q. He, Y. Yu, W. and Zheng, J., 2014 (modificado pelo autor).

Legenda: (a) Arranhão Claro

(b) Carepa

(c) Esfoliação

(d) Escalas de óxido

(e) Casca convexa

(f) Rachaduras longitudinais

1.2 Motivação

O processamento digital de imagens está presente em inúmeras aplicações, tanto acadêmicas quanto comerciais, industriais, residenciais ou até mesmo governamentais. Por ser um tema muito rico e com possibilidades muito amplas, sua utilização é cada vez mais comum. Isto faz com que o desenvolvimento de novas técnicas nessa área seja necessário, desafiador e, por isso, muito motivacional.

Sabe-se que em processos industriais a qualidade do produto produzido é um dos fatores mais importantes a serem controlados. Um sistema de controle de qualidade satisfatório deve ser ágil e preciso, de forma a garantir uma boa velocidade de reação, otimizando resultados e minimizando perdas.

A inspeção automática de defeitos em tiras laminadas a quente feita via processamento digital de imagens surge por dois motivos principais: precisa ser feita por sistemas computadorizados, de forma a evitar uma possível introdução de erros causados pela avaliação do ser humano devido à fadiga visual; precisa ser realizada em alta velocidade, devido às características do processo.

A relevância do projeto de pesquisa apresentado está em estudar diferentes formas de aplicação de filtros de imagem, bem como seus modelos matemáticos. Mais do que obter resultados específicos em relação ao mapeamento de defeitos em imagens, busca-se com este projeto aprender fundamentos de processamento de imagem, de forma a criar um embasamento teórico e prático para auxiliar no desenvolvimento de novas tecnologias.

1.3 Objetivo geral e objetivos específicos

Objetivo geral:

O objetivo geral deste projeto é realizar um estudo com testes práticos para detectar e classificar defeitos em tiras de aço laminadas a quente, utilizando técnicas computacionais de processamento de imagem. Para isso, algumas tarefas são necessárias: analisar a bibliografia sobre processamento digital de imagens; estudar características visuais de defeitos encontrados em amostras contidas em um banco de imagens fornecido por uma indústria siderúrgica; realizar simulações e testes, utilizando o software MATLAB.

Os objetivos específicos deste trabalho são:

- Determinar uma forma eficiente de identificar amostras defeituosas em imagens que contenham defeitos;
- desenvolver um algoritmo capaz de identificar defeitos distintos e classificá-los;
- avaliar o desempenho do sistema desenvolvido, comparando-o com o catálogo do banco de imagens.

1.4 Estrutura do trabalho

O Capítulo 1 apresenta a contextualização deste trabalho e, além de expor a motivação que resultou na decisão de abordar o assunto aqui tratado, estão listados os objetivos gerais e específicos.

O Capítulo 2 trata de aspectos teóricos relacionados ao Processamento Digital de Imagens e também algumas teorias relacionadas à Estatística.

O Capítulo 3 aborda alguns aspectos que foram estudados para a escolha das imagens utilizadas para realização do trabalho.

O Capítulo 4 exhibe o desenvolvimento do projeto, explicando o passo a passo de cada etapa realizada e mostrando as dificuldades encontradas e contornadas. Em cada item do

desenvolvimento há exemplos de imagens mostrando os resultados de aplicação das técnicas escolhidas.

O Capítulo 5 fornece conclusões para os resultados obtidos, assim como sugestões para projetos futuros.

Em seguida, serão apresentadas as referências bibliográficas pesquisadas durante a elaboração deste trabalho.

Finalmente, serão apresentados os apêndices contendo os códigos feitos em Matlab.

2 ASPECTOS TEÓRICOS

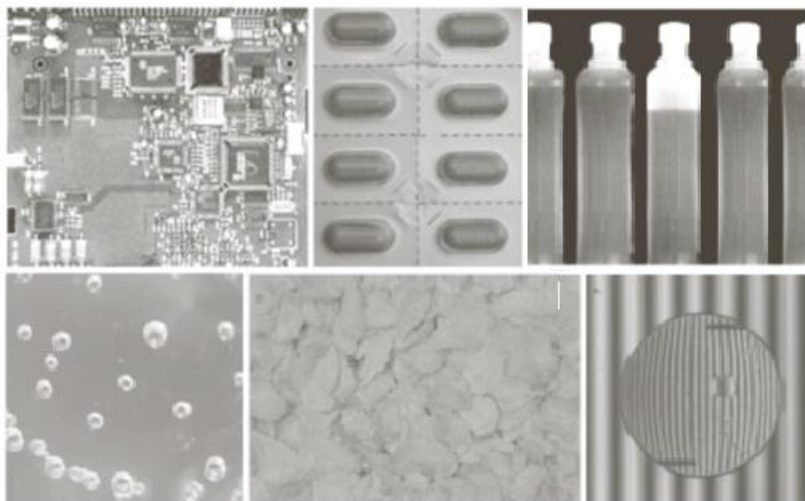
2.1 Introdução ao Processamento Digital de Imagens

O processamento digital de imagens está presente em diversas áreas, por exemplo: na melhoria das informações visuais para interpretação humana e no processamento de dados de imagens para armazenamento, transmissão e representação (GONZALEZ, 2008). Um caso típico de aplicação é na análise periódica de imagens aéreas de centros populacionais, para avaliação do crescimento populacional, ou então em áreas rurais, para avaliação e dimensionamento de desmatamentos.

A indústria de produtos manufaturados aplica constantemente a inspeção visual automatizada, com o objetivo de analisar se algum produto possui defeito, seja de forma, coloração ou tamanho (GONRING, 2013).

A Figura 4 ilustra alguns exemplos onde é feita a aplicação de processamento de imagem, tanto em indústrias quanto em supervisão de processos médicos (implante intra-ocular).

Figura 4 – Exemplos de produtos/serviços em que utiliza-se processamento de imagens



Fonte: GONZALEZ, 2008.

Legenda: (a) placa de circuito impresso

(b) pílulas

(c) garrafas

(d) bolhas de ar em produtos de plástico

(e) cereal

(f) implante intra-ocular

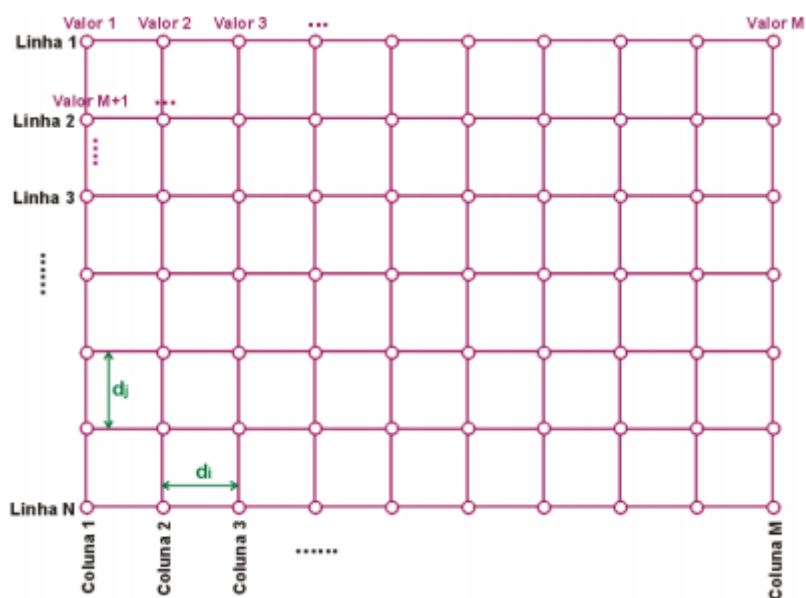
2.2 Representação de uma imagem digital

Uma imagem pode ser definida como uma função bidimensional $f(x,y)$, em que x e y são coordenadas espaciais. A amplitude f em determinado ponto representa a intensidade da imagem no mesmo. Caso os valores x , y e f sejam todos finitos e discretos, a imagem é dita digital.

Os finitos elementos que compõem uma imagem digital são chamados *elementos pictóricos*, *elementos de imagem*, *pels* ou *pixels* (termo mais conhecido e utilizado).

Para que possa ser feito armazenamento computacional, imagens digitais são referenciadas em *grids*, organizados em linhas e colunas. Em geral, as linhas são ordenadas de cima para baixo, com a primeira linha correspondendo à linha superior da imagem. As colunas iniciam no canto esquerdo, o que significa que a coordenada de origem de uma imagem digital fica localizada no canto superior esquerdo da imagem (ponto referente à Linha 1, Coluna 1, na Figura 5).

Figura 5 – Exemplo de grid de uma imagem digital, estruturada em Linhas e Colunas



Fonte: VASCONCELOS, 2002.

2.3 Amostragem e quantização

Os processos de amostragem e quantização são responsáveis por dar origem à imagem digital. A amostragem torna discretos os valores originalmente contínuos ao longo das direções x e y das imagens, resultando em uma representação matricial com N linhas e M colunas, totalizando $N \times M$ amostras, conforme visto no item 2.2. A quantização é responsável por associar um nível de brilho para cada amostra da matriz que representa a imagem real. Em outras palavras, a digitalização das coordenadas de uma imagem é chamada de *amostragem* e a digitalização dos valores de amplitude é chamada *quantização* (GONZALEZ, 2008).

2.4 Vizinhaça entre pixels

Um pixel p na coordenada (x,y) possui dois pixels vizinhos verticais, dois vizinhos horizontais e quatro vizinhos diagonais. As vizinhaças podem ser denotadas por *vizinhaça-4* de p ou *vizinhaça-8* de p (GONZALEZ, 2008). Quando p está localizado na borda da imagem, ignora-se os pixels que estão do lado de fora.

Os quatro vizinhos horizontais e verticais de p possuem coordenadas dadas por

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1). \quad (1)$$

Os quatro vizinhos diagonais de p possuem coordenadas dadas por

$$(x + 1, y + 1), (x - 1, y + 1), (x - 1, y - 1), (x + 1, y - 1). \quad (2)$$

A Figura 6 ilustra o conceito de vizinhaça.

Figura 6 – (a)Vizinhaça-4 de um pixel (b)Vizinhaça-8 de um pixel

	$(x, y+1)$		$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$
$(x-1, y)$	(x, y)	$(x+1, y)$	$(x-1, y)$	(x, y)	$(x+1, y)$
	$(x, y-1)$		$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$

Fonte: Produção do próprio autor.

2.5 Etapas do Processamento de Imagens

A primeira etapa do processamento digital de imagens é a aquisição da imagem, que pode ser feita pela captura da cena com uma câmera digital ou então pela recepção de uma imagem cujo formato já seja digital. Após a aquisição, pode ser feito um redimensionamento na imagem, para padronização e posterior aplicação de uma sequência de algoritmos (que podem representar operações lineares ou não lineares) necessários para obtenção do resultado objetivado.

A determinação da região de interesse, ou ROI (*region of interest*) é uma das etapas mais importantes do processamento, pois possibilita economizar recursos computacionais em casos onde apenas uma parte da imagem possui dados pertinentes à análise que estiver sendo realizada.

2.6 Segmentação

A segmentação é a etapa de separação da ROI do restante da imagem, sendo portanto fundamental para o processo de análise e resolução de problemas relacionados a imagem. A separação da região de interesse é feita de acordo com o problema a ser resolvido, até que sejam detectados os objetos (GONZALEZ, 2008). Quando há controle do ambiente de aquisição de imagens, por exemplo em relação à uniformidade da iluminação, a segmentação torna-se mais simples e eficiente.

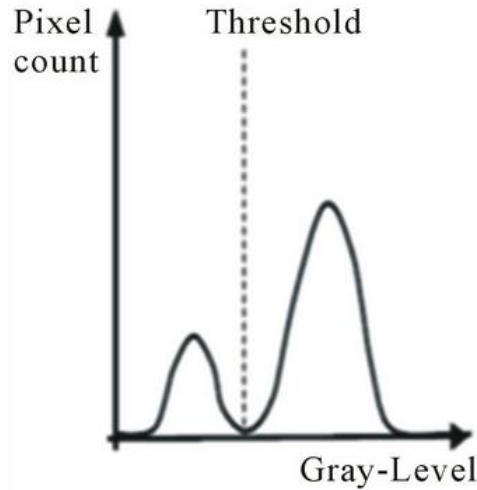
Os critérios utilizados para segmentar uma imagem variam de acordo com o problema, mas em geral baseiam-se na divisão, ou separação, de regiões que contenham atributos semelhantes, como por exemplo o nível de cinza.

A limiarização (*thresholding*) é um importante método de segmentação: consiste em dividir a imagem em relação ao nível de intensidade. Se a imagem for representada em níveis de cinza, por exemplo, a ideia da limiarização é dividir a imagem de acordo com limiares estabelecidos por algum método, seja manual ou automático. Para uma imagem representada em 256 níveis de cinza, caso o número de limiares seja igual a um, o processo resultará em uma imagem em apenas dois níveis: preto (0) e branco (255). Isto é comumente chamado de binarização.

2.7 Binarização

Regiões de uma imagem podem ser representadas de acordo com valores de intensidade.

Figura 7 – Gráfico com ocorrências de níveis de cinza e *threshold* para limiarização



Fonte: AL-BAYATI, 2013.

A Figura 7 mostra um gráfico de ocorrência de valores de níveis de cinza para uma determinada imagem $f(x,y)$. É possível notar que há dois picos e entre eles foi definido um limiar, ou *threshold* T , com o qual é possível dividir o gráfico em duas regiões em relação ao nível de cinza. Binarizar uma imagem consiste em dividi-la em dois níveis de cinza. Dessa forma, considerando que a imagem em pauta possua objetos claros sobre um fundo escuro, todo ponto (x,y) que possuir intensidade $f(x,y) \geq T$ será parte do objeto de interesse e o restante da imagem será pertencente ao fundo.

A imagem segmentada, representada por $r(x,y)$ é definida da seguinte forma:

$$r(x,y) = \begin{cases} 0, & \text{se } f(x,y) < T \\ 1, & \text{se } f(x,y) \geq T \end{cases} \quad (3)$$

Um processo de limiarização global é aquele em que o limiar T é aplicado à imagem inteira, ao passo que se T varia no decorrer da imagem, a limiarização é variável, local ou regional. Há também a limiarização global ótima, que pode ser utilizada para minimizar o erro médio incorrido na atribuição de pixels a cada região da imagem. Nesse sentido, uma abordagem amplamente utilizada é o do método de Otsu (GONZALEZ, 2008).

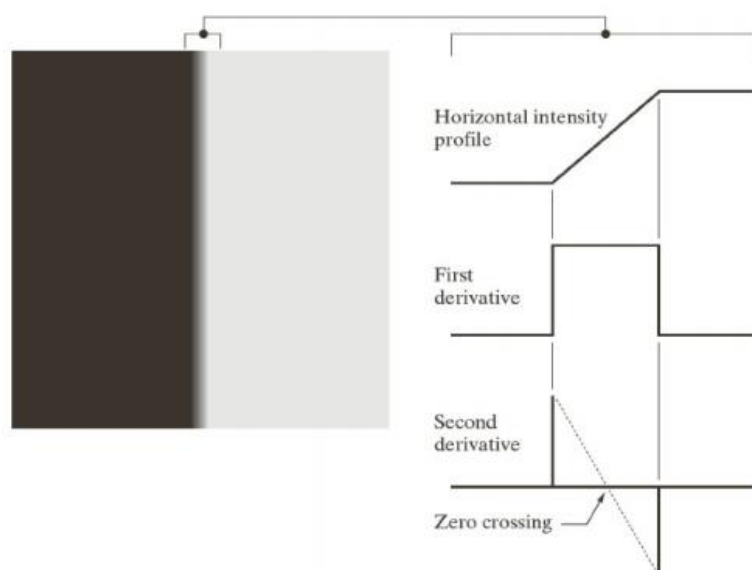
2.8 Filtro de realce: derivada

Há inúmeros filtros utilizados para realçar detalhes em imagens digitais. Uma forma de realçar transições de intensidade é utilizar filtros de derivadas, tanto de primeira quanto de segunda ordem. Os pontos de interesse na aplicação de tais filtros podem ser: áreas de intensidade constante, início e fim de discontinuidades (vide Figura 8).

Características da derivada de primeira ordem:

- Deve ser nula em áreas de intensidade constante;
- deve ser diferente de zero no início e fim de um degrau ou rampa de intensidade;
- deve ser constante e diferente de zero ao longo de uma rampa.

Figura 8 – (a) Regiões de intensidade constante separadas por uma borda vertical (b) Perfil horizontal da região de transição, juntamente com o sinal de resposta da primeira e da segunda derivadas



Fonte: GONZALEZ, 2008.

Seja X um vetor com M elementos tal que

$$X = [x(1) \ x(2) \ \dots \ x(M)],$$

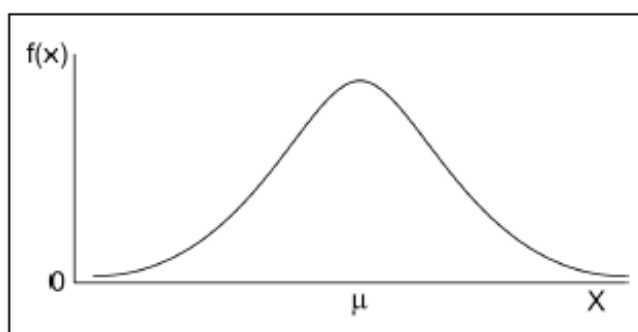
A derivada (*diff*) Y de primeira ordem de X pode ser calculada pela diferença entre os valores de elementos adjacentes, resultando em um vetor com $M-1$ elementos.

$$Y = \text{diff}(X) = [x(2) - x(1) \quad x(3) - x(2) \quad \dots \quad x(m) - x(m - 1)]. \quad (4)$$

2.9 Conceitos de Estatística

Probabilidade é a chance de ocorrência de um determinado evento, ou seja, a chance de ocorrer uma certa medida em um dado intervalo de tempo. A distribuição de frequências de ocorrência de um evento é a aproximação da distribuição de probabilidades, também chamada Função de Densidade de Probabilidade. Tal distribuição é considerada Normal quando sua curva tem a forma de "sino" (PORTAL ACTION, 2014), como se observa na Figura 9.

Figura 9 – Função de densidade de probabilidade com forma de ‘sino’



Fonte: PORTAL ACTION, 2014.

Muitos processos industriais possuem amostras com variação naturalmente aleatória. Ainda que as distribuições de tais processos possam assumir uma variedade de formas, diversas variáveis observadas geralmente possuem uma distribuição de frequências que é, aproximadamente, uma distribuição de probabilidade Normal. Alguns exemplos de casos em que as amostras seguem aproximadamente uma curva de distribuição Normal são verificados quando faz-se um estudo dos valores de altura, peso e pressão sanguínea de uma população.

