



INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Detecção de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (*Fuzzy Number Edge Detector*)

ANDRÉ LUIZ DE MELLO TEIXEIRA

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Ciências na Área de Tecnologia Nuclear - Reatores

Orientador:
Prof. Dr. **Roberto Navarro de Mesquita**

São Paulo
2023

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Detecção de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (*Fuzzy Number Edge Detector*)

Versão Corrigida

Versão Original disponível no IPEN

ANDRÉ LUIZ DE MELLO TEIXEIRA

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Ciências na Área de Tecnologia Nuclear - Reatores

Orientador:

Prof. Dr. Roberto Navarro de Mesquita

São Paulo

2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, para fins de estudo e pesquisa, desde que citada a fonte.

Como citar:

TEIXEIRA, A. L. d. M. ***Detecção de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (Fuzzy Number Edge Detector)***. 2023. 83 f. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN, São Paulo. Disponível em: <<http://repositorio.ipen.br/>> (data de consulta no formato: dd/mm/aaaa)

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN, com os dados fornecidos pelo(a) autor(a).

Teixeira, André Luiz de Mello

Detecção de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (Fuzzy Number Edge Detector) / André Luiz de Mello Teixeira; orientador Roberto Navarro de Mesquita. -- São Paulo, 2023. 83 f.

Dissertação (Mestrado) - Programa de Pós-Graduação em Tecnologia Nuclear (Reatores) -- Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2023.

1. Detecção de bordas. 2. Imagens digitais. 3. Número fuzzy. 4. Pertinência. I. Mesquita, Roberto Navarro de, orient. II. Título.



Universidade de São Paulo



ATA DE DEFESA

Aluno: 85133 - 10073630 - 2 / Página 1 de 1

Ata de defesa de Dissertação do(a) Senhor(a) André Luiz de Mello Teixeira no Programa: Tecnologia Nuclear, do(a) Instituto de Pesquisas Energéticas e Nucleares da Universidade de São Paulo.

Aos 22 dias do mês de março de 2023, no(a) Auditório Rui Ribeiro Franco realizou-se a Defesa da Dissertação do(a) Senhor(a) André Luiz de Mello Teixeira, apresentada para a obtenção do título de Mestre intitulada:

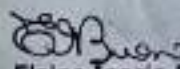
"Determinação de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (*Fuzzy Number Edge Detector*)"

Após declarada aberta a sessão, o(a) Sr(a) Presidente passa a palavra ao candidato para exposição e a seguir aos examinadores para as devidas arguições que se desenvolvem nos termos regimentais. Em seguida, a Comissão Julgadora proclama o resultado:

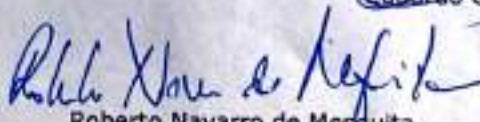
Nome dos Participantes da Banca	Função	Sigla da CPG	Resultado
Roberto Navarro de Mesquita	Presidente	IPEN(IPEN)	Aprovado
Elaine Inacio Bueno	Titular	IFSP - Externo	Aprovado
Eduardo Guy Perpétuo Bock	Titular	IFSP - Externo	Aprovado

Resultado Final: Aprovado

Eu, _____, lavrei a presente ata, que assino juntamente com os(as) Senhores(as) examinadores. São Paulo, aos 22 dias do mês de março de 2023.


Elaine Inacio Bueno


Eduardo Guy Perpétuo Bock


Roberto Navarro de Mesquita
Presidente da Comissão Julgadora

A defesa foi homologada pela Comissão de Pós-Graduação em _____ e, portanto, o(a) aluno(a) faz jus ao título de Mestre em Ciências obtido no Programa Tecnologia Nuclear - Área de concentração: Tecnologia Nuclear - Reatores.

Presidente da Comissão de Pós-Graduação

DEDICATÓRIA

Dedico este trabalho a Deus, aos meus pais Luiz Antônio Bartole Teixeira e Tânia Maria de Mello Teixeira por todos os ensinamentos, à minha amada esposa Natallie Zilio de Souza Teixeira por todo suporte e companheirismo e à minha querida filha Luiza Zilio Teixeira.

AGRADECIMENTOS

Ao meu orientador Dr. Roberto Navarro de Mesquita, por sua paciência, auxílio e aprendizado no desenvolvimento deste projeto e a todos os servidores e colaboradores do IPEN que muito prestativamente me auxiliaram todas as vezes que precisei neste percurso.

RESUMO

TEIXEIRA, A. L. d. M. ***Detecção de borda em imagens de escoamento bifásico de um circuito experimental de circulação natural pelo método FUNED (Fuzzy Number Edge Detector)***. 2023. 83 f. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN, São Paulo.

Este trabalho aplicou o método FUNED (do inglês “Fuzzy Number Edge Detector”) que é baseado em números *Fuzzy* para detecção de contornos em um conjunto de imagens digitais de escoamento bifásico obtidas em experimentos no Circuito de Circulação Natural (CCN) do Instituto de Pesquisas Energéticas e Nucleares (IPEN). O algoritmo se baseia na utilização de cada pixel da imagem adquirida como um número *Fuzzy* para gerar uma imagem que permita mostrar a pertinência do pixel em relação às regiões de borda e fundo. A imagem-pertinência gerada é utilizada como base para a identificação das bordas, onde um pixel branco significa valores acima do limiar e um pixel preto indica valores abaixo desse valor, processo conhecido como binarização da imagem. Após encontrar-se o limiar ótimo, é aplicada a técnica de supressão de não-máximos para criar uma borda mais fina. As imagens finais do método são comparadas com imagens idealmente classificadas (construídas pixel a pixel por um especialista). A mesma comparação foi realizada utilizando-se o clássico detector de bordas de Canny, permitindo-se mostrar a eficiência do detector de bordas FUNED.

Os resultados mostraram que o método FUNED detectou corretamente entre 19% e 25% dos pixels que foram idealmente classificados como borda pelo especialista e, de forma complementar detectou corretamente entre 96% e 98% dos pixels classificados como fundo. O detector de Canny foi aplicado nas mesmas imagens e sob a mesma análise, apresentou um percentual de acertos entre 25% e 39% na predição do que era borda e complementarmente, entre 98% e 99% do que era fundo.

Palavras-chave: Detecção de bordas, Imagens Digitais, Pertinência, Número fuzzy.

ABSTRACT

TEIXEIRA, A. L. d. M. *Edge detection in biphasic flow images of an experimental natural circulation circuit by the FUNED (Fuzzy Number Edge Detector) method.* 2023. 83 f. Dissertation (Master in Nuclear Technology), Institute of Energy and Nuclear Research, IPEN-CNEN, São Paulo.

This work applied the FUNED method (Fuzzy Number Edge Detector), which is based on Fuzzy numbers for contour detection in a set of digital images of two-phase flow obtained in experiments in the Natural Circulation Circuit (CCN) of the Instituto de Pesquisas Energéticas e Nucleares (IPEN). The algorithm uses each pixel of acquired image as a fuzzy number to generate an image that shows the pixel pertinence in relation to the border and background regions. The generated membership-image is used as a basis for the identification of edges, where a white pixel denotes values above the threshold and black pixel denotes values below this value, a process known as image binarization. After finding the optimal threshold, the non-maximum suppression technique is applied, allowing creating a thinner edge detainment. The final image of the method is compared with an ideally classified image, constructed pixel by pixel by an expert. The same comparison was performed with the classic Canny edge detector, showing the FUNED edge detector efficiency.

The results showed that the FUNED method correctly detected between 19% and 25% of the pixels that were classified as edge by the specialist and, in a complementary way, correctly detected between 96% and 98% of pixels that were classified as background. The Canny detector was Applied to the same images and, under the same analysis, presented correctly detected between 25% and 39% of what was edge and, additionally, between 98% and 99% of what was background.

Keywords: Edge detection, Digital Images, Relevance, Fuzzy number.

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVO.....	13
2.1	Justificativa	13
3	REVISÃO DA LITERATURA.....	15
4	FUNDAMENTAÇÃO TEÓRICA	24
4.1	Circuito Experimental de Circulação Natural (CCN).....	24
4.2	Teoria de conjuntos <i>fuzzy</i>	26
4.2	Número fuzzy triangular simétrico.....	26
4.2.1	Função pertinência.....	28
4.2.2	Números fuzzy.....	41
4.2.2.1	Número fuzzy triangular simétrico.....	41
4.3	Interpretação fuzzy de imagem digital.....	30
4.4	Orientação das bordas.....	33
4.5	Supressão de não máximos	37
4.6	Análise ROC para detectores de borda em imagens digitais	37
4.6.1	Gráficos e pontos no espaço ROC.	41
4.6.2	Curvas no espaço ROC.....	43
4.6.3	Área abaixo da curva no espaço ROC.....	46
5	METODOLOGIA.....	48
5.1	Desenvolvimento da Imagem Ground Truth.	48
5.2	Implementação do detector de bordas.....	50
6	RESULTADOS	54
7	CONCLUSÕES	64
8	REFERÊNCIAS	66
	APÊNDICE A – Algoritmo 1: Imagem pertinência	72
	APÊNDICE B – Algoritmo 2: Imagem sem supressão de não máximos	75
	APÊNDICE C – Algoritmo 3: Imagem final (FUNED)	79
	APÊNDICE D – Algoritmo 4: Imagem em <i>Dataset</i>.	82

1 INTRODUÇÃO

Desde o advento do processamento digital de imagens em 1964, com a utilização de técnicas computacionais para melhoramento de imagens produzidas por uma sonda espacial na *Jet Propulsion Laboratory* (Pasadena, Califórnia), e quando as imagens da Lua, transmitidas pelo *Ranger 7*, foram processadas por um computador para corrigir vários tipos de distorções de imagem inerentes à câmera de televisão utilizada, até os dias de hoje, a área de processamento de imagens cresceu rapidamente (GONZALES, 2010).

A área da visão computacional em geral e o processamento de imagens em particular, tem um papel importante na vida humana e nos processos que a cerca. Atualmente, o campo de processamento de imagens tem inúmeras aplicações comerciais, científicas, industriais, médicas, militares, de entretenimento e outras. A contínua interação de pesquisas fundamentais e o desenvolvimento de novas tecnologias tornou essa área de pesquisa bastante ampla e ativa (BOAVENTURA, 2010).

A detecção e análise de borda de uma imagem desempenha um papel importante no campo do processamento de imagens. Várias técnicas foram desenvolvidas para detecção de borda eficiente, como a de KIRSCH (1971) que descreve uma classe de algoritmos para que imagens quantizadas sejam decompostas, ou CANNY (1986) que desenvolveu uma abordagem computacional por meio de análise de padrões e inteligência de máquina associada a definição de valores que devem ser precisos o suficiente para delimitar o comportamento desejado do detector. Além desses, temos os trabalhos de ULUPINAR e MEDIONI (1990) que refinaram as arestas das bordas detectadas por um operador de filtro Laplaciano da Gaussiana, SHEN e CASTAN (1992) que desenvolveram um operador linear, ZHANG et al. (2009) que aplicaram seu método com base no melhor operador de Sobel e algoritmos genéticos e LEI et al. (2011) que aprimoraram o filtro de Prewitt para detecção de bordas. Esses métodos possuem em sua estrutura um grande volume de cálculo, sensibilidade a ruídos e dificuldade

na extração de bordas em imagens com iluminação não uniforme, presentes na maioria das imagens naturais (BOAVENTURA, 2010).

A necessidade em detectar bordas é um fundamento muito importante em diversas aplicações na busca de soluções, como exemplos no campo da robótica no trabalho de BRUNO et al. (2008), na análise de imagens médicas demonstrada por GOSHTASBY (2005), na ciência geográfica com LIN et al. (2004), no reconhecimento de padrões com RULANINGTYAS e AIN (2009), na inspeção e análise automática de objetos de XU e YU (2013), no diagnóstico de patologias de VEIGA et al. (2014), etc.

As bordas de uma imagem são contornos gerados como resultado de uma mudança repentina ou abrupta das características entre pixels vizinhos. Essas mudanças podem ser observadas devido à alteração da cor, textura, sombra ou absorção de luz. Essas características permitem estimar as informações de orientação, tamanho, profundidade e superfície de uma imagem (IZHAR et al., 2015).

Uma imagem digital possui diversas características que, ao serem verificadas e analisadas, podem produzir diversas informações úteis para processos, aplicações e pesquisas. As bordas em uma imagem são uma das mais importantes propriedades que permitem a interpretação visual dela e o consequente reconhecimento das características de uma cena (BOAVENTURA, 2010).

De acordo com DA SILVA (2020), uma imagem científica pode ser compreendida como uma representação completa ou parcial de uma “cena” em que se quer obter, quantitativa e qualitativamente, informações específicas que estejam relacionadas a fenômenos a serem avaliados, que inclusive, podem extrapolar as percepções humanas. O processamento de imagens possibilitou o surgimento de diversos recursos para a detecção de características importantes de um determinado evento.

O Centro de Engenharia Nuclear do IPEN possui um circuito experimental de circulação natural com tubos de vidro transparente que possibilita, por meio da observação e do imageamento digital, além de outros, o estudo das características de circulação bifásica do fluido. Muitas publicações têm relacionado essas duas áreas de estudo, como por exemplo, o setor de extração de petróleo

que utiliza tubos em modelos reduzidos para a simulação de fenômenos de escoamento multifásico (TAYLOR et al., 2010).

Sistemas de circulação natural são incluídos em projetos de usinas nucleares, como mecanismos de segurança e controle para a remoção de calor em caso de acidentes. Muitos projetos de reatores refrigerados à água levam em consideração o estudo do escoamento bifásico, composto por fases de vapor e líquido, nas condições limítrofes de transferência de calor (DA SILVA, 2020).

O estudo do fluxo bifásico foi relacionado por FU e LIU (2016) para prever o valor médio e a densidade do número de bolhas com o objetivo de mensurar a fração de vazio resultante do processo. O escoamento bifásico tem sido objeto de estudo em diversos trabalhos com base na aquisição e análise de imagens digitais (BUENO et al., 2018).

Com imagens obtidas em diferentes fases de instabilidades cíclicas do escoamento bifásico em circuitos de circulação natural, BUENO (2016) utilizou regras *fuzzy* associadas a um método probabilístico envolvendo técnicas resultantes do processamento de imagem, e isso possibilitou obter diversos parâmetros característicos do escoamento bifásico.

Dentro desse contexto, desenvolveu-se nesse trabalho uma metodologia para detecção de bordas em imagens digitais de escoamento bifásico por meio da utilização de técnicas *fuzzy* na operação direta com números *fuzzy*.

2 OBJETIVO

O objetivo deste trabalho consiste em aplicar uma metodologia utilizando o método baseado em números *Fuzzy*, conhecido como FUNED (do inglês, “*Fuzzy Number Edge Detector*”) para a detecção de bordas em imagens digitais de escoamento bifásico em imagens digitais do Circuito de Circulação Natural (CCN) do Centro de Engenharia Nuclear (CEN) do Instituto de Pesquisas Energéticas e Nucleares (IPEN-CNEN/SP). A implementação do algoritmo será desenvolvida em linguagem de programação *Python* em ambiente de rede aberta (do inglês “open-source”) na interface gráfica de edição web “Jupyter notebook”.

O algoritmo deverá fornecer como resultado uma imagem binária segmentada, de mesma dimensão da imagem analisada, com as bordas representadas em pixels pretos e o fundo representado com região homogênea de pixels brancos.

O método FUNED não se fundamenta em sistemas *fuzzy* baseados em regras, mas na operação direta no domínio espacial da imagem, ou seja, atua diretamente em cada pixel na janela de vizinhança como um número *fuzzy*. Por essas características do método, essa metodologia deverá apresentar uma solução compacta e intuitiva para detecção de bordas em imagens de escoamento bifásico.

2.1 Justificativa

Um dos grandes motivos que justifica a utilização de técnicas *fuzzy* em processamento de imagens está na característica natural em que se tem para gerenciar dados imprecisos. Em segmentação de imagem, por exemplo, isso ocorre quanto ao limiar que separa uma região de outra e que pode ser de difícil identificação, ou então, na definição sobre o que é fundo ou borda de um determinado objeto, ou ainda, sobre a necessidade da aplicação de técnicas que melhorem as imagens através de filtros, realces em áreas de interesse ou outras técnicas que permitem melhorar a imagem a fim de obter melhores resultados (BOAVENTURA, 2010).

Uma das grandes dificuldades experimentais para aquisição de imagens em escoamento bifásico é a avaliação da nitidez da imagem obtida que é relacionada diretamente ao controle do foco. O desenvolvimento de metodologias de software que possibilitem um maior controle e aferição das bordas em imagens experimentais é muito importante. Imagens de experimentos com sistemas óticos complexos costumam apresentar limiares e situações complexas que exigem um tratamento refinado e adaptado ao qual as técnicas que utilizam fuzzy tratam muito bem.

Outra razão para a abordagem é o fato da lógica *fuzzy* aproximar o processamento computacional do raciocínio humano. Isso faz com que as máquinas de inferência *fuzzy* possam ser desenvolvidas usando o conhecimento especialista. As técnicas tradicionais baseadas em regras, por exemplo, são expressas em termos de condições, em aplicações reais. Contudo, as condições são geralmente satisfeitas de maneira parcial. Por exemplo, a questão da homogeneidade em uma vizinhança nem sempre pode ser respondida utilizando a teoria da lógica tradicional. Para esses casos, as regras *fuzzy* permitem que ações sejam parcialmente tomadas, e isso implica numa boa alternativa para a solução de problemas como esse (BOAVENTURA, 2010).

Embora existam diversas técnicas para a extração de bordas, em geral, essas técnicas são complexas, pouco intuitivas, requerem grande esforço de implementação e são computacionalmente custosas. A interpretação direta da imagem por números *fuzzy* reflete bem as incertezas dos níveis de cinza em uma imagem. (BOAVENTURA, 2010).

O detector de bordas FUNED permite ao usuário os ajustes de três parâmetros para otimização do método em busca dos melhores resultados para cada imagem. Os ajustes dos parâmetros podem ser alterados conforme a necessidade do experimento.

3 REVISÃO DA LITERATURA

De acordo com MESQUITA (2009), a circulação natural é decorrente da ocorrência de forças direcionais (denominadas "*driving forces*") que surgem da variação da densidade do fluido como resultado do gradiente de temperatura em diferentes pontos de um circuito. Sob condições limítrofes pode ocorrer o escoamento bifásico com a presença de líquido e gás e esse regime modifica a eficiência do circuito de transferência de calor por circulação natural.

As pesquisas realizadas em sistemas de escoamento bifásico têm ocorrido com bastante frequência e o desenvolvimento de um modelo que consiga analisar o fenômeno e suas características têm sido uma busca por meio de variados métodos e abordagens.

Um circuito de circulação forçado empregando bombas centrífugas nas instalações experimentais do Laboratório de Engenharia Térmica e de Fluidos da Universidade de São Paulo em São Carlos foi utilizado por BARBOSA et al. (2010). O trabalho desenvolvido avaliou o uso de um mapa auto-organizável (rede neural) adaptado ao problema de identificação de regime em fluxos bifásicos horizontais. Para verificação dos resultados obtidos do modelo da rede neural, os testes consideraram dois tipos diferentes de escoamentos bifásicos, gás-sólido e gás-líquido. Os resultados mostraram que a rede neural é capaz de identificar corretamente os regimes de escoamento.

No circuito de circulação natural do Reator de Pesquisa da Universidade de Kyoto, ZHANG et al. (2015) investigaram experimentalmente a capacidade de resfriamento da circulação natural através das medições das temperaturas de entrada e saída do núcleo, sob operação em variadas potências até o estado de desligamento. A capacidade do resfriamento da circulação natural é representada pela velocidade média do fluxo de circulação natural no núcleo e foi calculada a partir da diferença de temperatura de entrada do núcleo.

Os efeitos dos parâmetros no comportamento da estabilidade e geração de entropia de um circuito de circulação natural bifásico foram investigados por GOUDARZI e TALEBI (2015). Para obter uma maior eficiência termodinâmica,

além do controle da estabilidade, a minimização da geração de entropia foi buscada e se tornou uma importante variável nesse tipo de sistema. Neste estudo, a investigação do comportamento da estabilidade ocorreu pela aplicação de um método não linear, conhecido como método da solução direta ou método no domínio do tempo.

Estudos foram realizados por WANG e DONG (2009) a partir de imagens digitais capturadas em um circuito experimental. As análises são baseadas na construção de dois modelos matriciais (matriz de co-ocorrência de nível de cinza - GLCM e gradiente da matriz de co-ocorrência de nível de cinza - GLGCM) com base nas características de texturas das imagens. Foi constatada a existência de uma relação com a densidade gás-líquido, mas nenhuma relação com a velocidade superficial do gás ou do líquido.

O algoritmo proposto por GUO et al. (2010) analisou as imagens por *Wavelet* em imagens de escalas de cinza com filtro gaussiano para a eliminação de ruídos. O algoritmo determina se o pixel é borda ou não por meio de um limiar e esse limiar específico é definido pelo cálculo do coeficiente e do ângulo de fase para encontrar o máximo local normalizado do gradiente de cinzas.

TORISAKI e SHUICHIRO (2020) reforçam a necessidade fundamental de identificar com detalhes o regime de vazão de um fluxo bifásico gás-líquido a fim de melhorar as análises termo-hidráulicas, bem como garantir a integridade dos sistemas de tubulação. Além disso, a transição do regime de escoamento bifásico ocorre gradualmente e é um fenômeno altamente instável, a identificação objetiva do regime de escoamento próximo à região de transição ainda permanece um desafio até hoje.

Em sua pesquisa, TORISAKI e SHUICHIRO (2020) desenvolveram um método de identificação de regime de fluxo baseado em inteligência artificial com técnica de aprendizado profundo (do inglês, "*Deep learning*"). A ferramenta foi desenvolvida com base em um algoritmo de aprendizado de máquina, que pode detectar características de fluxo e capturar posições de bolhas individuais de imagens fornecidas. Além de calcular os principais parâmetros de fluxo bifásico, como fração de vazios e velocidade de ascensão da bolha utilizando os resultados da detecção.

Dentre as diferentes técnicas que estão atualmente em uso para esse tipo de solução de problema, encontram-se as técnicas *fuzzy*. Muitas técnicas abordam a detecção de bordas como um problema de raciocínio *fuzzy*. Elas oferecem uma estrutura não-linear e baseada no conhecimento para lidar com a ambiguidade e a imprecisão frequentemente presentes em imagens digitais, são aplicadas principalmente na definição de vários elementos como bordas e regiões, ou às noções de brilho e contraste (FATEMEH et al., 2014).

Em geral, as técnicas *fuzzy* podem ser descritas como técnicas ou métodos que encontram sua origem na teoria de conjuntos *fuzzy* e têm sido usadas e/ou adaptadas para serem aplicadas no campo de processamento de imagem. O processamento de imagens envolvendo reconhecimento de bordas possui incertezas e imprecisões intrínsecas que podem ser tratadas por inferência *fuzzy* (BOAVENTURA, 2010).

Com base no algoritmo *fuzzy c-shells*, DAVE e BHASWAN (1992) apresentaram várias generalizações do algoritmo para caracterizar e detectar *clusters*. O algoritmo resultante tem um bom desempenho para formas parciais, no entanto, outra formulação baseada na equação quadrática de segunda ordem pode detectar melhor as elipses e círculos em dados 2D. A eficiência dos algoritmos de lógica *fuzzy* com um *clustering* na detecção de círculos e elipses é parecida quando comparada à transformada de *Hough*.

Utilizando-se da lógica *fuzzy*, HAN et al. (1994) detectaram formas em imagens ruidosas integradas à transformada de *Hough* e com isso obtiveram um modelo geométrico da transformada de *Hough fuzzy*. Esta técnica encontra formas ajustando os pontos da imagem e isso evita as formas espúrias detectadas ao usar a transformação de *Hough* convencional. Tem aplicação eficiente na detecção de linhas e círculos.

