

CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

Detecção e Caracterização Automatizada de Fraturas e Desmoronamentos em Imagens de Petróleo utilizando Inteligência Artificial

Luciana Olivia Dias

Orientador
Dr. Márcio Portes de Albuquerque
Coorientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2018

Luciana Olivia Dias

**Detecção e Caracterização Automatizada de Fraturas e
Desmoronamentos em Imagens de Petróleo utilizando
Inteligência Artificial**

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física com Ênfase em Instrumentação Científica.

CBPF – Centro Brasileiro de Pesquisas Físicas

Orientador: Dr. Márcio Portes de Albuquerque
Coorientador: Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2018

Dias, Luciana Olivia
Detecção e Caracterização Automatizada de Fraturas e Desmoronamentos em
Imagens de Petróleo utilizando Inteligência Artificial/ Luciana Olivia Dias. - 2018
155 f. : il.

Dissertação de Mestrado – CBPF – Centro Brasileiro de Pesquisas Físicas , Rio de
Janeiro, RJ, 2018.
Orientador: Dr. Márcio Portes de Albuquerque

1. Imagens Acústicas. 2. Fraturas Naturais. 3. Detecção. 4. Reconheci-
mento de padrões. 5. Inteligência Artificial. I. Título
CDU 02:141:005.7

*Dedico este trabalho
à minha filha.*

AGRADECIMENTOS

Primeiramente agradeço à minha segunda família CBPF, em especial ao meu co-orientador Dr. Clécio Roque De Bom, por todo seu tempo dedicado, inclusive finais de semana e por ser um exemplo inspirador pessoal e profissional. Ao meu orientador Dr. Márcio Portes de Albuquerque e seu irmão Dr. Marcelo Portes de Albuquerque por todo apoio e incentivo. As geólogas da *Petrobrás*[®] MSc. Anelise Lima e MSc. Cândida Menezes de Jesus por toda atenção com nosso grupo e por contribuir com dados iniciais. Aos meus amigos de projeto Felipe Pires, MSc. André Persechino e MSc. Elizângela Farias que prestaram suportes nas dificuldades do dia-a-dia. Ao MSc. Manuel Blanco Valentin, um amigo que admiro muito e que contribuiu com a parte de simulação e a Juliana Coelho que além de se tornar uma grande amiga, contribuiu para caracterização na imagens reais. Ao meu pai Altair José Dias que não está fisicamente presente, mas está no meu coração e sempre será o meu referencial de persistência. A minha querida mãe Cecilia Olivia Dias que me apoia com seu amor e posso contar com suas orações. Ao meu marido Alessandro de Souza Goulart que tem um papel importante no meu trabalho, dele veio o incentivo para buscar no CBPF minha iniciação científica. Em especial à Deus, em quem confio fielmente e à Yasmim Dias de Souza Goulart minha amada filha.

RESUMO

Durante o processo de exploração de um campo com potencial para produção de hidrocarbonetos, operadoras realizam uma extensa e exaustiva análise do reservatório. O objetivo dessas empresas é determinar a viabilidade econômica de extração. Durante essa fase, são realizadas análises sísmicas, perfurações, amostragem e perfilagens de poços que produzem um enorme fluxo de dados que precisam ser analisados por geólogos e geofísicos. Entre esse volume de dados são utilizadas imagens acústicas da parede do poço para identificar fraturas naturais e desmoronamentos. Fraturas naturais são o resultado de processos geológicos. Desmoronamentos ou breakouts são processos induzidos durante a perfuração decorrente da resultante de tensões da rocha. A detecção e a classificação dessas ocorrências, associados as demais informações obtidas através de outras ferramentas de sondagem durante os processos de avaliação, podem indicar a direção do fluxo de fluidos de interesse através das fraturas, assim como a determinação da variação de tensões entre camadas rochosas identificando desmoronamentos. Analisar toda essa informação demanda muito tempo do profissional e em muitos casos pode resultar em opiniões divergentes entre analistas. Neste trabalho propomos automatizar o processo de detecção e classificação de fraturas geológicas naturais e desmoronamentos utilizando a Inteligência Artificial (IA). Para essa tarefa, utilizamos a rede neural artificial convolucional profunda rápida (FRCNN) que determina as regiões de interesse da imagem, da sigla em inglês de *Fast Region-based Convolutional Neural Network*. Essa abordagem foi avaliada e comparada a outros métodos de detecção geralmente utilizados em imagens baseados no reconhecimento de padrões em imagem, como a rede neural artificial (RNA), Método de Classificação por Polinômio (MCP), Método Transformada de Radon (MTR), Método Transformada de Hough (MTH) e Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP). A avaliação da qualidade de cada método de classificação de fraturas e desmoronamentos, foi obtida através da área abaixo da Curva Característica de Operação (ROC), da sigla em inglês de *Receiver Operating Curve*, porém o MTR e o MTH que não alcançaram índices superiores à 50% foram apresentados em tabela. Com o uso da FRCNN a ROC apresentou uma área abaixo da curva (AUC) de 0,97 para classificação de desmoronamentos e 0,86 de fraturas.

Palavras Chaves: Perfilagem, Poço, Imagens Acústicas, Detecção, Reconhecimento, Fraturas Naturais, Desmoronamentos, Inteligência Artificial, *Fast R-CNN*.

ABSTRACT

The oil and gas companies carry out an extensive analysis of the reservoir during the process of exploring a field with potential for hydrocarbon production. The goal is to determine the feasibility of a given field. Drilling oil and gas wells for profile are performed during this phase and produce a huge flow of data that need to be analyzed by geologists and geophysicists. In this volume, acoustic images of the soil wall are used for the characterization of natural fractures and breakouts. The natural fractures are the result of geological processes. Breakouts are processes induced from the resultant of rock stress. The investigation of occurrences associated with a long process of evaluation of the probing tools during the evaluation processes can indicate a direction of the flow of fluids of interest through the fractures, as well as a selection of tendencies between the areas of formation. Analyzing all this information is very difficult and many cases can result in divergent opinions among analysts. We propose to automatize the process of detection and classification of natural geological fractures and breakouts using Artificial Intelligence (AI). We use a Fast Convolutional Artificial Neural Network based in Regions (FRCNN) of interest of the image for this purpose. This approach was evaluated and compared to other methods of detection in based on the Artificial Neural Network (ANN), Polynomial Classification Method (MCP), Radon Transform Method (MTR), Hough Transform Method (MTH) and Pattern Recognition and Detection Method (DRP). The evaluation of the promising methods was obtained through the Operating Characteristic Curve (ROC). Through the FRCNN a ROC, presented an area below the curve of 0,97 for breakouts and 0,86 fractures.

Key-Words: Well profile, Acoustic Images, Detection, Recognition, Natural Fractures, Breakouts, Artificial Intelligence and Fast R-CNN.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Esquema resumido da estrutura de perfuração do poço para avaliação de reservatório.	6
Figura 2.2 – Esquema ferramenta acústica localizada no poço.	7
Figura 2.3 – Trecho de perfil de imagem resistiva.	8
Figura 2.4 – Esquema conceitual de valores de amplitudes medidos convertidos em matriz em escala de cor.	9
Figura 2.5 – Trechos de perfil de imagem de tempo de trânsito e de amplitude em tons de cinza, referentes à mesma profundidade.	9
Figura 2.6 – Esquema da ferramenta acústica.	10
Figura 2.7 – Esquema representativo do recurso <i>dipmeter</i>	12
Figura 2.8 – Esquema com um exemplo de tensões responsáveis por fratura naturais.	13
Figura 2.9 – Esquema de tensões responsáveis por desmoronamentos.	15
Figura 2.10 – Trecho de imagem real com desmoronamento.	15
Figura 3.1 – Esquema comparativo entre o neurônio biológico e neurônio artificial.	20
Figura 3.2 – Esquema simplificado de classificador binário com função linear.	21
Figura 3.3 – Curva característica da função de ativação sigmóide e sua derivada.	23
Figura 3.4 – Curva característica da função tangente hiperbólica e sua derivada.	24
Figura 3.5 – Curva de função relu e sua derivada.	25
Figura 3.6 – Esquema comparativo dos principais tipos paradigmas de aprendizagem em RNA.	28
Figura 3.7 – Esquema geral de retropropagação de uma Rede Neural Artificial.	29
Figura 3.8 – Esquema função custo no processo de propagação direta e retropropagação da rede.	31
Figura 3.9 – Gráfico característico da função custo em função apenas dos pesos.	31
Figura 3.10 – Esquema exemplo comparativo de taxas de aprendizado.	32
Figura 3.11 – Esquema comparativo dos principais tipos de configuração de interconexão de neurônios.	33
Figura 3.12 – Esquema rede neural artificial rasa com uma camada oculta.	36
Figura 3.13 – Esquema comparativo entre redes neurais rasa e redes neurais profundas.	37

Figura 3.14–Esquema exemplo de arquitetura de rede neural convolucional.	40
Figura 3.15–Esquema exemplo de extração de características utilizando convolução.	41
Figura 3.16–Esquema de preenchimento de zeros e detalhes de operação de convolução usando filtro deslizante.	42
Figura 3.17–Convolução entre as matrizes referentes à imagem e os filtros.	43
Figura 3.18–Esquema subamostragem.	44
Figura 3.19–Esquema propagação direta e retropropagação.	46
Figura 3.20–Esquema de aceleração de trabalhos usando GPU.	47
Figura 3.21–Esquema gráficos de custo em função dos filtros e viéses.	48
Figura 3.22–Esquema comparativo de caminhos de descida na superfície de custo utilizando termo de momento.	49
Figura 3.23–Esquema exemplo de caracterização por fechamento.	50
Figura 3.24–Esquema com exemplo de trecho de imagem representada em grafos.	52
Figura 3.25–Exemplo com resultado da segmentação utilizando o método Felzenszwalb e Huttenlocher.	52
Figura 3.26–Esquema Geral FRCNN.	54
Figura 3.27–Comparativo RCNN com FRCNN.	55
Figura 3.28–Detalhes das camadas da arquitetura FRCNN.	58
Figura 3.29–Esquema referente à camada de subamostragem.	59
Figura 3.30–Esquema detalhes referente à ROI subamostragem.	60
Figura 3.31–Esquema Decomposição do Valor Singular Truncado.	61
Figura 3.32–Esquema exemplo decomposição do valor singular truncado.	62
Figura 3.33–Esquema representação multiescala e escala única.	63
Figura 3.34–Esquema de regressão de regiões de detecção.	64
Figura 3.35–Esquema exemplo de saídas da FRCNN.	65
Figura 3.36–Diferença entre a fase de treinamento e fase utilização do modelo de predição em novos dados.	66
Figura 4.1 – Etapas de PDI.	68
Figura 4.2 – Exemplo de gráficos com as curvas de atributo de Haralick contraste e aglomerado proeminente.	70
Figura 4.3 – Resultado de imagem utilizando filtro utilizando correlação.	71
Figura 4.4 – Imagens resultados de segmentação utilizando K-médias para 2 clusters.	73
Figura 4.5 – Imagens resultado de segmentação utilizando K-médias para 3 clusters.	74
Figura 4.6 – Exemplo de histograma característico utilizado pelo método Otsu.	75
Figura 4.7 – Exemplo de imagem e seu respectivo histograma com corte definido pelo processo Otsu.	76
Figura 4.8 – Imagens de resultados das etapas de PDI.	78
Figura 4.9 – Resultado da segmentação de Breakout sintético.	78