Portanto, grande parte das distribuições de frequência possuem “Tendência Central”, que é um dos conceitos fundamentais em análise estatística: quando a maior parte das observações se acumula na área entre os dois extremos (BUSSAB, 2006). Há algumas medidas principais de Tendência Central, duas delas são a Média e a Moda.

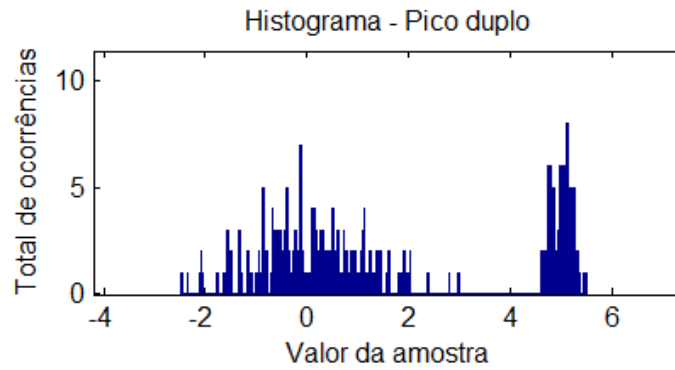
A Média é usada para distribuições simétricas ou quase simétricas, ou para distribuições que têm um único pico dominante, como a da Figura 9. Sua fórmula é dada por

$$\mu = \frac{\sum_{i=1}^n x_i}{n} \quad (5)$$

onde x_i é a i -ésima amostra e n é a quantidade de amostras.

A Moda é o valor que ocorre com maior frequência, sendo comumente usada para distribuições assimétricas, por exemplo em situações desregulares onde dois picos são encontrados, como visto na Figura 10.

Figura 10 – Histograma de dados distribuídos desregularmente



Fonte: Produção do próprio autor.

Há medidas que representam quão dispersos estão os dados, são as chamadas medidas de dispersão. Fundamental em análises estatísticas, uma dessas medidas é o Desvio-Padrão, dado por (estimador não polarizado pois, diferentemente de μ em (5), divide-se por $n-1$ e não por n)

$$\sigma = \frac{1}{n-1} \sqrt{\sum_{i=1}^n (x_i - \mu)^2} , \quad (6)$$

onde x_i é a i -ésima amostra, μ é a média e n é a quantidade de amostras.

Finalmente, utilizando os valores de Média e Desvio-Padrão é possível calcular a Função de Densidade de Probabilidade para uma distribuição normal, supondo uma variável aleatória x normal, conforme

$$\rho(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad (7)$$

onde μ é a Média e σ é o Desvio-Padrão.

2.10 Métodos de classificação

O processo de classificação de objetos em processamento de imagens é feito baseado em suas características ou estrutura, sendo geralmente realizado da seguinte forma:

- aquisição da imagem;
- tratamento da imagem;
- segmentação de objetos de interesse;
- cálculo e obtenção de atributos;
- construção de um classificador;
- testes de classificação.

Dois métodos de classificação bastante difundidos são o casamento por correlação (*template matching*) e a árvore de decisão (GONZALEZ, 2008).

2.10.1 Casamento por correlação (*template matching*)

Uma das principais aplicações de correlação em processamento de imagens é na área de casamento de *templates*, o que significa buscar encontrar em uma imagem desconhecida a região que mais se aproxima de uma imagem já conhecida. O casamento mais próximo pode ser considerado aquele cuja seleção da imagem forneça a função de correlação de maior valor (GONZALEZ, 2008).

Se $f(x,y)$ e $g(x,y)$ são funções de variáveis discretas (nesse caso, são imagens), a similaridade ou, em outras palavras, a correlação $c(s,t)$ entre elas está definida segundo a equação a seguir:

$$c(s, t) = \sum_x \sum_y f(x, y) g(x - s, y - t), \quad (8)$$

onde f é a imagem principal, o somatório é feito sobre as coordenadas x e y dentro da janela que contém a característica g , posicionada nas coordenadas (s,t) (GONZALEZ, 2008).

Se a energia da imagem, representada por $\sum f^2(x, y)$ variar com a posição, a análise pode falhar. Por exemplo, a correlação entre uma região de certa imagem e um *template* semelhante pode ser mais baixa que a correlação entre o *template* e uma região mais clara (LEWIS, 1995). Portanto, a equação 8 não é imune às possíveis variações causadas por alterações de iluminação ao longo da imagem analisada. (o que justifica a escolha de imagens uniformes, que será apresentada no item 3.2).

A normalização do coeficiente de correlação $\gamma(\cdot)$ é uma opção para vencer o obstáculo citado anteriormente, e pode ser feita como sugere a expressão

$$\gamma(s, t) = \frac{\sum_{x,y} [f(x,y) - \bar{f}_{s,t}] [t(x-s, y-t) - \bar{t}]}{\{\sum_{x,y} [f(x,y) - \bar{f}_{s,t}]^2 \sum_{x,y} [t(x-s, y-t) - \bar{t}]^2\}^{0.5}}, \quad (9)$$

onde $\bar{t}$ é a média da característica e $\bar{f}_{s,t}$ é a média de $f(x, y)$ na região da característica.

A equação 9 é denominada correlação cruzada normalizada (*normalized cross-correlation*) (LEWIS, 1995).

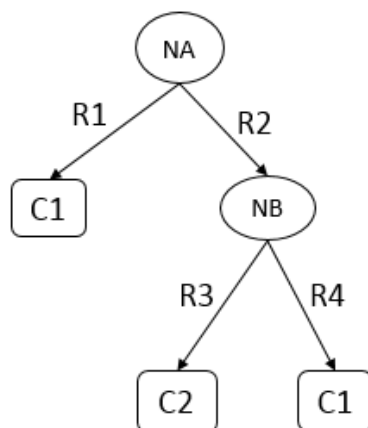
2.10.2 Árvores de decisão

Os algoritmos de classificadores baseados em árvore de decisão trabalham na sucessiva divisão de um problema em subproblemas, até que determinado subproblema seja constituído de apenas uma classe, ou até que esta classe seja dominante. Nesse ponto, é associada uma solução para o problema inicial.

Uma árvore de decisão é estruturada por nós, ramos e folhas. Os nós representam os atributos analisados, os ramos provenientes dos nós recebem os valores do atributo em questão, e os

nós folha representam as diferentes classes de um conjunto de exemplos. A Figura 11 mostra um exemplo de árvore de decisão, onde *nós*, *ramos* e *folhas* estão representados pelas letras *N*, *R* e *C*, respectivamente.

Figura 11 – Árvore de decisão



Fonte: Produção do próprio autor.

Do modelo visto na Figura 11 podem-se extrair três regras de diagnóstico: se $NA = R1$, então o resultado é $C1$; se $NA = R2$ e $NB = R3$, então o resultado é $C2$; se $NA = R2$ e $NB = R4$, então o resultado é $C1$.

A técnica de árvore de decisão tem como ponto forte sua eficiência em termos de tempo de processamento, além de uma representação simbólica simples e normalmente bastante compreensível, o que facilita o entendimento do problema em análise (QUINLAN, 1993).

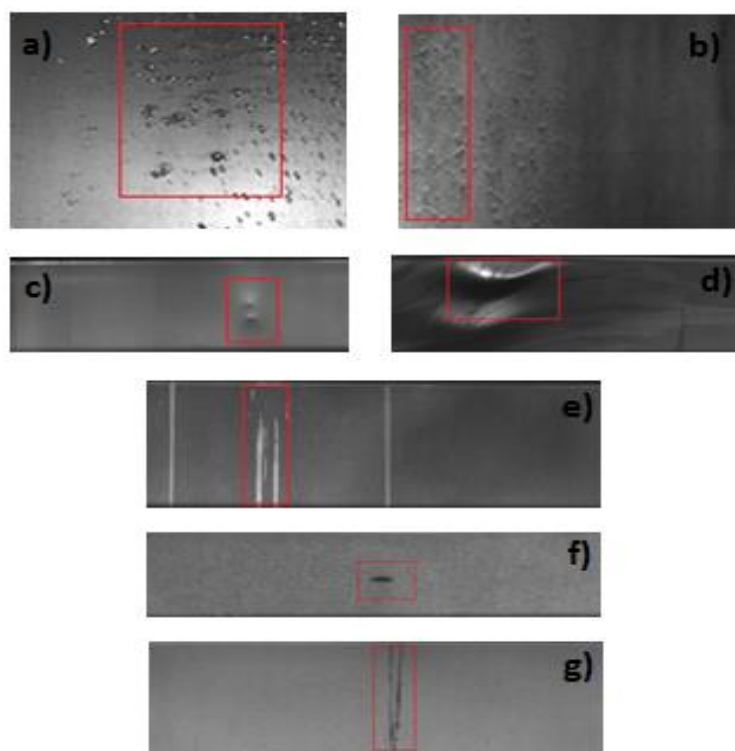
3 IMAGENS SELECIONADAS PARA ESTUDO

O primeiro passo para a realização do trabalho foi o estudo e a escolha das imagens a serem utilizadas. Estas imagens foram fornecidas por uma indústria siderúrgica, já classificadas e catalogadas em relação aos defeitos presentes nas tiras.

3.1 Avaliação de características para seleção das imagens

As classes de defeitos que podem surgir durante o processo de laminação de tiras a quente são muito variadas e podem ser visualmente bastante diferentes. Características como forma e nível de cinza, por exemplo, são bastante particulares para cada classe de defeito. Isto pode ser visto na Figura 12.

Figura 12 – Diferentes defeitos encontrados em tiras de aço



Fonte: Banco de imagens fornecido por indústria siderúrgica. (modificado pelo autor).

Legenda: (a) PS Gota

(b) Carepa de Temperatura

(c) Marca de Cilindro

(d) Repuxado

(e) Arranhão Claro

(f) Carepa Primária

(g) Esfoliação

Diante da ampla variedade de defeitos, foram escolhidas três diferentes classes como base para a realização do trabalho: *Arranhão Claro (ARC)*, *Carepa Primária (CA)* e *Esfoliação (ESF)*.

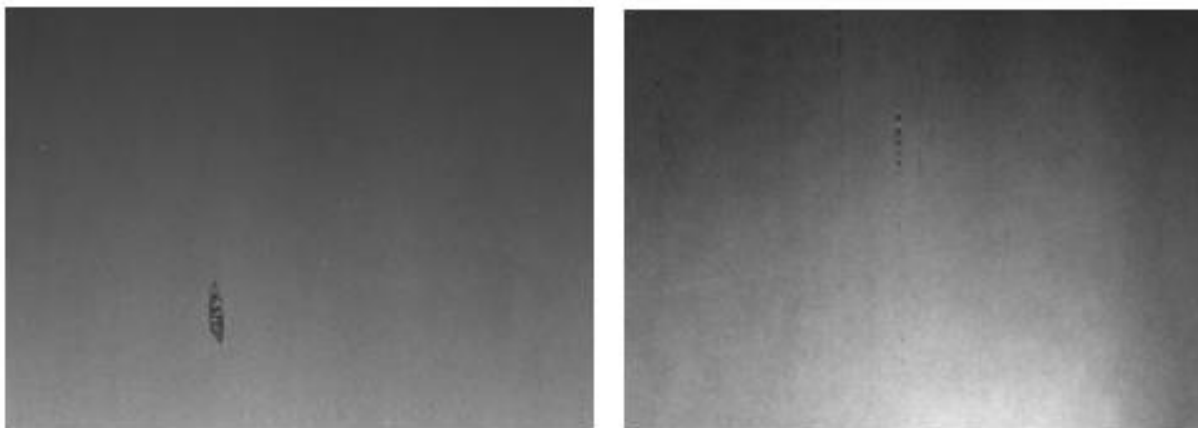
Além das diferentes classes de defeitos, há também outras questões importantes que devem ser consideradas durante os testes, como iluminação e também qual é a região da tira que é exibida na imagem (pode ser uma borda ou uma parte intermediária).

3.2 Iluminação

Em primeiro lugar, é preferível que a iluminação da imagem seja a mais uniforme possível, para que descontinuidades na iluminação não sejam confundidos com defeitos de superfície das tiras. Quanto maior é a uniformidade, mais evidentes são os defeitos, o que torna menos complexa a tarefa de localizá-los.

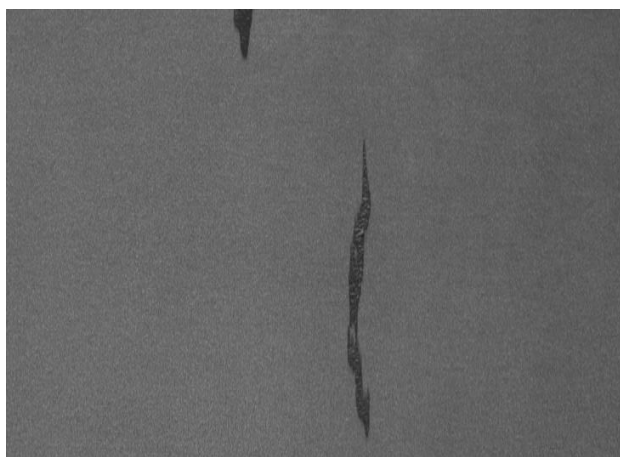
Para exemplificar a situação citada acima, a Figura 13 mostra imagens com iluminação pouco uniforme e a Figura 14 mostra uma imagem com iluminação mais uniforme.

Figura 13 – Tiras de aço com iluminação descontínua



Fonte: Banco de imagens fornecido por indústria siderúrgica.

Figura 14 –Tira de aço com iluminação uniforme

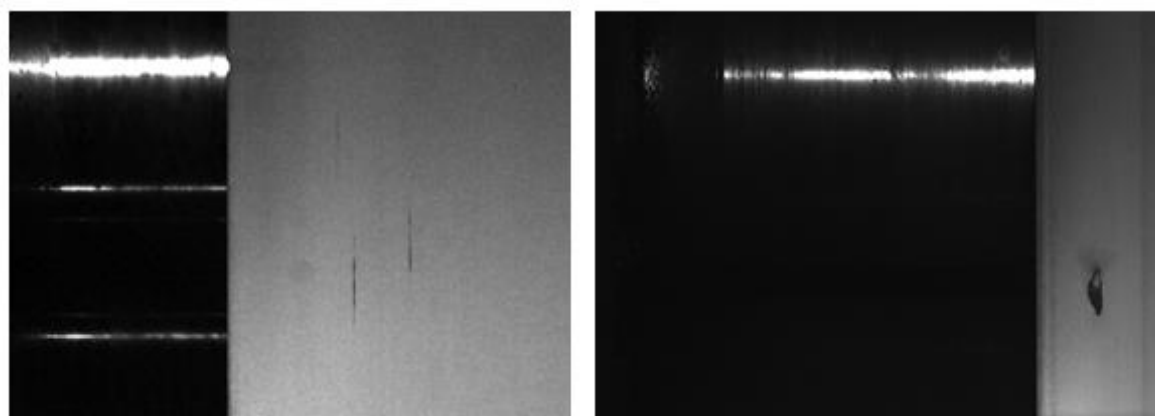


Fonte: Banco de imagens fornecido por indústria siderúrgica.

3.3 Posicionamento

Outra consideração importante é que há imagens que mostram uma das bordas laterais da tira e há imagens que mostram uma porção intermediária. Quando a imagem é feita na direção de uma borda da tira, os cilindros que a movimentam ficam expostos, tornando-se um elemento complicador para o trabalho, pois antes de procurar pelos defeitos o algoritmo precisa separar o que é tira e o que é cilindro. Isto é mostrado na Figura 15.

Figura 15 – Imagens de borda lateral



Fonte: Banco de imagens fornecido por indústria siderúrgica.

Para que o trabalho não se tornasse tão complexo, foram descartadas as imagens de borda e também aquelas que não apresentavam iluminação uniforme. Vale lembrar que os defeitos

escolhidos para análise foram: *Arranhão Claro (ARC)*, *Carepa Primária (CA)* e *Esfoliação (ESF)*.

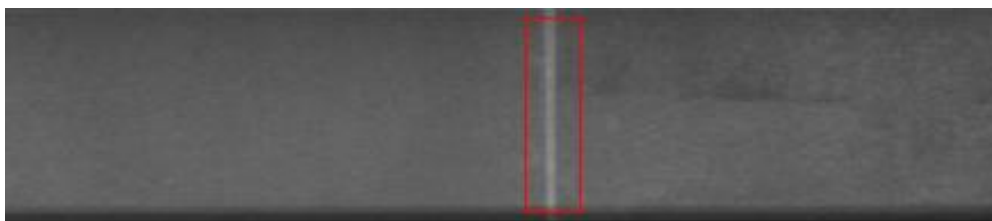
Terminado o processo de escolha das imagens, a próxima etapa foi a marcação dos defeitos, para manter uma base de dados onde fosse possível conferir os resultados ao fim de todo o trabalho.

3.4 Marcação de defeitos para criação de base de dados

Foi feito um banco de imagens com 60 amostras para cada defeito escolhido, totalizando 180 imagens. Com isso, foram criados dois grupos, cada um com 90 imagens. O primeiro grupo foi usado para marcação dos defeitos e estudo das propriedades dos mesmos (30 imagens para cada defeito). O segundo grupo foi usado para validação dos testes (30 imagens para cada defeito).

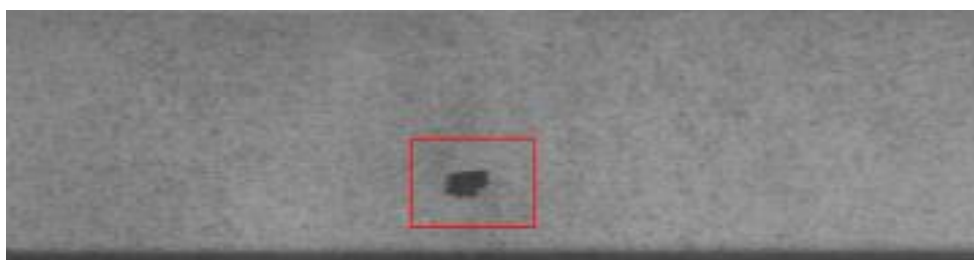
As Figuras 16, 17 e 18 mostram alguns exemplos de imagens do primeiro grupo, contendo alguns defeitos marcados. As regiões marcadas foram usadas como parâmetro para validação dos testes de correlação, etapa descrita no item 4.1.

Figura 16 – Imagem contendo defeito tipo ARC



Fonte: Banco de imagens fornecido por indústria siderúrgica (modificado pelo autor).

Figura 17 – Imagem contendo defeito tipo CA



Fonte: Banco de imagens fornecido por indústria siderúrgica (modificado pelo autor).

Figura 18 – Imagem contendo defeito tipo ESF



Fonte: Banco de imagens fornecido por indústria siderúrgica (modificado pelo autor).