O modelo denominado *neuro-fuzzy*, de SURAL e DAS (1999) permitiu o reconhecimento de caracteres em documentos digitalizados com a técnica de transformação de *Hough* de lógica *fuzzy*. As duas abordagens fundamentais para este problema são o reconhecimento de caracteres e o "*template matching*" (comparação de modelos).

A transformada de *Hough* associada à lógica *fuzzy*, de BASAK e PAL (2005), consegue lidar com a imprecisão e mal definição das formas em uma

imagem. Além de identificar as formas, a metodologia pode identificar a quantidade de distorção em cada forma, caracterizando adequadamente o espaço paramétrico. O método é aplicado em imagens com níveis de cinza utilizando a variação desses níveis de cinza e a distorção das formas. O resultado é uma *Fuzzy Hough*, ou "FHT - *Fuzzy Hough Transform*", que dá origem a um esquema de segmentação de imagens baseado no reconhecimento de padrões.

BLOCH et al. (1996) fizeram reconstrução 3D de vasos sanguíneos sem qualquer modelo geométrico prévio, com a utilização de conjuntos *fuzzy* como base de um método de extração de bordas morfológicas. RUSSO (1998) afirma que a técnica *fuzzy* tem melhor desempenho do que outros métodos na literatura do ponto de vista da sensibilidade ao ruído e da detecção de detalhes da imagem. Em seu trabalho, RUSSO (1998) apresentou uma nova abordagem para a detecção de bordas em imagens corrompidas por ruído de impulso. O método proposto adotou o raciocínio *fuzzy* para extrair bordas sem ser enganado pelo ruído presente na imagem.

Os filtros lineares de primeira ordem são amplamente aplicados na detecção de bordas digitais, mas de acordo com MIOSSO e BAUCHPIESS (2001), eles não permitem que bons resultados sejam obtidos em imagens com alta variação do contraste, devido à iluminação não uniforme, como ocorre em grande parte das imagens naturais. Nesse sentido, desenvolveram um sistema de inferência *fuzzy* de detecção de bordas e aplicaram em imagens com alta variação de contraste. Os resultados obtidos foram comparados com o operador linear de Sobel.

O classificador *fuzzy* proposto por LIANG e LOONEY (2003) detectou uma classe de pixels em uma imagem em função da variação dos níveis de cinza em várias direções. O desempenho do algoritmo é semelhante ao de Canny e computacionalmente mais rápido. Em ambos, o nível adequado de sensibilidade deve ser escolhido pelo usuário para obter os melhores resultados. O ajuste da sensibilidade permite obter mais bordas e mais ruídos ou menos bordas e menos ruídos.

O conjunto *fuzzy* intuicionista de Atanassov foi utilizado por CHAIRA e RAY (2008) para o desenvolvimento de um detector de bordas com imagens digitais. O conjunto *fuzzy* intuicionista considera a incerteza na atribuição do grau

de pertinência conhecido como grau de hesitação e uma medida de distância, chamada divergência difusa intuicionista. Com esta medida de distância proposta, a detecção de bordas é realizada e os resultados são definidos como bons, quando comparado a outros métodos.

O filtro de Prewitt foi adaptado por JACQUEY et al. (2008) utilizando o conceito de lógica *fuzzy* para aplicação em imagens omnidirecionais. O processamento de imagem convencional não está adaptado a esses tipos de imagens específicas. Nesse sentido, desenvolveu-se um detector de bordas adaptado à geometria da imagem associado aos conjuntos *fuzzy*, a fim de serem consideradas todas as imprecisões introduzidas pelo processo de amostragem.

Em seu trabalho, BUSTINCE et al. (2009) propuseram um método para construção de conjuntos *fuzzy* com intervalos valorados (IVFS) a partir de uma matriz ou imagem. O método obtém o comprimento do intervalo que representa a pertinência de qualquer elemento ao novo conjunto a partir das diferenças entre os valores atribuídos a esse elemento e seus vizinhos na matriz inicial.

Utilizando-se da visualização e captura de imagens em um circuito de circulação natural, MESQUITA et al. (2012) estudaram padrões de escoamento bifásico associados aos transientes de fase e de instabilidades de escoamento em circulação natural. Um sistema *fuzzy* de classificação de fluxo (*Fuzzy Flow-type Classification System – FFCS*) foi desenvolvido para classificar os padrões com base apenas nas características extraídas da imagem. Os experimentos foram realizados nas instalações experimentais para gerar uma série de ciclos periódicos de instabilidades de fluxos bifásicos característicos. A aquisição das imagens e medições de temperatura foram feitas simultaneamente e o FFCS demonstrou bons resultados com adequados ajustes nos parâmetros de aquisição e no pré-processamento da imagem.

A fim de identificar o padrão de escoamento bifásico de forma automática e precisa, SHI (2007) desenvolveu um método com rede neural e lógica *fuzzy*. As imagens de escoamento são capturadas em um tubo horizontal e as características das bolhas como área, largura e altura são obtidas por meio de técnicas de processamento de imagens. O método consiste na utilização de rede neural com lógica *fuzzy* para identificar o tipo e o padrão de escoamento livre, caótico e pistonado. O algoritmo otimizado de Levenberg-Marquart foi utilizado para

aprendizado e os resultados experimentais mostraram que o método *fuzzy* pode identificar com precisão os padrões de escoamento em um tubo horizontal.

BARKHODA e SHAHRYARI (2009) apresentaram um algoritmo de detecção de bordas baseado na análise *fuzzy* de duas fontes diferentes de informação para decidir se cada pixel é borda ou não. Os valores de gradiente e desvio padrão são calculados e formando dois conjuntos de bordas, utilizados como entradas para o sistema *fuzzy*. Em seguida, o sistema decide sobre cada pixel de acordo com as regras *fuzzy*. Os resultados experimentais mostram a habilidade e o alto desempenho do algoritmo proposto.

PATEL et al. (2011) apresentaram um algoritmo de detecção de bordas baseado em sistema e regras *fuzzy*. A saída do sistema decide se aquele pixel específico é uma parte da borda ou não. Os métodos utilizados foram baseados em gradiente da derivada de primeira ordem e detecção de cruzamentos no zero, usando o operador laplaciano aplicado à imagem suavizada gaussiana, que é o método derivado de segunda ordem. Usando essas duas abordagens, os primeiros valores são calculados e aplicados ao sistema *fuzzy* que decide para cada pixel, usando diferentes conjuntos de regras, se é borda ou não.

BUENO (2016) propôs a detecção de contornos em imagens em tons de cinza. A etapa de contraste enfatiza as propriedades estatísticas da característica de segmentação do objeto de interesse. O método *FUZCON* foi baseado na *Fuzzy logic* para adaptar os valores de contraste às classes de faixa dinâmica que melhoram o processamento do computador. O desenvolvimento desse sistema de detecção de bordas seguiu investigações anteriores que exploraram as vantagens da *Fuzzy logic*.

O sistema *Fuzzy* desenvolvido por BUENO (2016) utiliza uma abordagem estatística do Desvio Padrão do Contraste (DSC), denominado *Dynamic Fuzzification Base* (DFB), baseado no método descrito por BARKHODA e SHAHRYARI (2009). Essa técnica mostrou a eficácia do uso do desvio padrão para encontrar o gradiente de uma imagem. A metodologia *FUZCON* proposta por BUENO (2016) faz um ajuste automático das funções de pertinência na identificação de bordas baseado em propriedades estatísticas da imagem.

SERRA (2017) desenvolveu uma metodologia para estimar a fração de vazio a partir de imagens digitais capturadas no Circuito de Circulação Natural

(CCN) do Centro de Engenharia Nuclear (CEN) do Instituto de Pesquisas Energéticas e Nucleares (IPEN-CNEN / SP) por meio da inferência dos padrões geométricos para cada bolha detectada.

MESQUITA (2018) em comparação a um estudo de inferência fuzzy, com o objetivo de obter melhorias no poder de classificação, utilizou características da imagem no domínio da frequência para caracterizar três estágios de instabilidade de fluxo bifásico de circulação, a fim de servir como entrada em um algoritmo de agrupamento de mapas auto-organizáveis SOM (do inglês "*Self-Organizing Maps*"). O banco de dados de entrada para treinamento são os coeficientes da Transformada de Cosseno Discreta de Quadro Completo (FFDCT) que foram obtidos para 32 amostras de imagem para cada estágio de instabilidade. A metodologia de treinamento/teste foi utilizada para verificar o método de classificação e o banco de imagens foi obtido a partir de dois experimentos de escoamento bifásico realizados no circuito de Circulação Natural (CCN) do Instituto de Pesquisas Energéticas e Nucleares (IPEN/CNEN), Brasil.

MESQUITA (2018) conclui que a inferência *fuzzy* mostrou-se mais flexível e capaz de se adaptar a características estatísticas mais simples a partir de apenas um perfil de imagem. Os recursos FFDCT resultaram em resultados mais precisos quando aplicados a uma rede neural SOM, embora tenham que ser aplicados à matriz de escala de cinza original completa para que todas as imagens de escoamento sejam classificadas.

A seguir, ainda considerando as metodologias que podem identificar bolhas em imagens com alta fração de vazios, tem-se o trabalho de WU et al. (2006) baseado na aplicação da Tomografia de Resistência Elétrica (ERT) em escoamento bifásico gás / líquido de tubulação vertical. A transformada *Wavelet* é utilizada para analisar os dados medidos do sistema ERT, e de acordo com a análise de multi-resolução (MRA), o vetor de características de energia *Wavelet* de múltiplas escalas pode expressar as informações essenciais do escoamento bifásico gás / líquido. Por fim, o método de máquina de vetores de suporte SVM (do inglês "*Support Vector Machine*") em teorias estatísticas é adotado para validar a viabilidade do método de extração de características.

O método proposto por WENYIN et al. (2008) para segmentação de imagem bifásica gás/líquido baseia-se em informações de distribuição dos níveis

de cinza com filtro gaussiano, com a finalidade de manter as bordas e eliminar os ruídos. O método é eficaz para a análise e reconhecimento de escoamento de bolhas.

As medições de vazão bifásica são muito comuns em aplicações industriais, especialmente em áreas de petróleo e gás. Embora alguns trabalhos em segmentação de imagens tenham analisado o fluxo de gás-líquido ao longo de tubos verticais, poucas abordagens focaram em experimentos horizontais. Nesse sentido, AMARAL et al. (2013) descreveram uma técnica que estima automaticamente os parâmetros de bolha como, frequência, dimensão e velocidade por meio de análise de vídeo com câmeras de alta velocidade em tubos horizontais. Os dados experimentais são obtidos a partir de uma seção de teste de fluxo gerados em condições controladas. Por fim, os parâmetros estimados foram comparados com as previsões teóricas, mostrando boa concordância e indicando que a técnica proposta é uma boa ferramenta na investigação de escoamento bifásico.

O projeto hidráulico térmico de reatores nucleares necessita de extensivos estudos sobre o padrão de fluxo bifásico. Nesse sentido, SARKAR (2005) usou métodos de radiografia de nêutrons em tempo real para estudar as instabilidades do padrão de fluxo e desenvolveu um modelo de hidráulica térmica para o reator projetado pela Bhabha Atomic Research Center (BARC) em Bombaim, na Índia. Os experimentos foram conduzidos em seções de teste de subsistemas de diferentes diâmetros com pressões variando de 1 a 40 bar e em vários níveis de potência.

BHARDWAJ e MITTAL (2012) revisaram técnicas de detecção de bordas como Roberts, Sobel, Prewitt, Canny e especialmente *declivity* e apresentaram em seu artigo uma comparação entre vários detectores de borda para identificar qual detector de borda apresenta melhores resultados. O algoritmo desenvolvido demonstrou que o operador de declividade modificado oferece melhores resultados em comparação a outros detectores de borda.

O operador de declividade, em particular, tem sido utilizado em muitas investigações. MICHÉ e DEBRIE (1995), apresentaram um operador de segmentação de imagem auto-adaptativa para aplicações de visão em tempo real. Este operador não linear detecta limites de regiões de baixo contraste sem pré-

processamento. A declividade proposta envolve a avaliação das magnitudes de intensidade de um conjunto de pixels contíguos entre dois extremos locais em uma linha de imagem. Os resultados experimentais são produzidos com diferentes tipos de imagem e comparados com o operador básico de declividade e com o operador de Deriche.

CABANI et al. (2006) apresentaram um sistema de visão estéreo auto-adaptativo para detecção de obstáculos em estradas. Os pontos de borda verticais são extraídos usando um operador de declividade e em seguida, esses pontos são associados a uma análise dinâmica. As correspondências erradas são detectadas e corrigidas usando a restrição de continuidade de disparidade aplicada em formas 3D construídas, e por fim, as arestas 3D do obstáculo são extraídas.

Em visão computacional, várias técnicas, métodos e teorias são usadas para resolver problemas relacionados ao processamento digital de imagens. Algumas dessas aplicações trabalham com técnicas de geometria digital, morfologia matemática, abordagens estatísticas, teoria da probabilidade etc. Pela busca de abordagens que ofereçam formulações mais simples e intuitivas, muitos trabalhos surgiram de forma alternativa através de pesquisas em processamento *fuzzy* de imagens (BUENO, 2016).

BOAVENTURA (2010) mostrou que a lógica *fuzzy* permite que dados imperfeitos sejam adequadamente manipulados e quantificados. Além disso, permite também que dados sejam combinados para se tomar uma decisão final, a partir do conhecimento de regras heurísticas sem nenhuma relação analítica entre elas. A lógica *fuzzy* não é simplesmente uma solução pontual para uma tarefa especial, ela escreve uma nova classe de técnicas de processamento de imagens.

O reconhecimento de padrões em imagens utiliza diversas áreas do conhecimento humano para tornar possível a resolução de problemas mais complexos que envolvem a detecção de contornos e identificação de objetos de interesse. Com o avanço do desempenho dos computadores, diversas abordagens foram desenvolvidas agregando o acúmulo de técnicas, experiência e resultados. O desenvolvimento de *hardware* cada vez mais rápido permite o uso de técnicas inteligentes mais elaboradas para a detecção de objetos em imagens de difícil reconhecimento.

4 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a teoria utilizada no desenvolvimento desta pesquisa e fundamenta os conceitos aplicados no método proposto. Os assuntos abordados serão tratados com o objetivo de trazer ciência quanto ao conhecimento teórico necessário na construção do trabalho e no sucesso de sua execução. Os assuntos estão dispostos da seguinte forma e sequência: circuito experimental de circulação natural (CCN), teoria de conjuntos fuzzy, interpretação *fuzzy* de imagem digital, orientação das bordas, supressão de não máximos e análise ROC para detectores de borda em imagens digitais

4.1 Circuito Experimental de Circulação Natural (CCN)

Sistemas de resfriamento de circulação natural são atualmente usados em novos reatores nucleares e nas últimas décadas, pesquisas em sistemas como esse têm se concentrado no estudo de parâmetros de escoamento e transferência de calor. A refrigeração é um componente crítico em uma usina nuclear, que requer um processo eficiente de troca de calor. Em um sistema tão crítico, é importante ter conhecimento sobre as condições limitantes dos regimes de escoamento bifásico para gerenciar a transferência de calor do refrigerante do reator (BUENO, 2018).

A Figura 1 mostra o Circuito de Circulação Natural (CCN) instalado no Instituto de Pesquisas Energéticas e Nucleares, IPEN/CNEN. O CCN é um sistema experimental destinado a fornecer dados termo-hidráulicos relativos a escoamentos monofásicos e bifásicos em condições de circulação natural. Esse sistema é composto por um circuito retangular (com 2600 mm de altura e 850 mm de largura) de tubos de vidro borossilicato resistentes à temperatura, com 38,1 mm de diâmetro interno e 4,42 mm de espessura.

O conjunto possui uma seção aquecida (Fig. 1(a)), também feita de tubo de vidro com 76,2 mm de diâmetro interno e 880 mm de comprimento. Esta seção possui um aquecedor elétrico resistivo que pode fornecer até 8000W de potência. A energização do aquecedor elétrico reproduz de forma experimental o

aquecimento gerado no núcleo do reator, e sua potência elétrica é controlada por um variador de tensão de corrente alternada.

A seção de resfriamento (Fig. 1(c)) consiste em um trocador de calor/condensador, também feito de vidro, com duas serpentinas espirais internas por onde flui a água à temperatura ambiente para resfriamento do circuito.

Com o circuito em operação, a densidade do fluido próximo ao aquecedor ligado aumenta e isso gera o efeito de convecção natural, o líquido mais quente sobe e o mais frio desce, o aumento constante dessa temperatura gera instabilidade estática periódica. Os ciclos de instabilidade são geralmente divididos em três fases diferentes chamadas períodos de incubação, expulsão e reenchimento (ANDRADE et al., 2000). O CCN é ajustado para sustentar o comportamento cíclico e periódico dessa instabilidade.

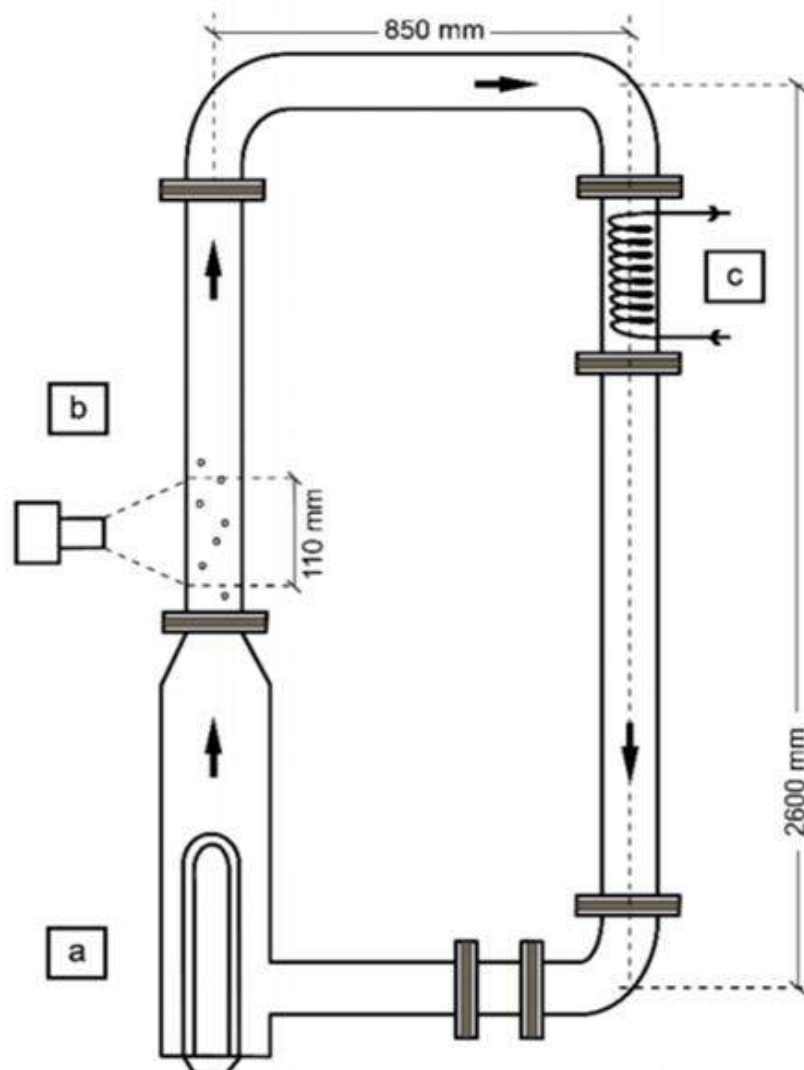
A fase de incubação não tem escoamento líquido quando as bolhas de vapor crescem em número e tamanho, o vapor permanece no tubo horizontal superior devido à sua menor densidade. Em decorrência do aumento da temperatura esse ciclo é substituído pelo *churn flow* (padrão de escoamento bifásico turbulento caracterizado por um fluxo altamente agitado, nesta fase as bolhas de gás são suficientes em número para interagir umas com as outras e, enquanto interagem, podem formar novas bolhas distorcidas e maiores com formas e comportamentos únicos no sistema) e dessa forma surge a fase de expulsão, onde o líquido arrastado pelo vapor é expelido do tubo vertical em que se encontra o resistor de aquecimento.

A fase final (reenchimento) é caracterizada pela inversão da direção da vazão causada pela diferença de carga hidrostática, substituindo a água quente do aquecedor por água fria proveniente do resfriador de serpentina. A produção de vapor no aquecedor diminui e a parte horizontal do tubo vertical em que se encontra o resistor de aquecimento é novamente preenchida com água, iniciando novamente o ciclo global (ANDRADE et al., 2000).

Este comportamento de oscilação periódico do escoamento pode ser observado nesta instalação devido à transparência de seus tubos feitos de vidro. Essa e outras características do CCN tem possibilitado o desenvolvimento de métodos de modelagem e medidas experimentais de parâmetros de escoamento bifásico por meio de imagens capturadas por uma câmera fotográfica digital.

BUENO, (2016, 2018); MASOTTI e MESQUITA, 2015; MESQUITA et al., (2009, 2012, 2018).

Figura 1 – Circuito de Circulação Natural (IPEN).



Fonte: MASOTTI; MESQUITA, 2015.

4.2 Teoria de conjuntos Fuzzy

A teoria *fuzzy* introduzida por Zadeh (1965) surgiu para tratar de forma adequada dados vagos e subjetivos presentes na comunicação humana. As expressões imprecisas definidas como “aproximadamente”, “em torno de”, dentre outros, que não encontravam solução apropriada ao serem tratadas como um conjunto de informações convencionais, podem com esse novo tratamento serem utilizadas para a execução de cálculos, a exemplo do que faz o ser humano.

As expressões linguísticas que expressam de forma inexata as quantidades, definindo-as como “muito”, “pouco”, “no mínimo”, “por volta de”, são

tratadas dentro da teoria *fuzzy* como dados numéricos *fuzzy* que podem ser representados por subconjuntos *fuzzy* reais, conhecido simplesmente como números ou variáveis *fuzzy*.

O objetivo deste capítulo não é exaurir a teoria *fuzzy* que é bastante ampla e abrangente, mas abordar os conceitos essenciais para a compreensão e implementação da técnica *fuzzy* aplicada a imagens, sendo eles:

- Suporte do conjunto *fuzzy*: Para A sendo um subconjunto *fuzzy* do conjunto universo X , o suporte de A é o subconjunto de X que possui elementos cuja pertinência é não-nula em relação a A , ou seja, são elementos que possuem algum grau de pertinência maior do que zero, conforme pode ser observado na equação (1).

$$\text{Suporte}(A) = \{x \in X: \mu_A(x) > 0\} \quad (1)$$

- Corte- α em um conjunto *fuzzy*: O corte- α em um conjunto *fuzzy* A , também conhecido como limiar (do inglês “*threshold*”) é definido como um conjunto que contém todos os elementos de A cujos elementos pertencem ao universo X e possuem grau de pertinência maior ou igual a α , ou seja, elementos cujos valores de pertinência sejam maiores do que o valor definido como limiar, conforme pode ser observado na equação (2).

$$A_\alpha = \{x \in X: \mu_A(x) \geq \alpha\} \quad (2)$$

4.2.1 Função pertinência.

Em conjuntos numéricos convencionais, a função pertinência (função característica) mostra se um dado valor pertence ou não a um determinado conjunto ou subconjunto. Neste caso, diz-se que pertence ou não pertence a um dado conjunto, ou seja, a função característica assume valores 1 (um) para elementos pertencentes ao conjunto e 0 (zero) para elementos não pertencentes.

Na teoria *fuzzy*, o conjunto é associado a uma função pertinência em que os elementos poderão não somente pertencer totalmente ou não a esse

conjunto, mas a pertinência poderá ocorrer de forma parcial, com valores de pertinência variando entre 1 (um) e 0 (zero), conforme mostra a equação (3).