Figura 4.10–Resultado da utilização dos critérios aplicados pelo Método de Classificação por Polinômio.	80
Figura 4.11–Resultado de detecção na imagem utilizando Método de Classificação por Polinômio.	81
Figura 4.12–Esquema com a forma representativa de uma curva pela transformada de Radon.	82
Figura 4.13–Parametrização da curva senoidal na imagem.	83
Figura 4.14–Sinograma utilizando Transformada de Radon.	83
Figura 4.15–Resultado imagem de detecção de fratura utilizando Transformada de Radon.	84
Figura 4.16–Resultado da detecção de fratura utilizando Método Transformada de Hough.	85
Figura 4.17–Resultados da detecção com Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP)	86
Figura 4.18–Análise de orientação do objeto.	87
Figura 4.19–Exemplo da utilização de critérios de distância entre o par de desmoronamento.	88
Figura 4.20–Esquema Image Quilting.	90
Figura 4.21–Esquema representativo da formação de imagem de textura simulada, realizado pela função Image Quilting	90
Figura 4.22–Esquema dilatação em imagens de tons de cinza.	92
Figura 4.23–Comparativo fratura simulada com parâmetro de descontinuidade.	92
Figura 4.24–Comparativo fratura simulada com parâmetro de foco.	93
Figura 4.25–Comparativo fratura simulada com parâmetro de ruído.	93
Figura 4.26–Comparativo fratura simulada com parâmetro de contraste.	94
Figura 4.27–Exemplo de trecho de imagem acústica com fratura sintética e seus parâmetros de construção.	95
Figura 4.28–Esquema de gerador de um único desmoronamento.	96
Figura 4.29–Exemplos de desmoronamentos sintéticos	96
Figura 4.30–Esquema ROC.	98
Figura 4.31–Esquema área sob a curva utilizando o conceito de integral definida.	98
Figura 4.32–Diagrama com o resumo das etapas do processamento digital de imagens PDI até avaliação da detecção de fraturas.	100
Figura 4.33–ROC de detecção de fraturas utilizando Método de Classificação por Polinômio (MCP).	100
Figura 4.34–Casos de verdadeiros positivos de detecção fraturas sintéticas método Método de Classificação por Polinômio (MCP).	101

Figura 4.35–Diagrama com o resumo das etapas do processamento digital de imagens do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).	102
Figura 4.36–ROC detecção do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).	103
Figura 4.37–Casos de sucesso na detecção de desmoronamento sintéticos com Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).	104
Figura 5.1 – Esquema matriz de co-ocorrência.	107
Figura 5.2 – Esquema arquitetura rede neural rasa utilizada para detectar imagens com fraturas e desmoronamentos.	108
Figura 5.3 – Diagrama geral com Rede Rasa.	110
Figura 5.4 – ROC do classificador de fraturas com RNA rasa com duas camadas ocultas.	110
Figura 5.5 – Casos de verdadeiros positivos de detecção de fraturas em imagens sintéticas utilizando RNA rasa.	111
Figura 5.6 – ROC do classificador de desmoronamentos com RNA rasa com duas camadas ocultas.	112
Figura 5.7 – Casos de sucesso de detecções de desmoronamentos com rede neural rasa.	113
Figura 5.8 – Diagrama resumido com etapas até a avaliação de detecções utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	115
Figura 5.9 – ROC de desempenho da detecção de fraturas sintéticas utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	116
Figura 5.10–Casos de verdadeiros positivos de detecção de regiões de fraturas sintéticas com Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	117
Figura 5.11–Casos de falsos positivos de detecção de regiões de fraturas com Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	118
Figura 5.12–ROC de detecções de desmoronamentos Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	119
Figura 5.13–Casos de verdadeiros positivos de desmoronamento em imagens simuladas utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	120
Figura 5.14–Casos considerados como falsos positivos em imagens simuladas de desmoronamento, utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).	121
Figura 5.15–Detecção de fraturas reais utilizando Método de Classificação por Polinômio.	124
Figura 5.16–Detecção de fraturas utilizando FRCNN em fraturas reais.	125

Figura 5.17–Detecção de desmoronamentos reais utilizando Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).	126
Figura 5.18–Detecção de desmoronamento utilizando FRCNN em imagens reais. . .	127

LISTA DE TABELAS

Tabela 1 – Tabela relação xor.	77
Tabela 2 – Tabela de intervalo de parâmetros das fraturas sintéticas.	94
Tabela 3 – Tabela de percentuais de área abaixo da curva <i>AUC</i> dos métodos PDI.	101
Tabela 4 – Tabela de detecções de fraturas com métodos MTH e MTR.	101
Tabela 5 – Tabela de tempo de processamento da Rede Neural Artificial (RNA) e da Rede Neural Convolutiva Rápida baseada em regiões de imagens (FRCNN).	106
Tabela 6 – Tabela de percentuais de área abaixo da curva <i>AUC</i> da Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN) e Rede Neural Artificial (RNA) por classes.	115
Tabela 7 – Tabela comparativa de percentuais de área abaixo da curva <i>AUC</i> com imagens sintéticas e número de acertos de detecção de fraturas reais.	123
Tabela 8 – Tabela comparativa de percentuais de área abaixo da curva <i>AUC</i> com imagens sintéticas e número de acertos de detecção de desmoronamentos reais.	123

LISTA DE ABREVIATURAS E SIGLAS

CBPF	Centro Brasileiro de Pesquisas Físicas
IA	Inteligência Artificial
AM	Aprendizado de Máquina
RNA	Rede Neural Artificial
CNN	Convolutional Neural Network
RCNN	Region-based Convolutional Neural Network
FRCNN	Fast Region-based Convolutional Neural Network
MCP	Método de Classificação por Polinômio
MTR	Método Transformada de Radon
MTH	Método Transformada de Hough
ROC	Receiver Operating Curve
ANP	Agência Nacional do Petróleo
TT	Tempo de Trânsito
TV	Tensões Verticais
THm	Tensões Horizontais Mínimas
THM	Tensões Horizontais Máximas
IBM	International Business Machines
FA	Função de Ativação
FTH	Função Tangente Hiperbólica

CPU	Central Processing Unit
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
AP	Aprendizagem Profunda
MC	Mapa de Características
FCL	Fully Connected Layer
GDE	Gradiente de Descida Estocástico
MFH	Método de Felzenszwalb e Huttenlocher
RP	Regiões Propostas
VGG	Visual Geometry Group
PDI	Processamento Digital de Imagens
DRP	Método de Detecção de Desmoronamentos por Reconhecimento de Padrões
AUC	Area Under the curve

SUMÁRIO

Lista de ilustrações	xiii
Lista de tabelas	xix
Sumário	xxiii
1 INTRODUÇÃO	1
2 PERFIS DE IMAGENS	5
2.0.1 Perfilagem e Imageamento de Poço	5
2.0.2 Perfis de Imagem	6
2.0.3 Aquisição da Imagem Acústica	9
2.0.4 Interpretação dos perfis de imagem	11
2.0.5 Fraturas Naturais	13
2.0.6 Desmoronamentos	14
3 INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL	17
3.1 Redes Neurais Artificiais	18
3.1.1 Peso e Viés	19
3.1.2 Função de Ativação	20
3.1.2.1 Ativação Sigmóide	21
3.1.2.2 Ativação Tangente Hiperbólica	22
3.1.2.3 Ativação Relu	22
3.2 Aprendizado de Máquina	26
3.2.1 Fluxo de Dados em Redes Supervisionadas	27
3.3 Tipos de Arquiteturas RNA	33
3.4 Topologias das Redes Neurais	35
3.4.1 Aprendizagem Profunda	37
3.4.2 Redes Neurais Convolucionais	38
3.4.2.1 Camada convolucional + Função de ativação	39
3.4.2.2 Camadas de Subamostragem	42
3.4.2.3 Camadas totalmente conectadas	43
3.4.3 Treinamento da Rede Neural Convolucional	44

3.4.3.1	Gradiente descendente Estocástico com Momento	47
3.4.4	Rede Convolutiva baseada em Propostas de Regiões de Imagem	49
3.4.4.1	Regiões propostas	50
3.5	Rede Neural Convolutiva Rápida baseada em regiões de imagem	52
3.5.1	Arquitetura	53
3.5.1.1	Decomposição do Valor Singular Truncado	58
3.5.1.2	Escala e rotação invariante	61
3.5.1.3	Classificação	62
3.5.2	Aferição em Novos Dados	65
4	MÉTODOS CLÁSSICOS DE DETECÇÃO AUTOMATIZADA DE	
	ESTRUTURAS	67
4.1	Processamento Digital de Imagens	67
4.1.1	Pré-processamento	68
4.1.1.1	Regiões de Interesse	69
4.1.1.2	Filtro Baseado em Correlação	70
4.1.2	Segmentação	71
4.1.2.1	Gradiente Direcional	72
4.1.2.2	K-médias	72
4.1.2.3	Otsu	73
4.1.2.4	Niblack	74
4.1.2.5	Entropia Local	75
4.1.3	Pós-processamento	77
4.1.4	Extração de Atributos	77
4.1.5	Classificação e Reconhecimento	78
4.1.5.1	Método de Classificação por Polinômio	79
4.1.5.2	Método Transformada de Radon	80
4.1.5.3	Método Transformada de Hough	84
4.1.5.4	Método de Detecção de Desmoronamentos por Reconhecimento de Padrões	85
4.2	Imagem Acústica Sintética	88
4.2.1	Faturas e Desmoronamentos Sintetizados	89
4.3	Característica de Operação do Receptor	97
4.4	Avaliação Detecção de Fraturas	99
4.5	Avaliação da Detecção de Desmoronamentos	102
5	DETECÇÃO AUTOMATIZADA UTILIZANDO INTELIGÊNCIA AR-	
	TIFICIAL	105
5.1	Modelo Proposto de Rede Neural Rasa	106
5.2	Modelo de Rede Neural Convolutiva com Proposta de Regiões	
	de Imagem	108

5.3	Avaliação Detecção de Fraturas e Desmoronamentos com RNA Rasa	109
5.4	Avaliação da Detecção de Fraturas e Desmoronamentos com FRCNN	114
5.5	Detecção de Fraturas e Desmoronamentos Reais	122
6	CONSIDERAÇÕES FINAIS	129
	REFERÊNCIAS	131
	APÊNDICES	149
APÊNDICE A	– DESCRITORES HARALICK	151
APÊNDICE B	– INFORMAÇÕES GERAIS ANP POÇO	155

CAPÍTULO 1

INTRODUÇÃO

Uma das metas estratégicas de empresas de petróleo e gás que buscam novas tecnologias e inovações, é minimizar as incertezas relacionadas a produção em reservatórios de hidrocarbonetos, promovendo a integração de dados, a qualidade e a confiabilidade das bases de informações dos reservatórios e dos sistemas petrolíferos. Para isso, são necessários processos que otimizem a concepção, operação e análises de produção para viabilizar a qualidade do fator de gestão de reservatórios [1, 2].