4 TESTES E RESULTADOS

4.1 Testes de correlação

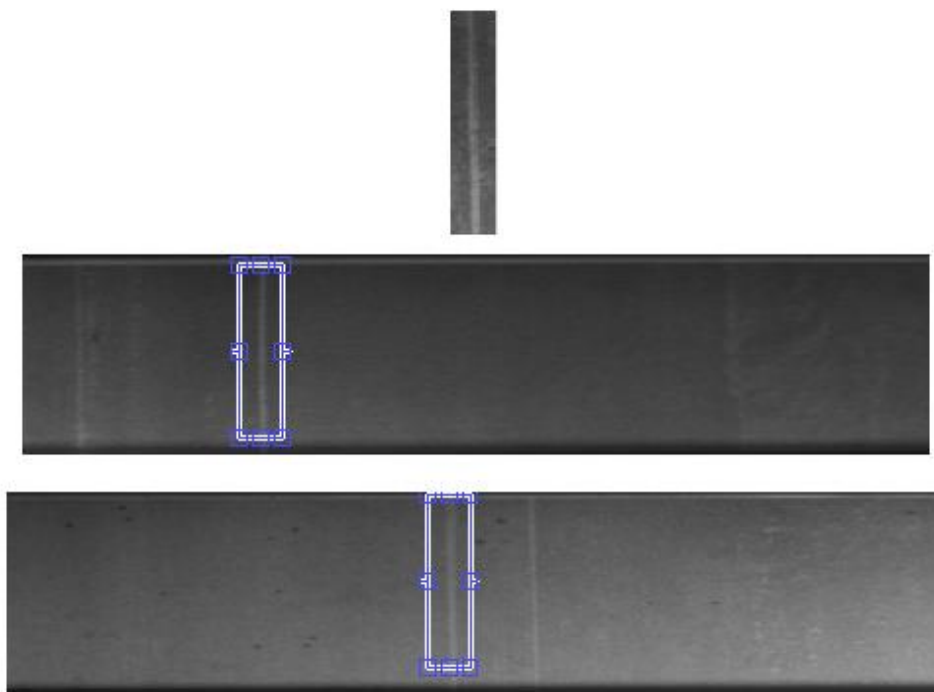
Em um primeiro momento, foram feitos testes para detectar defeitos em imagens avaliando o mapeamento da correlação entre os defeitos marcados e as imagens de validação. O algoritmo criado utiliza uma janela (*template*) e faz uma varredura desta em toda a imagem de teste. Para cada região testada é associado um valor de correlação entre ela e o *template*. O resultado é uma matriz, onde o termo de valor máximo, ou seja, de máxima correlação, é candidato a ter as coordenadas do centro do defeito.

Os testes resultaram em três diferentes situações: resultados positivos, em que o algoritmo localiza com sucesso os defeitos procurados e os classifica corretamente; falsos positivos, quando o algoritmo erra a região e escolhe um ponto em que não há defeito algum; erros de classificação, em que o algoritmo localiza um defeito de determinada classe mas o classifica de forma errada. No último caso, o resultado é considerado um problema crítico, pois na prática medidas devem ser tomadas para eliminar as causas dos defeitos, que são várias, por exemplo: oxidação, falhas em controle de temperatura ou de planicidade, cilindros de laminação defeituosos, etc. Com erros de classificação, pode ser que as medidas escolhidas para solucionar os problemas não sejam as mais indicadas.

4.1.1 Resultados positivos – Identificação e classificação correta

A Figura 19 mostra dois resultados positivos obtidos utilizando um *template ARC* para procurar defeitos em diferentes imagens de teste que contêm *ARC*. Os retângulos azuis mostram nas imagens testadas as regiões de máximo valor de correlação, locais marcados automaticamente pelo algoritmo. Em cada teste é marcada apenas uma região defeituosa, que corresponde à janela da imagem de teste que contém o máximo valor de correlação com o *template*.

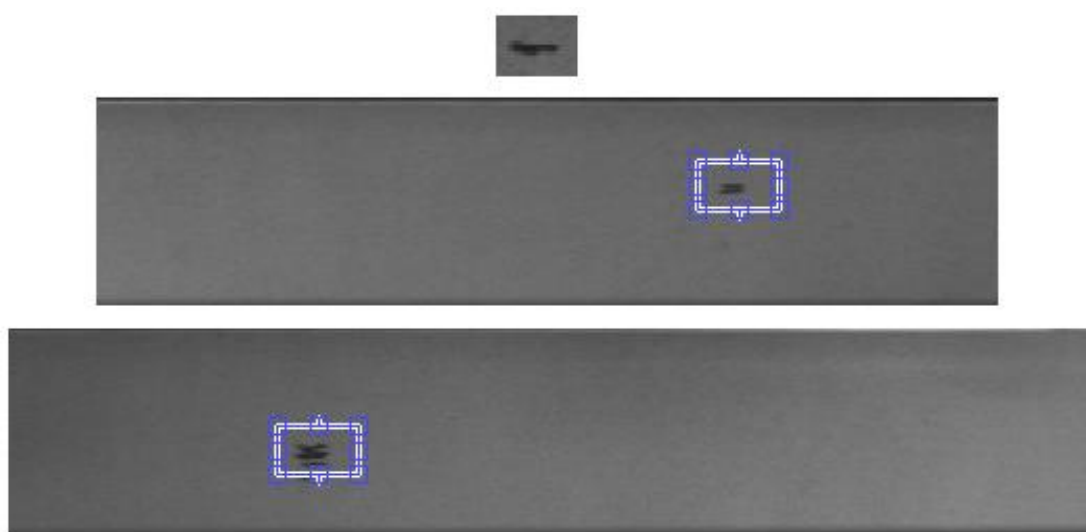
Figura 19 – Defeito tipo ARC encontrado via *template matching*



Fonte: Produção do próprio autor.

A Figura 20 mostra dois resultados obtidos utilizando um *template CA* para procurar defeitos em diferentes imagens que contêm CA. Os retângulos azuis mostram, novamente, regiões de máximo valor de correlação marcados pelo algoritmo.

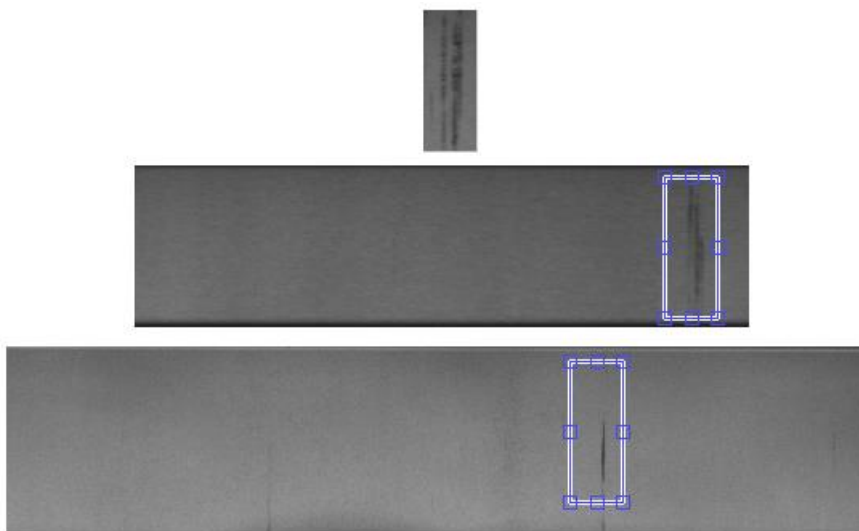
Figura 20 – Defeito tipo CA encontrado via *template matching*



Fonte: Produção do próprio autor.

A Figura 21 mostra dois resultados obtidos utilizando um template *ESF* para procurar defeitos em diferentes imagens que também contêm defeitos do tipo *ESF*.

Figura 21 – Defeito tipo *ESF* encontrado via *template matching*



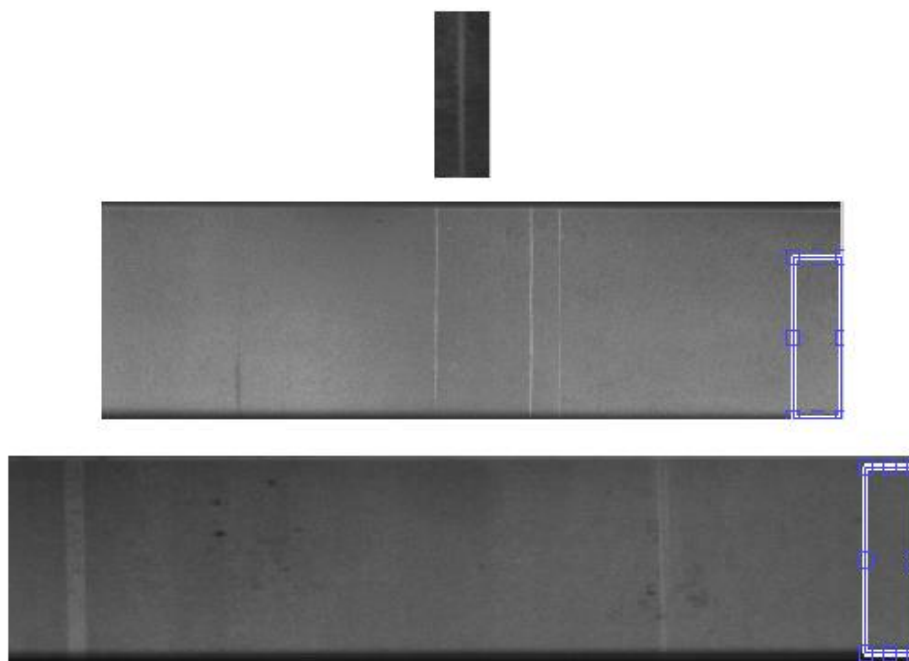
Fonte: Produção do próprio autor.

4.1.2 Primeiro problema: falsos positivos

Como o algoritmo sempre encontra um ponto onde o valor de correlação é máximo, foi necessário criar um limiar de decisão, para evitar falsos positivos. Com isso, seria possível impedir que o algoritmo decidisse que há defeito em uma região onde não há defeito algum.

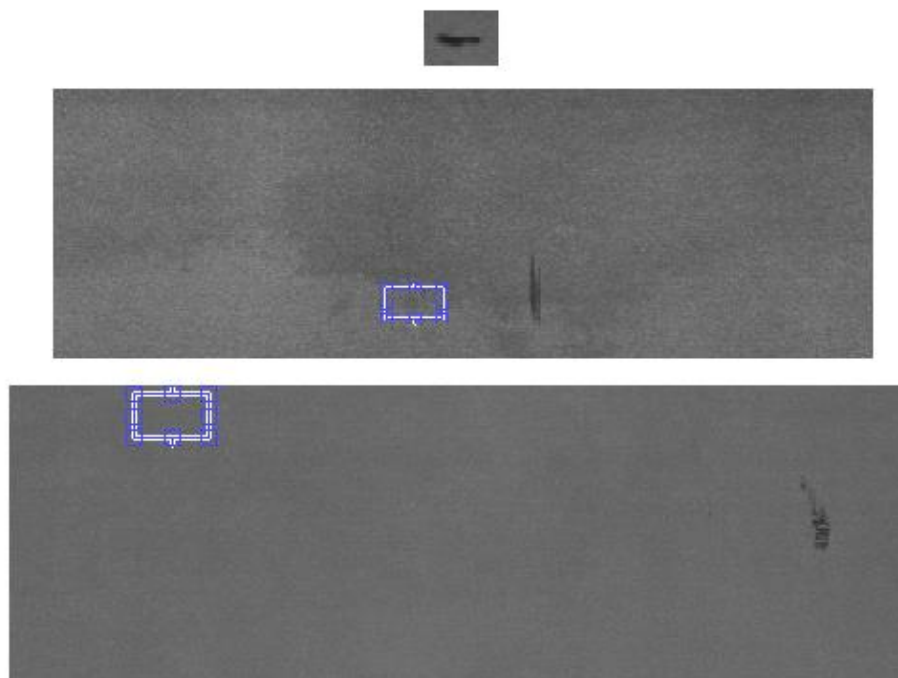
As Figuras 22, 23 e 24 mostram exemplos de casos que foram sinalizados como alerta para que fosse criado o limiar de decisão.

Figura 22 – Falsos positivos encontrados para defeito tipo ARC



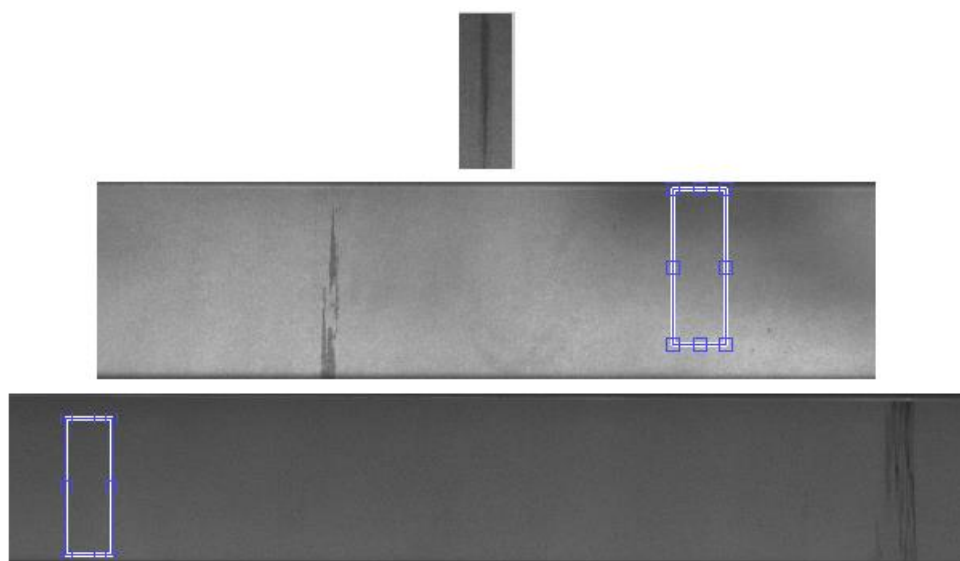
Fonte: Produção do próprio autor

Figura 23 – Falsos positivos encontrados para defeito tipo CA



Fonte: Produção do próprio autor

Figura 24 – Falsos positivos encontrados para defeito tipo ESF



Fonte: Produção do próprio autor.

Avaliando os resultados coerentes (aqueles em que não há ocorrência de falsos positivos), foi adotado um limiar de decisão igual a 0,7. Portanto, considera-se que o algoritmo deva marcar apenas as regiões em que o valor máximo de correlação seja no mínimo 0,7. Para valores abaixo, considera-se que na região avaliada não há o defeito procurado.

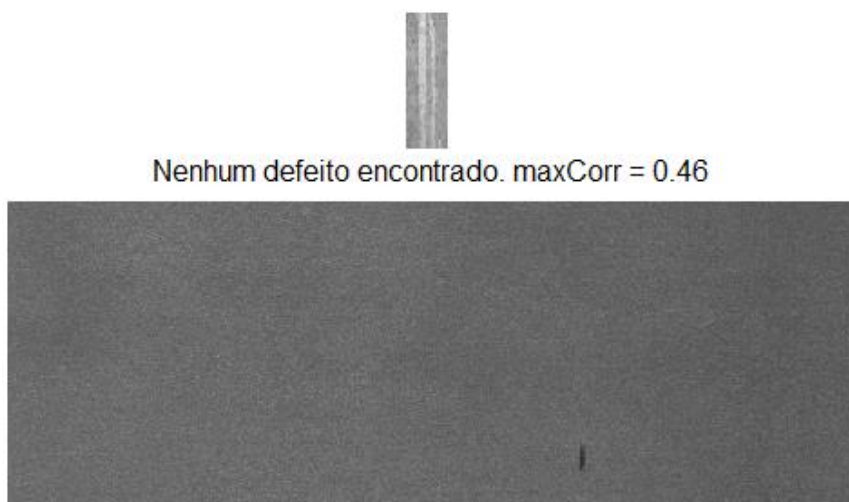
Com a utilização do limiar, o problema de falsos positivos foi solucionado completamente, ou seja, foram apontadas pelo algoritmo apenas regiões onde realmente há defeitos.

A próxima etapa foi utilizar *templates* de uma determinada classe de defeito para testes em imagens contendo outras classes de defeitos. Surgiu então um novo problema: erros de classificação, isto é, casos em que um defeito de determinada classe é encontrado, porém é classificado de forma errada.

4.1.3 Problema crítico: erros de classificação

Em diversos casos o comportamento do algoritmo é correto, por exemplo: utilizando templates do tipo *ARC* para procurar defeitos em imagens que contêm *CA*, nenhuma região é marcada, pois o limiar de decisão não é ultrapassado. A Figura 25 mostra um resultado desse tipo.

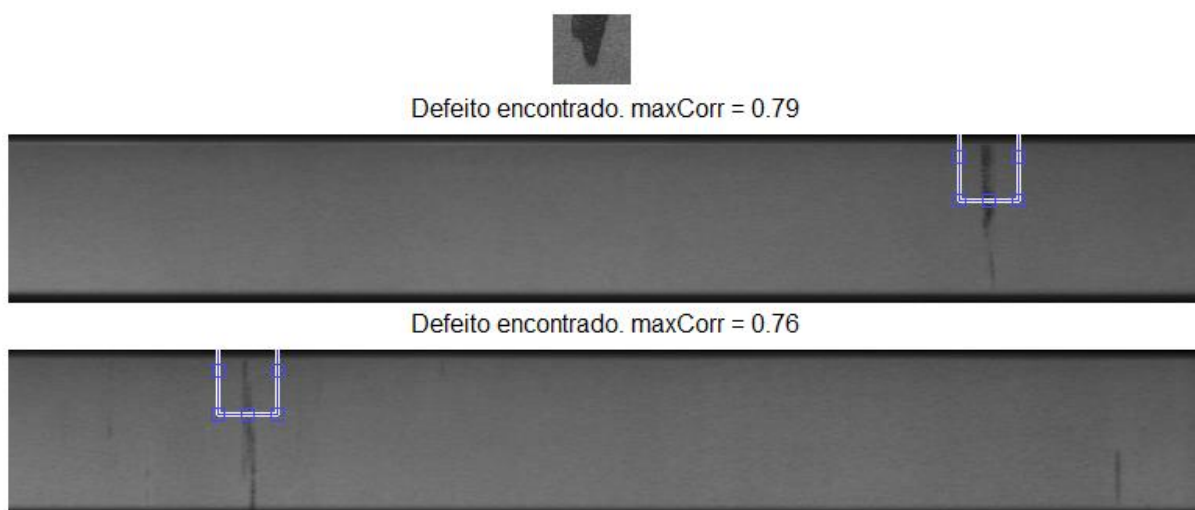
Figura 25 – Resultado correto na busca por defeitos ARC utilizando imagem contendo CA



Fonte: Produção do próprio autor.

Porém, há casos em que utilizando templates do tipo *CA*, por exemplo, o algoritmo encontra defeitos do tipo *ESF* com valores de máxima correlação acima de 0,76, ou seja, quase 10% acima do limiar de decisão escolhido anteriormente, igual a 0,7. Isto caracterizou um erro de classificação, ilustrado na Figura 26.