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (3)$$

Sendo X o conjunto universo e A um subconjunto de X , $\mu_A(x)$ é chamada de função pertinência em que atribui para cada elemento X um determinado grau de pertinência.

4.2.2 Números *fuzzy*.

O número *fuzzy* é um subconjunto de A que possui grau de pertinência não nulo definido em μ_A , cortes- α não vazios com intervalos fechados variando entre 0 (zero) e 1 (um) e que tenha suporte do conjunto *fuzzy*, ou seja, possua grau de pertinência maior ou igual a α . Os números *fuzzy* mais comuns são os triangulares, trapezoidais e aqueles em forma de sino (gaussianos), no método FUNED em questão, é utilizado o número *fuzzy* triangular simétrico cuja definição é dada a seguir (BARROS; BASSANEZI, 2006).

4.2.2.1 Número *fuzzy* triangular simétrico.

Problemas reais com aplicações em áreas diversas requerem com que aproximações numéricas sejam feitas a fim de obter os resultados esperados. Os números *fuzzy*, por características intrínsecas, permitem dentro de contextos como esses, quantificar a incerteza e imprecisão associada a uma dada informação com bastante eficiência.

Os números *fuzzy* são constituídos por conjuntos *fuzzy*, cujo universo é o conjunto dos números reais e os maiores valores de pertinência estão agrupados ao redor de um determinado número real chamado de valor médio, no caso da Figura 2, esse valor está definido como o ponto g . A função pertinência é monotônica em ambos os lados desse valor médio (BOAVENTURA, 2010).

Todo número real r é um número *fuzzy* particular cuja função de pertinência varia entre 0 e 1 sob determinadas condições, e nesse sentido, como mostra a equação (4), é a própria função característica.

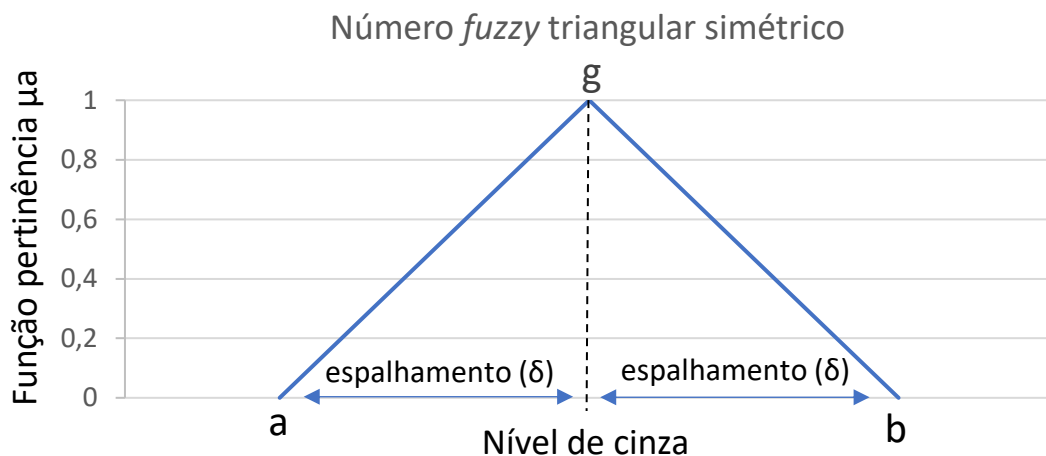
$$X_r(x) = \begin{cases} 1, & \text{se } x = r \\ 0, & \text{se } x \neq r \end{cases} \quad (4)$$

O número *fuzzy* triangular em particular, utilizado na aplicação do método FUNED, tem a seguinte função de pertinência, conforme mostra a equação (5). Possui pertinência “0” para valores superiores ao ponto b e inferiores ao ponto a . Isso significa que fora da base triangular do número *fuzzy* não há nenhuma pertinência e dentro do intervalo $[a,b]$, a pertinência varia em função da distância do ponto g .

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{g-a}, & \text{se } a < x \leq g \\ \frac{x-b}{b-g}, & \text{se } g < x \leq b \\ 0, & \text{se } x \geq b \end{cases} \quad (5)$$

O gráfico da função pertinência do número *fuzzy* triangular tem a forma de um triângulo, cuja base é construída entre os pontos $[a,b]$, e o vértice superior definido como ponto $[g, 1]$, conforme mostra a Figura 2.

Figura 2 – Gráfico da função do número *fuzzy* triangular.



Fonte: autor da dissertação.

Para que o número *fuzzy* triangular seja simétrico, a seguinte condição precisa ser verdadeira $g - a = b - g = \delta$. O parâmetro δ , também conhecido como o espalhamento do número *fuzzy*, é a variável que controla o aumento ou

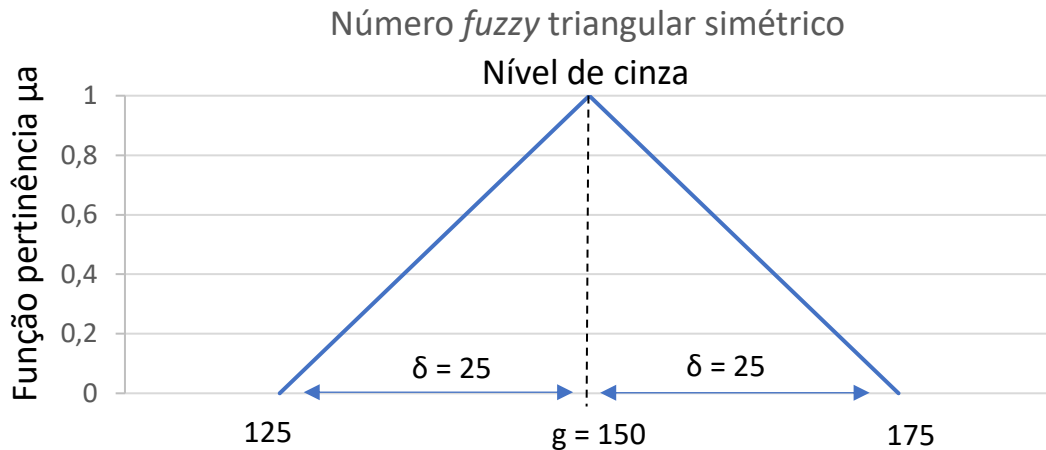
diminuição da base do triângulo, como mostra a Figura 2 e a função pertinência definida na equação (6).

$$\mu_A(x) = \begin{cases} 1 - \frac{|x-g|}{\delta}, & \text{se } g - \delta < x \leq g + \delta \\ 0, & \text{caso contrário} \end{cases} \quad (6)$$

A ilustração gráfica do número *fuzzy* triangular, como exemplo na Figura 3, é simétrica com o ponto $g = 150$ e espalhamento $\delta = 25$. Isso faz com que o ponto “a” esteja em 125 e o ponto “b” em 175, ou seja, 25 espalhados para a direita e para a esquerda em relação ao ponto “g”.

O modelo desenvolvido, em sua construção, utiliza como princípio para a interpretação dos pixels de uma imagem, números *fuzzy* triangulares simétricos, que segundo JAWAHAR e RAY (1996), são compactos e suficientes para representar a noção de valores de cinza ao redor do ponto “g”.

Figura 3 – Gráfico da função do número *fuzzy* triangular.



Fonte: autor da dissertação.

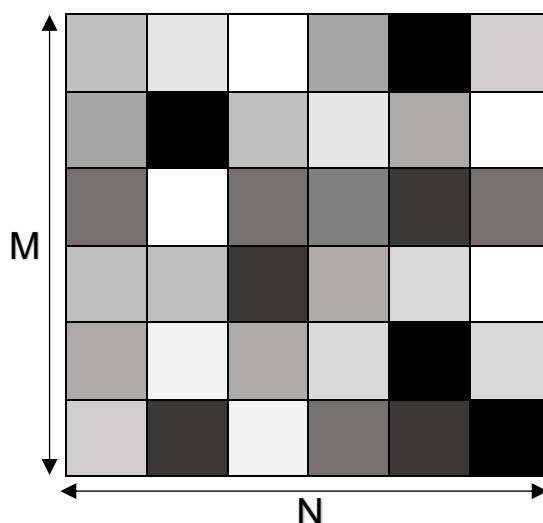
4.3 Interpretação *fuzzy* de imagem digital

O Processamento Digital de Imagens se inicia com a captura de uma imagem que corresponde à iluminação recebida dos objetos. Após a captura por um processo de digitalização, uma imagem precisa ser representada de forma apropriada para o tratamento computacional. Essa etapa pode envolver vários passos como filtragens e correções, a depender da imagem e sua qualidade (ACHARYA, 2005).

Uma imagem em tons de cinza é uma função bidimensional $f(x, y)$, onde x e y são coordenadas espaciais e o valor de f em qualquer ponto (x, y) é proporcional à intensidade luminosa. Cada ponto na grade bidimensional representado é denominado pixel, isso é necessário porque computadores não conseguem processar imagens contínuas (JÄHNE, 2002).

A Figura 4 apresenta uma imagem A de tamanho $M \times N$ pixels, com L valores de níveis de cinza variando entre 0 e $L - 1$, sendo "0" o pixel totalmente preto e " $L - 1$ " totalmente branco, usualmente, utiliza-se a representação com 256 níveis de cinza (0 a 255), esse valor é o número possível de ser representado para 8 bits no sistema binário (2^8). A imagem apresenta-se como uma composição matricial que define a localização de um pixel no arranjo em uma imagem bidimensional. O índice m aponta a posição da linha e o índice n a coluna. Uma imagem de M linhas e N colunas varia de 0 a $M - 1$ e de 0 a $N - 1$. O sentido de leitura adotado na representação espacial em uma imagem digital ocorre no índice m de cima para baixo e no índice n , da esquerda para a direita.

Figura 4 – Imagem A: Matriz ($M \times N$) com níveis de cinza.



Fonte: autor da dissertação

A imagem A é interpretada como uma matriz, onde cada um de seus elementos apresenta um valor de pertinência do nível de cinza g_{ij} em relação à uma região homogênea. A região homogênea se refere a um conjunto de pixels vizinhos que denotam uma região de fundo, que também pode ser definida como região de não borda. Portanto, a imagem pode ser representada pelo seguinte

conjunto *fuzzy* demonstrado na equação (7), ou de forma alternativa, através dos conjuntos discretos *fuzzy* através da equação (8) (BOAVENTURA, 2010).

$$A = \left\{ \frac{\langle g_{ij}, \mu_A(g_{ij}) \rangle}{g_{ij} \in \{0, \dots, L-1\}} \right\}, \quad (7)$$

$$i \in \{1, \dots, N\} \quad e \quad j \in \{1, \dots, M\}$$

ou

$$A = \sum_{i=1}^N \sum_{j=1}^M \frac{\mu_A(g_{ij})}{g_{ij}}, \quad (8)$$

A função de pertinência μ_A pode ser calculada por uma normalização simples, como descrita na equação (9).

$$\mu_A(g_{ij}) = \frac{g_{ij}}{\max_{i \in [1, N], j \in [1, M]} g_{ij}}, \quad (9)$$

Os níveis de cinza de uma imagem são interpretados como números *fuzzy*, podendo nesse contexto, inserir a variabilidade inerente aos valores de cinza e com isso, proporcionar uma abordagem mais adequada ao tratamento das imagens digitais.

Os valores de cinza podem assumir qualquer tipo de conjunto *fuzzy*. Neste caso, são adotados os números *fuzzy* simétricos e triangulares definidos pela equação (10).

$$\mu_A(x) = f(|x - g|, \delta), \quad (10)$$

onde $f(\dots)$, controla a forma do número *fuzzy* e $\delta \in R^+$ controla o espalhamento do número *fuzzy*. A equação (11) é uma outra forma de representar o número *fuzzy* triangular e simétrico.

$$\mu_A(x) = \text{MAX} \left(0, 1 - \frac{|x-g|}{\delta} \right), \quad (11)$$

A função pertinência proposta para a implementação do detector de bordas tem a seguinte formulação: para cada pixel $g(i, j)$ da imagem, interpretado como um número *fuzzy*, calcula-se a pertinência desse pixel em relação à região determinada pela vizinhança local, também conhecida como região homogênea.

Seja uma imagem A em que para cada pixel $g(i, j)$ tem-se uma janela de vizinhança $W \times W$. A função de pertinência, $\hat{\mu}_{g(i, j)}$ de cada pixel $g(i, j)$ do conjunto *fuzzy* da imagem é definida pela equação (12) (BOAVENTURA, 2010).

$$\hat{\mu}_{g(i, j)} = \frac{\sum_{k=1}^W \sum_{l=1}^W \max\left(0, 1 - \frac{|g(i, j) - A(k, l)|}{\delta}\right) - 1}{w^2 - 1}, \quad (12)$$

$i = 1 \dots N, j = 1 \dots M$

O pixel central da região em análise (janela de vizinhança 3x3 ou 5x5) é excluído do cálculo, a matriz resultante desta operação é chamada de matriz de pertinências e é definida pela equação (13) (BOAVENTURA, 2010).

$$A' = \sum_{i=1}^N \sum_{j=1}^M \frac{\hat{\mu}_{ij}}{g_{ij}}, \quad (13)$$

A função $\hat{\mu}_{ij}$ indica o grau de pertinência de cada pixel à região $W \times W$ predefinida da imagem. Valores de pertinências próximos a 1 “um” significam uma maior pertinência a região homogênea e 0 “zero” significam menor pertinência.

A matriz A' é observada como uma matriz em nível de cinza no intervalo 0 e 1, e representa as bordas presentes na imagem pelos tons de cinza mais escuros, diferentes das bordas tradicionais, cujas bordas são apresentadas pelos mais claros.

4.4 Orientação das bordas

O cálculo de orientação da borda considera a soma dos valores dos pixels na vizinhança de b_{ij} correspondente a cada orientação de referência predefinida, neste caso, cada orientação de referência possui uma soma

correspondente. O valor máximo dessas somas define o ângulo de orientação local da borda.

Para uma vizinhança de tamanho 3×3 , são definidas quatro orientações de bordas que correspondem aos ângulos $0^\circ, 45^\circ, 90^\circ, 135^\circ$. A soma dos valores de todos os pixels da borda b_{ij} e seus vizinhos numa determinada região são definidas como d_1, d_2, d_3 e d_4 representando os respectivos ângulos. Os cálculos das possíveis orientações d_i , para $i = 1..4$, em uma vizinhança (3×3 pixels) são fornecidas pelas equações (14), (15), (16) e (17).

$$d_1 = b_{i,j} + b_{i,j-1} + b_{i,j+1} \quad (14)$$

$$d_2 = b_{i,j} + b_{i-1,j+1} + b_{i+1,j-1} \quad (15)$$

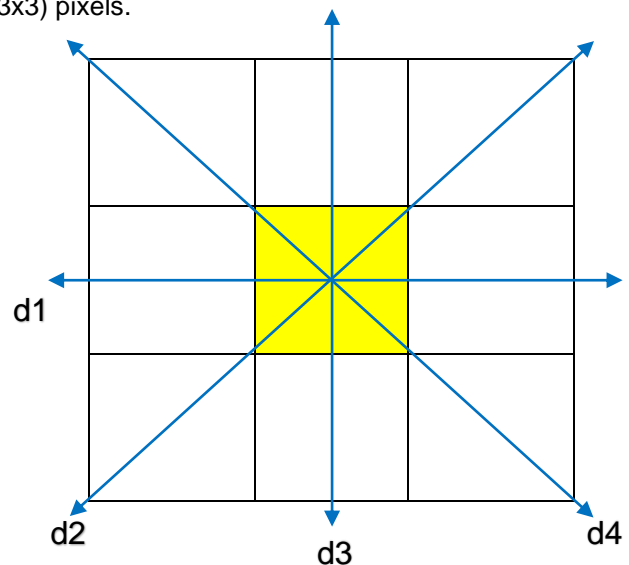
$$d_3 = b_{i,j} + b_{i-1,j} + b_{i+1,j} \quad (16)$$

$$d_4 = b_{i,j} + b_{i-1,j-1} + b_{i+1,j+1} \quad (17)$$

As quatro direções que representam os ângulos e os respectivos pixels na janela de vizinhança 3×3 , são demonstradas na Figura 5.

Figura 5 – Matriz de direções na vizinhança (3×3) pixels.

$b_{i-1,j-1}$	$b_{i-1,j}$	$b_{i-1,j+1}$
$b_{i,j-1}$	$b_{i,j}$	$b_{i,j+1}$
$b_{i+1,j-1}$	$b_{i+1,j}$	$b_{i+1,j+1}$



Fonte: autor da dissertação.

Para uma vizinhança de tamanho 5×5 , são definidas oito orientações de bordas que correspondem aos ângulos 0° , $22,5^\circ$, 45° , $67,5^\circ$, 90° , $112,5^\circ$, 135° e $157,5^\circ$. A soma dos valores de todos os pixels da borda b_{ij} e seus vizinhos numa determinada região são denominados como d_1 , d_2 , d_3 , d_4 , d_5 , d_6 , d_7 e d_8 representando os respectivos ângulos. Os cálculos das possíveis orientações d_i , para $i = 1..8$, em uma vizinhança (5×5 pixels) são fornecidas pelas equações (18), (19), (20), (21), (22), (23), (24) e (25).

$$d_1 = b_{i,j} + b_{i,j-2} + b_{i,j-1} + b_{i,j+1} + b_{i,j+2} \quad (18)$$

$$d_2 = b_{i,j} + b_{i-1,j-2} + (b_{i-1,j+1} + b_{i,j+1})/2 + b_{i+1,j-2} + (b_{i,j-1} + b_{i+1,j-1})/2 \quad (19)$$

$$d_3 = b_{i,j} + b_{i-2,j+2} + b_{i-1,j+1} + b_{i+1,j-1} + b_{i+2,j-2} \quad (20)$$

$$d_4 = b_{i,j} + b_{i+2,j-1} + (b_{i+1,j-1} + b_{i+1,j})/2 + b_{i-2,j+1} + (b_{i-1,j} + b_{i-1,j+1})/2 \quad (21)$$

$$d_5 = b_{i,j} + b_{i-1,j} + b_{i-2,j} + b_{i+1,j} + b_{i+2,j} \quad (22)$$

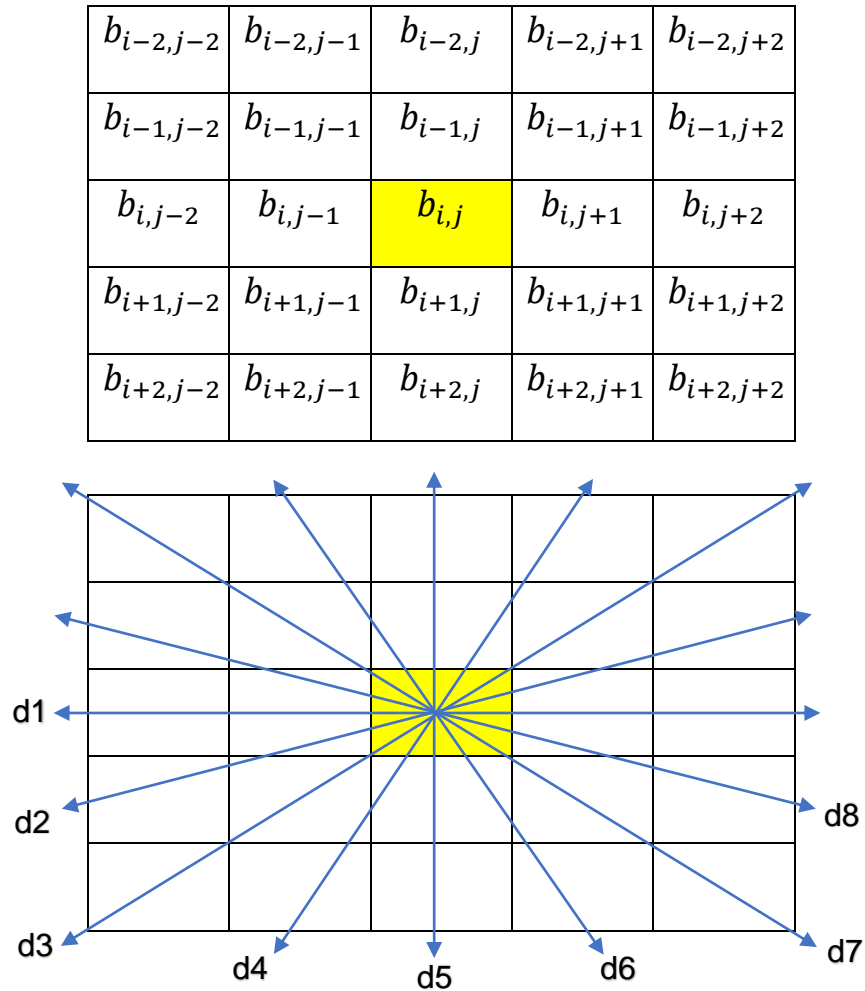
$$d_6 = b_{i,j} + b_{i+2,j+1} + (b_{i+1,j} + b_{i+1,j+1})/2 + b_{i-2,j-1} + (b_{i-1,j-1} + b_{i-1,j})/2 \quad (23)$$

$$d_7 = b_{i,j} + b_{i-2,j-2} + b_{i-1,j-1} + b_{i+1,j+1} + b_{i+2,j+2} \quad (24)$$

$$d_8 = b_{i,j} + b_{i-1,j-2} + (b_{i-1,j-1} + b_{i,j-1})/2 + b_{i+1,j+2} + (b_{i,j+1} + b_{i+1,j+1})/2 \quad (25)$$

As oito direções que representam os ângulos e os respectivos pixels na janela de vizinhança 5x5, são demonstrados na Figura 6.

Figura 6 – Matriz de direções na vizinhança (5x5) pixels



Fonte: autor da dissertação.

Com base nos cálculos das direções dk , sendo $k = 1 \dots 4$ para janela de vizinhança 3×3 e $k = 1 \dots 8$ para a janela 5×5 , define-se a direção de máximo valor $\max\{dk\}$ sendo o ângulo de referência definido como θ_l de b_{ij} na direção l . Cada elemento da matriz direcional θ é definido pela equação (26) ou (27).

$$\theta(i, j) = \theta_l \left[\left(\max \left\{ \sum_W (b_{ij})_k \right\} \right) \right] \quad (26)$$

ou

$$\theta(i, j) = \theta_l(\max\{d_k\}), \text{ sendo } d_k = \sum_{k=1}^W b_{ij} \quad (27)$$

4.5 Supressão de não máximos

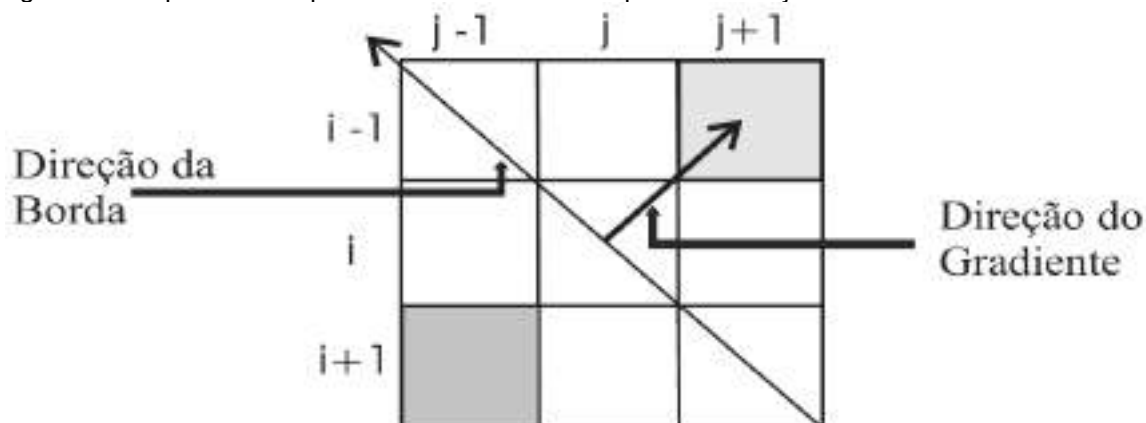
O processo de supressão de não máximos foi introduzido por CANNY (1986) com o objetivo de suprimir da matriz os valores das linhas de gradiente que não são os valores máximos. Com isso é possível afinar o cume de magnitude, ou seja, tornar as bordas da imagem mais finas.