Um reservatório de hidrocarbonetos é definido por rochas reservatório compostas de matriz rochosa sedimentar carbonática ou formada por arenito que, em geral, são porosas e permeáveis, i.e. com vazios conectados, permitindo a reserva do petróleo ou gás em volume considerável [1, 2]. Em particular, as fraturas naturais localizadas em rochas carbonáticas podem indicar permeabilidade e a presença de petróleo ou gás [3, 4].

Os responsáveis pelos setores de estudo de reservatórios coordenam pessoal de campo e de laboratório, a fim de avaliar as melhores condições para exploração de fluidos de interesse. Para isso, são necessários, mão-de-obra especializada e maquinários de alto custo para realizar estudos aéreos, terrestres e marítimos. Com o crescente ganho computacional e o desenvolvimento de novas ferramentas de operação em campo, a qualidade da aquisição de dados oferece aos profissionais de petróleo e gás um ganho significativo em informações de análise [3].

Um dos recursos disponíveis é a perfuração de poço para estudo de reservatórios, possibilitando duas etapas de investigação. A primeira consiste em retirar do poço através de uma ferramenta perfuradora rotativa, parte do cilindro de preenchimento do furo. Esse material é chamado de testemunho do poço e é levado para pesquisas em laboratórios utilizando técnicas para obter dados sobre a litologia do bloco rochoso da região [4]. Na segunda etapa, utilizam sondas para perfilar a parede do poço adicionando informações ao conjunto de dados utilizados para consolidação das informações [4, 5]. Essas ferramentas de sondagem são resistentes à pressões e temperaturas típicas do interior do poço, utilizam

emissores e receptores com a finalidade de mensurar parâmetros da formação através das propriedades elétricas, acústicas e radioativas sensíveis à complexidade de materiais existentes na rocha [3, 4].

Os parâmetros físicos da formação rochosa como a resistividade elétrica e a impedância acústica, são dados salvos em arquivos e posteriormente convertidos em curvas e imagens [6, 7]. Esses dados são utilizados para indicar entre outras ocorrências, a presença de fraturas naturais e desmoronamentos ou *breakouts*.

Porém, não é trivial encontrar essas ocorrências em alguns casos, pois a perfuração pode provocar rupturas ou marcas que se assemelham ou mascaram a existência dos eventos de interesse [4, 6].

Outro desafio é que reservatórios são encontrados em rochas sedimentares até em grandes profundidades [4]. Com isso, uma única imagem digitalizada de poço pode ter da ordem de centenas de milhares de linhas de pixels. Isso torna o trabalho do especialista muito demorado e limitado a sua interpretação visual [7].

Detectar e reconhecer ocorrências de interesse em imagem [8, 9, 10] de perfil tem como objetivo colaborar com as demais informações sobre a direção de escoamento de fluidos e a variação de tensão entre camadas rochosas. Por isso, tornar o trabalho do petrofísico o mais automatizado possível, contribui para diminuir o tempo e as divergências de classificação. Esse ganho, possibilita a análise de um maior volume de dados, relacionado diretamente ao aumento do aproveitamento do potencial dos campos de petróleo.

Nesse sentido, buscamos automatizar os processos aplicando modelos de detecção automatizados com o objetivo de aumentar a eficiência dos profissionais responsáveis pela análise das imagens, estabelecendo controle das informações referentes ao processo, sem esquecer a importância de reduzir o esforço ou a interferência humana [10]. Por isso, apresentamos meios de automatizar o processo de detecção e reconhecimento de padrões de fraturas naturais e desmoronamentos em imagens acústicas de amplitudes obtidas da parede de poço.

Os modelos clássicos propostos para detectar objetos de forma automática ou semi-automática em imagens, envolvem etapas de processamento digital de imagens determinadas pelo pré-processamento, separação entre o fundo e o objeto (i.e. segmentação), pós-processamento e reconhecimento de padrão. Os atuais modelos de detecção automatizados são baseados em aprendizado de máquina.

Entre os trabalhos baseados em modelos clássicos de detecção de fraturas naturais temos os apresentados por [11], [12] e [13]. O estudo de [11] propõe morfologia matemática, envolvendo segmentação Niblack para detecção de bordas e reconhecimento de padrões de fraturas tectônicas em imagens acústicas sintéticas à partir de identificação de curvas correlacionadas à polinômios de 4° grau. A justificativa da autora para utilização do polinômio é que dessa forma é possível encontrar resultados em casos onde a fratura sofreu distorção do seu padrão de senoide, devido ao alargamento do poço. O trabalho

de [12] sugere pré-processamento de imagens reais acústicas e de resistividade, utilizando filtragem, segmentação baseada em agrupamento de dados e determinação de padrões senoidais com o auxílio das transformadas de Radon e Hough. Em [13] a autora propõe processos que envolvem filtragem, segmentação utilizando histograma e reconhecimento de padrão com a transformada de Hough, para detecção em imagens reais. Contudo, os autores [11], [12] e [13] identificam que as técnicas empregadas são muito sensíveis a presença de outras ocorrências geológicas e artefatos presentes nas imagens decorrentes da aquisição. Esses estudos apresentaram resultados parciais, pois não utilizam um modelo estatístico de avaliação do classificador de fraturas que defina a acurácia dos métodos aplicados em um número de amostras significativo.

O estudo de [14] utiliza duas etapas de detecção. A primeira apresenta metodologia clássica na detecção de fraturas candidatas utilizando transformada de Hough e dessa maneira propõe regiões. O método utiliza essas regiões para avaliar o modelo de aprendizado de máquina, utilizando rede conhecida como Alexnet, treinada com imagens reais classificadas. Apresentou precisão de 97,5% na detecção de fraturas em imagens acústicas. Contudo, para cada região proposta é aplicado toda rede, exigindo muito processamento. Além disso, o autor não deixa claro o processo de validação e avaliação do classificador da rede. O método de detecção, assim como os dos demais autores, foi limitado apenas às fraturas, não identifica simultaneamente os desmoronamentos.

Os trabalhos relacionados à análise e reconhecimento de desmoronamentos foram apresentados por [15], [16], [17] e [18]. Em [15] o autor apresenta detalhes sobre a forma e as propriedades mecânicas dos desmoronamentos, apresentando alguns exemplos de ocorrências. Nos trabalhos de [16] e [17] são apresentados procedimentos para interpretação visual de desmoronamentos utilizando curvas e imagens resultados de medidas geradas por ferramentas de perfilagem utilizada para determinar o diâmetro do poço por profundidade. Em [19] o autor apresenta técnicas de processamento de imagem para corrigir problemas de aquisição em dados de nove poços. As curvas de medidas acústicas e de diâmetro do poço por profundidade, são comparadas, considerando seus valores originais e os novos valores corrigidos. A diferença entre o valores originais e corrigidos detecta os pontos de ovalização do poço indicando os locais de desmoronamentos.

O modelo, contudo, não apresenta nenhum método estatístico para avaliar a forma de classificar as ocorrências como desmoronamentos, tornando os resultados baseados em alguns casos.

No nosso trabalho, inicialmente utilizamos técnicas clássicas de processamento digital em imagens (PDI) para detecção e reconhecimento de objetos com os padrões das fraturas naturais e desmoronamentos, baseados em etapas de processamento digital de imagens [5, 11, 12]. O método foi avaliado com imagens simuladas, porém os resultados não atenderam à todos os níveis de dificuldades no reconhecimento de fraturas [11, 12] e desmoronamentos. Ocorrências com níveis muito baixo de contraste não foram reconhecidas

como esperado.

O atual estado da arte voltado para uso de redes neurais [20, 21, 22] permite detectar múltiplos objetos de interesse de classes distintas em imagens. Para obter uma quantidade de dados significativa é possível recorrer a simulação de imagens [5, 11]. As imagens sintéticas foram adotadas para o aprendizado e teste das arquiteturas destinadas à detecção, classificação de fraturas e desmoronamentos [21, 22, 23]. Assim, foi possível avaliar estatisticamente a qualidade dos classificadores utilizando modelos de Inteligência Artificial (IA).

Este estudo é organizado da seguinte maneira. No capítulo 2 descrevemos os processos de aquisição de imagens reais e sintéticas. No capítulo 3, é apresentada uma breve introdução à inteligência artificial e seus modelos de detecção em imagens. No capítulo 4 são descritos os modelos clássicos para detecção de estruturas utilizando processamento digital de imagens. No capítulo 5, são apresentadas as metodologias e resultados dos métodos de inteligência artificial e no capítulo 6 são discutidas as conclusões desse trabalho.

CAPÍTULO 2

PERFIS DE IMAGENS

Imagens digitais possuem papel importante na indústria, comércio e em centros de pesquisas, sendo poderosa ferramenta de caracterização [24]. O seu potencial compreende a maioria dos domínios científicos relacionadas as visualizações de fenômenos. As imagens que ajudam a observar e a interpretar são produzidas a partir do registro de fenômenos físicos. Em medicina, por exemplo, a radiografia, graças à utilização dos raios x, permite explorações não invasivas mais específicas do corpo humano [25]. Outros instrumentos de observação extremamente poderosos, como os microscópios eletrônicos ou os telescópios, têm como objetivo observar e depois reproduzir em imagens, respectivamente, o universo infinitamente pequeno de moléculas e a grandeza de galáxias [26, 27]. A interpretação destas imagens não correspondem a uma simples observação. Exige o apoio da tecnologia para classificar os fenômenos observados. Algoritmos e métodos cada vez mais robustos e sofisticados permitem o processamento contribuindo para a rapidez no reconhecimento do seu conteúdo e da sua interpretação [2].