Figura 26 – Erro de classificação.



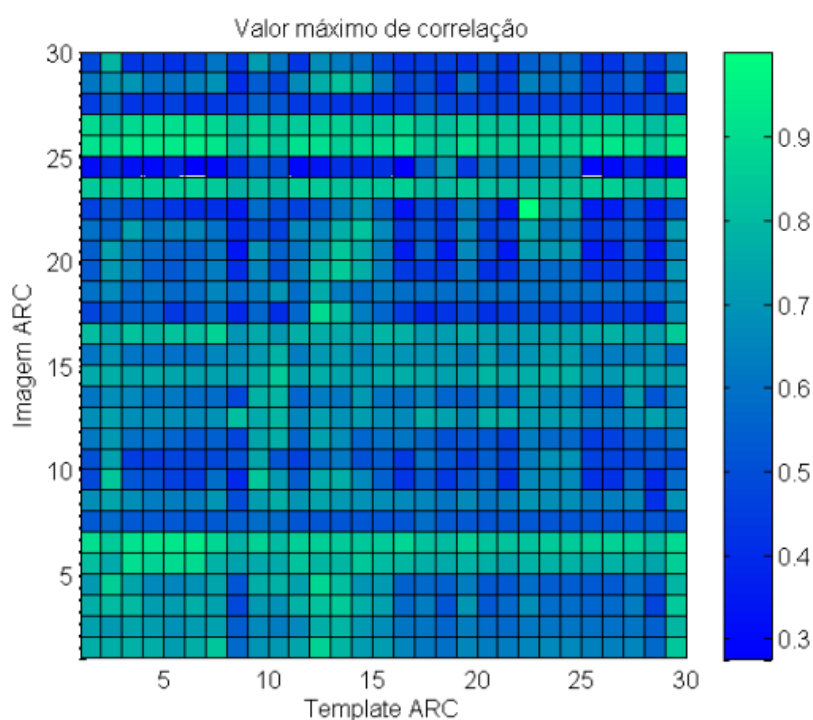
Fonte: Produção do próprio autor

4.1.4 Conclusão dos testes de correlação

Para fazer uma análise geral dos resultados, foi feito um gráfico com o mapeamento da máxima correlação em cada teste efetuado.

A Figura 27 mostra o mapeamento do valor máximo de correlação encontrado para cada template *ARC* testado em imagens que contêm defeitos tipo *ARC*. Cada quadrado marcado na imagem mostra o teste de determinado template (número no eixo horizontal) em certa imagem teste (número no eixo vertical).

Figura 27 – Mapeamento dos valores de correlação entre templates *ARC* e imagens *ARC*

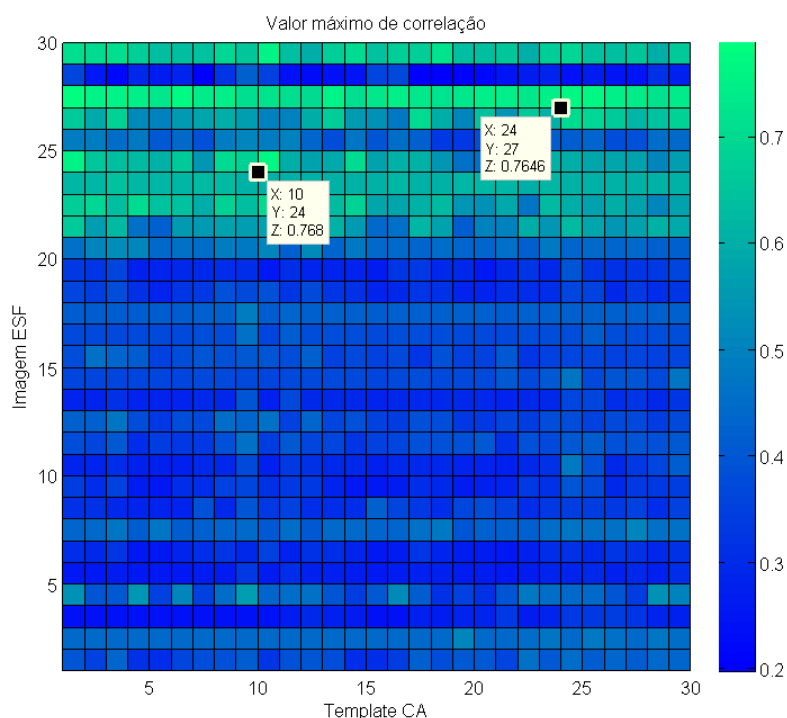


Fonte: Produção do próprio autor

Verifica-se que, exceto para as imagens 7, 11, 24 e 28, para todas as outras há templates que cumprem o objetivo de encontrar uma região com correlação máxima acima de 0,7.

A Figura 28 mostra o resultado do mapeamento dos testes com templates de CA em imagens contendo ESF.

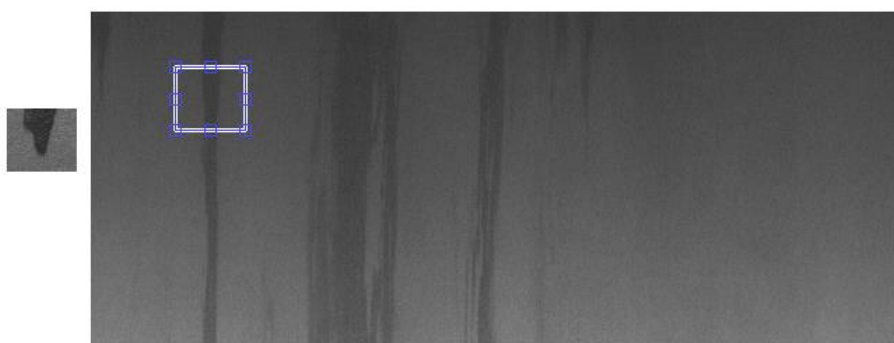
Figura 28 – Mapeamento dos valores de correlação entre templates *CA* e imagens *ESF*



Fonte: Produção do próprio autor

Em uma situação ideal, esperava-se que testes de *templates* de determinada classe em imagens de classes diferentes, como visto na Figura 28, resultassem em baixos valores de máxima correlação. Porém, em várias amostras isso não aconteceu. Ao analisar a Figura 28 verifica-se que, em alguns casos, foram obtidos valores acima de 0,75, por exemplo para os pares *template-imagem 10-24* e *24-27*. As Figuras 29 e 30 mostram os resultados para os dois pares citados.

Figura 29 – Resultado para o par *template CA_10 - imagem ESF_24*



Fonte: Produção do próprio autor

Figura 30 – Resultado para o par *template CA_24* - *imagem ESF_27*.



Fonte: Produção do próprio autor

A conclusão com os testes de correlação é que essa abordagem de busca por defeitos no contexto deste trabalho não é confiável, tamanha a quantidade de erros de classificação. Mesmo que os inúmeros falsos positivos tenham sido solucionados com a inclusão do limiar mínimo de correlação, classificações erradas aparecem em mais de 50% dos testes. Problemas desse tipo comprometem a confiabilidade do sistema de monitoramento de qualidade.

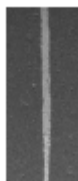
Por conta dos problemas supracitados, a busca por defeitos avaliando valores de correlação foi substituída por uma abordagem diferente.

4.2 Média do nível de cinza por coluna

Ao analisar as imagens dos defeitos estudados nesse trabalho, verifica-se que uma das principais características é que na região onde eles ocorrem há uma descontinuidade no nível de cinza.

A Figura 31 mostra um exemplo de defeito do tipo *ARC*, onde fica evidente a descontinuidade causada pelo arranhão, que faz com que o nível de cinza da região do defeito seja diferente (nesse caso ele é maior) em relação à média do nível de cinza do restante da imagem.

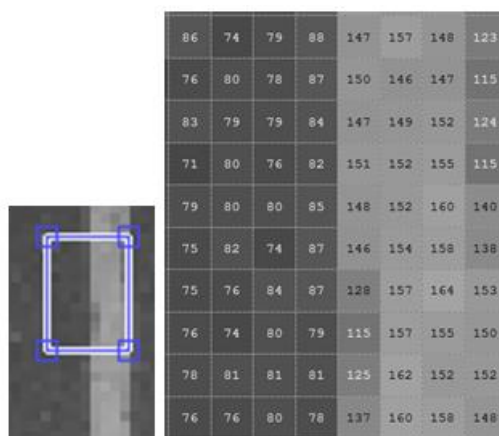
Figura 31 – Defeito tipo ARC



Fonte: Produção do próprio autor.

A descontinuidade do nível de cinza pode ser melhor visualizada na Figura 32, que mostra um zoom na região de transição do defeito (região interna ao retângulo azul), exibindo os valores do nível de cinza de cada pixel.

Figura 32 – Descontinuidade no nível de cinza entre colunas de imagem com defeito tipo ARC

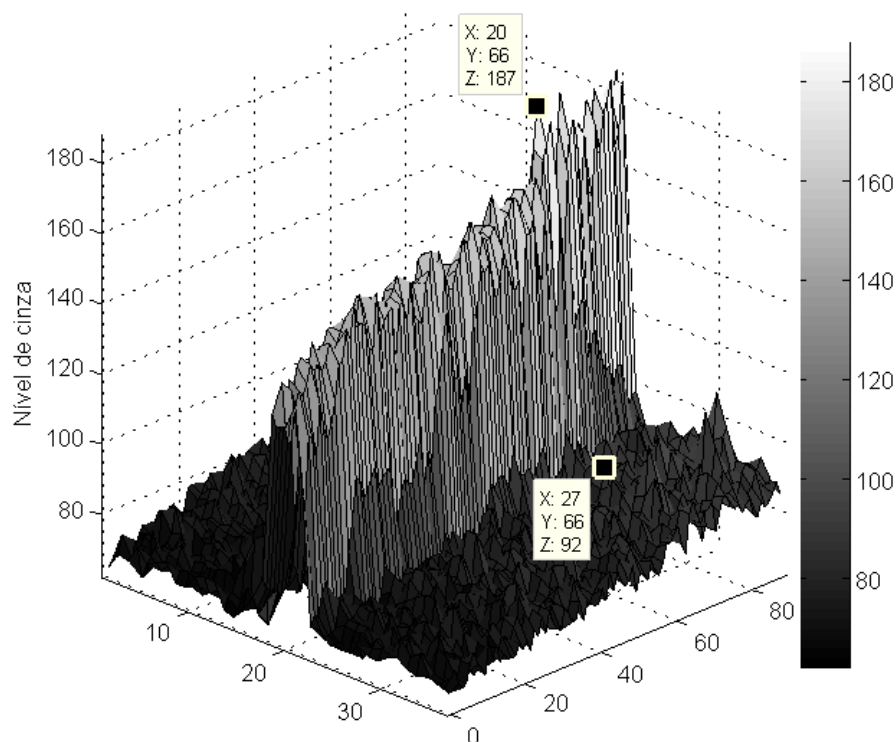


Fonte: Produção do próprio autor.

A partir do ponto em que o defeito aparece, o nível de cinza aumenta de forma abrupta, passando de valores próximos de 90 para valores acima de 120.

Outra forma de ver a descontinuidade citada é analisar a Figura 33, que mostra um gráfico em três dimensões da imagem com o defeito.

Figura 33 – Visão 3D de imagem contendo defeito tipo ARC



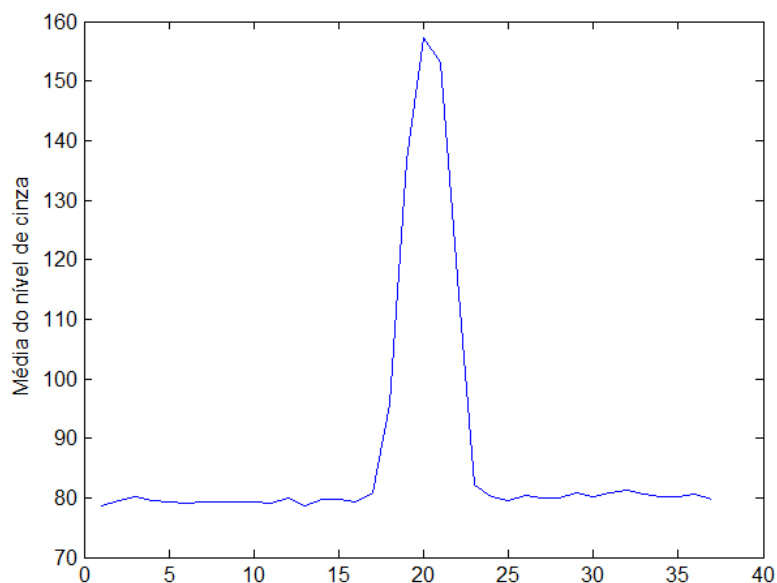
Fonte: Produção do próprio autor.

Uma imagem digital representada por 8 bits pode ter até 256 níveis de cinza, variando de 0 (preto) até 255 (branco). No espaço de coordenadas (x,y) , uma imagem digital tem seus valores distribuídos em linhas e colunas, conforme citado no item 2.2.

Ao calcular a média do nível de cinza em cada coluna das imagens, defeitos verticais podem ficar evidentes e serem melhor detectados, ao passo que com a média do nível de cinza das linhas é fácil detectar descontinuidades horizontais.

A Figura 34 mostra o gráfico da média do nível de cinza calculado para cada coluna da imagem contendo o defeito ARC que foi exibida na Figura 31.

Figura 34 – Média do nível de cinza por coluna



Fonte: Produção do próprio autor.

Como esperado, pelo fato de o defeito ser mais claro que o restante da imagem, há uma descontinuidade com um pico positivo no valor médio do nível de cinza das colunas. Com este resultado, a tarefa de detectar a região defeituosa de forma automática por um algoritmo torna-se possível, por exemplo, buscando encontrar o ponto de máximo na curva mostrada na Figura 34. Visualmente, tal região está compreendida aproximadamente entre os pixels 15 e 25 (eixo horizontal). A técnica de detecção automática de regiões defeituosas será explicada mais adiante, na seção 4.3.

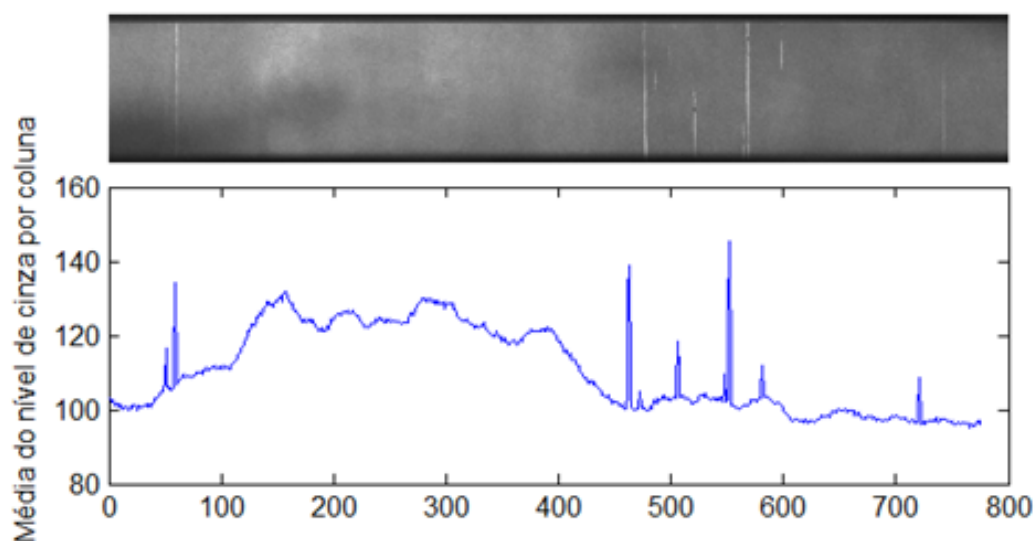
O teste citado acima foi feito para diversas imagens onde os defeitos já haviam sido manualmente separados e as imagens recortadas, ou seja, para aquelas imagens que serviram como templates nos testes de correlação. Terminada essa etapa, foram feitos testes análogos, porém em imagens completas contendo diversas amostras de regiões de tiras de aço defeituosas.

A escolha do cálculo da média por colunas, e não por linhas, foi feita pela seguinte razão: as imagens utilizadas neste trabalho possuem, em geral, entre 108x760 pixels e 200x840 pixels. Sabe-se que quão menor for a quantidade de amostras do vetor utilizado para o cálculo da média, mais evidentes serão os valores que fogem da normalidade. Como em todas as

imagens o número de pixels contido nas linhas é mais de quatro vezes maior que o número de pixels contidos nas colunas, foi escolhido trabalhar com a média por colunas.

A Figura 35 mostra um exemplo de imagem contendo diversas amostras de defeitos tipo *ARC*, além do gráfico do resultado contendo os valores médios de nível de cinza em cada coluna da imagem.

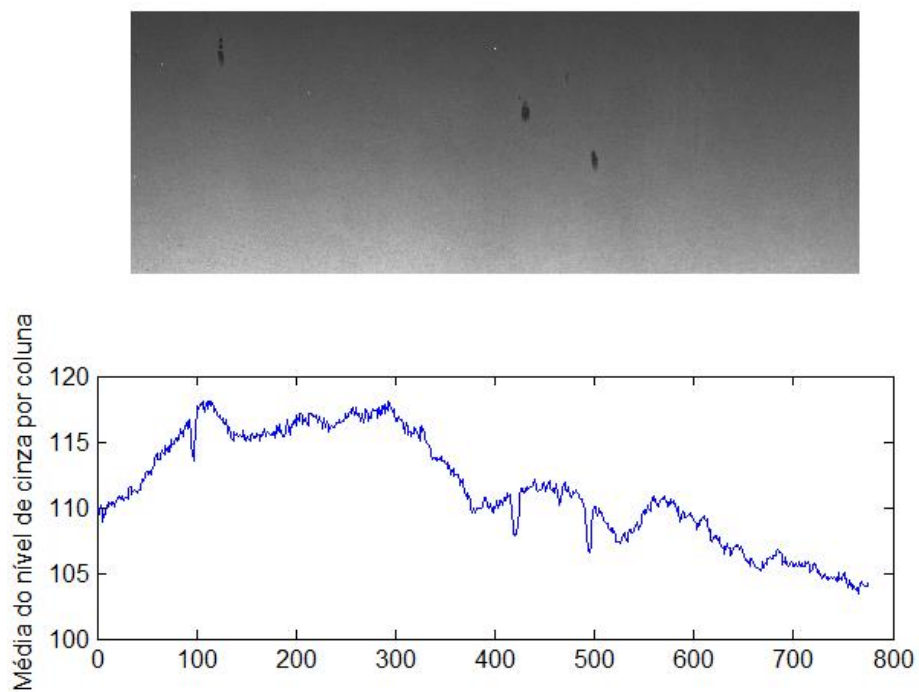
Figura 35 – (a) Imagem com defeitos tipo *ARC*. (b) Gráfico com valores de média de nível de cinza por coluna



Fonte: Produção do próprio autor.