A supressão de não máximos proposta por BOAVENTURA (2010) baseia-se na anulação dos pixels que não são máximos locais na direção perpendicular à borda. A técnica de supressão de não máximos adaptada (ANMS) utiliza-se de uma máscara de 3×3 de modo a comparar o pixel central b_{ij} com seus dois vizinhos ao longo da linha perpendicular à borda determinada pela matriz direcional. A imagem complementar B' é calculada pela equação (28).

$$B'(i, j) = ANMS(B(i, j), \theta(i, j)) \quad (28)$$

A Figura 7 ilustra o caso em que o pixel central (i, j) é examinado para verificar se o seu valor é máximo local, sendo a direção da borda de 135° .

Figura 7 – Esquema de supressão de não máximos quando a direção da borda é de 135° .



Fonte: BOAVENTURA, 2010.

4.6 Análise ROC para detectores de borda em imagens digitais

Desenvolvida com base na teoria de decisão estatística entre 1950 e 1960 para detectar sinais em radares e na psicologia social, METZ (1986) afirma que a análise ROC (*Receiver Operating Characteristic*) é utilizada por vários autores para avaliar diagnósticos médicos e em particular, na análise de imagens médicas como pode ser demonstrada nos trabalhos de PARKER (1995), SHUI e

ZHANG (2012) e LIU e FENG (2014) e em pesquisas mais recentes, na análise de classificadores de borda de SERRA (2017).

O desempenho do detector de bordas FUNED será avaliado usando o método ROC para comparar o seu desempenho ao clássico detector de borda Canny, que se mostrou no trabalho de BUENO (2016) como um dos melhores classificadores de borda.

A comparação é baseada em imagens classificadas manualmente, geralmente conhecidas como imagens de *Ground Truth* (GT). Os GTs geralmente indicam uma observação de detecção ideal (HEATH et al., 1998). Esse tipo de comparação resulta em classificações relativas que não mudam, mesmo com diferentes GTs produzidos manualmente por indivíduos diferentes ou em momentos diferentes. A posição absoluta dos pontos de ROCs podem variar, mas a ordem dos resultados é estável (HEATH et al., 1998). Na imagem GT, o valor 255 (branco) representa os pixels de borda e o valor zero (preto) indica regiões sem interesse (região homogênea).

O detector de bordas FUNED classifica os pixels de uma imagem em borda ou região homogênea (não borda). Portanto, em análise ROC, o modelo de classificação ocorre em duas classes, também conhecido como classificador binário que consiste em mapear instâncias da classe verdadeira para a classe preditiva, podendo ter dois possíveis resultados.

Para o conjunto binário da classe verdadeira rotulado como P (Positivo - Borda) e N (Negativo - Região homogênea) e o conjunto da classe preditiva (classificador) representado por p (instâncias positivas) e n (instâncias negativas), a classificação das instâncias gera quatro diferentes resultados que compõem a matriz de contingência demonstrada na Tabela 1.

A Tabela 1 da matriz de contingência mostra a classe verdadeira definida pelos pixels na imagem *Ground Truth*, também conhecida como “Padrão Ouro”, sendo as linhas P – Borda na imagem verdadeira e N – Região homogênea na imagem verdadeira. As colunas, por sua vez mostram a classe preditiva definida pelos pixels na imagem resultante do classificador (p – Borda na imagem predita e n – Região homogênea na imagem predita).

Tabela 1 – Classe preditiva.

		Classe Preditiva (Classificador)		
		p	n	
Classe Verdadeira (<i>Ground Truth</i>)	P	TP	FN	Pos
	N	FP	TN	Neg
		PPos	PNeg	Total

Fonte: autor da dissertação.

Os valores resultantes da intersecção que mostram se os resultados das predições estão corretos ou incorretos, são os seguintes:

- TP (Verdadeiro Positivo): pixel de borda nas duas imagens.
- FP (Falso Positivo): pixel de borda somente na imagem predita.
- FN (Falso Negativo): pixel de borda somente na imagem verdadeira.
- TN (Verdadeiro Negativo): pixel de borda em nenhuma imagem.

Se a classe verdadeira é definida como positiva e a predição é positiva, isso a torna TP (Verdadeira Positiva), pois foi predita como positiva e a informação é verdadeira, caso a predição ocorra como negativa para a mesma classe verdadeira positiva, isso a torna FN (Falsa Negativa), pois foi predita como negativa de maneira falsa.

Sob o mesmo raciocínio, para uma classe verdadeira negativa com a predição negativa, isso a torna TN (Verdadeira Negativa), pois foi predita como negativa e a informação é verdadeira, caso a predição ocorra como positiva para a mesma classe verdadeira negativa, isso a torna FN (Falsa Negativa), pois foi predita como positiva, mas é falsa.

A análise ROC aplicada especificamente aos detectores de borda deste trabalho, respondem da seguinte forma: se o detector reportar um pixel de borda e o GT também, o TP é aumentado. Se o pixel de borda é reportado no detector como positivo, mas no GT ele é região homogênea (não borda), o FP é incrementado. Se ambos, detector e GT reportarem como pixels de região homogênea (não borda), o TN é incrementado. Se houver pixels de bordas no GT, mas não no detector, o FN é incrementado.

Pos e Neg são as quantidades totais de instâncias positivas e negativas da classe verdadeira e PPos e PNeg são as quantidades totais de instâncias positivas e negativas da classe preditiva, respectivamente.

Os valores obtidos na matriz de contingência permitem obter algumas métricas. A TPR, também conhecida como *recall* ou sensibilidade, determina a habilidade que o classificador possui de prever corretamente a condição verdadeira, ou seja, o quão sensível é o classificador para detectar os valores positivos, conforme equação (29).

- TPR: taxa de verdadeiros positivos, sensibilidade ou *recall*.

$$TPR = \frac{TP}{Pos} \quad (29)$$

A FPR, também conhecida como (1 - especialidade), determina a habilidade que o classificador possui de prever corretamente a ausência da condição negativa da classe verdadeira, em outras palavras, a falta de especialidade do classificador, conforme mostra a equação (30).

- FPR: taxa de falsos positivos; (1 - especialidade).

$$FPR = \frac{FP}{Negat} \quad (30)$$

A taxa de acerto considera a variação da proporção das classes, por isso a relação entre TPR e a FPR com o total de positivos (Pos) e negativos (Neg) pelo valor total das instâncias (Total), conforme mostra a equação (31).

- Taxa de acerto:

$$Taxa\ de\ acerto = \frac{Pos \times TPR + Negat \times (1 - FPR)}{Total} \quad (31)$$

A Acurácia expressa a proporção dos positivos classificados corretamente e a soma da condição positiva da classe preditiva, ou seja, a proximidade entre o que é predito e o que é verdadeiro, conforme mostra a equação (32).

- Acurácia:

$$\text{Acurácia} = \frac{TP}{PPos} \quad (32)$$

A taxa de erros é o inverso da taxa de acerto, ou seja, a distância entre o que é predito e o verdadeiro, conforme mostra a equação (33).

- Taxa de Erros.

$$\text{Taxa de erro} = 1 - \text{taxa de acerto} \quad (33)$$

4.6.1 Gráficos e pontos no espaço ROC.

O gráfico ROC possui duas dimensões e relaciona a taxa de falsos positivos (*FPR*) no eixo x com a taxa de verdadeiros positivos (*TPR*) no eixo y, além disso possui algumas regiões de interesse no espaço ROC que demonstram a performance do classificador, e quanto mais próximo ou distante desse ponto, melhor ou pior é o classificador.

A primeira região de interesse representa uma classificação perfeita na Figura 8, ou seja, o ponto $A = (0, 1)$ demonstra que todos os rótulos de todas as instâncias positivas e negativas são classificados corretamente. A condição do eixo x (*Taxa de falsos positivos – FPR*) em 0 “zero”, mostra que nenhum valor foi classificado como positivo de maneira falsa, de maneira oposta, o eixo y (*Taxa de verdadeiros positivos – TPR*) em 1 “um”, mostra que todos os valores classificados como positivo, são verdadeiros. Isso mostra que os classificadores que se aproximarem do ponto A são melhores classificadores.

O ponto de interesse $B = (1, 0)$ apresenta classificações ruins, ou seja, todas as classificações incorretas. A condição do eixo x (*Taxa de falsos positivos – FPR*) em 1 “um”, mostra que todos os valores foram classificados como positivo de

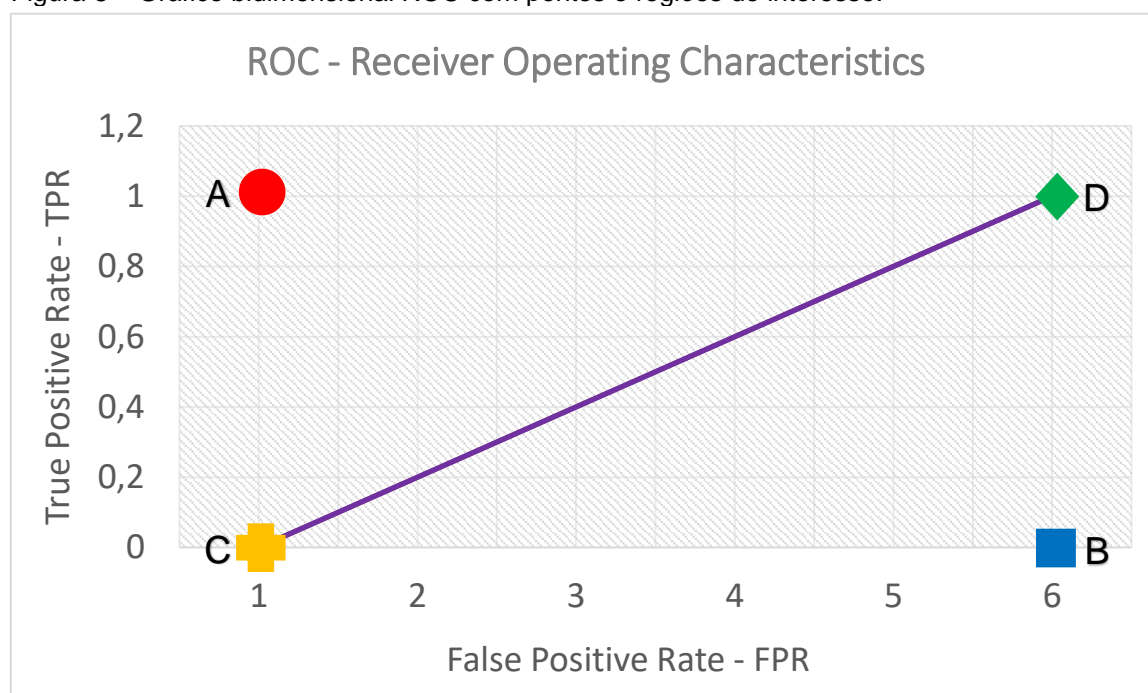
maneira errada e em contrapartida, o eixo y (*Taxa de verdadeiros positivos – TPR*) em 0 “zero” mostra que todos os valores classificados como positivo, são falsos. Isso mostra que os classificadores que se aproximarem do ponto B , são piores classificadores.

Os pontos sobre ou próximos à linha diagonal representam uma classificação aleatória, no qual o número de acertos (TPR) e erros (FPR) são semelhantes, ou seja, para cada classificação, existe 50% de chance de acerto ou erro. Isso mostra que os classificadores que se aproximarem da diagonal, são classificadores com características aleatórias.

A região próxima ao ponto $C = (0,0)$ caracteriza um classificador que erra e acerta muito pouco, se o classificador estiver exatamente no ponto C , significa que ele não definiu uma condição verdadeira como falsa, e isso é bom por não ter errado, mas da mesma forma, não definiu uma condição verdadeira como positiva, o que não é bom por não ter acertado. Esse é o típico caso de um classificador que não está classificando.

De forma oposta ao ponto $C = (0,0)$, há o ponto $D = (1,1)$ que representa um classificador que possui um número de acertos e erros muito altos. Esse é o típico classificador que classifica quase tudo, por esse motivo acerta, mas também erra muito.

Figura 8 – Gráfico bidimensional ROC com pontos e regiões de interesse.



Fonte: autor da dissertação.

Analisando os pontos exemplificados e a diagonal principal na Figura 8, verifica-se que as melhores classificações são aquelas que estão mais próximas do ponto $A(1,1)$, no entanto, para ter certeza sobre essa informação, também é necessária a verificação dos impactos dos erros e acertos da classificação final.

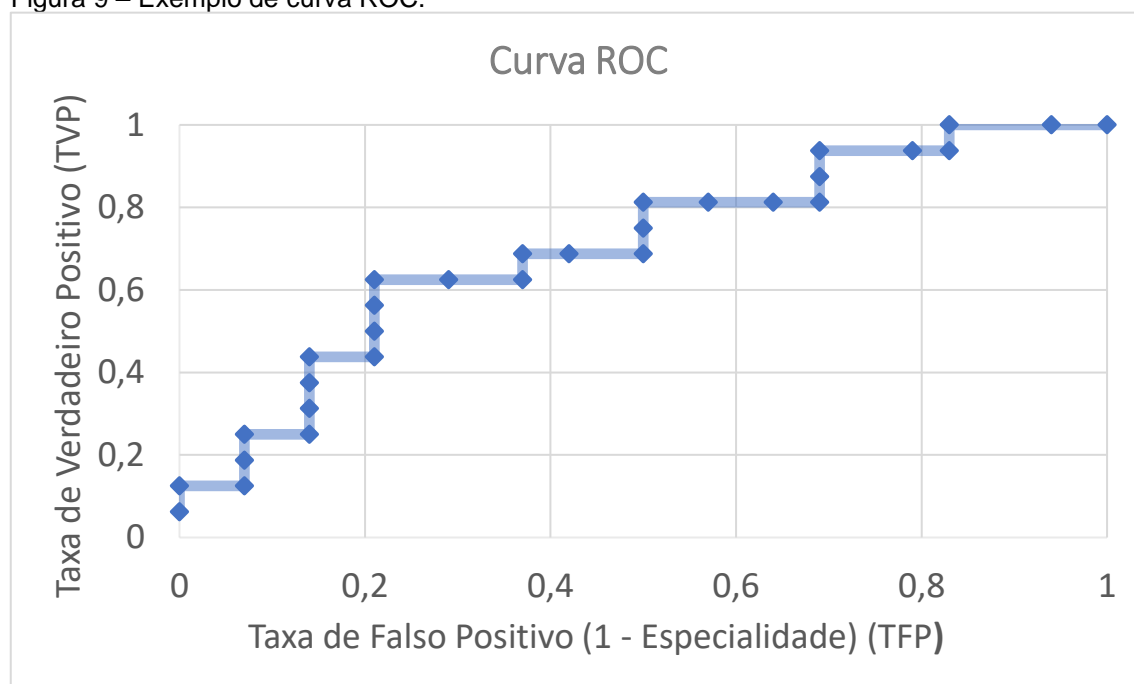
4.6.2 Curvas no espaço ROC.

O classificador discreto gera apenas um ponto no espaço, ou seja, é desenvolvido para produzir uma única classe por instância. Nesse sentido, a aplicação de mais limiares produzirá conjuntos distintos de pontos no espaço ROC.

Para construir uma Curva ROC é preciso relacionar os dados em ordem decrescente da probabilidade estimada do modelo, calcular a taxa de verdadeiro positivo (*TVP*) e a taxa de falso positivo (*TFP*) para cada linha de forma cumulativa. O gráfico considera a taxa de verdadeiro positivo (*TVP*) no eixo *Y* e a taxa de falso positivo (*TFP*) no eixo *X*.

Para cada matriz de contingência são calculados valores de *FPR* e *TPR* que estão dispostos no gráfico para formar a curva ROC do classificador, conforme exemplo apresentado na Figura 9.

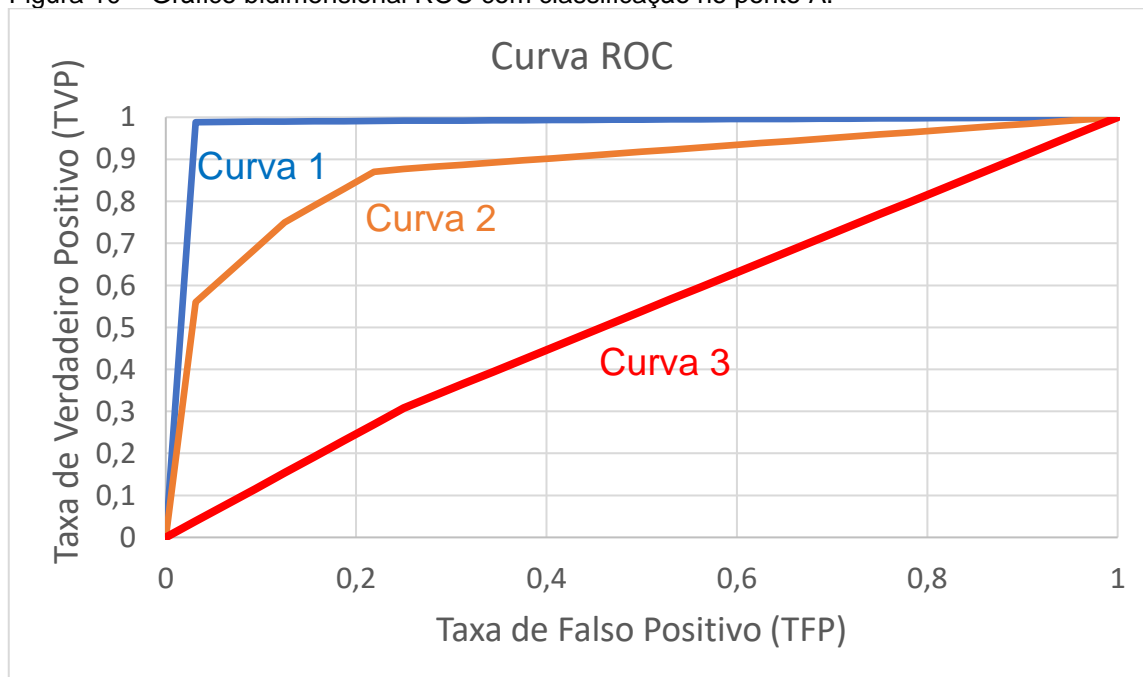
Figura 9 – Exemplo de curva ROC.



Fonte: autor da dissertação.

É possível, no mesmo gráfico, inserir várias curvas ROCs para comparar os resultados entre classificadores. A Figura 10 mostra como exemplo, três classificadores de bordas hipotéticos e suas respectivas curvas.

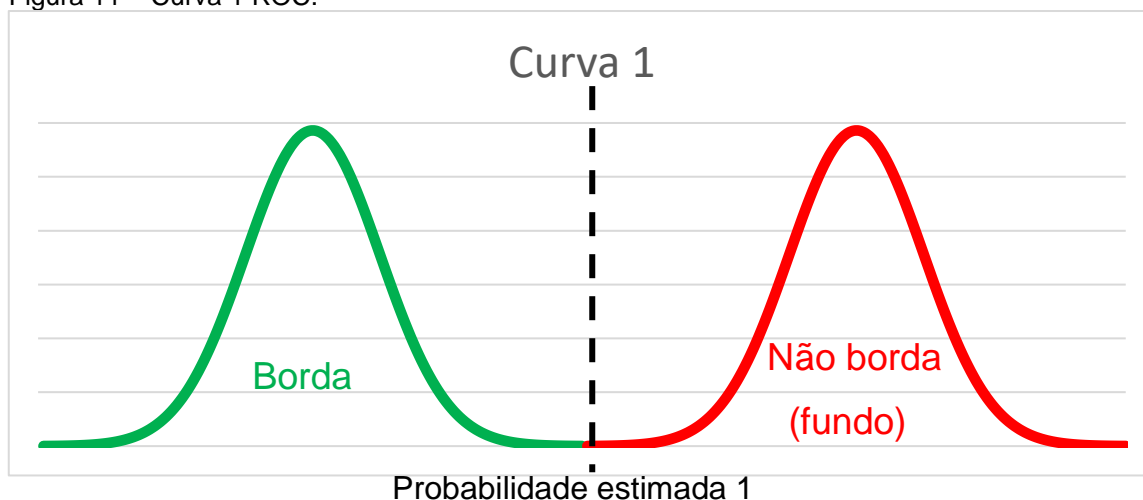
Figura 10 – Gráfico bidimensional ROC com classificação no ponto A.



Fonte: autor da dissertação.

A curva 1 apresenta um modelo próximo do ideal, mesmo com *threshold* baixo o modelo quase não apresenta falsos positivos e a taxa de verdadeiro positivo é muito alta, próximo de 100%, ou seja, quase todos os pixels de borda e região homogênea são classificados corretamente, conforme mostra a Figura 11.

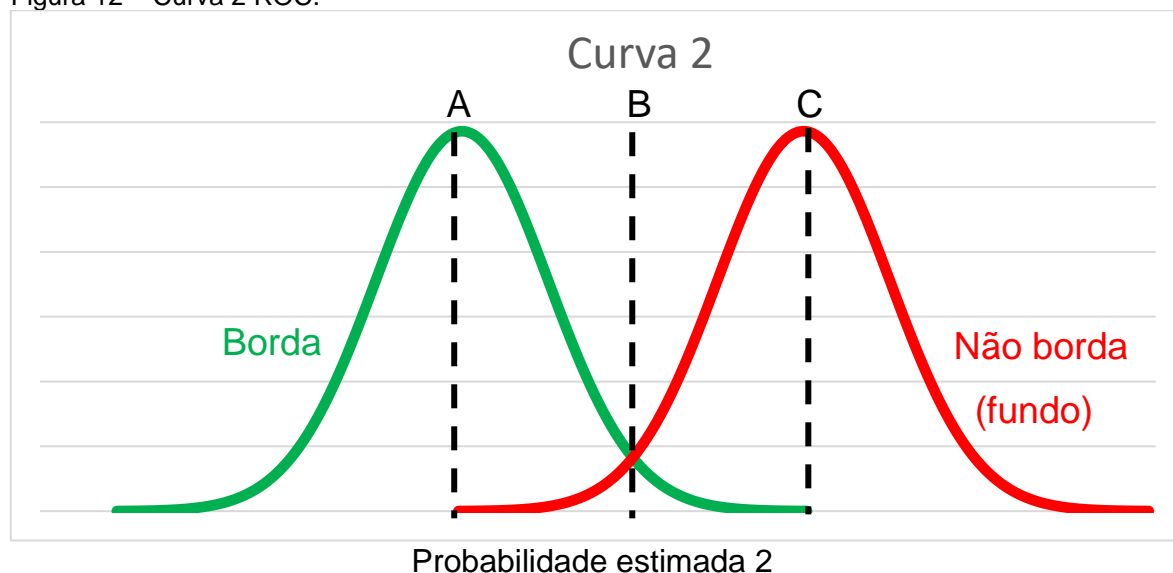
Figura 11 – Curva 1 ROC.



Fonte: autor da dissertação.

A curva 2, apresenta alguns falsos positivos e falsos negativos. A Figura 12 mostra uma região entre o ponto A e C em que o modelo apresentado pode não classificar da maneira correta.