2.0.1 Perfilagem e Imageamento de Poço

Setores da indústria do petróleo estudam as características das formações geológicas [2, 3] perfurando o bloco rochoso com estruturas como apresentado no esquema da figura 2.1. Com isso, obtém informações por meio da utilização de ferramentas de perfis através de diferentes propriedades físicas da rocha, tais como radioatividade, resistividade e densidade. Esse processo é conhecido como perfilagem. No geral, apresenta características estáticas das formações como as litologias, indicação de fluidos, porosidade, orientação de fraturas e camadas [2, 3]. A perfilagem também avalia a dinâmica das formações do local, realizando testes de danos e estimando parâmetros tais como permeabilidade, pressões, vazões e potencial de produtividade. A perfilagem avançou nos últimos anos e, como consequência, temos hoje os imageamentos de poço com escala de aproximadamente 0,2:1 polegada [3, 24, 28]. Representações das imagens da parede do poço são possíveis a

partir das medidas relacionadas as suas características resistivas e acústicas, conhecidas respectivamente como perfil de imagem elétrica e perfil de imagem acústica [4, 29]. Elas podem fornecer, após análise especializada, informações tais como orientação de camadas, ocorrência e direção de fraturas e desmoronamentos. Esses perfis também são utilizados para caracterizar litologia, discordâncias, identificar acamamentos e zonas com porosidade vugular ou cavernas, quando encontradas em matrizes rochosas [7]. Dados como esses são utilizados em conjunto com outros dados de perfis, amostras oriundas de testemunhos (i.e. amostras do cilindro central retirados do poço) e histórico de produtividade em estudos precursores à perfilagem [8].

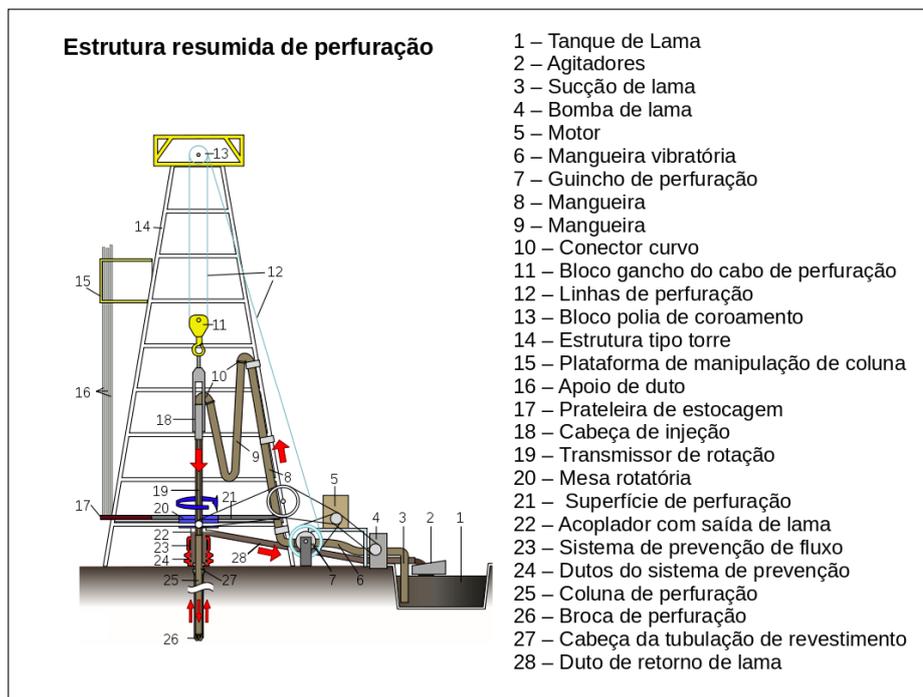


Figura 2.1 – Esquema resumido de estrutura de perfuração do poço para avaliação de reservatório, com descrição de alguns de seus componentes. Adaptada de [9].

2.0.2 Perfis de Imagem

Os perfis de imagem fornecem informações visuais relevantes por profundidade, determinada pelo deslocamento da ferramenta na parede do poço. Os perfis de imagem resistivas fornecem análise da parede do poço, e são produzidas por ferramenta composta por sensores e contatos deslizantes conhecidos como patins, para inferir as medidas de resistividade da rocha. Cada um dos patins possui da ordem de centenas de eletrodos que emitem corrente elétrica na formação rochosa. Assim, um sensor localizado na parte superior da ferramenta capta medidas de correntes, após percorrerem a parede do poço [7, 10].

As imagens de perfil acústico são produzidas por uma sonda para caracterizar poços utilizando ferramentas para medir a impedância acústica. Com ela são obtidas medidas

de Tempo de Trânsito (TT) e de Amplitude de ondas acústicas [4, 7]. Durante o processo de perfilagem acústica, a ferramenta desce rotacionando o poço até a profundidade determinada [3, 24]. Assim, é muito importante a calibração com os parâmetros iniciais, principalmente para manter a centralização do equipamento. Essa etapa depende do operador da unidade de perfilagem em campo [4, 28]. Uma vez que as medidas acústicas podem fornecer detalhes da rocha, sem esse cuidado, podem ocorrer medidas imprecisas, já que os sensores estão mais próximo ou mais afastados da parede do poço [3, 11]. Quando a ferramenta acústica começa a funcionar ao iniciar o movimento rotatório, emite ondas acústicas e capta com os sensores as medidas TT e de Amplitudes das ondas refletidas. A figura 2.2 apresenta um diagrama de ferramenta corretamente centralizada no poço, indicando como os pulsos sonoros podem ser medidos, visto que as ondas são enviadas e as reflexões retornam para a ferramenta que registra o tempo que a onda propagou e a atenuação do pulso [30].

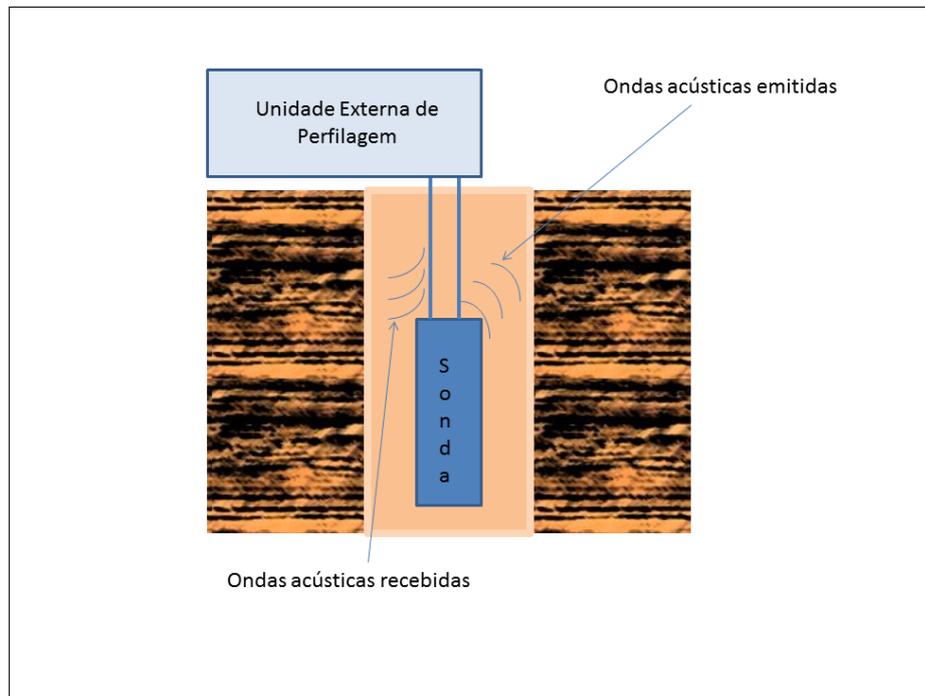


Figura 2.2 – Esquema ferramenta acústica localizada no poço.

As imagens de perfil resistivo e acústico são utilizadas para identificar características petrofísicas e geológicas de interesse, principalmente em rochas carbonáticas e arenitos. Essas rochas são conhecidas por suas propriedades porosas e permeáveis [4]. As rochas carbonáticas e arenitos são rochas sedimentares cuja composição primária são, respectivamente, os carbonatos e os quartzos. Essas formações apresentam baixos valores TT e altos valores de Amplitudes obtidos pela ferramenta acústica [10]. Em oposição, ocorrências geológicas naturais ou induzidas pela perfuração, produzem altos valores TT e baixos valores de Amplitudes.

A sonda acústica, diferente dos equipamentos de imageamento elétrico, não apresenta

restrição quanto à sua utilização de fluido de perfuração não condutivo [3]. A utilização de sondas com função de resistividade elétrica, funcionam apenas com lama condutiva [31]. Isso é uma característica importante, visto que, nos últimos anos houve um crescente preferência por lamas não condutivas na perfuração, decorrente ao ganho com a eficiência operacional e a estabilidade do poço [32].

Outro aspecto significante é a cobertura contínua por imageamento acústico de toda a parede do poço [32]. As ferramentas resistivas não apresentam descontinuidades laterais na imagem, onde os contatos dos patins estão presentes no momento da medição, contudo algoritmos são utilizados para apresentar uma imagem como a apresentada na figura 2.3, onde observamos uma leve alteração horizontal entre os segmentos.

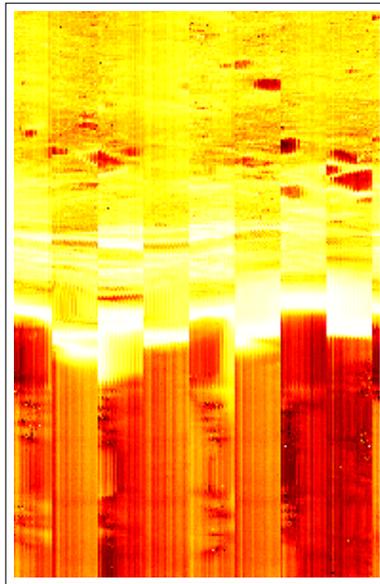


Figura 2.3 – Exemplo de trecho de imagem apresentada com mapa de cor do RGB *IBM*[®] denominada pelo termo inglês *hot*, obtida pelo dado ANP de perfil resistivo.

O processamento digital das medidas acústicas de Amplitude começa com a confecção de mapas de cores conforme apresentamos na figura 2.4, onde cada medida representa um valor de pixel referente à uma escala de cor utilizada. As feições com baixos valores de Amplitudes, tais como, irregularidades no poço (desmoronamentos) e fraturas naturais são apresentadas em escala de cor mais próximo ao preto, com valores mais próximo de 0, como visto na imagem em escala de cinza, à direita da figura 2.5. Essas ocorrências apresentam um tom mais claro na escala de cor em imagem de *TT*, conforme a imagem à esquerda da figura 2.5. As matrizes rochosas, são representadas em tons mais claros com valores mais próximo ao 1. Outras escalas de cores são utilizadas durante a visualização da imagem.