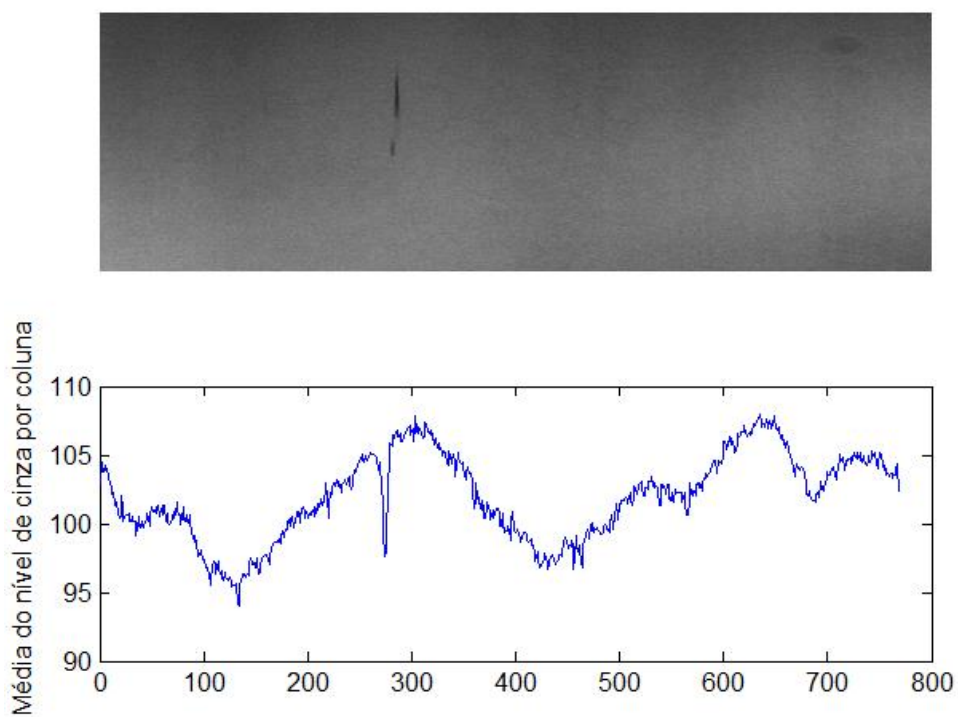
Verifica-se que sempre que há um defeito, há também um pico no valor médio do nível de cinza da coluna. Nesse caso, os picos são positivos pois o defeito é mais claro que a tira. Em casos em que o defeito é mais escuro que a tira, os picos são negativos, como mostram as Figuras 36 e 37 para defeitos tipo *CA* e *ESF*, respectivamente.

Figura 36 – (a) Imagem com defeitos tipo CA (b) Gráfico com valores de média de nível de cinza por coluna



Fonte: Produção do próprio autor.

Figura 37 - Imagem com defeito tipo ESF (b) Gráfico com valores de média de nível de cinza por coluna



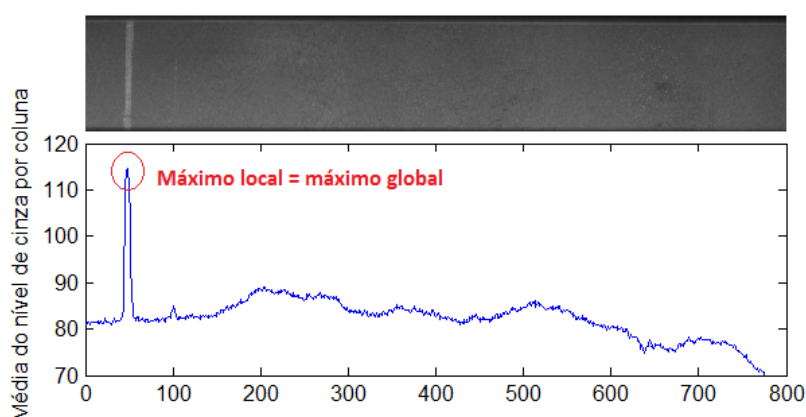
Fonte: Produção do próprio autor.

Com os resultados obtidos, localizar automaticamente a região onde o defeito se encontra seria uma tarefa simples, por exemplo procurando por valores máximos (para defeitos claros) ou mínimos (para defeitos escuros) no gráfico da média do nível de cinza.

Contudo, mesmo tendo selecionado imagens em que a uniformidade de iluminação era boa, ainda assim surgiu um problema: em fotos onde a iluminação da imagem não é perfeitamente uniforme, máximos e mínimos globais acabam mascarando os defeitos das tiras.

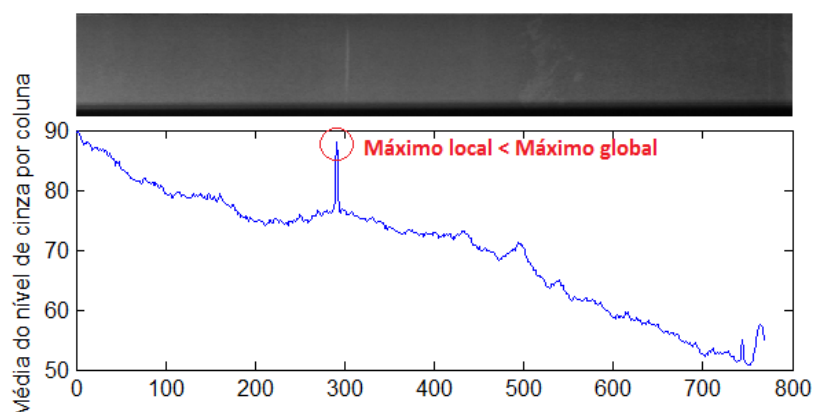
Para exemplificar o problema, as Figuras 38 e 39 mostram a diferença entre dois casos: primeiro, uma imagem com iluminação uniforme, onde o defeito destaca-se no gráfico com um valor máximo global; segundo, uma imagem com iluminação menos uniforme, situação em que o algoritmo apontaria um defeito em uma região onde não há anomalia alguma.

Figura 38 – (a) Imagem com iluminação uniforme (b) Gráfico com máximo local igual ao máximo global



Fonte: Produção do próprio autor.

Figura 39 – (a) Imagem com iluminação descontínua (b) Gráfico com máximo local mascarado pelo máximo global



Fonte: Produção do próprio autor.

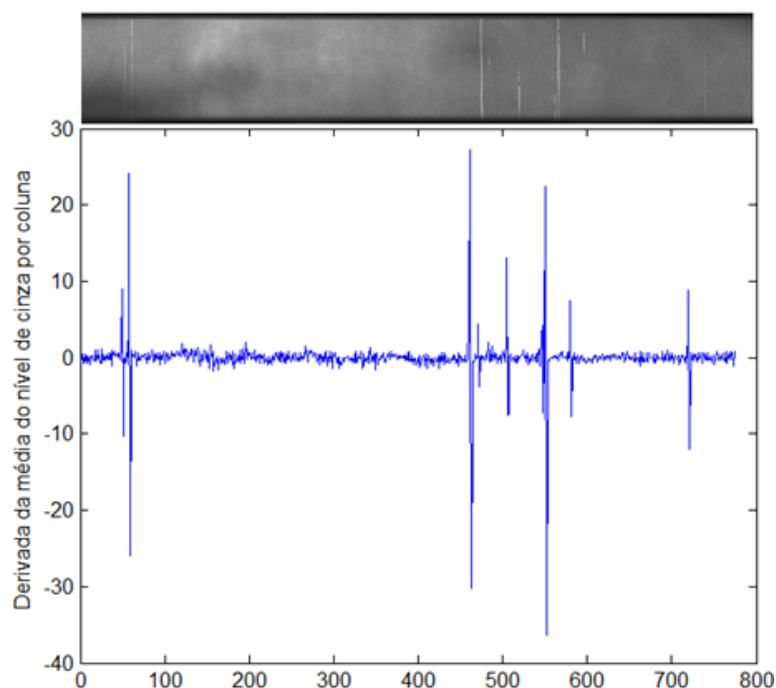
A forma encontrada para solucionar o problema foi usar a derivada da média do nível de cinza por coluna, para assim detectar melhor regiões de transições mais discrepantes.

4.3 Derivada da média do nível de cinza por coluna

Defeitos criam regiões de descontinuidade no nível de cinza das imagens. Uma forma de detectar descontinuidades é utilizar a derivada de primeira ordem, calculada pela diferença entre o nível de cinza do pixel $i+1$ e do pixel i (GONZALEZ, 2008).

A Figura 40 mostra o resultado gráfico da derivada da média do nível de cinza para cada coluna de uma imagem contendo amostras de defeito tipo ARC.

Figura 40 – (a) Imagem com defeitos tipo ARC (b) Gráfico da derivada da média do nível de cinza por coluna



Fonte: Produção do próprio autor.

Assim como esperado, as regiões de transição ficaram evidentes utilizando a derivada. Com este resultado foi possível criar um algoritmo que detectasse os defeitos em estudo, *ARC*, *CA* e *ESF* de maneira correta.

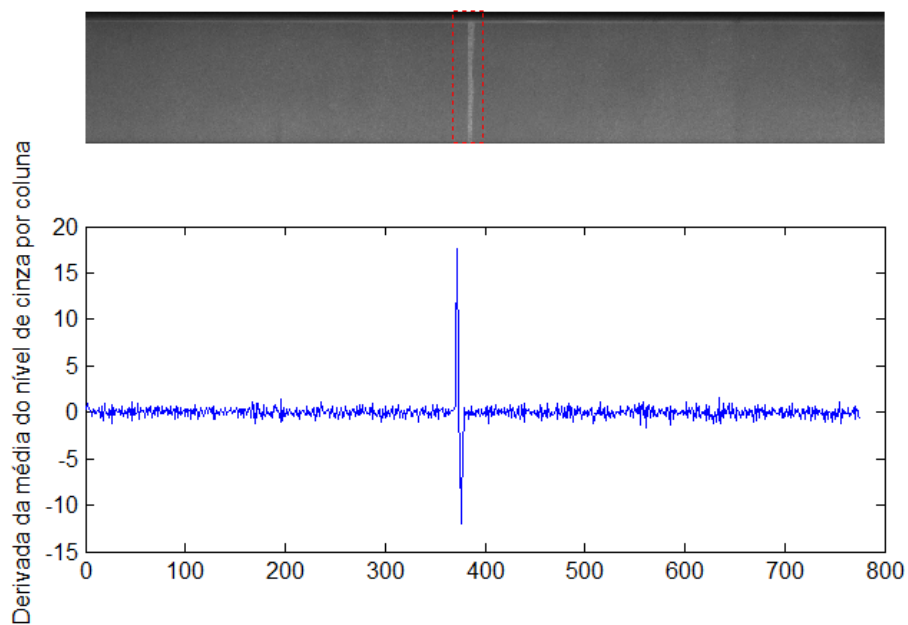
Foi criado um script no Matlab cujo funcionamento está descrito abaixo (código DETECTA_SIMPLES.M disponível no Apêndice).

- 1) Carregar imagem teste;
- 2) calcular a média do nível de cinza para cada coluna da imagem;
- 3) calcular a derivada da média do nível de cinza para cada coluna da imagem;
- 4) procurar o ponto de máximo valor em módulo e armazenar sua coordenada;
- 5) marcar um retângulo na região centrada no ponto encontrado na quarta etapa.

A largura do retângulo da marcação foi determinada empiricamente, conforme um valor que facilitasse a visualização do resultado, sendo fixada em 30 pixels, exceto para defeitos muito próximos de alguma borda da imagem. Nesse caso a largura é limitada à borda.

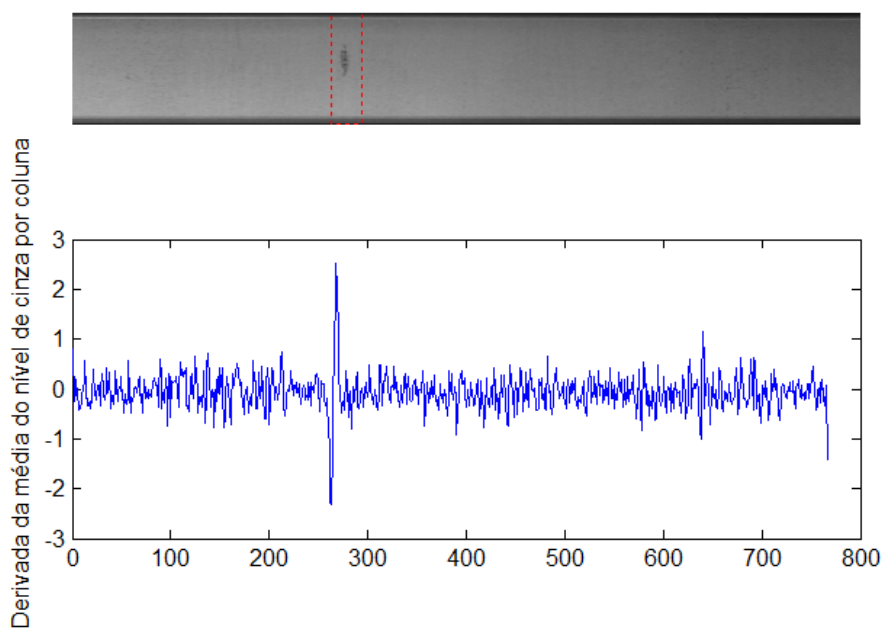
As Figuras 41, 42 e 43 mostram alguns resultados dos testes feitos com o algoritmo citado acima.

Figura 41 – Defeito tipo ARC localizado automaticamente.



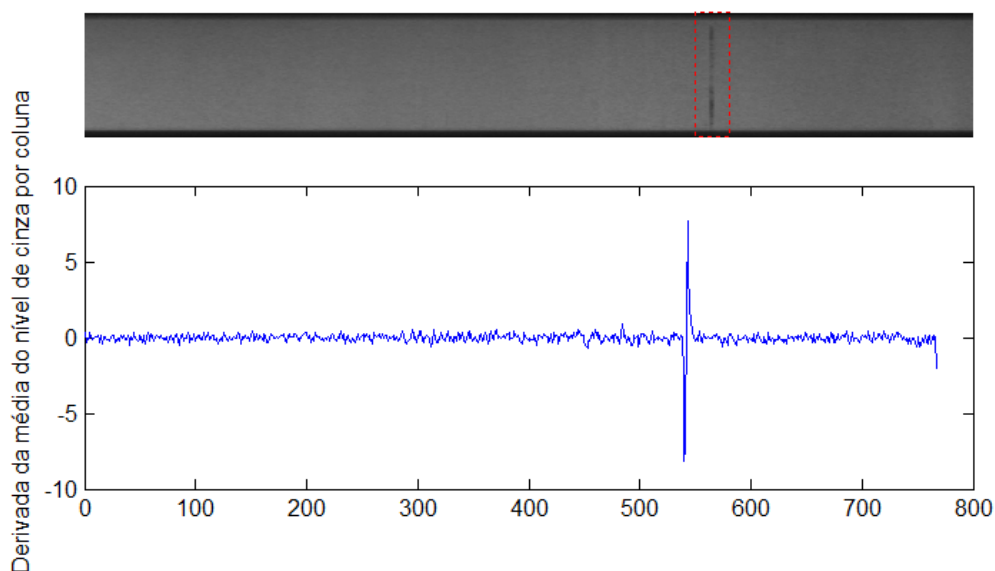
Fonte: Produção do próprio autor.

Figura 42 – Defeito tipo CA localizado automaticamente



Fonte: Produção do próprio autor.

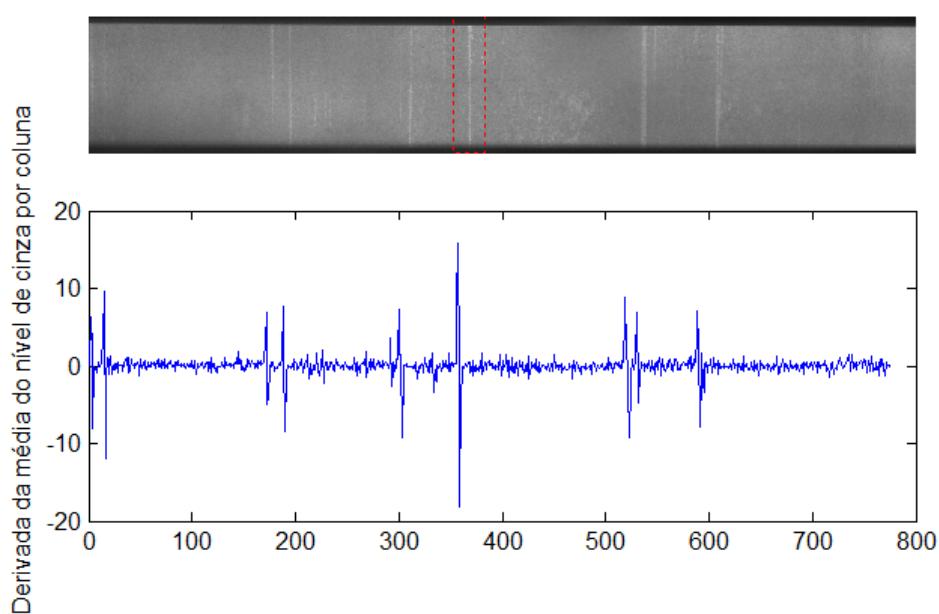
Figura 43 – Defeito tipo ESF localizado automaticamente



Fonte: Produção do próprio autor.

Apesar do algoritmo localizar corretamente os defeitos, ainda havia uma melhoria a ser feita, pois a detecção estava limitada a um defeito por imagem, como é o caso exemplificado na Figura 44, em que apenas o defeito mais evidente é marcado, já os demais são ignorados.

Figura 44 – (a) Apenas um defeito é localizado automaticamente na imagem. (b) Gráfico exibindo vários picos de derivada de média do nível de cinza



Fonte: Produção do próprio autor.

Para que fossem localizados múltiplos defeitos por imagem, sem acrescentar falsos positivos, foi feita uma melhoria no algoritmo, explicada no item 4.4.