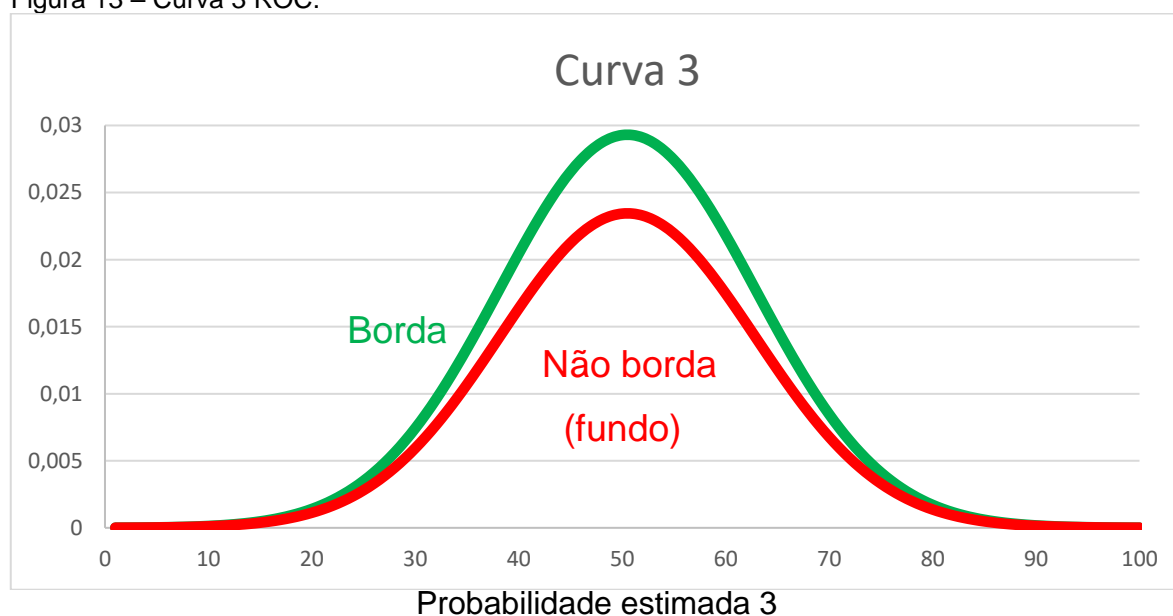
Figura 12 – Curva 2 ROC.



Fonte: autor da dissertação.

A curva 3 classifica de maneira aleatória, conforme mostra Figura 13, mas há curvas piores, elas não foram demonstradas graficamente, mas estariam abaixo da curva 3.

Figura 13 – Curva 3 ROC.



Fonte: autor da dissertação.

Para *threshold* em 1, todos os valores de qualquer classificador são preditos como 0 “zero”, isso implica na taxa de verdadeiros positivos e falsos positivos nulos, ou seja, $TVP = 0$ e $TFP = 0$. Com a diminuição do *threshold*, a taxa de verdadeiro positivo e falso positivo aumenta.

Se o *threshold* for 0 “zero”, todos os valores são preditos como 1 “um”, isso faz com que todos os verdadeiros positivos sejam classificados corretamente ($TVP = 100\%$) e de forma contrária, nenhum verdadeiro negativo seja classificado ($TFP = 100\%$).

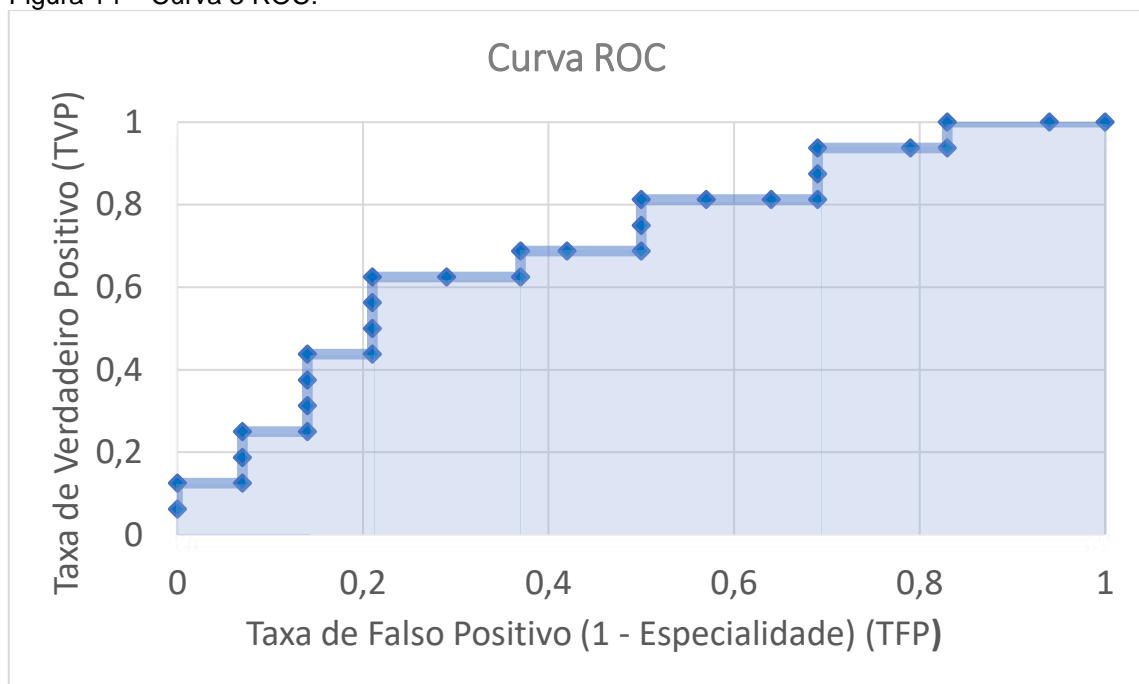
O *threshold* ótimo pode ser escolhido no ponto em que se tem a maior TVP para uma menor TFP . O modelo perfeito é aquele que classifica todos os verdadeiros positivos corretamente ($TVP = 100\%$) sem nenhum falso positivo ($TFP = 0\%$).

4.6.3 Área abaixo da curva no espaço ROC.

Comparar curvas ROC graficamente pode ser uma tarefa difícil, principalmente quando uma curva está muito próxima da outra. Como solução, é possível calcular a área abaixo da curva ROC, conhecida como AUC (*Area Under the Curve*). Esse é um método que permite medir a área para comparar os classificadores. A medida do desempenho consiste no resultado de uma grandeza escalar que varia entre 0 e 1, sendo a pior e a melhor classificação respectivamente. Uma área próxima de 1, mostra que a maioria dos pontos de construção da curva estão próximos ao ponto $(1,0)$, tendo uma taxa alta de verdadeiros positivos e uma taxa baixa de falsos positivos. Isso mostra a melhor classificação em função do maior valor da AUC.

A AUC pode ser calculada por dois métodos numéricos: a regra dos trapézios ou por métodos estatísticos. A Figura 14 mostra a curva ROC e a AUC com o cálculo realizado pelo método de cálculo de trapézio.

Figura 14 – Curva 3 ROC.



Fonte: autor da dissertação

5 METODOLOGIA

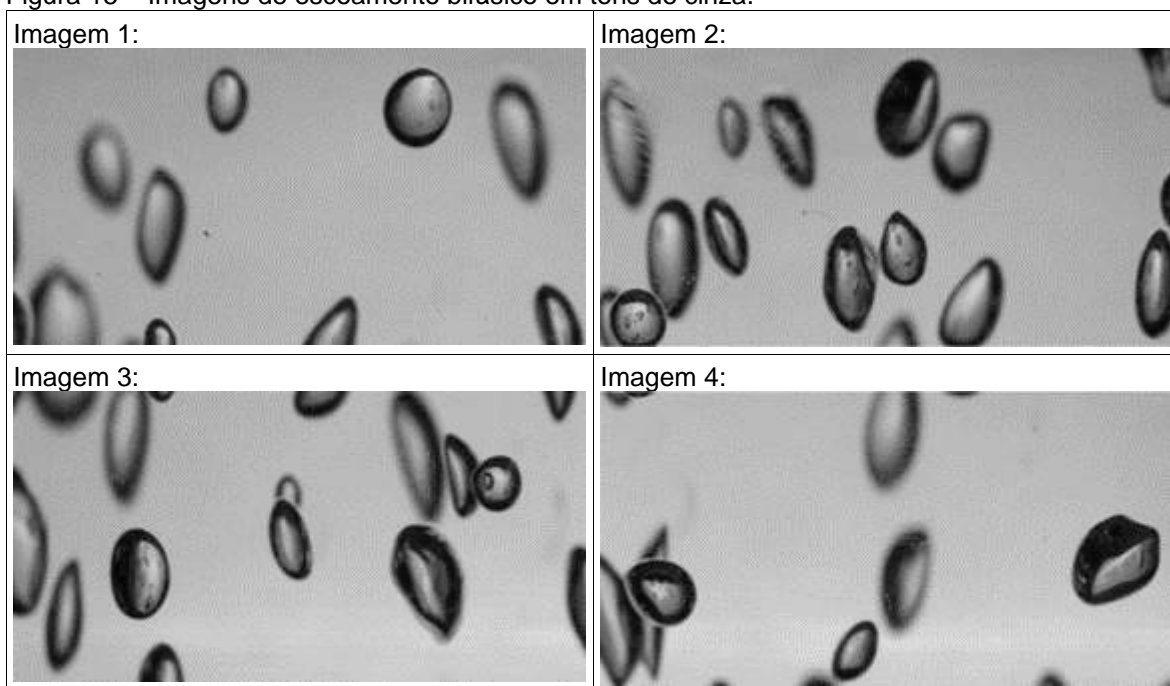
5.1 Desenvolvimento da Imagem *Ground Truth*.

O desempenho dos detectores de contornos pode ser avaliado com o uso do *Receiver-Operating Characteristics (ROC)* para comparar o resultado com imagens produzidas manualmente por um ser humano com base na observação visual e a definição de cada pixel como borda ou região homogênea. Essas imagens são intituladas de *Ground Truth (GT)* e indicam uma detecção ideal de observação (HEATH et al. 1998). O ranking relativo não muda mesmo com *GTs* distintos produzidos por pessoas ou ocasiões diferentes, apenas a posição absoluta dos pontos *ROCs* variam, mantendo-se os resultados estáveis (BOWYER et al, 2001).

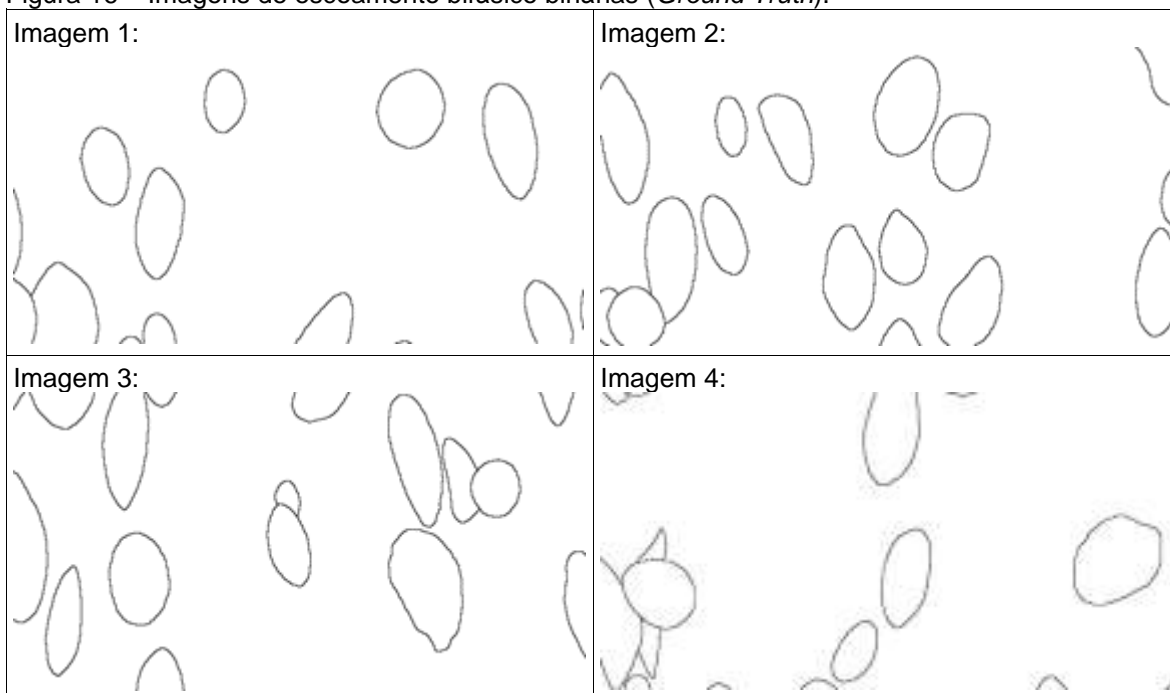
Para este trabalho foram criadas 20 imagens *Ground Truth* no software “*ImageJ*” através de um especialista que definiu pixel por pixel o que era borda e fundo. As 20 imagens de 360 x 690 pixels geradas foram produzidas a partir de quatro imagens em tons de cinza. Para cada imagem em tons de cinza, foram criadas cinco imagens *GTs* de mesma dimensão com bordas variando entre 1 e 5 pixels para análise da influência das bordas mais finas e mais grossas no resultado de cada método.

As simulações permitiram verificar que os resultados *ROCs* foram os mesmos na variação da espessura das bordas, não interferindo diretamente nos resultados de cada método. Sendo assim, as imagens *GTs* utilizadas foram as construídas com as bordas mais finas, ou seja, de apenas um pixel. A Figura 15 mostra as imagens em tons de cinza e a Figura 16 as imagens *GTs* utilizadas na análise.

Figura 15 – Imagens de escoamento bifásico em tons de cinza.



Fonte: BUENO, 2016.

Figura 16 – Imagens de escoamento bifásico binárias (*Ground Truth*).

Fonte: autor da dissertação.

Após a criação das imagens *GTs*, implementou-se o código em *Python* do método FUNED. O algoritmo desenvolvido foi dividido em quatro partes para uma melhor compreensão das etapas do método, conforme pode ser observado nos Apêndices A, B, C e D. O primeiro algoritmo gera a imagem pertinência, o

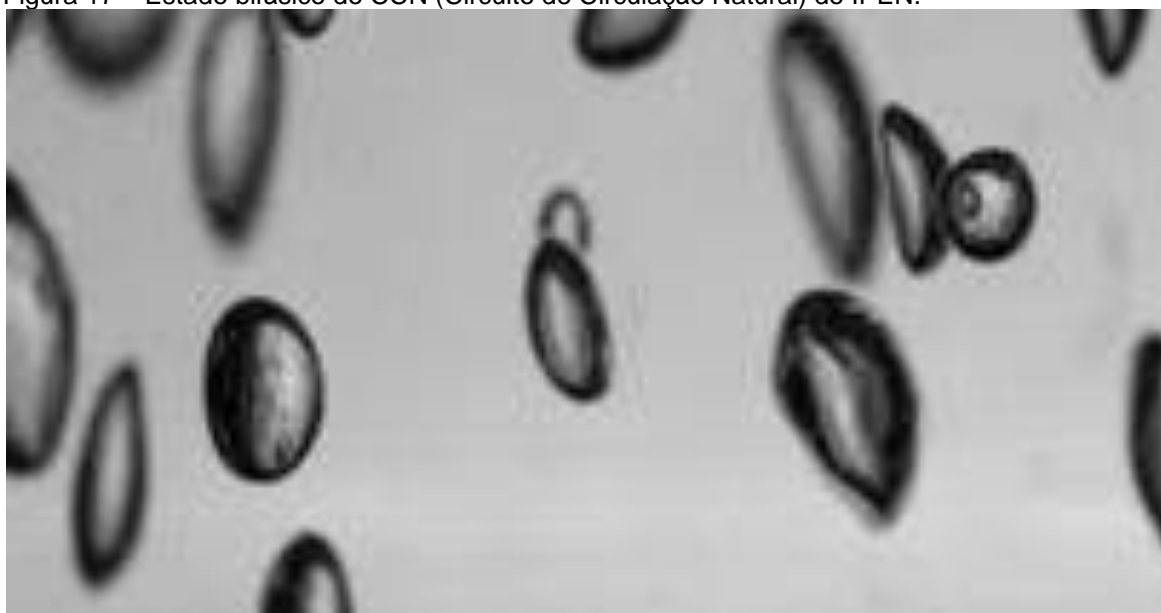
segundo a imagem segmentada sem a supressão de não máximo, o terceiro algoritmo a imagem finalizada e o quarto algoritmo é destinado a transformação da imagem em *Dataset* para análise na curva ROC.

5.2 Implementação do detector de bordas.

O detector de bordas FUNED foi executado em linguagem de programação *Python* no ambiente computacional *Notebook Jupyter*, uma ferramenta que permite a criação de “*software* de código aberto”. O algoritmo é aplicado na detecção de bordas em imagens digitais de escoamento bifásico. As imagens utilizadas foram capturadas por uma câmera profissional de alta resolução com sensor CMOS de 21 megapixels, no Circuito de Circulação Natural (CCN) presente no IPEN/CNEN-SP, como descrito no item 4.1.

As imagens utilizadas (690 x 360 pixels) relacionam diferentes fases e pertencem aos ciclos de instabilidade que são divididos em três fases, intitulados como períodos de incubação, expulsão e reenchimento, contendo padrões de escoamento instáveis, descritos detalhadamente em outras investigações (ANDRADE *et al.*, 2000; MESQUITA *et al.*, 2012). A imagem representada na Figura 17 mostra o escoamento bifásico de fase de incubação em níveis de cinza capturada do CCN, como exemplo.

Figura 17 – Estado bifásico do CCN (Circuito de Circulação Natural) do IPEN.



Fonte: BUENO, 2016.

O algoritmo exige como informação de entrada por parte do usuário a inserção de uma imagem como dado de entrada e a definição do espalhamento do número *fuzzy* triangular. Essa definição permite aumentar ou diminuir a base do triângulo para ajustar os resultados do detector FUNED, o aumento do espalhamento permite a detecção de uma faixa mais ampla na pertinência dos níveis de cinza para pixels definidos como fundo e a diminuição, uma faixa mais estreita, o que conseqüentemente acaba gerando uma sensibilidade maior aos pixels de borda.

Para resultados com elevadas detecções de ruídos, bordas grossas e falsas, sugere-se o aumento do espalhamento para a obtenção de bordas mais finas e com menos ruídos. O ajuste é definido pelo usuário em função da análise dos resultados obtidos e permite uma melhor adequação aos mais variados tipos e qualidades de imagens.

Outra variável que deve ser inserida pelo usuário é a matriz de varredura, podendo ser 3×3 ou 5×5 . A escolha desses valores permite, em conjunto com o espalhamento, uma melhor detecção de bordas mais grossas na vizinhança 5×5 e bordas mais finas na vizinhança 3×3 . A matriz considerada como uma vizinhança local define o pixel central e o retira da análise. Todos os outros pixels são definidos como números *fuzzy*. Considera-se a pertinência de cada pixel em relação à vizinhança e constrói-se a partir destes valores uma nova imagem definida como imagem pertinência.

A imagem pertinência é uma matriz intermediária que tem a mesma quantidade de pixels da imagem original, porém, os níveis de cinza que variavam entre 0 e 255 na imagem original, agora variam entre as pertinências "0" e "1", sendo "0" o pixel que menos pertence à região da vizinhança (borda), e "1" o que mais pertence (não borda).

Criada a imagem pertinência e fixado o limiar (do inglês "*threshold*" ou também definido como corte- α) entre o que é borda e fundo, obtém-se uma imagem segmentada sem supressão de não máximos. Essa imagem intermediária é utilizada como entrada de dados para calcular a direção da borda através do valor máximo das somas nas direções de forma complementar, uma vez que a função pertinência de maior valor identifica o pixel de fundo e não o pixel de borda. Elas podem ser 4 possibilidades na matriz 3×3 , correspondentes aos ângulos 0° , 45° ,

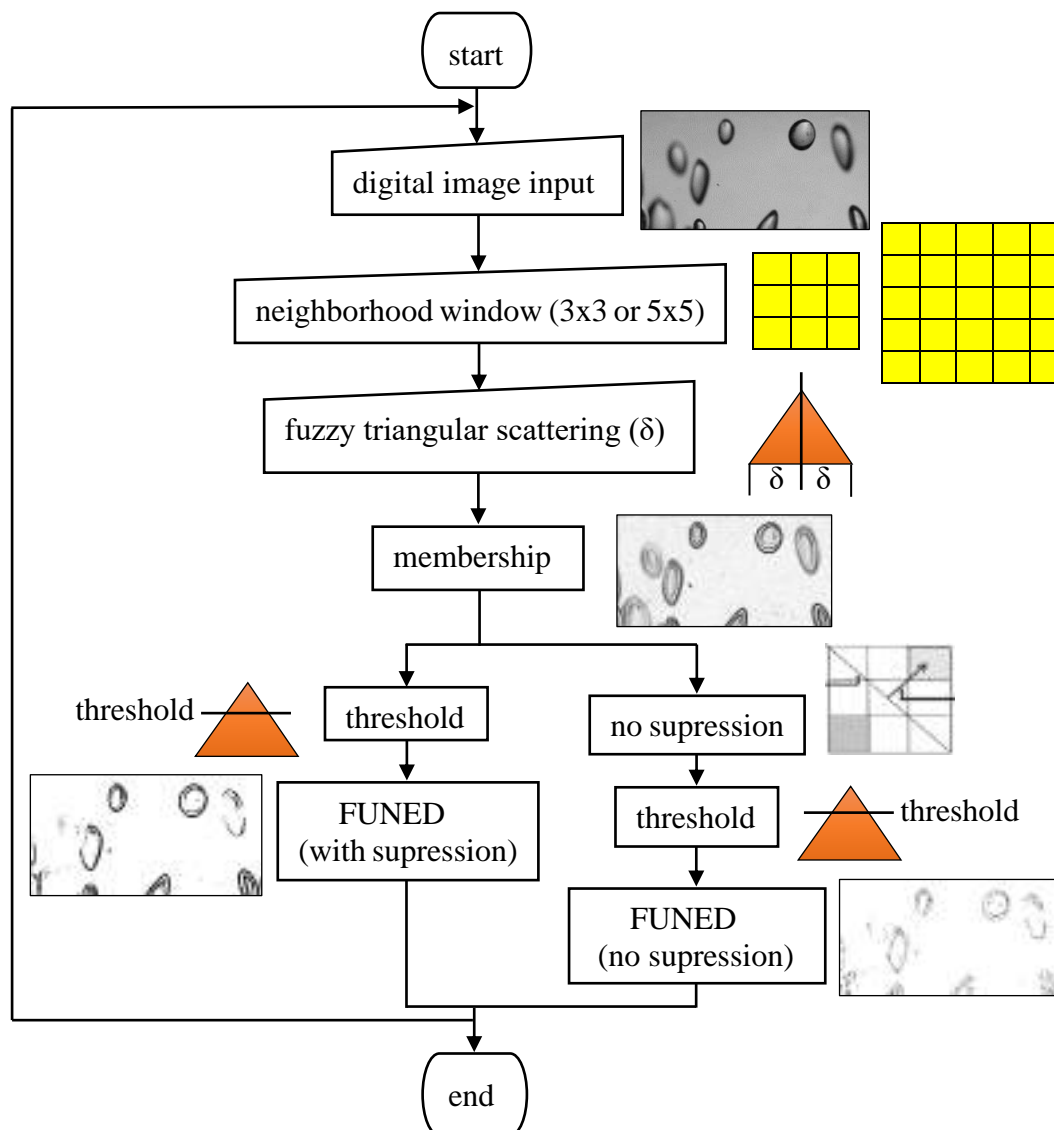
90°, 135° ou 8 possibilidades na matriz 5x5, correspondentes aos ângulos 0°, 22,5°, 45°, 67,5°, 90°, 112,5°, 135°. A direção somada complementar de maior valor corresponde à orientação da borda.