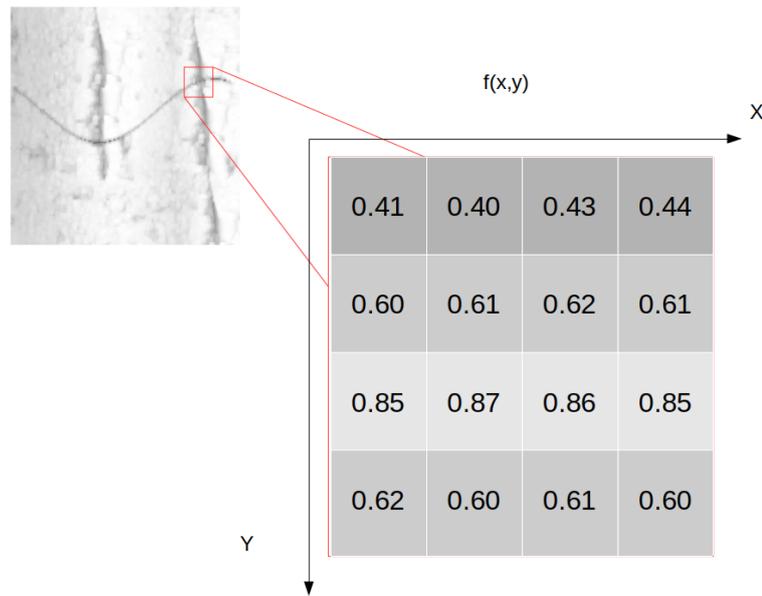


Figura 2.4 – Esquema conceitual de matriz de valores medidos convertidos em matriz, onde cada medida representa um valor de pixel referente à uma escala de cor. Nesse caso, utilizamos escala em tons de cinza.

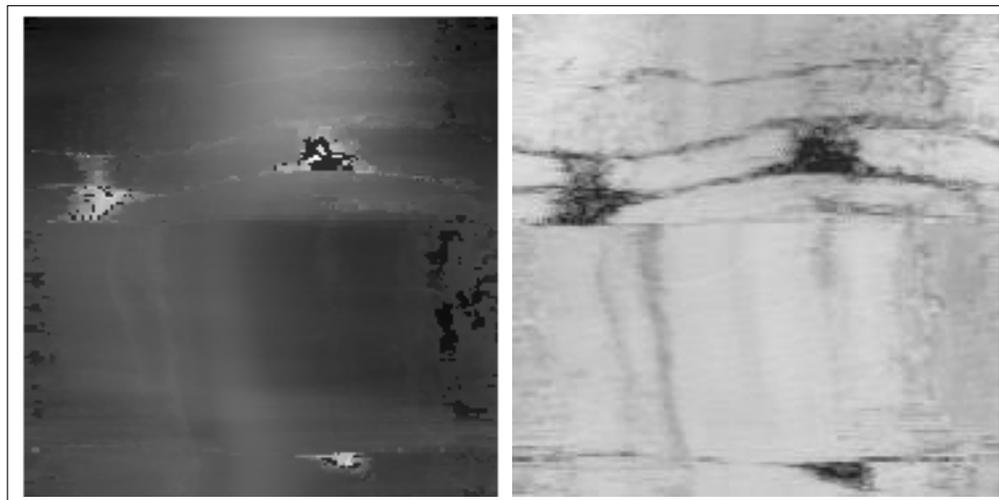


Figura 2.5 – A imagem à esquerda corresponde a imagem de tempo de trânsito e ao lado a de amplitudes. As duas são referentes a mesma profundidade e obtida de [12] referente as imagens obtidas dos dados de [12].

2.0.3 Aquisição da Imagem Acústica

O equipamento de imageamento acústico, visto em 2.6, é uma ferramenta composta por um transdutor rotacional, acelerômetro e magnetômetro. O acelerômetro realiza medidas de velocidade e o magnetômetro estima a inclinação da perfuração [33]. O transdutor é o dispositivo responsável por emitir e registrar os pulsos acústicos após reflexão na rocha. Essas ondas sonoras são emitidas por um material piezoelétrico, presente no dispositivo, que tem a função de converter energia elétrica em energia mecânica em forma de pulsos

acústicos que são direcionados à parede do poço. O transdutor possui, também, um motor que realiza a rotação, permitindo a ferramenta emitir pulsos em várias direções [33]. A ferramenta envia os dados para uma unidade de campo móvel na superfície, que controla os parâmetros de aquisição. Após obter os dados da formação rochosa da parede do poço pela unidade de campo, os mesmos são enviados para uma estação de trabalho do especialista. No entanto ainda são necessárias correções durante a pós-aquisição.

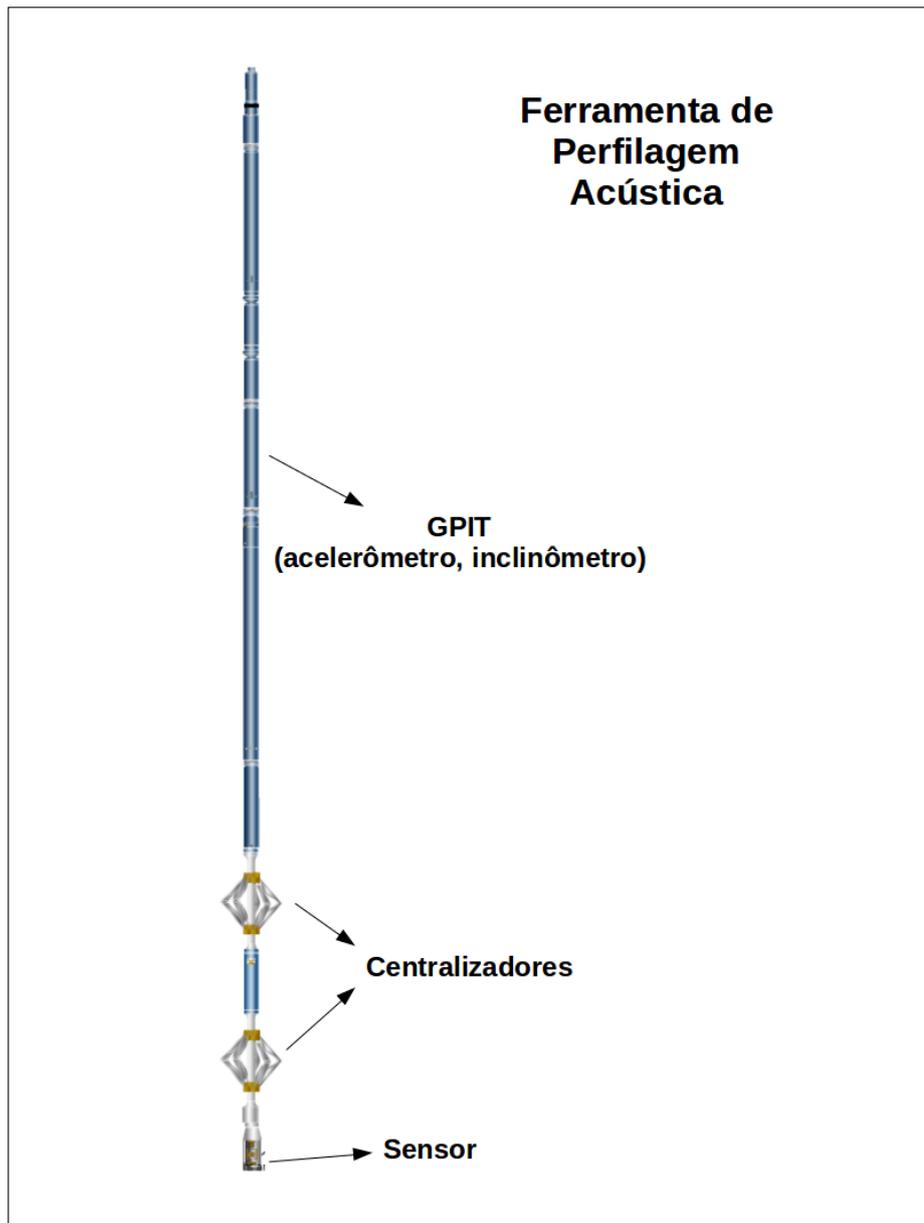


Figura 2.6 – Esquema ferramenta da acústica obtido das informações cedidas pela ANP.

A finalidade da pós-aquisição é utilizar recursos computacionais e matemáticos para converter e corrigir as informações obtidas em um conjunto de dados que possam ser interpretados [34].

Esses dados passam por correções de velocidade [35] que durante a aquisição, é medida pelo acelerômetro em intervalos regulares. A etapa de correção é utilizada para corrigir

essas medidas, quando comprometidas pela ocorrência de deslizamento da coluna devido a diferença de camadas de rocha [35]. Então no pós-aquisição, a correção de velocidade é uma etapa altamente recomendável para obtenção de uma imagem com contorno e nitidez contínuas. A correção de excentricidade também é realizada para garantir que as medidas não estejam comprometidas pela descentralização da ferramenta causada pela ovalização do poço devido a deformações produzidas pela perfuração [36]. As possíveis correções de velocidade e excentricidade referente a medidas realizadas pela ferramenta acústica são realizadas em *software* específico.

2.0.4 Interpretação dos perfis de imagem

A etapa de interpretação das características de interesse realizada pelo profissional é de grande importância para a caracterização do reservatório [37]. Antes da existência das imagens de perfil ou imageamento do poço, era utilizada a ferramenta de mergulho ou *dipmeter* que mede a microresistividade da parede do poço utilizando apenas três eletrodos. O diagrama de *dipmeter* apresentado na figura 2.7 determinado pelas medidas de microresistividade da rocha, as variações de velocidade e inclinação da ferramenta [38], que geram gráficos permitindo que o especialista interprete e determine a direção e a inclinação de camadas rochosas e fraturas interceptadas pelo poço. Nesse diagrama a inclinação de cada profundidade rochosa é determinada em uma escala de 0 a 90 graus por um marcador do tipo ponto, que representa a magnitude de inclinação ou mergulho de termo em inglês (*dip*) da camada ou fratura [39]. Uma linha ligada a cada ponto é apontada no mesmo sentido da direção do mergulho da camada em relação ao Norte da rosa-dos-ventos, definindo o azimute [39], que no exemplo da figura 2.7 é de 45 graus, com um dip de 23 graus na profundidade 1000. O eixo Y é a profundidade e o eixo X é escalonado em graus, onde o mergulho é determinado pela inclinação horizontal. Um ponto é demarcado e o azimute associado é projetado com o eixo Y como norte verdadeiro, formando uma exibição gráfica em forma de pontos. As cores e formas dos pontos podem ser variadas para distinguir diferentes tipos de ocorrências.