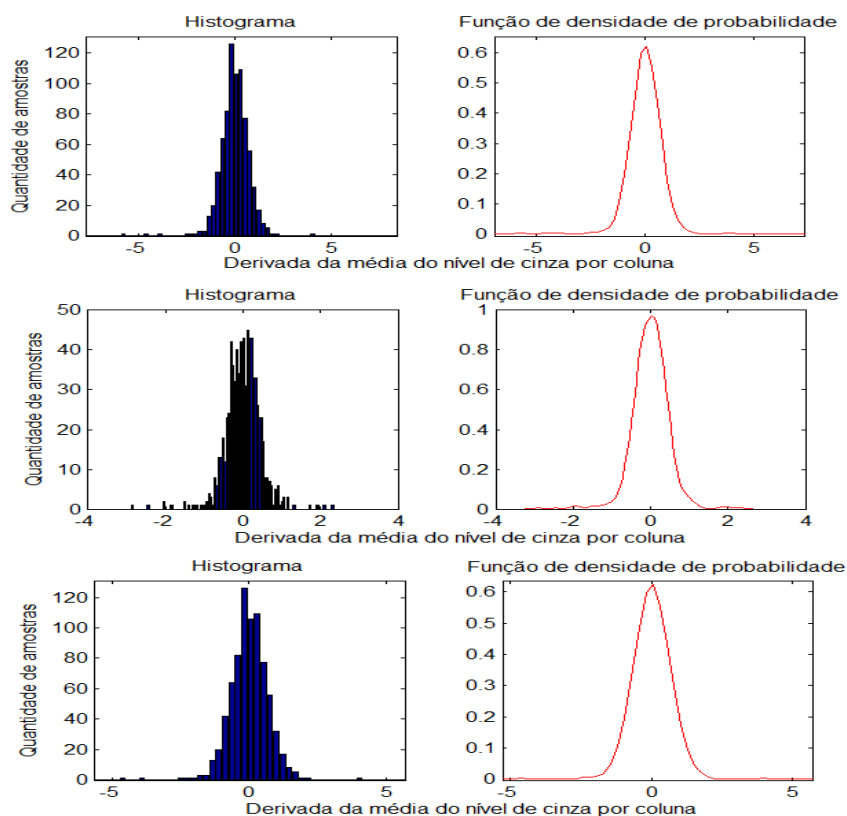
4.4 Algoritmo criado para detecção de múltiplos defeitos

Conforme explicado anteriormente, analisando os gráficos da derivada da média do nível de cinza por colunas, percebe-se que os dados seguem um determinado padrão e apenas onde há defeitos é que existem diferenças maiores nos valores. Os histogramas desses gráficos abrem portas para uma forma diferente de avaliar os dados.

4.4.1 Distribuição das amostras

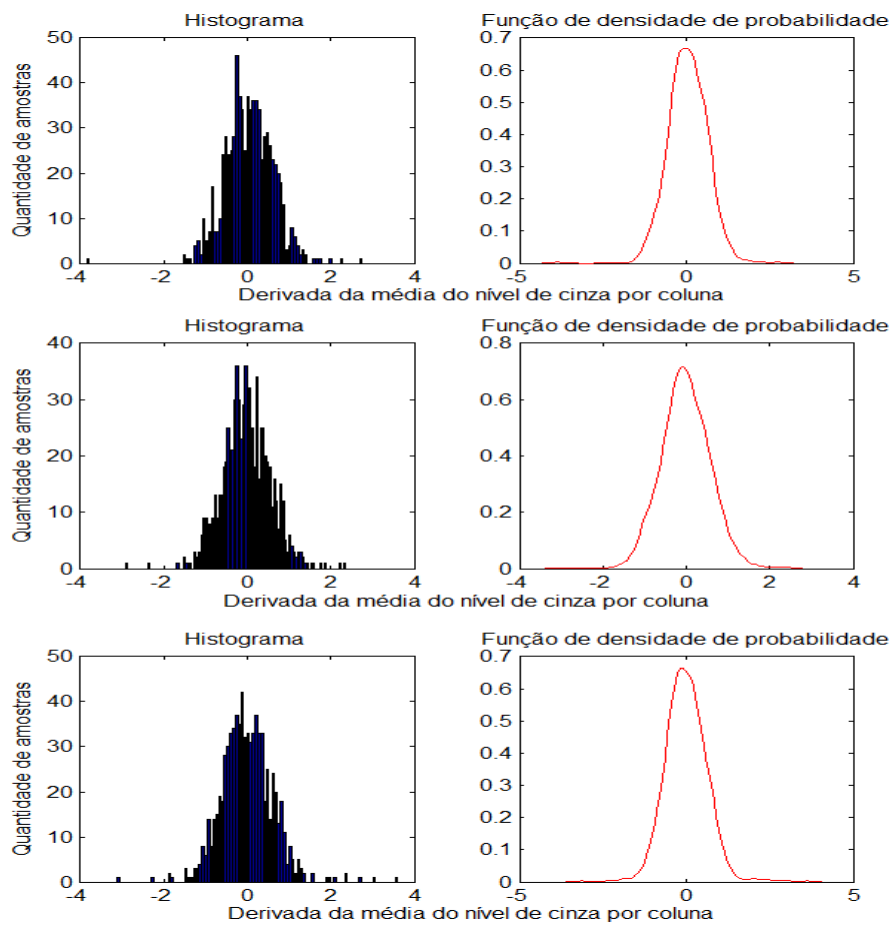
As Figuras 45, 46 e 47 mostram alguns exemplos de histogramas de dados obtidos de imagens contendo múltiplos defeitos tipo *ARC*, *CA* e *ESF*. Para cada histograma há também as respectivas funções de densidade de probabilidade.

Figura 45 – Histograma e função de densidade de probabilidade para imagens contendo defeito tipo *ARC*



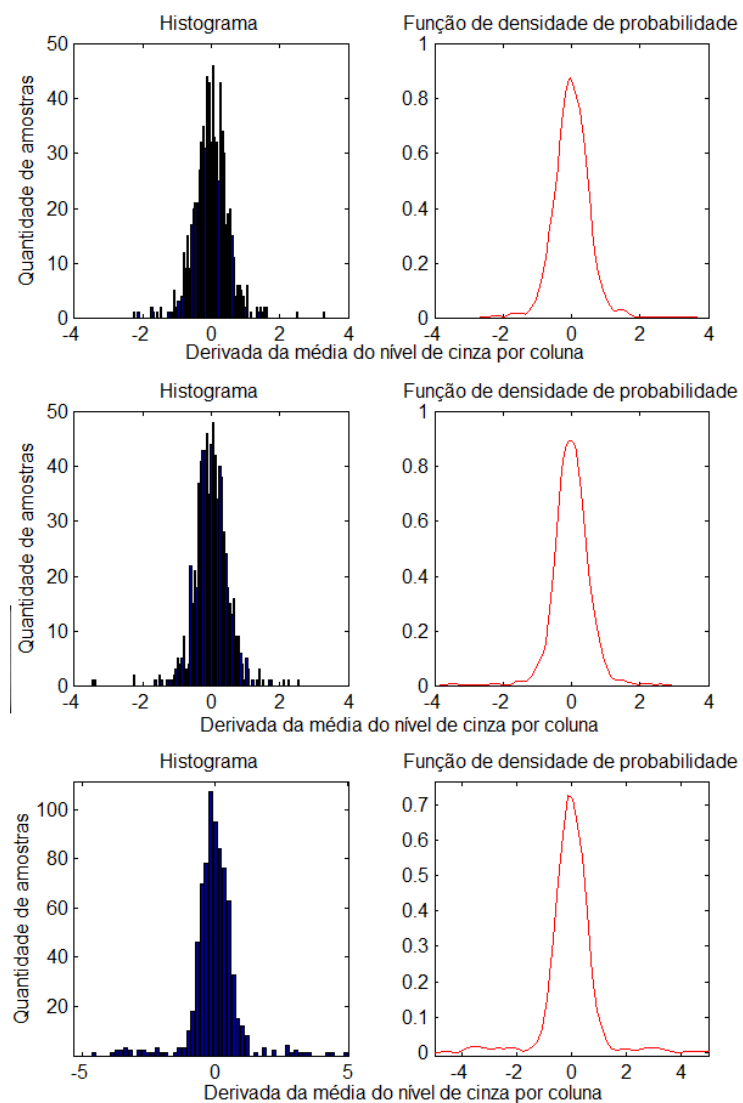
Fonte: Produção do próprio autor.

Figura 46 – Histograma e função de densidade de probabilidade para imagens contendo defeitos tipo CA



Fonte: Produção do próprio autor.

Figura 47 – Histograma e função de densidade de probabilidade para imagens contendo defeitos tipo ESF



Fonte: Produção do próprio autor.

Os gráficos de função de densidade de probabilidade possuem forma de ‘sino’ e são simétricos em relação à média, que para a maioria das imagens é muito próxima de zero, como mostra a Tabela 1 para cada uma das 90 imagens analisadas (30 para cada defeito).

Tabela 1 – Valores de média dos dados para todas as imagens utilizadas durante os testes

Imagem	Média		
	Arranhão Claro	Carepa Primária	Esfoliação
1	-0,045	0,044	-0,026
2	-0,032	0,025	-0,021
3	-0,008	-0,024	-0,020
4	-0,008	-0,030	-0,023
5	0,060	0,034	-0,018
6	0,072	0,030	-0,022
7	0,027	0,049	0,007
8	-0,021	0,027	0,007
9	-0,118	0,044	-0,001
10	-0,041	0,038	-0,037
11	-0,017	0,005	-0,009
12	-0,014	-0,019	-0,006
13	0,009	-0,024	-0,001
14	0,006	-0,008	0,000
15	-0,011	-0,065	-0,012
16	0,033	0,004	0,006
17	0,016	-0,021	-0,004
18	0,034	-0,003	0,004
19	0,000	-0,003	0,001
20	-0,064	-0,008	-0,003
21	-0,001	0,010	-0,003
22	-0,007	-0,013	0,016
23	0,032	-0,020	-0,024
24	0,032	-0,019	-0,015
25	0,015	-0,022	-0,017
26	0,015	-0,018	-0,022
27	0,017	-0,005	-0,028
28	-0,051	-0,004	-0,019
29	-0,002	-0,018	-0,017
30	0,000	-0,004	-0,019

Fonte: Produção do próprio autor.

Para avançar no projeto, analisando as características de forma, simetria e média para as funções de densidade de probabilidade, foi considerado que as distribuições são do tipo normal.

Sabendo o tipo de distribuição dos dados, é possível calcular o intervalo de confiança (limites superior e inferior) dentro do qual encontra-se, estatisticamente, determinado percentual do conjunto de amostras. O cálculo do intervalo é feito utilizando o vetor de dados a ser analisado, a média, o desvio padrão e um coeficiente α , que representa o percentual de amostras que estão fora do intervalo calculado.

Para defeitos mais claros que o fundo da imagem, dados acima do limite superior representam o início de um defeito, ao passo que dados abaixo do limite superior representam o fim da região defeituosa. Para defeitos mais escuros vale a relação contrária.

Foi criada uma rotina para calcular o intervalo de confiança e a partir disso detectar as falhas nas tiras. Amostras acima do limite superior do intervalo são tratadas como defeitos nas tiras (amostras abaixo do limite inferior foram ignoradas para evitar redundância). Dessa forma foi possível detectar múltiplos defeitos em uma única imagem.

O funcionamento do algoritmo de detecção múltipla está descrito abaixo (código DETECTA_MULTIPLOS.M disponível no Apêndice).

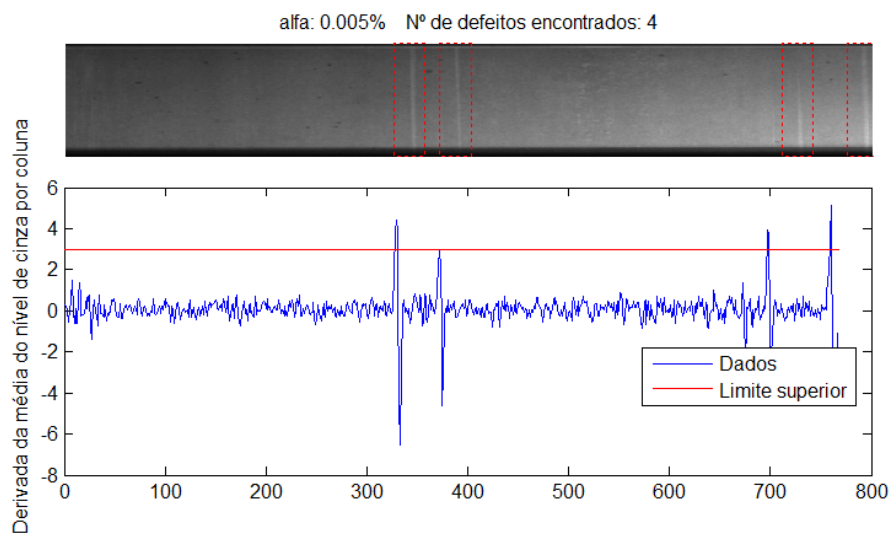
- 1) Carregar imagem teste;
- 2) calcular a média do nível de cinza para cada coluna da imagem;
- 3) calcular a derivada da média do nível de cinza para cada coluna da imagem;
- 4) calcular a média e o desvio padrão da derivada, segundo (5) e (6), supondo, em ambos os casos que x_i compõe um vetor de amostras já derivadas;
- 5) escolher o valor do parâmetro α ;
- 6) calcular o limite superior do intervalo de confiança;
- 7) armazenar amostras que estejam fora do limite superior;
- 8) marcar um retângulo na região centrada nos pontos encontrados na sétima etapa.

A Média e o Desvio-Padrão foram calculados utilizando as expressões citadas no item 2.9.

Os valores utilizados para o parâmetro α foram escolhidos empiricamente durante os testes, conforme a qualidade dos resultados. As Figuras 48, 49 e 50 mostram resultados de testes

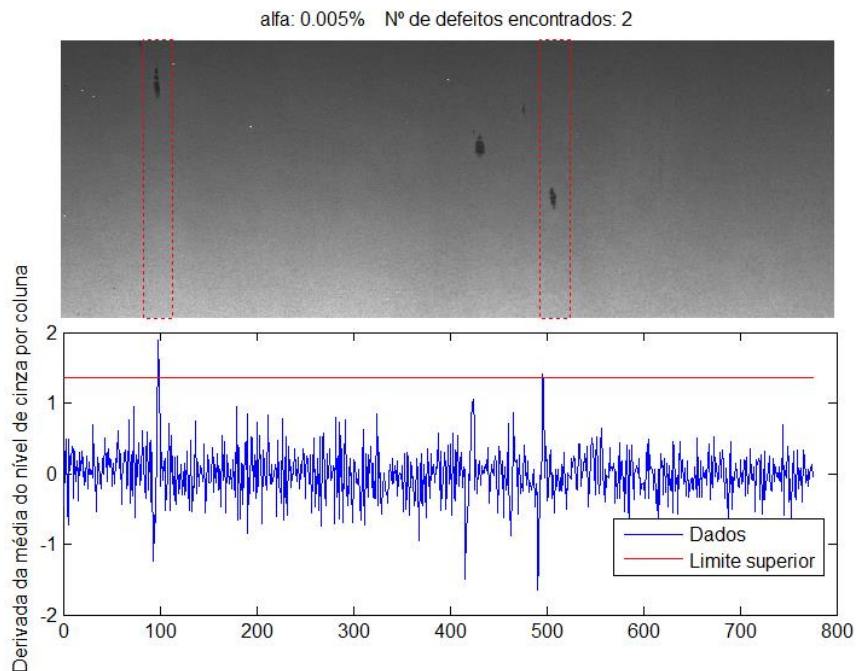
para α igual a 0,005%, ou seja, considerando que estatisticamente 99,995% das amostras não representam defeitos nas tiras.

Figura 48 – Detecção de múltiplos defeitos tipo ARC



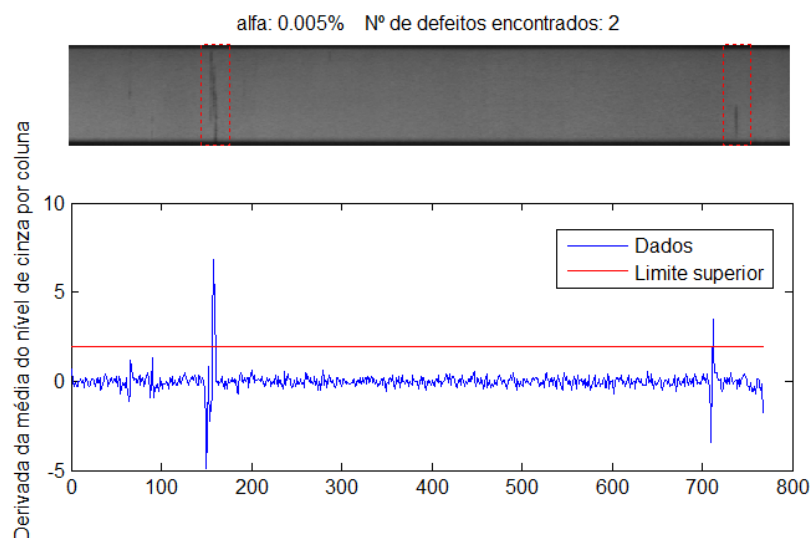
Fonte: Produção do próprio autor.

Figura 49 – Detecção de múltiplos defeitos tipo CA



Fonte: Produção do próprio autor.

Figura 50 – Detecção de múltiplos defeitos tipo ESF

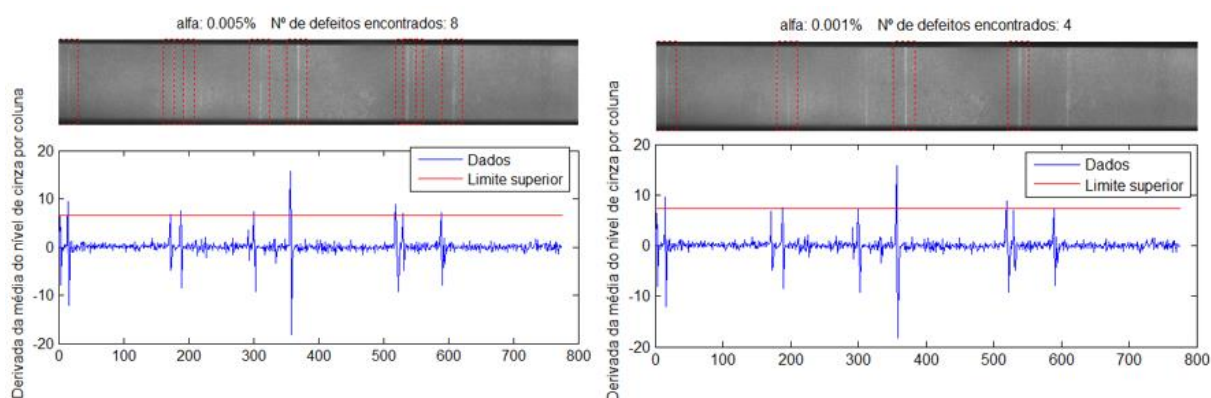


Fonte: Produção do próprio autor.

A sensibilidade do algoritmo varia de acordo com o valor do termo α . Quanto menor for o valor do coeficiente, menos sensível é o algoritmo, fazendo com que sejam detectados apenas os defeitos mais evidentes.