Após a definição da orientação da borda, realiza-se o processo de supressão de não máximos com o objetivo de suprimir da matriz os valores das linhas de gradiente que não são os valores máximos. Essa técnica permite afinar os contornos (bordas) pois todos os valores de pixel que não estão na direção da borda serão definidos como fundo.

A técnica de supressão de não máximos adaptada (*ANMS*) utiliza-se de uma máscara de modo a comparar o pixel central b_{ij} com seus dois vizinhos ao longo da linha perpendicular à borda determinada pela matriz direcional, isso permite o afinamento das bordas.

Para uma visualização adicional dos passos dados no detector de bordas FUNED, construiu-se um fluxograma que mostra de forma esquemática e visual o caminho da imagem digital desde a entrada no algoritmo até à saída como imagem predita. Com o objetivo de demonstrar a importância de todos os passos envolvidos no detector, criaram-se imagens intermediárias (imagem pertinência e sem supressão de não máximos) para melhor compreensão da função de cada passo no processo. A imagem FUNED (sem supressão de não máximos) é mostrada apenas para demonstrar a influência da técnica de supressão de não máximos no resultado final do método. O fluxograma esquemático é demonstrado na Figura 18.

Figura 18: Fluxograma de funcionamento.

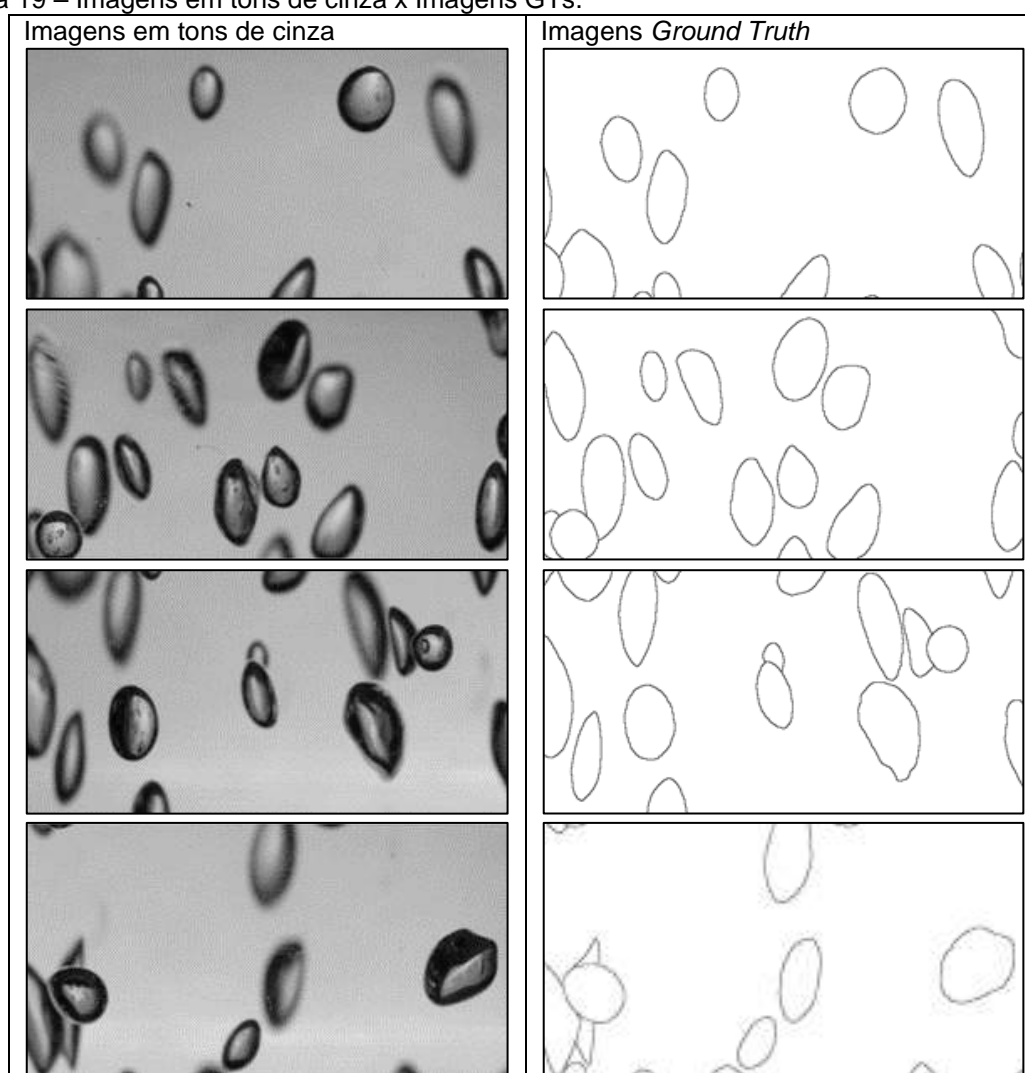


Fonte: autor da dissertação.

6 RESULTADOS

Os resultados do detector de bordas FUNED aplicado em imagens de escoamento de circulação natural do circuito experimental do IPEN teve o seu desempenho avaliado pelo método de representação gráfica das características de operação do receptor (*Receiver Operating Characteristic Curve - ROC*) com base nas imagens *GTs* e em comparação com o clássico detector de Canny. A Figura 19 apresenta as imagens utilizadas para análise do detector de bordas e suas respectivas imagens *Ground Truth*.

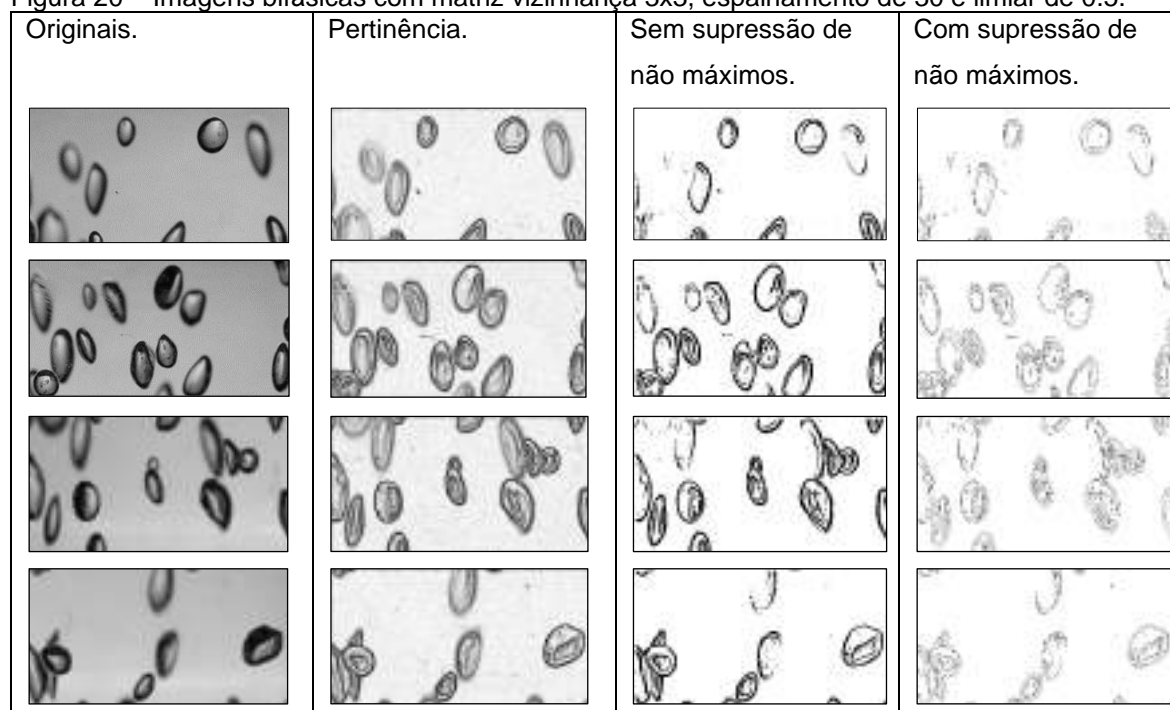
Figura 19 – Imagens em tons de cinza x Imagens *GTs*.



Fonte: autor da dissertação.

A Figura 20 apresenta as imagens utilizadas e os resultados intermediários e finais do método FUNED. A simulação ocorreu para uma janela de vizinhança 3×3 , sob as condições de espalhamento *fuzzy* triangular de 30 e *threshold* de 0,5. As quatro imagens dispostas nas figuras são respectivamente, a imagem pertinência que demonstra o grau de pertencimento do pixel à região homogênea (fundo), a segunda, uma imagem segmentada sem a supressão de não máximo para demonstrar a importância da supressão na visualização dos contornos e a terceira imagem, o resultado do algoritmo completo com a supressão de não máximos.

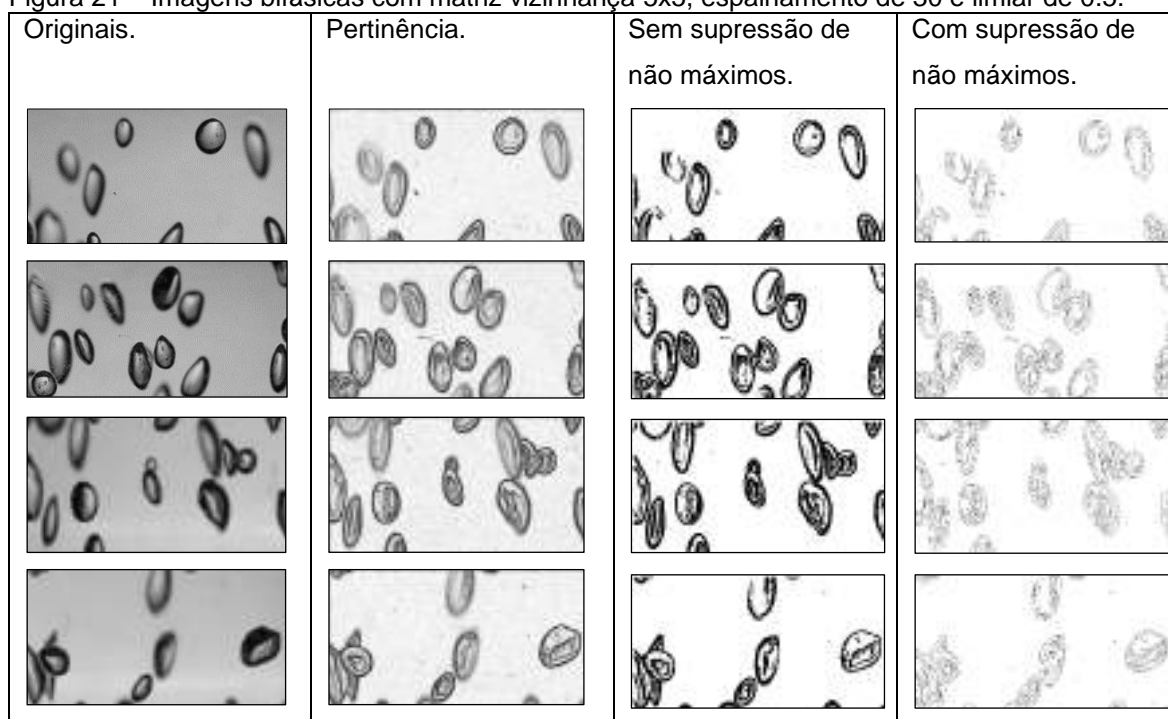
Figura 20 – Imagens bifásicas com matriz vizinhança 3×3 , espalhamento de 30 e limiar de 0.5.



Fonte: autor da dissertação.

As imagens na Figura 21 são resultantes da aplicação da janela de vizinhança 5×5 e demonstram o impacto na variação do tamanho da janela sobre a detecção de bordas. O resultado de bordas mais grossas mostra que o aumento da janela de análise de 3×3 para 5×5 permite uma maior detecção de pixels como borda. Isso resulta em bordas mais espessas e maior detecção de ruídos, quando comparada à janela 3×3 .

Figura 21 – Imagens bifásicas com matriz vizinhança 5x5, espalhamento de 30 e limiar de 0,5.



Fonte: autor da dissertação.

As imagens na Figura 22 mostram os impactos na detecção de bordas em função da variação do espalhamento *fuzzy*. Com o aumento do espalhamento *fuzzy*, a base triangular aumenta e conseqüentemente menos bordas são detectadas em função da maior pertinência em relação à região homogênea. Os resultados mostram a influência da diminuição de pixels como borda em função do aumento no espalhamento *fuzzy* triangular.

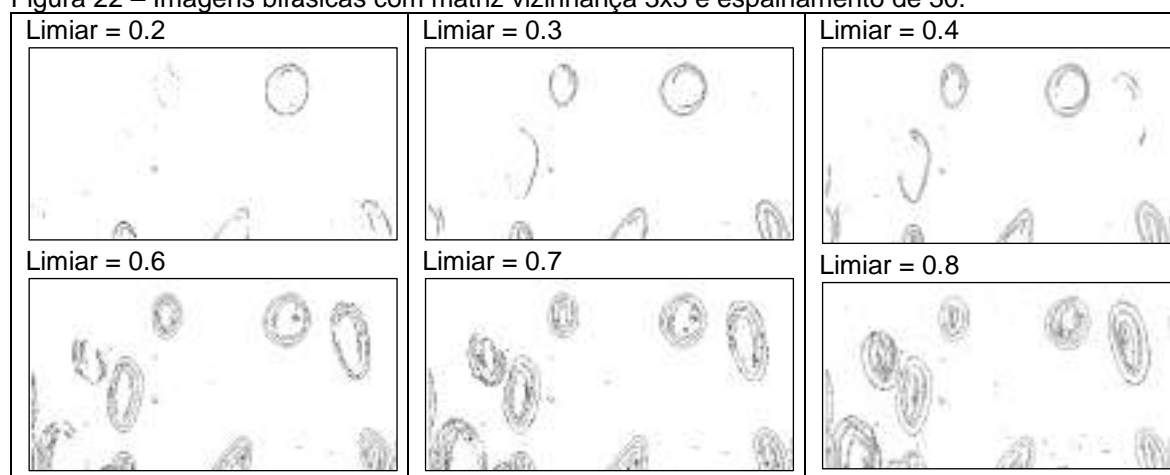
Figura 22 – Imagens bifásicas com matriz vizinhança 3x3 e limiar de 0,5.



Fonte: autor da dissertação.

A seguir são demonstradas as mudanças geradas na detecção de bordas em relação à variação do limiar (*threshold*). Com a variação do limiar é possível controlar o nível de pertinência a ser definido como borda ou fundo. O aumento do limiar, como pode ser observado na Figura 23, mostra que mais pixels da imagem são definidos como borda e isso conseqüentemente faz com que mais ruídos também sejam detectados como borda.

Figura 22 – Imagens bifásicas com matriz vizinhança 3x3 e espalhamento de 30.



Fonte: autor da dissertação.

O algoritmo FUNED, neste trabalho, compara quatro imagens preditas pelo método com imagens ideais *GTs*. A mesma avaliação ocorre com o clássico detector de bordas Canny para posterior análise pelo método ROC.

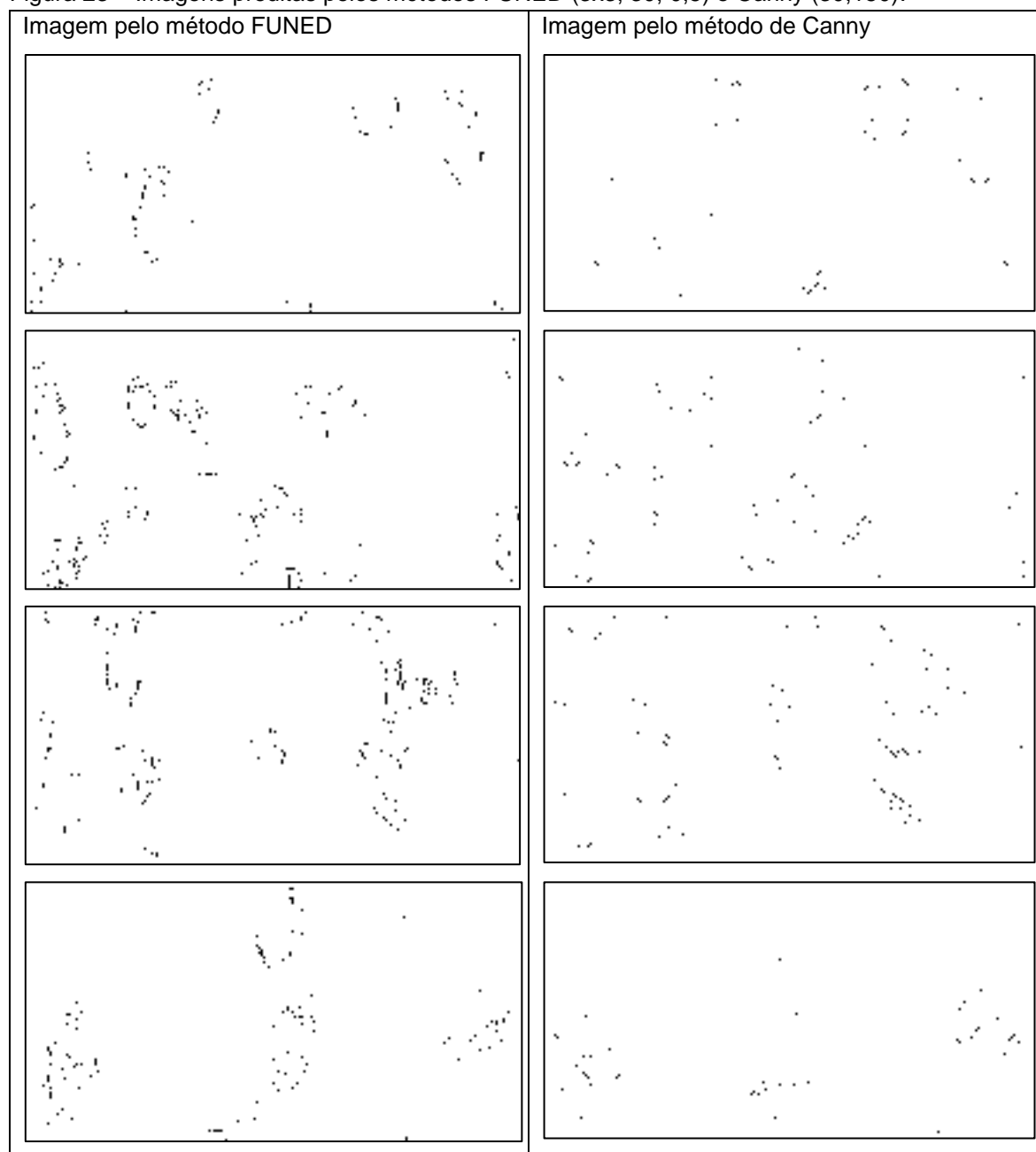
O detector de bordas de Canny realiza uma filtragem para atenuação de ruídos através de um operador de primeira ordem em função do gradiente da imagem. Além disso, o algoritmo faz o uso de dois limiares, um inferior e outro superior para considerar o que é borda ou fundo. As informações que forem filtradas pelo operador de primeira ordem só serão consideradas caso o pixel em questão esteja conectado a um pixel acima do limiar mais alto, do contrário, ele é desprezado.

O ajuste dos limiares superior e inferior permite flexibilidade. A definição de valores muito altos pode fazer com que informações sejam perdidas e de forma contrária, os valores baixos podem fazer com que informações falsas sejam consideradas como importantes.

Os ajustes dos parâmetros dos dois algoritmos em análise, FUNED e Canny, foram definidos de forma padrão como sugeridos em seus respectivos

trabalhos de origem. Canny em 1976 recomendou o valor de 30 para baixo limiar e 150 para alto limiar e Boaventura em 2010 sugeriu a matriz 3x3, espalhamento de 30 e limiar de 0,5 para o detector FUNED. A Figura 23 mostra as imagens resultantes dos detectores.

Figura 23 – Imagens preditas pelos métodos FUNED (3x3, 30, 0,5) e Canny (30,150).



Fonte: autor da dissertação.

Os resultados de desempenho dos detectores de bordas FUNED e de Canny são avaliados pelo método de representação gráfica das características de operação do receptor (*Receiver Operating Characteristic Curve - ROC*). As imagens

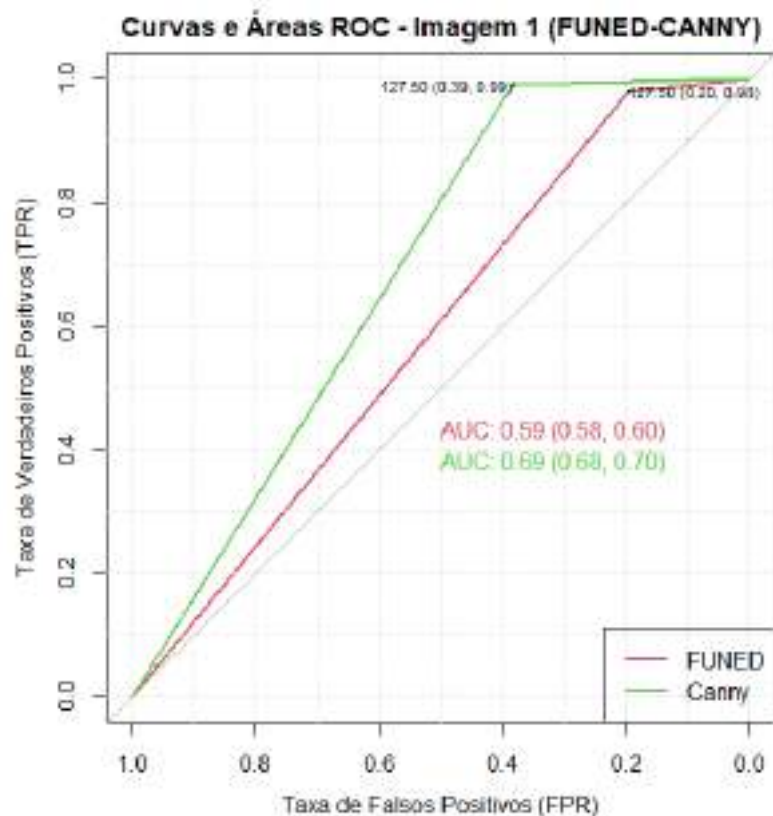
finais geradas pelos detectores juntamente com suas respectivas imagens *Ground Truth* - *GTs* foram transformadas em um *Dataset* para implementação da análise *ROC*.

Os detectores FUNED e Canny, analisados na curva *ROC* geraram boas respostas, ambos desenvolveram boa detecção de bordas, contornos bem localizados e com poucos ruídos, sendo que o de Canny teve destaque por apresentar bordas mais definidas, menos ruídos e borda mínima por conta de seu duplo limiar, como mostram os gráficos *ROC* na Figura 24, Figura 25, Figura 26 e Figura 27, os quais apresentam os resultados obtidos nas curvas geradas, as respectivas áreas abaixo da curva e a sensibilidade e especialidade de cada método aplicado no conjunto de imagens.

A análise da curva FUNED no gráfico da Figura 24, permitiu observar que o método aplicado alcançou um grau de sensibilidade de 0,98, isso mostra a eficiência em determinar corretamente os pixels de fundo com 98% de acertos. Além disso, a especialidade do método que determina o percentual de acertos na definição dos pixels como borda, foi de 0,20, ou seja, eficiência de 20% na detecção das bordas.

A curva característica do método de Canny conforme demonstrado na Figura 24, demonstrou um grau de sensibilidade de 0,99 e especialidade de 0,39. Ainda observando a Figura 24, foi analisada a área abaixo da curva (do inglês "*Area Under the Curve* - *AUC*") que mostra de forma complementar a eficiência dos métodos apresentados, nesse caso, as respectivas *AUCs* dos métodos FUNED e Canny foram de 0,59 e 0,69. Isso significa que 59% dos pixels no método FUNED e 69% dos pixels no método de Canny, foram definidos como borda ou fundo de maneira correta. Todos os valores e análise são demonstrados no gráfico da Figura 24.