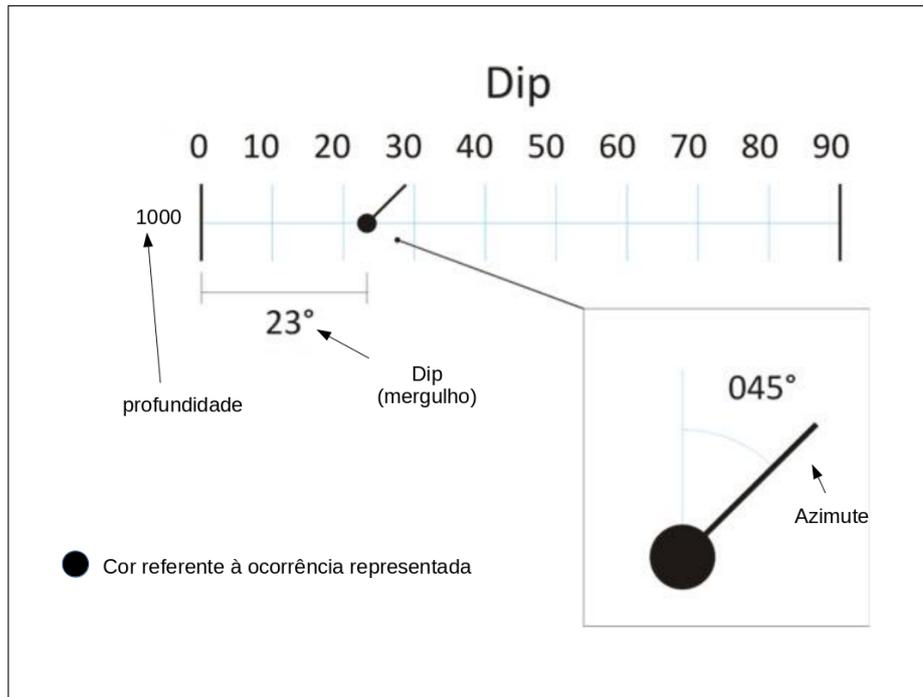


Figura 2.7 – Esquema representativo do recurso *dipmeter*. Um método em duas dimensões usando pinos para ilustrar a orientação de ocorrências como camadas de rochas diferentes, fraturas e desmoronamentos derivados de perfilagem de microresistividade com medidores de mergulho em uma determinada profundidade.

O desenvolvimento de recursos de imageamento, contribuiu para obtenção de informações adicionais com maior precisão, para consolidação com outras fontes de dados geológicos [38].

Assim, os dados das medidas podem ser convertidos digitalmente em escala de cores e em imagens. Imagens acústicas de tempo de trânsito apresentam um mapa do tempo que a onda acústica emitida pela ferramenta percorreu até a parede do poço, refletiu e atingiu o sensor [4, 7]. As imagens acústicas de amplitudes modelam atenuações das ondas acústicas. Os materiais existentes na rocha possuem coeficientes de reflexão diferentes. Com isso, a atenuação medida está relacionada a complexidade de elementos presentes na parede do poço. As imagens estão entre as que mais se destacam visualmente a localização de ocorrências geológicas e induzidas, por isso há o interesse dos profissionais por esse recurso. Isso porque o material de preenchimento de ocorrências como fraturas naturais e desmoronamentos diferem da rocha intacta, principalmente pela presença de fluido [4, 6]. O resultado é o contraste, algumas vezes perceptível na imagem, entre as bordas e texturas de diferentes fases.

As imagens acústicas de amplitudes são as mais utilizadas para obter as inclinações e direções das camadas e fraturas, permitindo também analisar e determinar desmoronamentos e a litologia do poço, já que apresentam maior contraste entre essas ocorrências e a matriz rochosa [39]. Assim, são analisadas as direções, inclinações e aberturas das fraturas e a área aproximada do desmoronamento através da área retangular [40, 41].

2.0.6 Desmoronamentos

As formações rochosas estão sujeitas a um estado de tensão compressivas chamado de tensões locais, compreendidas por tensões verticais, tensões horizontais máximas, tensões horizontais mínimas [16], a tensão de pressão de poros e a pressão de formação. A tensão vertical, em geral, é a maior tensão atuante na formação. Sua direção aponta sempre para o centro da Terra e sua origem é o peso das camadas sobrejacentes à rocha [50]. A pressão de formação é entendida como a pressão do fluido contido no volume poroso da rocha. As tensões horizontais, são o resultado do peso vertical, levando em consideração as partes de rocha que se deformam lateralmente limitada pelo material vizinho [15, 51]. Avaliar as tensões horizontais máximas e mínimas não é possível através de modelos matemáticos, sem adotar uma série de aproximações e simplificação do comportamento do maciço rochoso [52]. A determinação das direções das tensões horizontais é fundamental na detecção de desmoronamentos nas paredes do poço, pois as direções dessas ocorrências, conforme apresentamos no esquema na figura 2.9, são definidas pelas direções das duas tensões horizontais mínimas [50, 53]. Os desmoronamentos, conhecidos pelo termo em inglês *breakouts*, são detectados na perfilagem de poço em imagens acústicas [51, 52, 54]. Quando a rocha está sujeita a ação das tensões locais, regiões do poço tendem a desmornar devido a instabilidade, apontando a direção de atuação da máxima e mínima tensão de compressão horizontal da rocha [55, 56]. A geometria do poço, em geral é cilíndrica, mas em decorrência da perfuração em determinadas rochas pode ocorrer a ovalização devido aos desmoronamentos laterais da parede do poço. Essa ocorrência é caracterizada nas imagens acústicas normalmente em pares localizados a 180 graus um referente ao outro [57]. Se a imagem do poço não passou por um rigoroso pré processamento com as devidas correções, alguns desmoronamentos podem não apresentar o seu devido par. Por isso, avaliar a orientação de ocorrência de possíveis desmoronamentos sem o par é necessário, comparando sua orientação com os demais desmoronamentos próximos, pois a direção de tensão é a mesma para toda formação [57]. Avaliando as ocorrências de desmoronamento do poço o analista de reservatório pode estimar a ação de tensão da rocha. Os desmoronamentos possuem um padrão peculiar na imagem acústica de amplitudes [57, 58]. São identificados como dois prolongamentos de tonalidades mais escuras conforme exemplificado na figura 2.10.

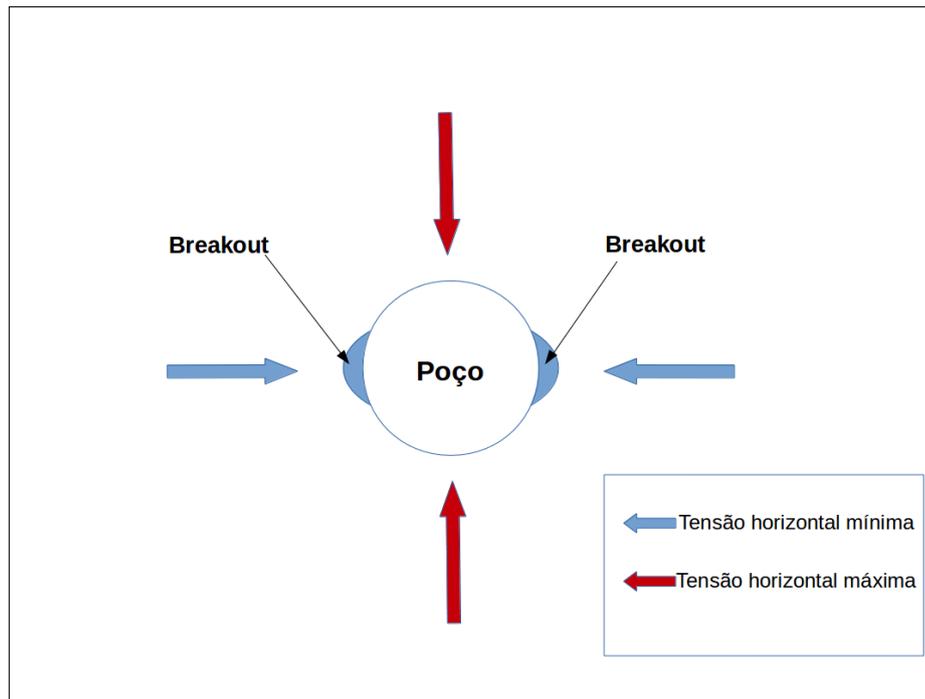


Figura 2.9 – Esquema tensões horizontais de desmoronamento no poço. As tensões horizontais mínimas são responsáveis por desmoronamentos que podem ocorrer durante a perfuração.

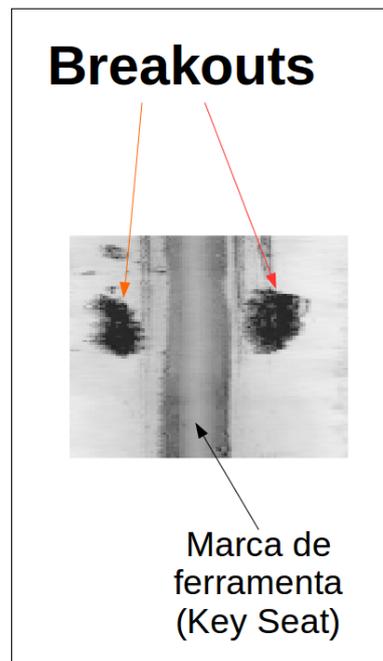


Figura 2.10 – Trecho de imagem real com desmoronamento. Entre esse par de desmoronamentos observamos uma marca de ferramenta conhecida pelo termo *key seat*. Essa marca foi provocada pela perfuração e apresenta esse formato devido o tipo de material presente e não relacionado à variação de tensão da rocha.

CAPÍTULO 3

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (*IA*) visa solucionar problemas de regressão, classificação, decisão, modelagem, construção de dados sintéticos e mapeamento de correlações baseado em métodos de aprendizado de máquina (*AM*) [59, 60], sem que exista, necessariamente, um modelo matemático prévio. Para isso, utiliza algoritmos que permitem realizar processos de decisão e aprendizado. Uma aplicação típica está na automatização de processos [14, 22]. O surgimento de sistemas computacionais de grande porte deu início à modelos de aprendizado automático inicialmente introduzidos pela indústria durante a década de 1950 [59]. Pesquisas dos laboratórios da *IBM*[®] foram algumas das pioneiras nessa direção. Porém, devido a limitações de estudos teóricos e de poder de processamento dos computadores da década de 50, associadas à falta de embasamento teórico de suporte, a implementação desses modelos falharam [60]. No entanto, por volta de 1970, com o aumento do poder de processamento dos computadores, uma nova abordagem ganhou destaque, utilizando aprendizado de máquina (*AM*) [60]. Dessa forma, todas as operações lógicas foram reproduzidos pelo modelo desenvolvido com aplicações limitadas pela configuração das redes [61]. O esforço de pesquisadores da área médica na busca do entendimento dos mecanismos biológicos responsáveis pela inteligência, no início dos anos 80, levou a uma descoberta que demonstrou o funcionamento do neocórtex [61]. Esses pesquisadores demonstraram que os meios de aprendizado do cérebro humano, processam informações em uma sequência hierárquica utilizando a repetição e o paralelismo, realizados de forma organizável. Esse conceito de repetição e aprendizado destaca a principal noção das redes neurais artificiais. Segundo os autores [62, 63] existem parâmetros gerais de percepção que seriam predominantes no processamento da informação visual e que se sobrepõe as peculiaridades de cada contexto. O cérebro busca parâmetros como por exemplo cor, forma, contornos, textura, ou quaisquer similaridade, tentando extrair propriedades essenciais e

relevantes, enquanto descarta as características secundárias e inconstantes [64]. Segundo este pressuposto, o cérebro seria naturalmente atraído por repetições e ritmos e estaria à procura de padrões que o levassem a antecipar ações, reduzindo a energia empregada no processamento de novas informações. [62]. Diante de imagens ambíguas ou confusas, o cérebro realiza um esforço organizacional para que a informação visual faça sentido, sempre tendo como referência informações já aprendidas. Este fenômeno foi descrito por psicólogos da Gestalt [65] na década de 80. Outro fato é que tendo em vista que o cérebro tem limitação na capacidade de armazenamento de informações, procura sempre simplificar e agrupar, para que se utilize da memória de forma econômica [64]. Contudo, apenas na década de 90 surgiram pesquisas visando o aprendizado de máquina em analogia ao processo de aprendizado biológico, utilizando a estratégia de repetição e transformação dos problemas mais complexos em questões mais simples, com o propósito de encontrar padrões, facilitando seu reconhecimento [65, 66, 67, 68]