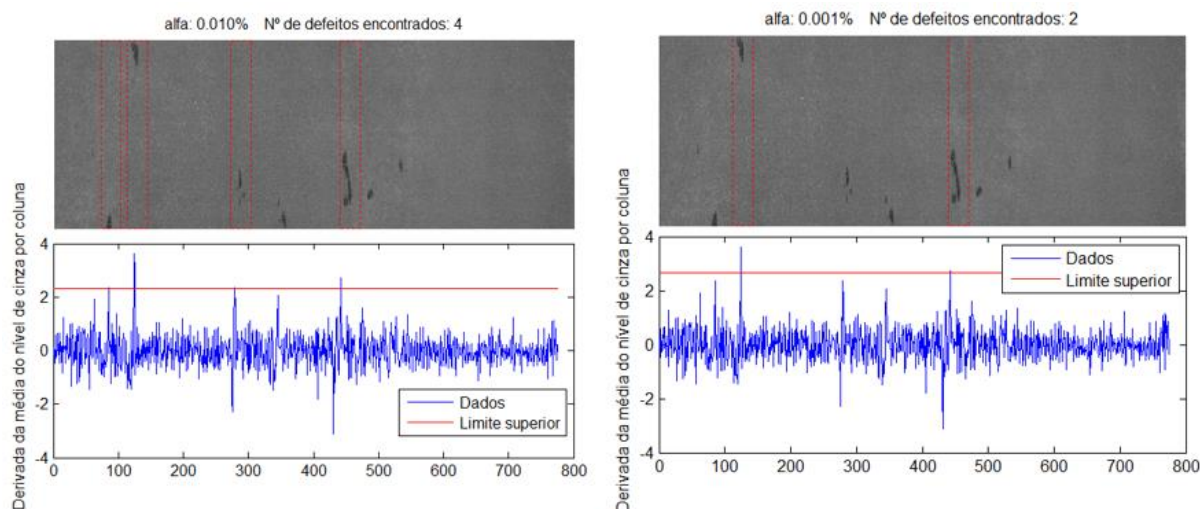
As Figuras 51, 52 e 53 mostram diferentes resultados em testes com a versão final do algoritmo para imagens iguais, variando apenas o valor do coeficiente α .

Figura 51 – Comparativo de sensibilidade do algoritmo para defeitos tipo ARC



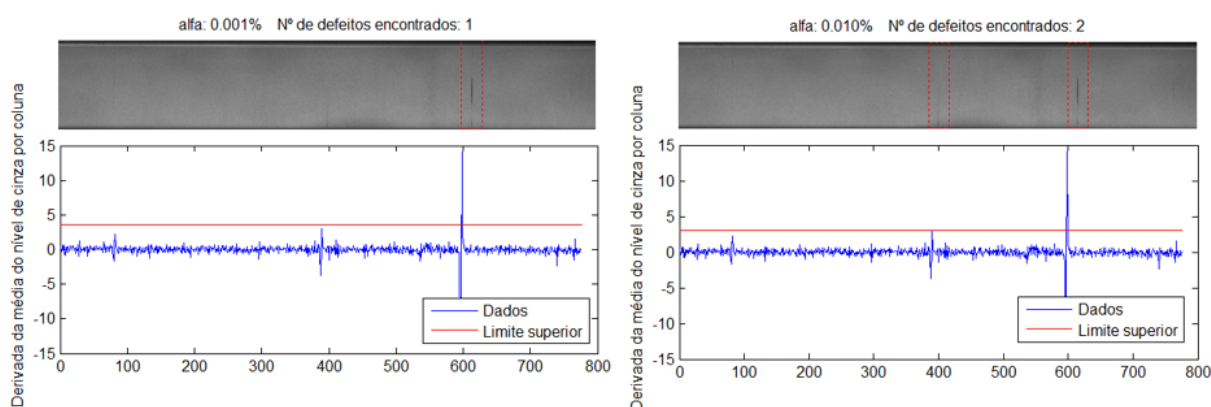
Fonte: Produção do próprio autor.

Figura 52 – Comparativo de sensibilidade do algoritmo para defeitos tipo CA



Fonte: Produção do próprio autor.

Figura 53 –Comparativo de sensibilidade do algoritmo para defeitos tipo ESF



Fonte: Produção do próprio autor.

Portanto, com o algoritmo final foi possível detectar corretamente defeitos de três classes distintas nas tiras. Terminada essa etapa, foi dado início ao processo de classificação automática dos defeitos detectados.

4.5 Classificação

Terminada a etapa de detecção das falhas, foi iniciada a preparação dos dados para que fosse possível classificar os defeitos. O primeiro passo foi isolar a região que contém a falha e processá-la separadamente, a fim de economizar esforço computacional.

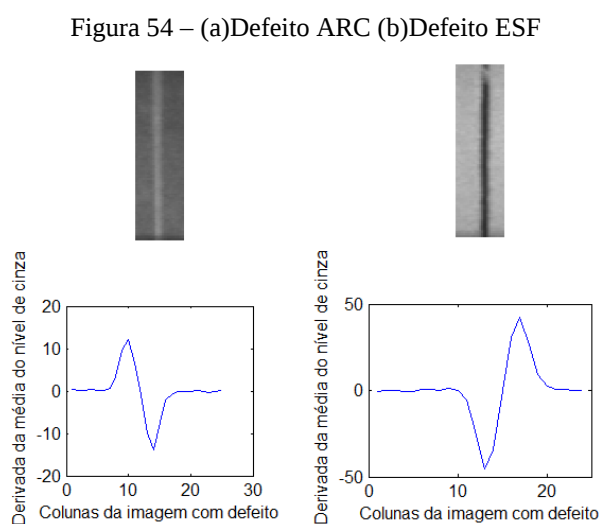
A principal diferença entre os defeitos aqui estudados é em relação ao nível de cinza: uma classe contém defeitos mais claros que o fundo das imagens (defeitos tipo *ARC*); as outras duas classes (*CA* e *ESF*) são caracterizadas por defeitos mais escuros que o restante da imagem.

Portanto, após acessar apenas a região da imagem que contém o defeito, o próximo passo para a classificação foi diferenciar defeitos claros de defeitos escuros.

4.5.1 Classificação entre defeitos claros e escuros

O algoritmo criado para diferenciação entre defeitos claros e escuros é simples: calcula-se globalmente o valor máximo e o valor mínimo para a derivada da média do nível de cinza da região defeituosa; caso a coordenada onde ocorre o valor máximo esteja à esquerda da coordenada onde ocorre o valor mínimo, trata-se de um defeito mais claro que o fundo da imagem; caso contrário, é um defeito escuro.

A Figura 54 ilustra as duas situações citadas.



Fonte: Produção do próprio autor.

Com a análise acima foi possível concluir a classificação de defeitos tipo *ARC*, pois esta é a única classe analisada de defeito mais claro que o fundo da imagem.

Para defeitos escuros, existem duas classes possíveis, *CA* e *ESF*. Para diferenciá-las, foi necessário seguir algumas etapas a mais, começando pela binarização.

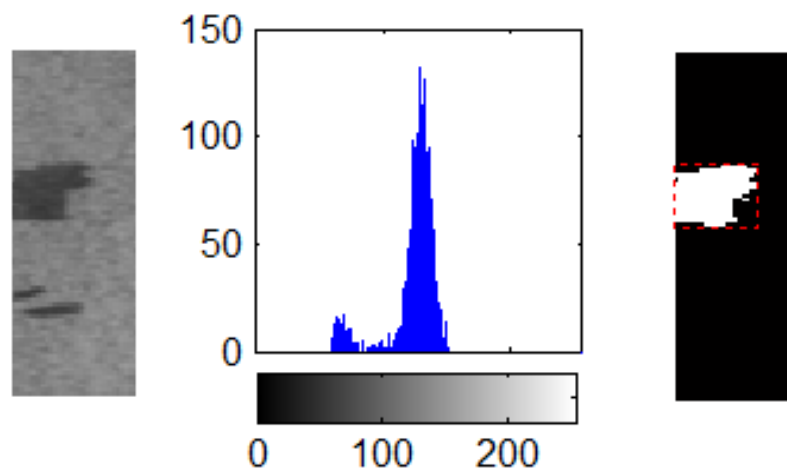
4.5.2 Binarização para definir ROI

Uma forma interessante de classificar imagens é analisando atributos da ROI. Para que pudessem ser calculadas algumas características comuns para as duas classes restantes de defeitos e diferenciá-las, iniciou-se pelo processo de binarização.

A abordagem escolhida para binarização foi o método de Otsu, pois este apresenta eficácia comprovada em casos onde a iluminação da imagem não é perfeitamente uniforme (GONZALEZ, 2008).

A Figura 55 mostra o resultado obtido pelo método de Otsu para uma imagem com Carepa.

Figura 55 – (a)Imagem contendo defeito CA (b)Histograma da imagem com defeito CA (c)Resultado da binarização pelo método de Otsu



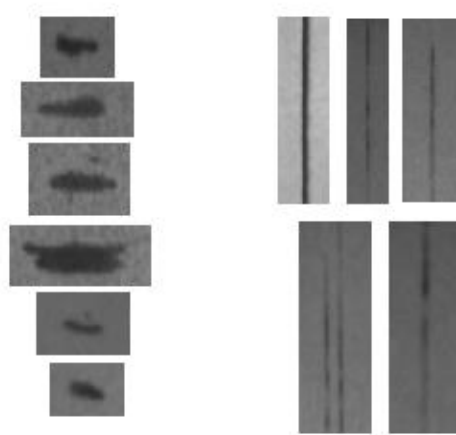
Fonte: Produção do próprio autor.

Terminada a etapa de binarização das imagens contendo defeitos, foi passado para a próxima etapa, referente aos cálculos dos atributos necessários para a construção do classificador.

4.5.3 Obtenção de atributos para montar classificador

Com as imagens binarizadas, o próximo passo foi calcular características que possibilitassem a diferenciação entre as classes de defeitos CA e ESF. Por inspeção visual, verificou-se que os defeitos tipo CA possuem formato mais arredondado que os ESF, além da orientação predominantemente horizontal, ao passo que defeitos tipo ESF são mais alongados e predominantemente verticais. A Figura 56 ilustra essas características.

Figura 56 – (a) Defeitos tipo CA (b) Defeitos tipo ESF



Fonte: Produção do próprio autor.

A Figura 57 mostra alguns defeitos tipo CA e ESF binarizados.

Figura 57 – (a) Defeitos CA binarizados (b) Defeitos ESF binarizados

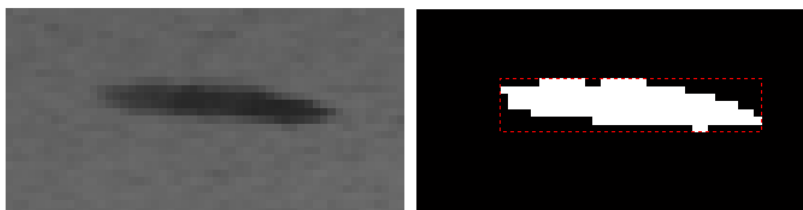


Fonte: Produção do próprio autor.

Após a binarização, foi utilizado um algoritmo para detectar o objeto de interesse na imagem binária. Foi selecionado para análise apenas o maior objeto encontrado em cada imagem desse

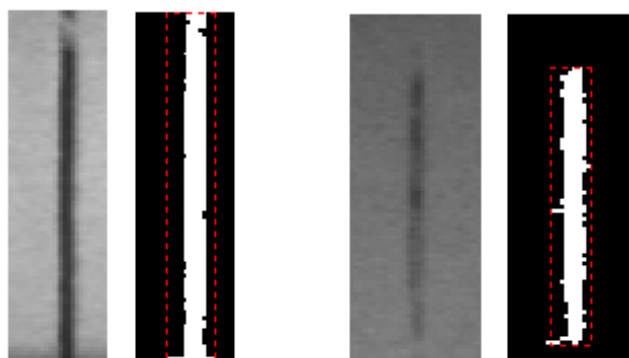
tipo. As Figuras 58 e 59 mostram alguns resultados da binarização seguida de detecção de maior objeto.

Figura 58 – Carepa binarizada e detectada para cálculo de atributos



Fonte: Produção do próprio autor.

Figura 59 – Esfoliações binarizadas e detectadas para cálculo de atributos



Fonte: Produção do próprio autor.

Os atributos utilizados para montar o classificador foram: *orientação* e *relação entre largura e altura*. A orientação é medida em relação ao eixo horizontal (0°), variando de -90° até $+90^\circ$. Um objeto é considerado totalmente horizontal quando sua orientação é igual a 0° . As medidas de largura e altura do objeto são medidas em pixels. Um objeto é considerado vertical se sua altura for consideravelmente maior que sua largura, ao passo que um objeto horizontal tem sua largura maior.

Com base nesses atributos e nas informações obtidas através de visualização dos defeitos, foi criado um classificador baseado em árvore de decisão.

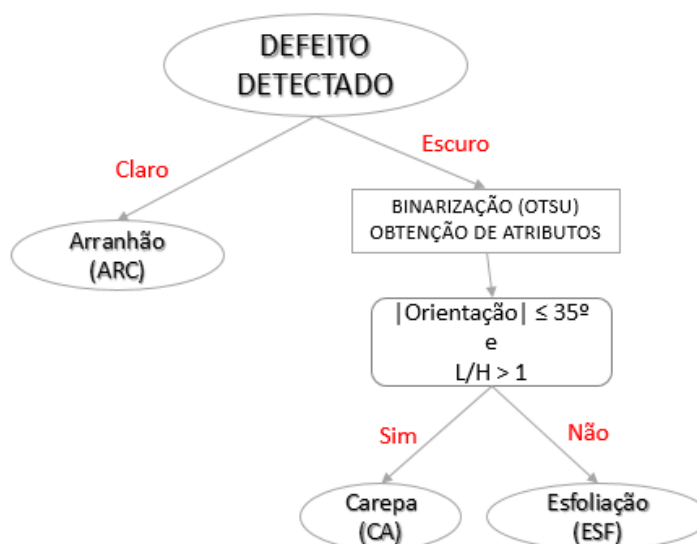
4.5.4 Classificador – Árvore de decisão

Defeitos tipo Carepa, em geral, aparecem orientados horizontalmente e têm sua largura maior que a altura, ao passo que Esfoliações aparecem verticalmente e com uma relação *largura/altura* menor que 1.

Considerando os dados obtidos com 30 amostras de cada defeito (imagens idênticas àsquelas usadas como templates nos testes de correlação) que foram utilizadas para treinamento, foram elaboradas as seguintes regras:

Carepas apresentam módulo da *orientação* menor ou igual a 35° , além de *relação entre largura/altura* maior que 1. Tais valores foram testados e fixados empiricamente, considerando o caso em que foram obtidos os melhores resultados. A Figura 60 mostra como ficou a disposição da árvore de decisão utilizada no algoritmo final de classificação dos defeitos.

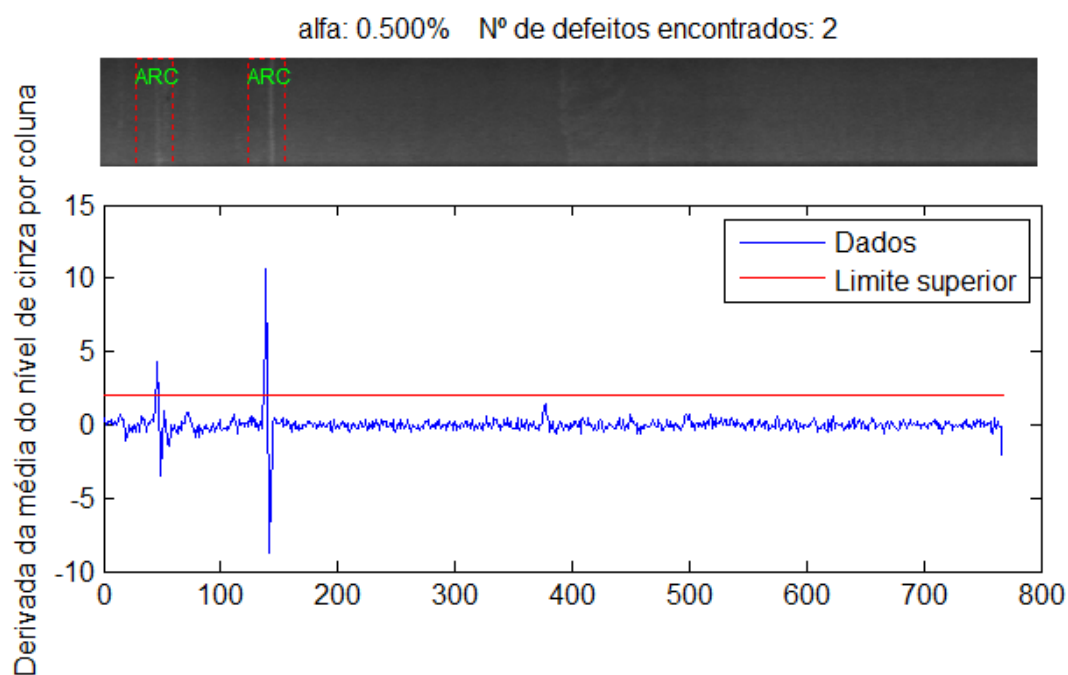
Figura 60 – Árvore de decisão utilizada para classificação dos defeitos



Fonte: Produção do próprio autor.

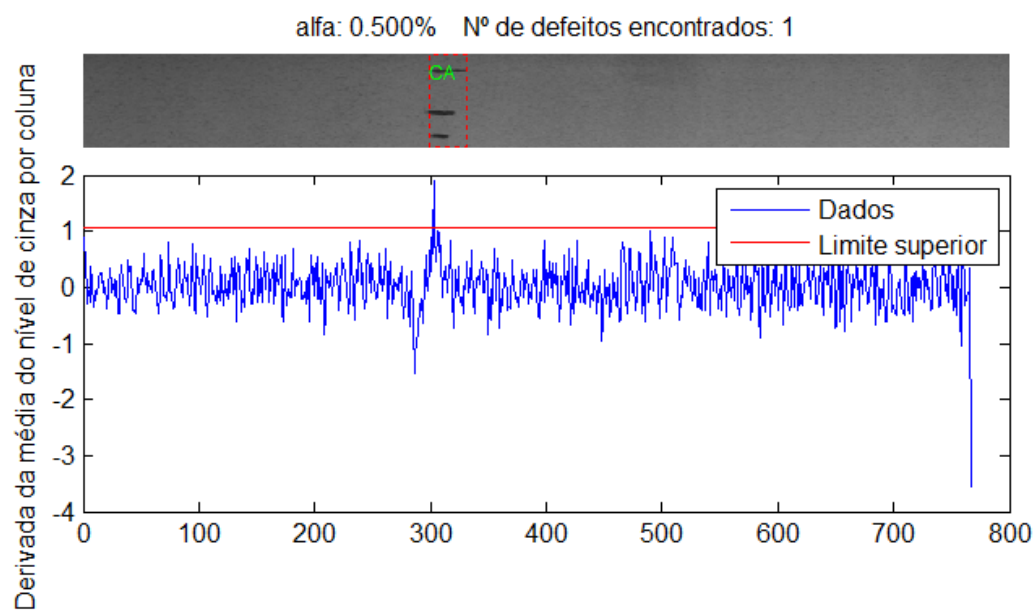
As Figuras 61, 62 e 63 mostram alguns resultados de classificação com imagens do conjunto de testes, para cada tipo de defeito.