Figura 24 – Gráfico bidimensional ROC - Imagem 1.

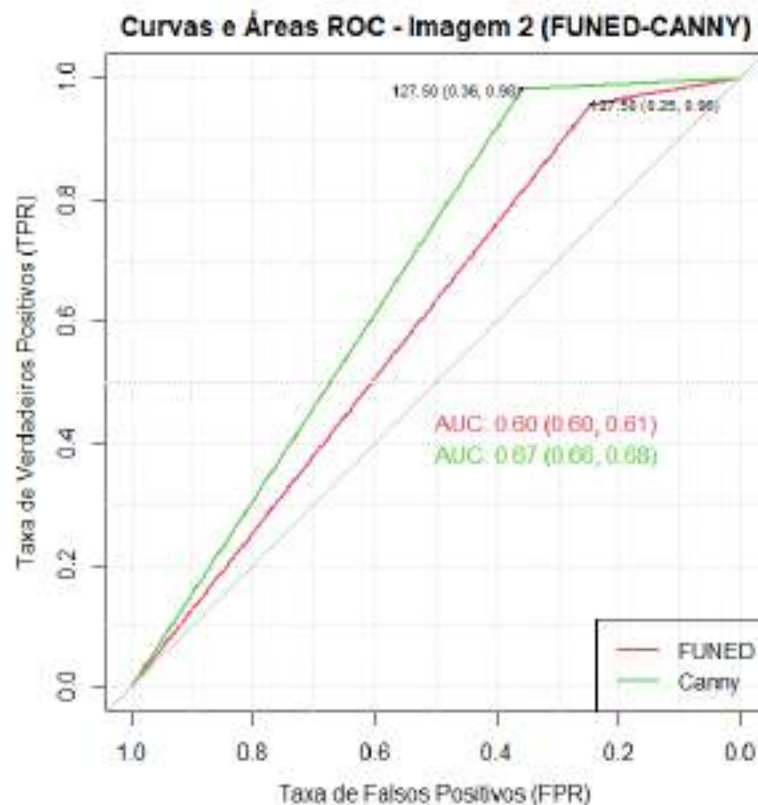


Fonte: autor da dissertação.

A partir da análise dos dados obtidos na curva do gráfico da Figura 25, observou-se que para a segunda imagem, também capturada do circuito de circulação natural, o método FUNED obteve um grau de sensibilidade de 0,96 e especialidade de 0,25. Pelo método de Canny, o grau de sensibilidade foi de 0,98 e a especialidade de 0,36.

A área abaixo da curva *AUC* do método FUNED e Canny verificado na Figura 25, respectivamente, foi de 0,60 e 0,67. Isso mostra que nesta imagem 60% dos pixels no método FUNED e 67% dos pixels no método de Canny, foram definidos como borda ou fundo de maneira correta.

Figura 25 – Gráfico bidimensional ROC - Imagem 2.

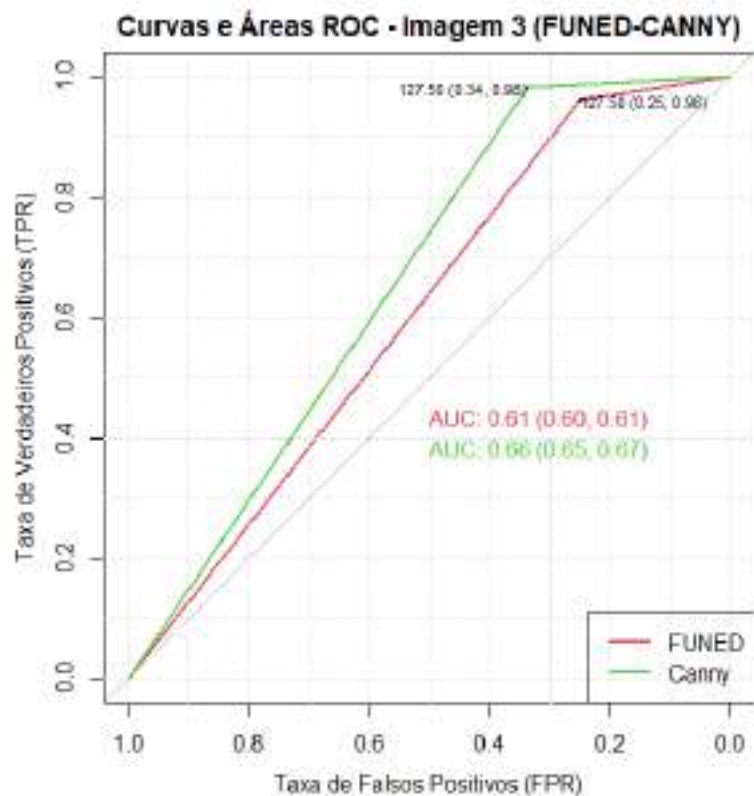


Fonte: autor da dissertação.

As curvas nos gráficos analisadas na Figura 26 e Figura 27, mostrou que o método FUNED, apresentou um grau de sensibilidade de $0,96$ e $0,98$ e especialidade de $0,25$ e $0,19$, respectivamente. Pelo método de Canny, o grau de sensibilidade foi de $0,98$ e $0,99$ e a especialidade de $0,34$ e $0,35$.

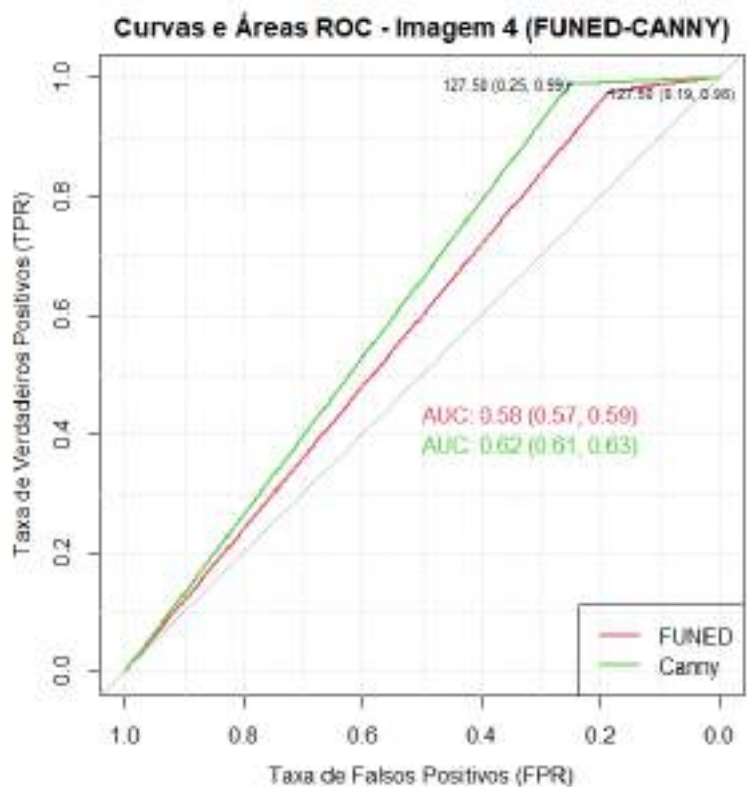
A área abaixo da curva *AUC* do método FUNED e Canny verificado nos respectivos gráficos, foram de $0,61$ e $0,58$ para o primeiro método e $0,66$ e $0,62$ para o segundo método.

Figura 26– Gráfico bidimensional ROC - Imagem 3.



Fonte: autor da dissertação.

Figura 27 – Gráfico bidimensional ROC - Imagem 4.



Fonte: autor da dissertação.

Considerando os quatro gráficos ROC, foi verificado que os valores de sensibilidade variaram de 0,96 nos gráficos da Figura 24 e Figura 33 e 0,98 nos gráficos da Figura 31 e Figura 34. Esses números mostram uma probabilidade de acerto de 96% a 98% na predição dos pixels que não são bordas. O detector de Canny, sob a mesma análise, apresentou uma taxa de acertos de não bordas entre 98% e 99%. Isso demonstra um destaque em relação à sensibilidade do preditor de Canny comparado ao de FUNED.

Como característica complementar à sensibilidade, a especialidade do FUNED variou de 19 a 25% e a de Canny, de 25 a 39%, sendo essa, a variação percentual da probabilidade que cada algoritmo tem de definir um pixel como borda e acertar na decisão. Os resultados apontam maior habilidade do detector de Canny em comparação ao FUNED.

A área abaixo da curva no detector de bordas FUNED variou de 0.58 a 0.61 e o de Canny de 0,62 e 0,69. Essa é uma métrica muito importante na análise, pois comparar curvas ROC graficamente pode ser uma tarefa difícil, principalmente quando uma curva está muito próxima da outra. A medida do desempenho consiste em resultado de 0 a 1, sendo a pior e a melhor classificação, respectivamente. Uma área próxima de 1, por exemplo, demonstra alta taxa de verdadeiros positivos e uma baixa taxa de falsos positivos.

As *AUCs* medida nos dois métodos apresentaram altas taxas de verdadeiros positivos comparadas as taxas de falsos positivos. Isso mostra eficiência dos dois detectores comparados, com destaque aos resultados de Canny que apresentou maiores áreas abaixo da curva.

7 CONCLUSÕES

Este trabalho aplicou o detector de bordas FUNED em imagens digitais de escoamento bifásico e apresentou uma solução compacta, com baixo custo computacional e intuitiva utilizando técnicas *fuzzy* não fundamentada em sistemas baseados em regras, mas na operação direta do domínio espacial da imagem atuando diretamente com cada pixel na janela de vizinhança como número fuzzy.

O método foi desenvolvido na linguagem de programação *Python* e aplicou números *fuzzy* triangulares simétricos em processamento de imagens para a obtenção de bordas nas imagens digitais do Circuito de Circulação Natural (CCN) do Centro de Engenharia Nuclear (CEN) do Instituto de Pesquisas Energéticas e Nucleares (IPEN-CNEN / SP).

O detector de bordas FUNED permitiu ajustar três parâmetros na busca de resultados otimizados para cada imagem através do tamanho da janela de vizinhança, o ajuste do espalhamento triangular *fuzzy* e o limiar. Ajustados os parâmetros e definida a imagem para a análise, o algoritmo forneceu como resultado uma imagem binária segmentada, de mesma dimensão, com bordas de pixels pretos e região de fundo com pixels brancos.

O método FUNED demonstrou a aplicabilidade da técnica FUNED em imagens de escoamento bifásico de 360 x 690 *pixels* e o tempo de execução do algoritmo foi de 0,001517 segundos. Este foi o menor valor obtido em 1000 testes de desempenho. O método de Canny, submetido ao mesmo número de testes e com as mesmas imagens de entrada, executou em um tempo de 0,024705 segundos.

Os tempos medidos mostraram que o método FUNED é 16 vezes mais rápido e possui um custo computacional conseqüentemente menor que o de Canny. O método, portanto, demonstra ser intuitivo, performático e computacionalmente viável. Além disso, o método Funed tem potencial para detectar aspectos específicos da imagem, preservando propriedades estruturais da imagem e descartando informações menos relevantes.

Os resultados analisados graficamente pela curva ROC - *Receiver Operating Characteristics* (FUNED x Canny) mostraram que o detector FUNED, mesmo sem a utilização de um algoritmo de suavização, teve uma taxa de acerto na definição do que é borda próxima ao detector de Canny, apresentou contornos bem definidos, poucos ruídos e bordas estreitas.

Embora o objetivo deste trabalho não fosse comparar métodos, foi utilizado o clássico método de Canny como referência de resultados. Com base nessa análise e observação, os dois métodos mostraram eficiência na detecção dos contornos em imagens de escoamento bifásico, apresentando alta sensibilidade e razoável especialidade, o que confere aos dois algoritmos uma excelente capacidade de não errar na predição do que não é borda.

Os conhecimentos obtidos no desenvolvimento deste trabalho e na implementação do método, com base na análise dos resultados obtidos, permite considerar que um algoritmo de suavização pode otimizar os resultados de detecção do FUNED. Portanto, como sugestão para trabalhos futuros, sugere-se o desenvolvimento de um algoritmo de suavização que trabalhe em conjunto com o método apresentado neste trabalho.

8 REFERÊNCIAS

ACHARYA, T.; RAY, A. K. **Image Processing-Principles and Applications**. John Wiley & Sons, Inc. 2005.

AMARAL, C.E.F.; ALVES, R.F., DA SILVA, M.J., ARRUDA, L.V.R., Dorini, L., Morales, R.E.M., Pipa, D.R. **Image processing techniques for high-speed videometry in horizontal two-phase slug flows**. *Flow Meas. Instrum.* p. 257-264, 2013.

ANDRADE, D.; BELCHIOR, J. A.; BASTOS, J.L.F. **Two-Phase Instabilities in a Natural Circulation Rectangular Loop**. *Proc. 8th International Conference on Nuclear Engineering, Baltimore – Maryland, 2000*.

BARBOSA, P.R.; CRIVELARO, K.C.O.; SELEGHIM JR., PAULO. **On the Application of Self-Organizing Neural Networks in Gas-Liquid and Gas-Solid Flow Regime Identification**. In.: *J. of the Braz. Soc. of Mech. Sci. & Eng.*, Vol. XXXII, March 2010.

BARKHODA, W.; TAB, F.A.; SHAHRYARI, O.K. **Fuzzy edge detection based on pixel's gradient and standard deviation values**. *Proc. Int. Multiconference Comput. Sci. Inf. Technol. IMCSIT*. 09, p. 4, 7– 10, 2009.

BARROS, L.; BASSANEZI, R. C. **Tópicos de Lógica Fuzzy e Biomatemática. Coleção**. IMECC Textos Didáticos, volume 5, 2006.

BASAK, J.; PAL, S. K. **Theoretical quantification of shape distortion in fuzzy Hough transform**. *Fuzzy Sets and Systems*. 154, p. 227-250, 2005.

BENSRHAIR, A.; MICHÉ, P.; DEBRIE, R. **Fast and automatic stereo vision matching algorithm based on dynamic programming method**. *Pattern Recognit. Letter* 5. p. 457-466, 1996.

BHARDWAJ, S.; MITTAL, A. **A Survey on Various Edge Detector Techniques**. *Procedia Technol.* p. 220–226, 2012.

BLOCH, I.; et al. **Fuzzy modelling and fuzzy mathematical morphology applied to 3d reconstruction of blood vessels by multi-modality data fusion**. *In Fuzzy Set Methods in Information Engineering*. New York: John Wiley Sons., 1996.

BOAVENTURA I. A. G. **Números Fuzzy em Processamento de Imagens Digitais e Suas Aplicações na Detecção de Bordas**. Escola de Engenharia de São Carlos da Universidade de São Paulo, 2010.

BRUNO, S.; LORENZO, S.; LUIGI, V.; GIUSEPPE, O. **Robotics: Modelling, Planning and Control**. Springer-Verlag London, 2008.

BUENO, R. C. **Detecção de contornos em imagens de padrões de escoamento bifásico com alta fração de vazio em experimentos de circulação natural com o uso de processamento inteligente**. Instituto de Pesquisas Energéticas e Nucleares, 2016.

BUENO, R.C.; et al. **Two-Phase Flow Bubble Detection Method Applied to Natural Circulation System Using Fuzzy Image Processing**. *Nuclear Engineering and Design*, vol. 335, 2018, pp. 255–264.

BUSTINCE, H.; et al. **Interval-valued fuzzy sets constructed from matrices: Application to edge detection**. *Fuzzy Sets and System*, 2009.

CABANI, I.; TOULMINET, G.; BENSRAHAIR, A. **A Color Stereo Vision System for Extraction of 3D Edges of Obstacle**. *Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference (ITSC)* 17 p. 307-312, 2006.

CANNY J. **A Computational Approach to Edge Detection**. *Pattern Analysis and Machine Intelligence*. IEEE, 1986.

CHAIRA, T.; RAY, A. K. **A new measure using intuitionistic fuzzy set theory and its application to edge detection**. *Applied Soft Computing*, 2008.

DAVE, N.; BHASWAN, K. **Adaptive Fuzzy c-Shells Clustering and Detection of Ellipses**. *IEEE transactions on neural networks*, vol. 3, 5, Setembro, 1992.

EI ANSARI, M.; MOUSSET, S.; BENSRAHAIR, A. **Temporal consistent real-time stereo for intelligent vehicles**. *Pattern Recognit. Lett.* p. 31, 1226-1238, 2010.

FATEMEH, A.; ESFANDIAR, E.; POOYA, E. **Interval-valued intuitionistic fuzzy generators: Application to edge detection, Faculty of Mathematics and Computer Science**. *Shahid Bahonar University of Kerman, Journal of Intelligent & Fuzzy Systems*, 2014.

FU, Y.; LIU, Y. **Development of a robust image processing technique for bubbly flow measurement in a narrow rectangular channel**. *International Journal of Multiphase Flow*, v. 84, p. 217–228, 2016.

GONZALEZ, R. C. **Processamento de Imagens**. 3 ed. São Paulo: Pearson Prentice Hall, 2010.

GOSHTASBY A. A. **2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications**. Wiley-Interscience, 2005.

GOUDARZI, N.; TALEBI, S. **Improving performance of two-phase natural circulation loops by reducing of entropy generation**. *Energy* 93 p. 882-899, 2015.

GUO, F.; YANG, Y.; CHEN, B.; GUO, L. **A novel multi-scale edge detection technique based on wavelet analysis with application in multiphase flows.** *Powder Technology*, 2010.

HAN, J. H.; KÓCZY, L. T.; POSTON, T. **Fuzzy Hough transform.** *Pattern Recognition Letters*. p. 649 - 658, 1994.

HEATH, M.; SARKAR, S.; SANOCKI, T.; BOWYER, K. **Comparison of Edge Detectors**, 1998.

IZHAR, H.; SHAHZAD, A.; KAMRAN, S.; MUHAMMAD, T. K.; SHAUKAT, A. S. **Fuzzy Logic Based Edge Detection in Smooth and Noisy Clinical Images.** *Institute of Mechatronics Engineering, University of Engineering and Technology, Peshawar, Pakistan, Department of Mechanical Engineering, University of Engineering and Technology*, 2015.

JACQUEY, F.; COMBY, F.; STRAUSS, O. **Fuzzy edge detection for omnidirectional images.** *Fuzzy Sets and System*, 2008.

JÄHNE, B. **Digital Image Processing.** Springer-Verlag, 2002.

JAWAHAR, C.; RAY, A. **Fuzzy statistics of digital images.** *IEEE Signal Processing Letters*, v. 3, n. 8, p. 225–227, 1996.

KIRSCH R. A. **Computer determination of the constituent structure of biological images.** *Computers and Biomedical Research*. 1971.

LEI, Y.; DEWEI, Z.; XIAOYU, W.; HUI, L.; JUN, Z. **An improved Prewitt algorithm for edge detection based on noised image.** *Image and Signal Processing (CISP)*, 2011.

LIANG, L.; LOONEY, C. **Competitive fuzzy edge detection.** *Applied Soft Computing*, 2003.

LIN, H.; DU P. J.; ZHAO C. S.; SHU N. **Edge detection method of remote sensing images based on mathematical morphology of multi-structure elements.** *Chinese Geographical Science*, 2004.

LIN, J. S.; CHENG, K. S.; MAO, C. W. **A fuzzy Hopfield neural network for medical image segmentation.** *IEEE Trans. Nucl. Sci.* 43, p. 2389–2398, 1996.

MASOTTI, P.H.F.; DE MESQUITA, R.N. **Paraconsistent logic study of image focus in cylindrical refraction experiments.** *Intelligente System Reference Library*, v. 94, p. 175-205, 2015.

MESQUITA, R.N.; et al. **Two-Phase Flow Patterns Recognition and Parameter Estimation through Natural Circulation Test Loop Image Analysis.** *Proc. 7th ECI international Conference on Boiling Heat Transfer (ICBHT 2009)*, Florianopolis - SC- Brazil, 3-7 May, p. 541-546, ISBN: 978-1-61567-794-8, 2009.

MESQUITA, R.N.; MASOTTI, P.H.F.; PENHA, R.M.L.; ANDRADE, D. a., SABUNDJIAN, G.; TORRES, W.M., MACEDO, L. a.: **Classification of natural circulation two-phase flow patterns using fuzzy inference on image analysis.** *Nucl. Eng. Des.* 250, 592–599, 2012.

MESQUITA, R.N.; CASTRO L.F.C.; TORRES, W.M., ROCHA, M.D.S., UMBEHAUN, P.E., ANDRADE, D.A., SABUNDJIAN, G., MASOTTI, P.H.F. **Classification of Natural Circulation Two-phase Flow Image Patterns Based on Self-organizing Maps of Full Frame DCT Coefficients.** *Nuclear Engineering and Design* 335 (2018): 161-71. Web.

METZ, C. E. **Statistical Analysis of ROC Data in Evaluating Diagnostic Performance. Multiple Regression Analysis: Applications in the Health Sciences,** number 13, edited by Donald E. Herbert and Raymond H. Myers. 365–384, 1986.

MICHÉ, P.; DEBRIE, R. **Fast and self-adaptive image segmentation using extended declivity.** *Ann. Des Télécommunications* 50, 401–410, 1995.

MIOSSO, C.; BAUCHPIESS, A. **Fuzzy inference system applied to edge detection in digital images.** In: *Proceedings of the V Brazilian Conference on Neural Networks*, 2001.

PARKER, J., et al. **Classification of Ductal Carcinoma in Situ by Image Analysis of Classifications from Digital Mammograms.** *The British Journal of Radiology*, 68:150–159, 1995.

PATEL, J.; PATWARDHAN, J.; SANKHE, K.; KUMBHARE, R. **Fuzzy Inference Based Edge Detection System using Sobel and Laplacian of Gaussian Operators.** *Int. Conf. and Workshop on Emerg. Trends in Tech (ICWET)*, 2011.

RULANINGTYAS R, AIN K. **Edge detection for brain tumor pattern recognition.** *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 2009.

RUSSO, F. **Edge detection in noyse images using fussy reasoning.** *IEEE Transaction on Intrumentation and Measurement*, 1998.

SARKAR, P.S., KASHYAP, Y., SINHA, A., VIJYAN, P.K., RAO, G.S.S.P. **Applications for real-time neutron radiography for convection driven flow pattern transition studies.** *IEEE Trans. Nucl. Sci.* 52, 290–294, 2005.

SCHLEGEL, J.P.; SAWANT, P.; PARANJAPE, S.; OZAR B.; HIBIKI, T.; ISHII, M. **Void fraction and flow regime in adiabatic upward two-phase flow in large diameter vertical pipes.** *Nuclear Engineering and Design.* 239, p. 2864-2874, 2009.

SERRA, P. L. S. **Cálculo da fração de vazio em escoamentos bifásicos (gás/líquido) a partir da identificação de bolhas em imagens digitais.** Tese

(Doutorado) - Curso de Doutorado em Ciências na Área de Tecnologia Nuclear, Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2017.

SHEN, J.; CASTAN, S. **An optimal linear operator for step edge detection. Computer Vision, Graphics, and Image Processing: Graphical Models and Understanding.** 1992.

SHI, L. **Fuzzy Recognition for Gas-liquid Two-phase Flow Pattern Based on Image Processing, IEEE International Conference on Control and Automation.** IEEE, pp. 1424–1427, 2007.

SURAL, S.; DAS, P. K. **Fuzzy Hough transform and MLP with fuzzy input/output for character recognition. Fuzzy Sets and Systems.** 105, p. 489 - 497, 1999.

TORRES, W. M.; TORRES, E. **Análise Experimental do Fenômeno de Circulação Natural. 11th Brazilian Congress of Thermal Sciences and Engineering: Proceedings of ENCIT,** 2006.