Pouco tempo depois, ocorreu um avanço na computação paralela [61, 69], possibilitando o treinamento de modelos mais complexos com desempenho em um tempo menor relacionado aos anteriores. Jeff Hawkins em 2004, fundador da *Palm*[®] alegou que as descobertas de AM funcionam como reconhecedores de padrões [69], utilizadas de forma a aprender qualquer conceito e realizar previsões e generalizar sobre algo desconhecido.

3.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (doravantes RNA), são modelos matemáticos inspirados no modelo biológico, desenvolvidas com objetivo de resolver problemas de tomada de decisões, processamento de informação, aprendizagem, classificação e otimização [70]. As RNAs buscam simular algumas funcionalidades dos neurônios biológicos, com a atribuição de solucionar questões lineares e não lineares de forma paralela tolerante a erros, com poder de adaptabilidade, de aprendizado e generalização de forma robusta [71]. Os neurônios artificiais são as unidades básicas da RNA e os pesos possuem como uma das atribuições o armazenamento das características extraídas no processo de aprendizado de máquina [71], atualizados de forma que a rede consiga diferenciar os dados de entrada e reconheça os padrões durante o treinamento. O treinamento de uma RNA é uma etapa onde muitos exemplos dos dados de entrada são apresentados para a rede até que a rede aprenda e atinja um estado onde não haja mais mudanças significativas. Para isso, durante o treinamento os dados de entrada são acompanhados dos respectivos dados de saídas esperadas [72]. Assim, de acordo com a construção do modelo de aprendizagem é possível avaliar e atualizar os parâmetros. Ao apresentar novos dados ao modelo, ele deve ser capaz de prever o dado ou a probabilidade para cada uma das classes definidas durante o treinamento [72]. Na figura 3.1 apresentamos os componentes do neurônio biológico e o modelo artificial, as conexões entre as entradas (x_1, x_2, \dots, x_m) e a saída (y) do neurônio biológico

são representados pelos axônios e dendritos [70]. No caso do neurônio artificial as entradas (x_1, x_2, \dots, x_m) são ponderadas respectivamente pelos pesos sinápticos (w_1, w_2, \dots, w_m) e somadas ao valor de um bias b_k na junção aditiva, assim uma função restritiva é aplicada para limitar o intervalo do sinal de saída (y) [70]. Os pesos sinápticos são ponderados de forma coerente com o exemplo de entrada, onde processos iterativos se repetem até que a rede neural alcance um resultado satisfatório, aceitável para a solução do problema [73].

A função de ativação é responsável pela saída de um neurônio e introduzem um componente não linear, que faz com que elas possam aprender mais do que relações lineares entre as variáveis dependentes e independentes será detalhada na seção 3.1.2. A configuração de uma RNA é composta pela interconexão dos neurônios e pode variar em relação ao número de camadas, número de neurônios em cada camada, tipo de função de ativação e a forma como as camadas são conectadas [73, 74]. Os algoritmos de RNA determinam as regras para o processo de aprendizagem, definindo os parâmetros iniciais, a propagação do sinal nas camadas, o ajuste ou aprendizagem e a ativação dos neurônios. Essas regras variam de acordo com o tipo de aplicação. As RNAs podem ser de arquiteturas rasas, conhecidas pelo termo em inglês *shallow architectures* [74], onde apresentamos um exemplo desse tipo de rede na figura 3.12 e nela podemos entender a sua relação com um único neurônio [74]. Também existem as arquiteturas profundas conhecidas pelo termo em inglês *deep learning*. A nomenclatura rasa ou profunda atribuída à essas redes está relacionada a diferenças de entre tipo de entradas e o número de camadas ocultas [75], porém cada arquitetura possui as suas peculiaridades, detalhes na seção 3.4.1. As redes necessitam de aprendizado ou treinamento cada aplicação define o tipo de aprendizagem e a forma que a rede se adapta às informações recebidas de maneira supervisionada, não supervisionada ou por reforço [75], mais detalhes na seção 3.2. Nesse sentido, os pesos e os vieses tem papel fundamental no desempenho de uma RNA, principalmente em relação as suas definição inicial, pois tem influência na qualidade da solução, na generalização de desempenho e no tempo de treinamento.

3.1.1 Peso e Viés

O peso w e o viés b são parâmetros inicialmente pré-definidos ou são escolhidos aleatoriamente [77]. Esses parâmetros definem o modelo do neurônio ou das diferentes camadas e, após o processo de treinamento, são estes parâmetros que irão armazenar as informações relevantes da aplicação na qual a RNA foi proposta [77]. Para ilustrar apresentamos um processo simplificado, o qual uma RNA pode ser treinada para separar classes, com intuito de exemplificar a utilização dos pesos e dos vieses, consideremos dois grupos distintos [78]. O primeiro representado em vermelho, A, e o segundo de azul, B, como na figura 3.2, podem ser classificados em duas classes de forma linear [78]. Na primeira interação do código os parâmetros (w_1, b_1) que são inicializados aleatoriamente definem a reta z_1 , como ilustrado na figura 3.2. O processo de treinamento, e otimização definidos inicialmente,

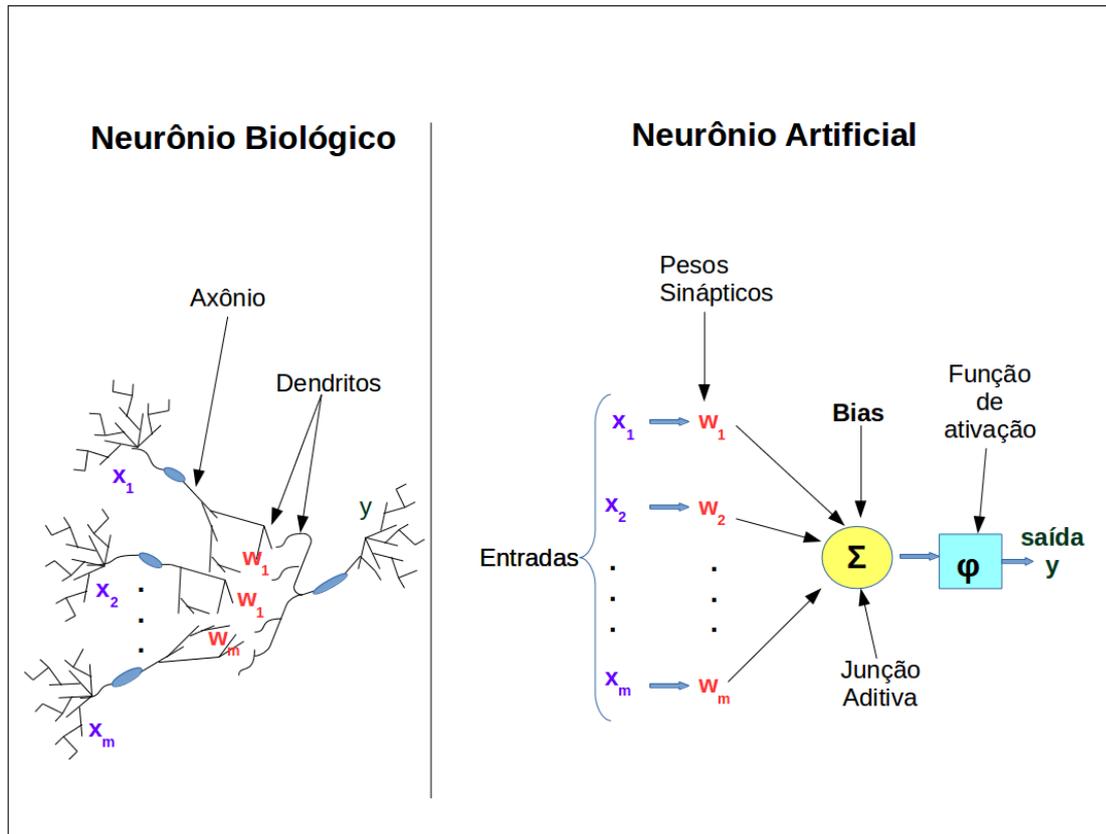


Figura 3.1 – Esquema adaptado de [76] apresenta comparativo entre o neurônio biológico à esquerda que inspirou o neurônio artificial à direita. As conexões entre as entradas (x_1, x_2, \dots, x_m) e a saída (y) do neurônio biológico são feitas pelos axônios e dendritos. No caso do neurônio artificial as entradas (x_1, x_2, \dots, x_m) são ponderadas por seus respectivos pesos sinápticos (w_1, w_2, \dots, w_m) e somadas ao valor de um bias b_k na junção aditiva, assim uma função restritiva é aplicada para limitar o intervalo do sinal de saída (y) .

estabelecem o quão distante z está do objetivo de definir uma reta que divida duas classes, utilizando novos pares (w, b) e estabelecendo novas retas durante cada interação. Nesse caso, após a quarta interação a reta z_4 , definida pelos parâmetros w_4 e b_4 divide as duas classes com nenhum erro [78]. Nesse exemplo podemos verificar que o peso tem o papel de ponderar a entrada e o viés tem o papel de deslocar a reta. Sem o viés todas as retas passariam pela origem e não cumpririam o objetivo. Dessa forma, o viés é um elemento que serve para aumentar o grau de liberdade dos ajustes dos pesos [78].