Figura 61 – Resultado do classificador para imagem contendo ARC



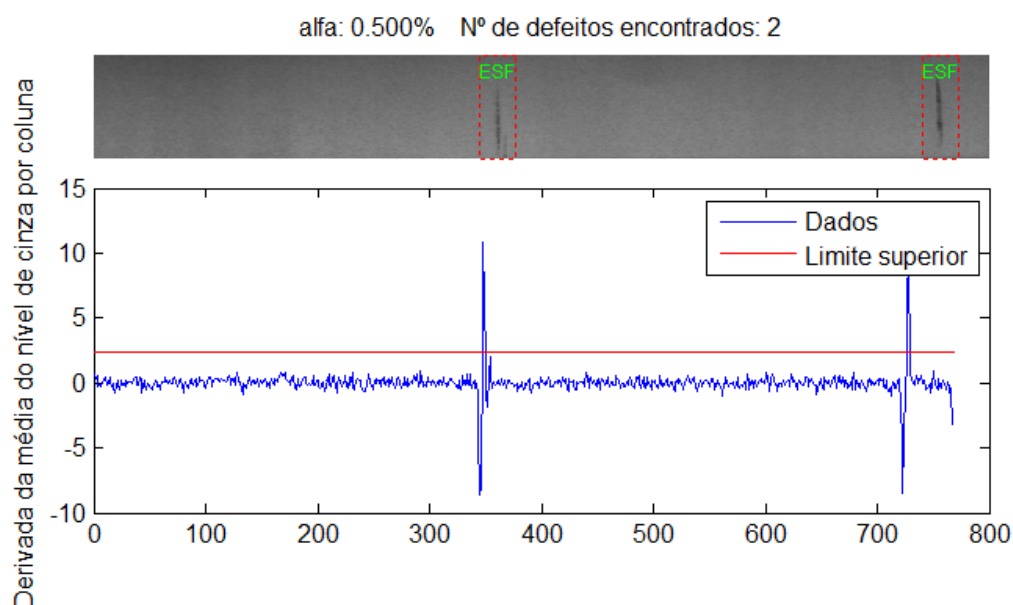
Fonte: Produção do próprio autor.

Figura 62 – Resultado do classificador para imagem contendo CA



Fonte: Produção do próprio autor.

Figura 63 – Resultado do classificador para imagem contendo ESF



Fonte: Produção do próprio autor.

A Tabela 2 mostra o resultado final da classificação para as 90 imagens utilizadas para testes, com os valores de número total de defeitos detectados, quantidade de erros de classificação e, finalmente, o percentual de acerto do classificador para cada defeito analisado.

Tabela 2 – Resultados de classificação dos defeitos.

	ARC	CA	ESF
IMAGENS	30	30	30
DEFEITOS DETECTADOS	95	23	62
ERROS DE CLASSIFICAÇÃO	6	4	4
% ACERTO	93,68421	82,6087	93,54839

Fonte: Produção do próprio autor.

Vale ressaltar que, para o estudo dos atributos utilizados no classificador, foram utilizadas 90 imagens (30 para cada classe de defeito) que faziam parte do grupo de *templates* (os mesmos usados como parâmetro nos testes de correlação). Já na validação dos testes da árvore de decisão, as 90 imagens utilizadas e citadas na Tabela 2 foram diferentes dos templates, ou seja, separadas exclusivamente para testes e registro de resultados.

5 CONCLUSÕES E PROJETOS FUTUROS

A busca automática por defeitos no processo de inspeção de qualidade de tiras de aço é de grande importância, visto que dessa forma é possível melhorar o desempenho da tarefa, aumentando a velocidade da análise e reduzindo a possibilidade de falhas causadas por erros humanos.

Uma etapa primordial para a realização do trabalho foi a seleção das imagens com melhor iluminação e o descarte de imagens de borda. Vale ressaltar que a intenção desse trabalho foi estudar as imagens para que fosse desenvolvida uma técnica capaz de detectar defeitos nas tiras, não necessariamente uma solução comercial, e sim um estudo acadêmico.

5.1 Testes de correlação

Os primeiros testes, envolvendo técnicas de correlação mostraram que apesar de funcionar para muitos casos, sobretudo aqueles em que o *template* e a imagem teste são do mesmo tipo de defeito, há dois problemas nesse tipo de abordagem. Primeiramente, regiões de imagens onde não há defeitos podem ser apontadas pelo algoritmo computacional, caracterizando falsos positivos. Outro problema é o erro de classificação: ao testar *templates* de determinada classe em imagens contendo defeitos de outras classes, há casos em que o algoritmo aponta regiões com altos valores de correlação, indicando portanto um erro de classificação, situação que deve ser evitada.

Abordagens baseadas em correlação são bem usuais em casos onde o processo segue um padrão fixo, por exemplo na fabricação de lotes de produtos visualmente idênticos, como placas de circuito impresso ou rótulos de embalagens, o que não é o caso da fabricação de tiras de aço laminadas a quente.

5.2 Testes estatísticos

Os testes baseados em derivadas e médias de nível de cinza mostraram que é possível detectar defeitos nas superfícies das tiras. Em um primeiro momento, era possível apenas detectar um único defeito por imagem. Após inserir métodos estatísticos na análise dos dados das

imagens, foi possível detectar múltiplos defeitos em uma mesma foto. Além disso, o método adotado possibilitou o ajuste da sensibilidade do algoritmo de detecção, podendo encontrar apenas as falhas mais evidentes, ou então encontrar toda e qualquer possibilidade de defeito. Cabe ao usuário selecionar um valor adequado para ajustar um nível de sensibilidade que atenda às suas necessidades.

5.3 Classificador baseado em árvore de decisão

Os resultados obtidos com o classificador construído nesse trabalho demonstram que os objetivos aqui propostos foram alcançados com sucesso. Para treinamento foram utilizadas imagens idênticas aos templates dos testes de correlação e, para validação, foram utilizadas imagens do conjunto separado para avaliação de resultados. Para duas classes de defeitos (*ARC* e *ESF*), o resultado final de classificação alcançou valores acima de 90% de acertos, ao passo que para uma terceira classe (*CA*) o resultado ficou acima de 80%.

5.4 Projetos futuros

Um passo adiante em relação ao presente trabalho pode ser a classificação de diferentes tipos de defeitos, além dos aqui já analisados. É necessário estudar a viabilidade de montar um classificador para operar com os dados obtidos utilizando novas amostras de defeitos.

Para criar um classificador é necessário primeiramente isolar os defeitos do restante da imagem, o que sem dúvida pode ser a principal dificuldade em analisar novas classes.

Além disso, poderiam ser inseridas imagens de borda nas análises, a fim de ampliar a gama de defeitos a serem detectados, visto que há falhas que ocorrem fundamentalmente nas extremidades laterais das tiras.

REFERÊNCIAS BIBLIOGRÁFICAS

AL-BAYATI, M. **Mammogram Images Thresholding for Breast Cancer Detection Using Different Thresholding Methods**. 2013

BUSSAB, W. MORETTIN, P. **Estatística Básica**. 5 Ed. Editora Saraiva. 2006.

GONRING, M. M. P. **Detecção de fissuras em pelotas de minério de ferro por meio de processamento de imagem**. 2014. Projeto de Graduação, Universidade Federal do Espírito Santo, Vitória, 2013

GONZÁLEZ, R. **Digital Image Processing**. 3. Ed. New Jersey: Pearson Prentice Hall. 2008.

IBOPE. **Modelos Supervisionados de Árvores de Decisão**. 2012. Disponível em: <<http://www.ibope.com.br/pt-br/conhecimento/artigospapers/Documents/paper%20completo%20vfinal.pdf>> Acesso em 14 dez. 2014.

INSTITUTO AÇO BRASIL. **Parque Siderúrgico**, 2013. Disponível em: <<http://www.acobrasil.org.br/site/portugues/aco/parque.asp>>. Acesso em 26 nov. 2014.

LEWIS, J. P. **Fast Normalized Cross-Correlation**. Industrial Light & Magic. 1995

Luo, Q. He, Y. Yu, W. and Zheng, J. 2014. **A Real-time and Cost-Effective Image Processing System for Quality Assurance of Sheet Materials**. Disponível em: <<http://scialert.net/fulltext/?doi=itj.2014.683.689>>. Acesso em: 07 jul. 2014.

MACHADO, M. L. P. **Laminação**. 2006. Disponível em: <<http://www.ebah.com.br/content/ABAAAgOckAK/apres-laminacao>>. Acesso em: 07 jul. 2014.

NEVES, R. P. de O. **Classificação de imagens e obtenção de atributos**, 2000. Disponível em: <http://www.lsi.usp.br/~rponeves/work/Proc_Imagens/>. Acesso em: 11 dez. 2014.

PARSYTEC. **Most advanced inspection technology**, 2009. Disponível em: <http://www.parsytec.de/index.php?id=1264&no_cache=1&L=KWovyYdg#c7588>. Acesso em: 07 jul. 2014.

PORTAL ACTION. **Modelos probabilísticos contínuos**, 2014. Disponível em: <<http://www.portalaction.com.br/content/62-distribui%C3%A7%C3%A3o-normal>>. Acesso em: 11 dez. 2014.

Processos de Fabricação: Laminação. 2008. Disponível em: <<http://www.ebah.com.br/content/ABAAAA4uwAH/aulas-laminacao>>. Acesso em: 07 jul. 2014.

QUINLAN, J. **Programs for machine learning**. San Mateo: Morgan Kaufmann 1993

VASCONCELOS, R. M. de. G. **Uma abordagem Resumida do Formato TIFF**, 2002. Disponível em: <www.cprm.gov.br/publique/media/geotiff.pdf> Acesso em: 11 dez. 2014.

APÊNDICE A – CÓDIGOS

TESTE_CORRELACAO.M

```

clear all;
close all;
clc;

load correlacao_test.mat

%% Procurar Defeitos por correlação

maxAbsCorr=zeros(30,30);
meanCorr=zeros(30,30);
meanAbsCorr=zeros(30,30);

for ii=1:30

    template = template_ARC{ii};
    %     template = template_CA{ii};
    %     template = template_ESF{ii};

    figure
    imshow(template)

    for jj=1:30

        img_teste = img_test_ARC{jj};
        %         img_teste = img_test_CA{jj};
        %         img_teste = img_test_ESF{jj};

        % pause(1);

        c=normxcorr2(template,img_teste); % cálculo de correlação
        [ypeak, xpeak] = find(c==max(c(:))); % coordenadas do ponto de
máxima correlação
        yoffSet = ypeak-size(template,1);
        xoffSet = xpeak-size(template,2);

        hFig = figure;
        hAx = axes;
        figure
        imshow(img_teste,'Parent', hAx); title('Resultado TESTE')
        imrect(hAx, [xoffSet, yoffSet, size(template,2),
size(template,1)]);

```

```

% Determinação dos limites da região de máxima correlação - Begin
xi=xoffSet;
yi=yoffSet;
xf=xi+size(template,2);
yf=yi+size(template,1);

if(xi<=0)
    xi=1;
end
if(yi<=0)
    yi=1;
end

cRecorte = c(yi:yf,xi:xf);

% Determinação dos limites da região de máxima correlação - End

maxAbsCorr(ii,jj) = max(abs(cRecorte(:)));
meanAbsCorr(ii,jj) = mean(abs(cRecorte(:)));

end
figure

imshow(img_teste([xoffSet:xoffSet+size(template,1),yoffSet:yoffSet+size(template,2)]))
end

figure
subplot(1,2,1)
surf(maxAbsCorr);
xlabel('Template ARC');
ylabel('Imagem ARC');
title('Valor máximo de correlação');
axis equal;view(0,90);

subplot(1,2,2)
surf(meanAbsCorr);
xlabel('Template ARC');
ylabel('Imagem ARC');
title('Valor médio de correlação');
axis equal;view(0,90);

colormap('winter');

```

DETECTA_SIMPLES.M

```

close all;
clear all;
clc;

load imagens.mat

% Script 1 - Encontra um único defeito por imagem
% com ARC, CA e ESF.

I = img_test_ARC;
% I = img_test_CA;
% I = img_test_ESF;

% I = template_ARC;
% I = template_CA;
% I = template_ESF;
% I = template_MC;

% Método de busca com base na DERIVADA da média do nível de cinza por
% coluna

for kk=1:30

    Media=mean(double(I{kk}));

    figure(kk)
    subplot(2,1,1)
    imshow(I{kk})

    subplot(2,1,2)
    plot(diff(Media)),ylabel('Derivada da média do nível de cinza por
coluna')

    % Trecho de script simples -> Encontra apenas 1 defeito por
    % imagem

    % Derivada da média
    difmediaCol = diff(Media);
    coord_defeito=0;
    largura=30;

    for jj=1:(size(I{kk},2)-1) %procura coluna por coluna
        if (difmediaCol(jj)) == max(difmediaCol)
            coord_defeito=jj;
            subplot(2,1,1)
            rectangle('Position',[jj-
(largura/2),0,largura,(size(I{kk},1))],'EdgeColor','r','LineStyle',':');
        end
    end

end

```

DETECTA_MULTIPLOS.M

```

close all;
clear all;
clc;

load imagens.mat

% Script principal. Encontra multiplos defeitos em imagens
% com ARC, CA e ESF.

I = img_test_ARC;
% I = img_test_CA;
% I = img_test_ESF;

% I = template_ARC;
% I = template_CA;
% I = template_ESF;

% Método de busca com base na DERIVADA da média do nível de cinza por
% coluna

for kk=1:30

    Media=mean(double(I{kk}));

    figure(kk)
    subplot(2,1,1)
    imshow(I{kk})

    subplot(2,1,2)
    plot(diff(Media)),ylabel('Derivada da média do nível de cinza por
coluna')

    % Trecho de Script para encontrar multiplos defeitos,
    % calculando intervalo com media, desvio padrão e alfa
    difmediaCol = diff(Media);
    %testhip=ttest(difmediaCol) %Teste de normalidade para
    %intervalo alfa=5%
    mu = mean(difmediaCol); % média
    sigma = std(difmediaCol); % desvio padrão
    alfa = 0.01; % percentual, altera sensibilidade. Quanto menor alfa,
    menos sensível!

    % Cálculo do intervalo de confiança para distribuição normal
    p = norminv([alfa/100 (1-alfa/100)],mu,sigma);

    UCL = p(2); % limite superior
    LCL = p(1); % limite inferior

    subplot(2,1,2)
    hold all;
    n = 1:length(difmediaCol)+1;
    plot(n,UCL,'r') % limite superior
    legend('Dados','Limite superior')

```

```

rectDefeito = 0;

X=0;
Xanterior=0;
for jj=1:size(difmediaCol,2) % procura coluna por coluna

    if (((difmediaCol(jj)) > UCL) && (jj-Xanterior > 5))

        X(jj) = 1;
        Xanterior=jj;
        subplot(2,1,1)
        hold all;
        largura=30;
        % marca a região do defeito
        rectangle('Position',[jj-
(largura/2),0,largura,(size(I{kk},1))],'EdgeColor','r','LineStyle',':');

        if(jj-largura/2 > 0 && jj+(largura/2)-1 < size(I{kk},2))
            imD = I{kk}(1:(size(I{kk},1)),jj-largura/2:jj+(largura/2)-
1);
        elseif (jj-largura/2 <= 0)
            imD = I{kk}(1:(size(I{kk},1)),1:largura/2);
        elseif (jj+(largura/2)-1 > size(I{kk},2))
            imD = I{kk}(1:(size(I{kk},1)),jj-largura/2:size(I{kk},2));
        end

        %figure,imshow(imD)
        BorW = class_BorW(imD); %Classifica Claro e Escuro
        num=num+1;                totalDefeitos = totalDefeitos + 1;

        if(BorW == 1)
            disp('ARC');
            text(jj-
largura/2,size(imD,2)/2,'ARC','Color','g','FontSize',7);
        else

            bw= adaptivethreshold(imD,30,0.05,0);
            bw=~bw;
            bw=bwareaopen(bw,100);
            cc = bwconncomp(bw, 4);

            % Implementação da Árvore de Decisão
            % INÍCIO

            if (cc.NumObjects > 0)
                clear d;
                d = regionprops(cc,'all');

                if(size(d,1)~=0) % d = regionprops do defeito
                    if(size(d,1)>1)

                        if((abs(d(1).Orientation) <= 35) &&
(d(1).BoundingBox(3)>d(1).BoundingBox(4)))
                            disp('CA');

```

```

        text(jj-largura/2-
largura/2,size(imD,2)/2,'CA','Color','g','FontSize',7);
        elseif(d(1).BoundingBox(3)>d(1).BoundingBox(4))
            disp('CA');
            text(jj-
largura/2,size(imD,2)/2,'CA','Color','g','FontSize',7);
        else
            disp('ESF');
            text(jj-
largura/2,size(imD,2)/2,'ESF','Color','g','FontSize',7);
        end

        else
            if((abs(d.Orientation) <= 35) &&
(d.BoundingBox(3)>d.BoundingBox(4)))
                disp('CA');
                text(jj-
largura/2,size(imD,2)/2,'CA','Color','g','FontSize',7);
            elseif(d.BoundingBox(3)>d.BoundingBox(4))
                disp('CA');
                text(jj-
largura/2,size(imD,2)/2,'CA','Color','g','FontSize',7);
            else
                disp('ESF');
                text(jj-
largura/2,size(imD,2)/2,'ESF','Color','g','FontSize',7);
            end
        end
    end
end
% Implementação da Árvore de Decisão
% FIM

        end
    end
end

subplot(2,1,1)
if(sum(X)==0)
    title(sprintf('alfa: %.3f%%      Nº de defeitos encontrados:
%d',alfa,sum(X)));
else
    title(sprintf('alfa: %.3f%%      Nº de defeitos encontrados:
%d',alfa,sum(X)));
end

end
end

```

CLASS_BORW.M

```
function [ BorW ] = class_BorW( I )
%class_BorW Classifica um defeito, se é branco ou escuro
% Baseado na derivada da média do nível de cinza por colunas
% Entrada: I - imagem para classificação
% Saída: BorW (1 = defeito claro, 2 = defeito escuro)
% figure
media = mean(I);
derivada = diff(media);

% plot(derivada)

for (count = 1:size(derivada,2))
    if (derivada(count) == max(derivada) )
        coordMax = count;
    end
    if (derivada(count) == min(derivada) )
        coordMin = count;
    end
end

if (coordMax < coordMin)
%     disp('claro');
    BorW = 1;
elseif (coordMax > coordMin)
%     disp('escuro');
    BorW = 0;
end

end
```