TAYLOR, P.; ALI, S.F.; YEUNG, H.: **Experimental Investigation and Numerical Simulation of Two-Phase Flow in a Large-Diameter Horizontal Flow Line Petroleum Science and Technology.** 37–41, 2010.

TORISAKI, S.; SHUICHIRO, M. **Gas-Liquid Two-Phase Flow Image Analysis Using Bubble Detection Algorithm.” Japanese Journal of Multiphase Flow,** vol. 34, no. 1, 2020, pp. 125–133.

ULUPINAR, F.; MEDIONI, G. **Refining edges detected by a LoG operator. Computer Vision, Graphics, and Image Processing.** 1990.

WANG, H.; DONG, F. **Image features extraction of gas/liquid two-phase flow in horizontal pipeline by GLCM and GLGCM. 2009 9th Int. Conf. Electron. Meas. Instruments.** 2-135-2–139, 2009.

WENYIN, Z.W.Z.; NINGDE, J.N.J.; XIA, L.X.L.; ZHIQIANG, N.Z.N.; 2008. **Bubble Image Segmentation of Gas/Liquid Two-Phase Flow Based on Improved Canny Operator. Int. Conf. Comput. Sci. Softw. Eng.** 1, 799–801, 2008.

WU, M.; DONG, F.; QI, G. **Feature Extraction Method for Gas / Liquid Two-Phase Flow. Mach. Learn.** 13–16, 2006.

XU, Y.; YU, F. **Research on the image acquisition and camera control of machine vision camera based on LabVIEW. Proceedings. 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2013,** v. 2, p. 499–502, 2013.

ZHANG, J.Y.; CHEN, Y.; HUANG, X.X. **Edge detection of images based on improved Sobel operator and genetic algorithms. Image Analysis and Signal Processing,** 2009.

ZHANG, J.; SHEN, X.; FUJIHARA, Y.; SANO, T.; YAMAMOTO, T.; NAKAGIMA, K. ***Experimental study on the safety of Kyoto University Research Reactor at natural circulation cooling mode.*** *Annals of Nuclear Energy*, 76, p. 410-420, 2015.

APÊNDICE A – Algoritmo 1: Imagem pertinência

FUNED - Algoritmo 1 - Imagem pertinência

```

#FUNED - Algoritmo 1: Imagem pertinência
import numpy as np #biblioteca numpy.
import skfuzzy as fuzz #biblioteca skfuzzy.
import matplotlib.pyplot as plt #biblioteca matplotlib.
import cv2 #biblioteca opencv
def funed(imagem, matriz, espalhamento): #função para entrada das variáveis

    z = espalhamento/2 #espalhamento do número fuzzy.
    img = cv2.imread(imagem, 0) #leitura da imagem de entrada.
    n = 1 #linha inicia em 1
    m = 1 #coluna inicia em 1
    mat = [] #cria matriz.

    for i in range(n): # para i indo até n (loop)
        for y in range(0, img.shape[0], 1): #percorre linhas.
            for x in range(0, img.shape[1], 1): #percorre colunas.
                x_pixel = np.arange(0, 255, 1) #níveis de cinza..

                if matriz == 5: #se matriz de varredura for de 5x5
                    w = img[y-2, x-2] #pixel central da matriz 5x5.
                    pixelc = fuzz.trimf(x_pixel, [w-z, w, w+z]) #função
                    ↳triangular simétrica.

                    part_px1 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
↳x]) #pertencimento do pixel 1.
                    part_px2 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
↳x-1]) #pertencimento do pixel 2.
                    part_px3 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
↳x-2]) #pertencimento do pixel 3.
                    part_px4 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
↳x-3]) #pertencimento do pixel 4.
                    part_px5 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
↳x-4]) #pertencimento do pixel 5.
                    part_px6 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
↳x]) #pertencimento do pixel 6.
                    part_px7 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
↳x-1]) #pertencimento do pixel 7.
                    part_px8 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
↳x-2]) #pertencimento do pixel 8.
                    part_px9 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
↳x-3]) #pertencimento do pixel 9.
                    part_px10 = fuzz.interp_membership(x_pixel, pixelc,
↳img[y-3, x-4]) #pertencimento do pixel 10.

```



```

        pert_px11 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x]) #pertencimento do pixel 11.
        pert_px12 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-1]) #pertencimento do pixel 12.

        pert_px14 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-3]) #pertencimento do pixel 14.
        pert_px15 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-3]) #pertencimento do pixel 15.
        pert_px16 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x]) #pertencimento do pixel 16.
        pert_px17 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-1]) #pertencimento do pixel 17.
        pert_px18 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-2]) #pertencimento do pixel 18.
        pert_px19 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-3]) #pertencimento do pixel 19.
        pert_px20 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-4]) #pertencimento do pixel 20.
        pert_px21 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x]) #pertencimento do pixel 21.
        pert_px22 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-1]) #pertencimento do pixel 22.
        pert_px23 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-2]) #pertencimento do pixel 23.
        pert_px24 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-3]) #pertencimento do pixel 24.
        pert_px25 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-4]) #pertencimento do pixel 25.

        pert_px13 = ((pert_px1 +pert_px2 +pert_px3 +pert_px4_
->+pert_px5 +pert_px6 +
                    pert_px7 +pert_px8 +pert_px9_
->+pert_px10+pert_px11+pert_px12+
                    _
->+pert_px14+pert_px15+pert_px16+pert_px17+pert_px18+pert_px19+
                    _
->+pert_px20+pert_px21+pert_px22+pert_px23+pert_px24+pert_px25)/24)*255 #media_
->= pertencimento do pixel 13.

        mat.append ([pert_px13]=m) #Adiciona o "pert_px13" na_
->matriz "mat".

    else:
        w = img[y-1, x-1] #pixel central da matriz 3x3.
        pixelc = fuzz.trimf(x_pixel, [w-z, w, w+z]) #função_
->triangular simétrica.
        pert_px1 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,
->x-2]) #pertencimento do pixel 1.
        pert_px2 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,
->x-1]) #pertencimento do pixel 2.

```

```

        pert_px3 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,0
->x]) #pertencimento do pixel 3.
        pert_px4 = fuzz.interp_membership(x_pixel, pixelc, img[y-1,0
->x-2]) #pertencimento do pixel 4.
        pert_px6 = fuzz.interp_membership(x_pixel, pixelc, img[y-1,0
->x]) #pertencimento do pixel 6.
        pert_px9 = fuzz.interp_membership(x_pixel, pixelc, img[y,0
->x]) #pertencimento do pixel 9.
        pert_px8 = fuzz.interp_membership(x_pixel, pixelc, img[y,0
->x-1]) #pertencimento do pixel 8.
        pert_px7 = fuzz.interp_membership(x_pixel, pixelc, img[y,0
->x-2]) #pertencimento do pixel 7.

        pert_px5 = ((pert_px1+pert_px2+
                    pert_px3+pert_px4+
                    pert_px6+pert_px7+
                    pert_px8+pert_px9)/6)+255 #media =
->pertencimento do pixel 5.

        mat.append ([pert_px5]+m) #Adiciona o "pert_px5" na matriz
->"mat".

    alt, lar = np.shape(img) #encontra as dimensões da "img".
    print ([alt]) #escreve a quantidade de linhas da "img".
    print ([lar]) #escreve a quantidade de colunas da "img".
    mat1 = np.reshape (mat, (alt, lar)) #estrutura a matriz "mat" com o mesmo
->numero de linhas e colunas que "img"

    inm = plt.imshow(mat1) #plota a matriz "mat1"

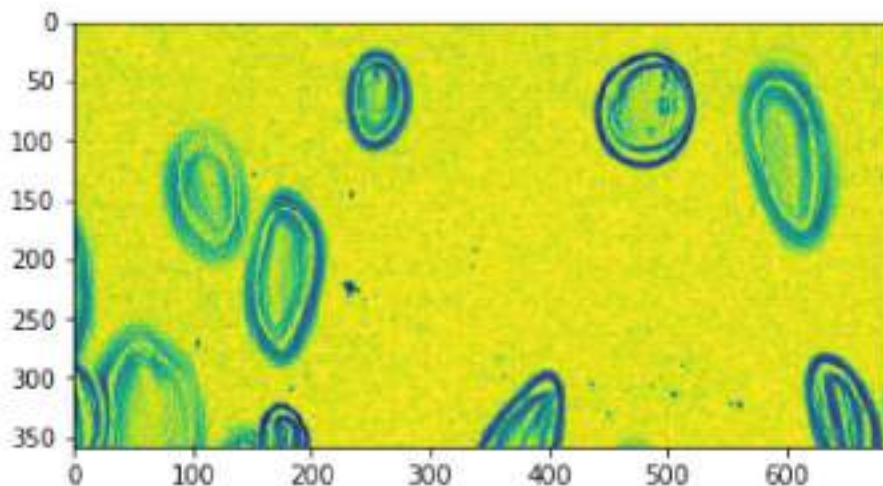
    cv2.imwrite('imagem_pertinencia.png', mat1) #salva a imagem pertinencia.

funed('imagem_cinza.png', 3, 30) #recebe as variáveis de entrada (imagem e
->matriz de varredura)

```

[359]

[690]



APÊNDICE B – Algoritmo 2: Imagem sem supressão de não máximos

FUNED - Algoritmo 2 - Imagem sem supressão de não máximos

```

#FUNED - Algoritmo 2: Imagem sem supressão de não máximos
import numpy as np #biblioteca numpy.
import skfuzzy as fuzz #biblioteca skfuzzy.
import matplotlib.pyplot as plt #biblioteca matplotlib.
import cv2 #biblioteca opencv
def funed(imagem, matriz, espalhamento, limiar): #função para entrada das
    →variáveis

    z = espalhamento/2 #espalhamento do número fuzzy.
    img = cv2.imread(imagem, 0) #leitura da imagem de entrada.
    n = 1 #linha inicia em 1
    m = 1 #coluna inicia em 1
    mat = [] #cria matriz.

    for i in range(n): # para i indo até n (loop)
        for y in range(0, img.shape[0], 1): #percorre linhas.
            for x in range(0, img.shape[1], 1): #percorre colunas.
                x_pixel = np.arange(0, 255, 1) #níveis de cinza.

                if matriz == 5: #se matriz de varredura for de 5x5
                    w = img[y-2, x-2] #pixel central da matriz 5x5.
                    pixelc = fuzz.trimf(x_pixel, [w-z, w, w+z]) #função
    →triangular simétrica.

                    pert_px1 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
    →x]) #pertencimento do pixel 1.
                    pert_px2 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
    →x-1]) #pertencimento do pixel 2.
                    pert_px3 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
    →x-2]) #pertencimento do pixel 3.
                    pert_px4 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
    →x-3]) #pertencimento do pixel 4.
                    pert_px5 = fuzz.interp_membership(x_pixel, pixelc, img[y-4,
    →x-4]) #pertencimento do pixel 5.
                    pert_px6 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
    →x]) #pertencimento do pixel 6.
                    pert_px7 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
    →x-1]) #pertencimento do pixel 7.
                    pert_px8 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
    →x-2]) #pertencimento do pixel 8.
                    pert_px9 = fuzz.interp_membership(x_pixel, pixelc, img[y-3,
    →x-3]) #pertencimento do pixel 9.

```

```

        pert_px10 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-3, x-4]) #pertencimento do pixel 10.
        pert_px11 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x]) #pertencimento do pixel 11.
        pert_px12 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-1]) #pertencimento do pixel 12.

        pert_px14 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-3]) #pertencimento do pixel 14.
        pert_px15 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-2, x-3]) #pertencimento do pixel 15.
        pert_px16 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x]) #pertencimento do pixel 16.
        pert_px17 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-1]) #pertencimento do pixel 17.
        pert_px18 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-2]) #pertencimento do pixel 18.
        pert_px19 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-3]) #pertencimento do pixel 19.
        pert_px20 = fuzz.interp_membership(x_pixel, pixelc,
->img[y-1, x-4]) #pertencimento do pixel 20.
        pert_px21 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x]) #pertencimento do pixel 21.
        pert_px22 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-1]) #pertencimento do pixel 22.
        pert_px23 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-2]) #pertencimento do pixel 23.
        pert_px24 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-3]) #pertencimento do pixel 24.
        pert_px25 = fuzz.interp_membership(x_pixel, pixelc, img[y,
->x-4]) #pertencimento do pixel 25.

        pert_px13 = ((pert_px1 +pert_px2 +pert_px3 +pert_px4,
->+pert_px5 +pert_px6 +
                    pert_px7 +pert_px8 +pert_px9,
->+pert_px10+pert_px11+pert_px12+
                    ,
->+pert_px14+pert_px15+pert_px16+pert_px17+pert_px18+pert_px19+
                    ,
->+pert_px20+pert_px21+pert_px22+pert_px23+pert_px24+pert_px25)/24) #media =,
->pertencimento do pixel 13.

        if pert_px13 > limiar: #se o pert_px13 for maior que o,
->limiar.

            pert_px13 = 255 #escreva 255 no pert_px13.

        else: #sendo
            if x<-3 or y<-3: #se as duas primeiras colunas ou as,
->três primeiras linhas.
                pert_px13 = 255 #escreva 255 no pert_px13.
            else: #sendo

```

```

        else: #senão
            pert_px13 = 0 #escreva 0 no pert_px13.

            mat.append ([pert_px13]*m) #Adiciona o "pert_px13" na
→matriz "mat".

            else:
                w = img[y-1, x-1] #pixel central da matriz 3x3.
                pixelc = fuzz.trimf(x_pixel, [w-z, w, w+z]) #função
→triangular simétrica.
                pert_px1 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,
→x-2]) #pertencimento do pixel 1.
                pert_px2 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,
→x-1]) #pertencimento do pixel 2.
                pert_px3 = fuzz.interp_membership(x_pixel, pixelc, img[y-2,
→x]) #pertencimento do pixel 3.
                pert_px4 = fuzz.interp_membership(x_pixel, pixelc, img[y-1,
→x-2]) #pertencimento do pixel 4.

                pert_px6 = fuzz.interp_membership(x_pixel, pixelc, img[y-1,
→x]) #pertencimento do pixel 6.
                pert_px9 = fuzz.interp_membership(x_pixel, pixelc, img[y,
→x]) #pertencimento do pixel 9.
                pert_px8 = fuzz.interp_membership(x_pixel, pixelc, img[y,
→x-1]) #pertencimento do pixel 8.
                pert_px7 = fuzz.interp_membership(x_pixel, pixelc, img[y,
→x-2]) #pertencimento do pixel 7.

                pert_px5 = ((pert_px1+pert_px2+
                    pert_px3+pert_px4+
                    pert_px6+pert_px7+
                    pert_px8+pert_px9)/8) #media = pertencimento
→do pixel 5.

                if pert_px5 > limiar: #se o pert_px5 for maior que
→limiar.
                    pert_px5 = 255 #escreva 255 no pert_px5.

                else: #senão
                    if x<-3 or y<-3: #se as duas primeiras colunas ou as
→duas primeiras linhas.
                        pert_px5 = 255 #escreva 255 no pert_px5.
                    else: #senão
                        pert_px5 = 0 #escreva 0 no pert_px5.

                mat.append ([pert_px5]*m) #Adiciona o "pert_px5" na matriz
→"mat".

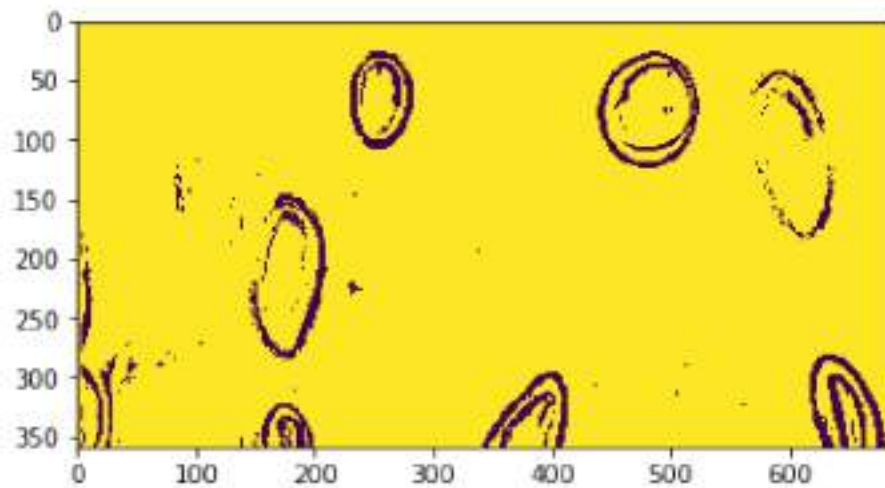
            alt, lar = np.shape(img) #encontra as dimensões da "img".
            print ([alt]) #escreve a quantidade de linhas da "img".
            print ([lar]) #escreve a quantidade de colunas da "img".
            mat1 = np.reshape (mat, (alt, lar)) #estrutura a matriz "mat" com o mesmo
→numero de linhas e colunas que "img"

```

```
imm = plt.imshow(mat1) #plots a matriz "mat1"  
cv2.imwrite('imagem_sem_supressao.png', mat1) #salva a imagem pertinencia,  
fused('imagem_cinza.png', 3, 30, 0.5) #recebe as variáveis de entrada (imagem, u  
-matriz de varredura, espalhamento e limiar)
```

[359]

[690]



APÊNDICE C – Algoritmo 3: Imagem final (FUNED)

FUNED - Algoritmo 3 - Imagem com supressão de não máximos (Final)

```

#FUNED - Algoritmo 3: Imagem com supressão de não máximos (30/10/2021)
import numpy as np #biblioteca numpy
import skfuzzy as fuzz #biblioteca skfuzzy
import matplotlib.pyplot as plt #biblioteca matplotlib
import cv2 #biblioteca opencv
def funed(imagem): #função para entrada da variável

    img1 = cv2.imread(imagem, 0)
    img = np.float32(img1)
    n = 1 #linha inicia em 1
    m = 1 #coluna inicia em 1
    mat = [] #cria uma matriz chamada mat

    for i in range(n): # para i indo até n (loop infinito)
        for y in range(0, img.shape[0], 1): #percorre linhas
            for x in range(0, img.shape[1], 1): #percorre colunas

                px3 = img[y-2, x] #posição do pixel 3
                px2 = img[y-2, x-1] #posição do pixel 2
                px1 = img[y-2, x-2] #posição do pixel 1

                px6 = img[y-1, x] #posição do pixel 6
                px5 = img[y-1, x-1] #posição do pixel 5
                px4 = img[y-1, x-2] #posição do pixel 4

                px9 = img[y, x] #posição do pixel 9
                px8 = img[y, x-1] #posição do pixel 8
                px7 = img[y, x-2] #posição do pixel 7

                d1 = (px4+px6) #direção da borda 0°
                d2 = (px3+px7) #direção da borda 45°
                d3 = (px2+px8) #direção da borda 90°
                d4 = (px1+px9) #direção da borda 135°

                direcao = [d1, d2, d3, d4] #matriz direcao

                d = min(direcao) #valor mínimo da matriz é inserido em "d"

                if x<=3 or y<=3: #se as duas primeiras colunas ou as três
└primeiras linhas.
                    px=255

```

```

        elif d == d1: #se d1 for o valor mínimo
            if px5<px2 or px5<px8: #verifico a condição dos pixels
--vizinhos
                px=0 #escrevo "0" em px
            else: #senão
                px=255 #escrevo "255" em px

        elif d == d2: #se d2 for o valor mínimo
            if px5<px1 or px5<px9: #verifico a condição dos pixels
--vizinhos
                px=0 #escrevo "0" em px
            else: #senão
                px=255 #escrevo "255" em px

        elif d == d3: #se d3 for o valor mínimo
            if px5<px4 or px5<px6: #verifico a condição dos pixels
--vizinhos
                px=0 #escrevo "0" em px
            else: #senão
                px=255 #escrevo "255" em px

        elif d == d4: #se d4 for o valor mínimo
            if px5<px3 or px5<px7: #verifico a condição dos pixels
--vizinhos
                px=0 #escrevo "0" em px
            else: #senão
                px=255 #escrevo "255" em px

        else: #senão
            px=255 #escrevo "255" em px

        mat.append ([px]*n) #adiciona px na matriz

    alt, lar = np.shape(img) #verifico a altura e a largura da imagem e guardo
--nas variáveis criadas
    print ([alt]) #mostro a altura
    print ([lar]) #mostra a largura
    nat1 = np.reshape (mat, (alt, lar)) #coloco a matriz "mat" no mesmo formato
--da imagem "img" e crio "nat1"

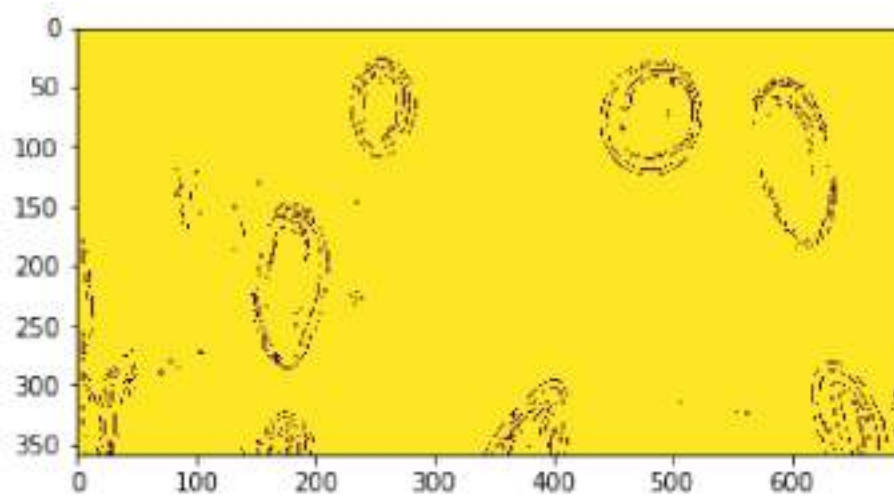
    inm = plt.imshow(nat1) #mostro a imagem gerada

    cv2.imwrite('imagen_funed.png', nat1) #salvo a imagem
funed('imagen_sem_supressao.png') #recebe a variável de entrada (imagem)

```

[359]

[690]



APÊNDICE D – Algoritmo 4: Imagem em Dataset

FUNED - Algoritmo 4 - Dataset da imagem

```
#FUNED - Algoritmo 4: Dataset da imagem em .csv
import numpy as np #biblioteca numpy.
import cv2 #biblioteca opencv

def funed(imagem): #função para entrada da variável

    img = cv2.imread(imagem, 0) #le uma imagem grayscale
    n = 1 #linha inicia em 1
    m = 1 #coluna inicia em 1
    mat = [] #cria uma matriz chamada mat

    for i in range(n): # para i indo até n (loop infinito)
        for y in range(0, img.shape[0], 1): #percorre linhas (start, stop e
        ..step)-(início, último elemento, intervalo)
            for x in range(0, img.shape[1], 1): #percorre colunas /////

                mat.append (img[y, x]*m) #Adiciona o pixel na matriz "mat".

    alt, lar = np.shape(img) #verifico a altura e a largura da imagem e guardo
    ..nas variáveis criadas
    mat1 = np.reshape (mat, ((alt*lar), 1)) #crio uma matriz coluna com os
    ..pixels da matriz "mat"

    np.savetxt('imagem_dataset_gt.csv', mat1, delimiter='') #salvo a matriz
    ..coluna como dataset em um arquivo .csv.
    funed('imagem_gt.png') #recebe a variável de entrada (imagem)
```

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Diretoria de Pesquisa, Desenvolvimento e Ensino
Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000
Fone/Fax(0XX11) 3133-8908
SÃO PAULO – São Paulo – Brasil
<http://www.ipen.br>

O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento, associada à Universidade de São Paulo e gerida técnica e administrativamente pela Comissão Nacional de Energia Nuclear, órgão do Ministério da Ciência, Tecnologia, Inovações e Comunicações.