3.1.2 Função de Ativação

O desempenho da rede neural depende de vários fatores como a quantidade de camadas ocultas, a definição do conjunto de parâmetros, o tipo de aprendizagem e a função de ativação [79]. As funções de ativação introduzem componentes de não linearidade na RNA, definindo a amplitude do sinal de saída à um valor finito [79], que faz com essas redes possam aprender mais do que relações lineares entre as variáveis dependentes e

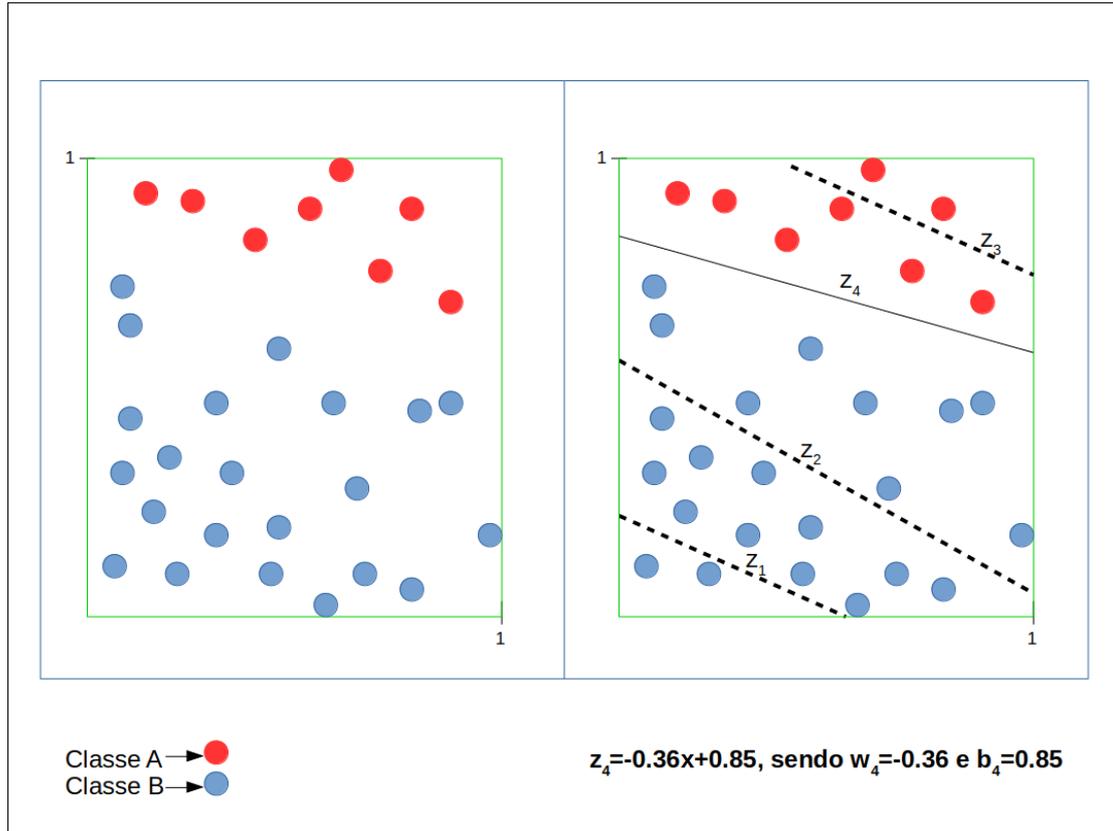


Figura 3.2 – Esquema simplificado de classificador binário com função linear. Esse esquema simplificado apresenta o caso de um grupo de dados e o processo linear de separação em duas classes. Os parâmetros de ajustes w e b foram atualizados até encontrar a reta z_4 que separa com maior eficiência as classes A e B.

independentes. A escolha da função de ativação ou (doravante FA) pode influenciar fortemente a complexidade computacional e o desempenho durante o processo de convergência do algoritmo de aprendizagem [80]. A convergência é a capacidade da rede aprender todos os padrões do conjunto de treinamento. As características dos dados, geralmente, definem a necessidade de uma FA que melhor se adapte ao método de otimização da arquitetura durante a aprendizagem [80]. Nesse sentido a derivada da função de ativação precisa ser previamente avaliada, pois indica a característica da convergência do aprendizado. As funções de ativação mais comuns em RNA são conhecidas como sigmóide, tangente hiperbólica (\tanh) e relu [81].

3.1.2.1 Ativação Sigmóide

As primeiras RNAs [81, 82] utilizavam a função sigmoide ou conhecida como função logística. A justificativa era o comparativo com a forma que os neurônios biológicos são apresentados, em dois estados conhecidos como ativados ou não ativados, logo de modo binário [79, 81]. Dessa forma, a função de ativação sigmoide é uma boa forma de modelar

esse comportamento através da equação:

$$\text{Sigmóide}(x) = \frac{LM}{1 + \exp^{-k(x-x_0)}}, \quad (3.1)$$

onde LM é o limite máximo da curva igual a 1, k é a declividade da curva igual a 1 e x_0 é o valor de x no ponto médio da curva sigmoide [82, 83]. Para valores de x no domínio dos números reais de $-\infty$ a $+\infty$, a curva sigmoide à direita é obtida com o gráfico de $\text{sigmóide}(x)$ se aproximando à LM conforme o x se aproxima de $+\infty$ e se aproximando à zero conforme o x se aproxima de $-\infty$) [82]. A representação dessa curva como função de ativação assume valores entre 1 e 0, conforme é possível visualizar a curva em vermelho na figura 3.4. O comportamento da derivada de uma função de ativação pode indicar o comportamento da convergência da rede. Nesse sentido, podemos observar o comportamento da derivada da função sigmóide atingindo valores máximos [80, 81], sempre menor que 0.5, conforme podemos visualizar a curva em azul na figura 3.4. Esse comportamento é uma desvantagem visto que a curva da derivada tende à 0 em valores de x acima de 5 e menores de -5, apresentando um fenômeno característico relacionado à um problema de estagnação da convergência, responsável pela saturação da rede [81, 84]. Por isso, a utilização da função sigmóide é limitada à alguns casos de RNA, pois com valores fixos da função de ativação, a não-linearidade é determinada pelos pesos das conexões, reduzindo a capacidade de aprendizado a ser explorada.

3.1.2.2 Ativação Tangente Hiperbólica

A função tangente hiperbólica (FTH) é similar a sigmóide, porém varia de 1 ativado e -1 não ativado através da equação,

$$\text{Tanh}(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{\exp^{-2x} - 1}{\exp^{-2x} + 1}, \quad (3.2)$$

porém a curva da derivada dessa função tende a zero quando x são valores acima de 2 e menores de -2, sofrendo do mesmo problema de estagnação de convergência característico da ativação sigmóide. No entanto, a derivada da FTH, diferente da sigmóide, consegue atingir valores próximos à 1 [85]. Devido o seu ganho em relação à sigmóide, com o decorrer dos anos, mais redes passaram à utilizar FTH. Contudo, uma desvantagem dessa função de ativação, é relacionada ao tempo computacional e testes anteriores são necessário para definição dos parâmetros w e b que se ajustam à função [85].

3.1.2.3 Ativação Relu

A função relu é a mais indicada em redes que utilizam método de otimização e recentemente mais utilizada [86]. A aplicação dessa função é um recurso que acelera consideravelmente a convergência devido a sua região linear, que não satura conforme é apresentado

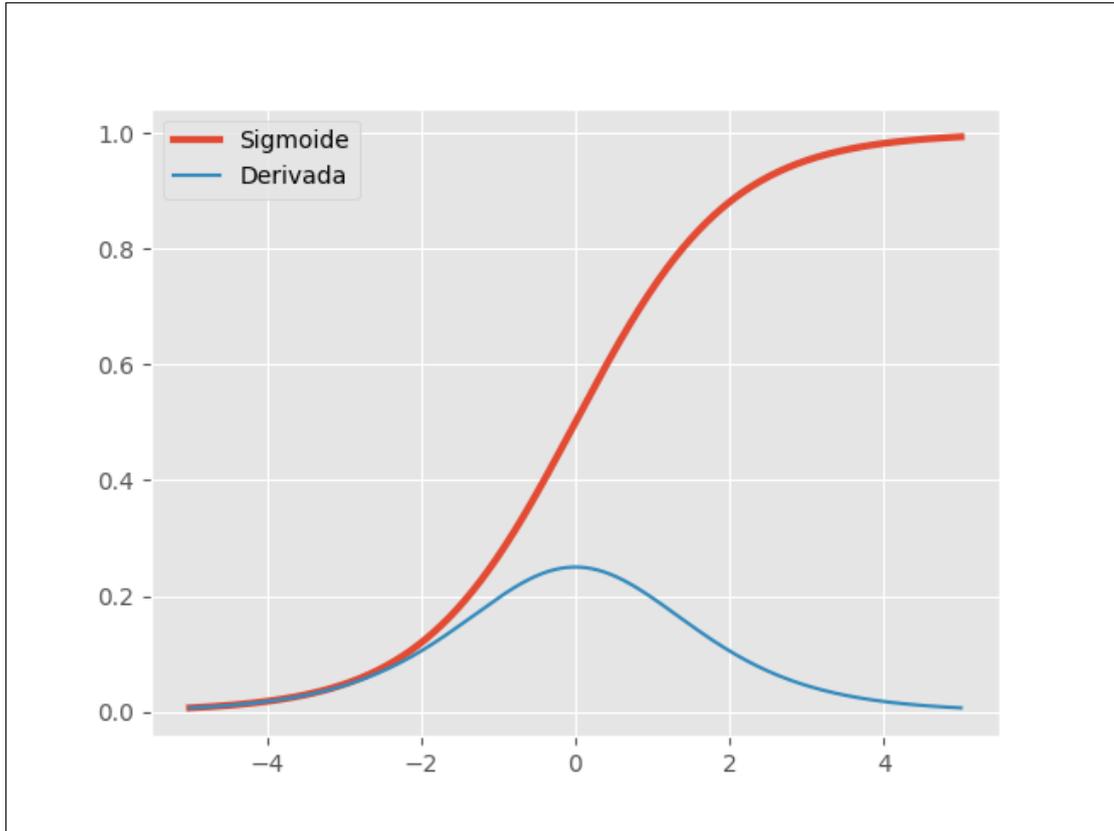


Figura 3.3 – Curva característica da função de ativação sigmoide ou logarítmica e sua derivada. Através da derivada podemos avaliar pontos onde a convergência estagna e satura, nesse caso, em valores acima de 5 e abaixo de -5.

em 3.5. Aplicando essa função, inativa os valores negativos, substituindo por zero, dessa forma é responsável por inativar neurônio de saída negativa [86] pois,

$$\text{Relu}(x) = \max(0, x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.3)$$

e sua não linearidade permiti um ganho no processamento, já que essa função trabalha com a maioria da imagem da função em zero, com isso as derivadas se mantêm com valores máximos em 1 enquanto estiver ativa, ou seja, x maior que 0, conforme figura 3.5. Essa função facilita a otimização, por conta da sua similaridade com a função identidade, já que para valores de x maiores de 0, a imagem de cada elemento é o próprio elemento [86].

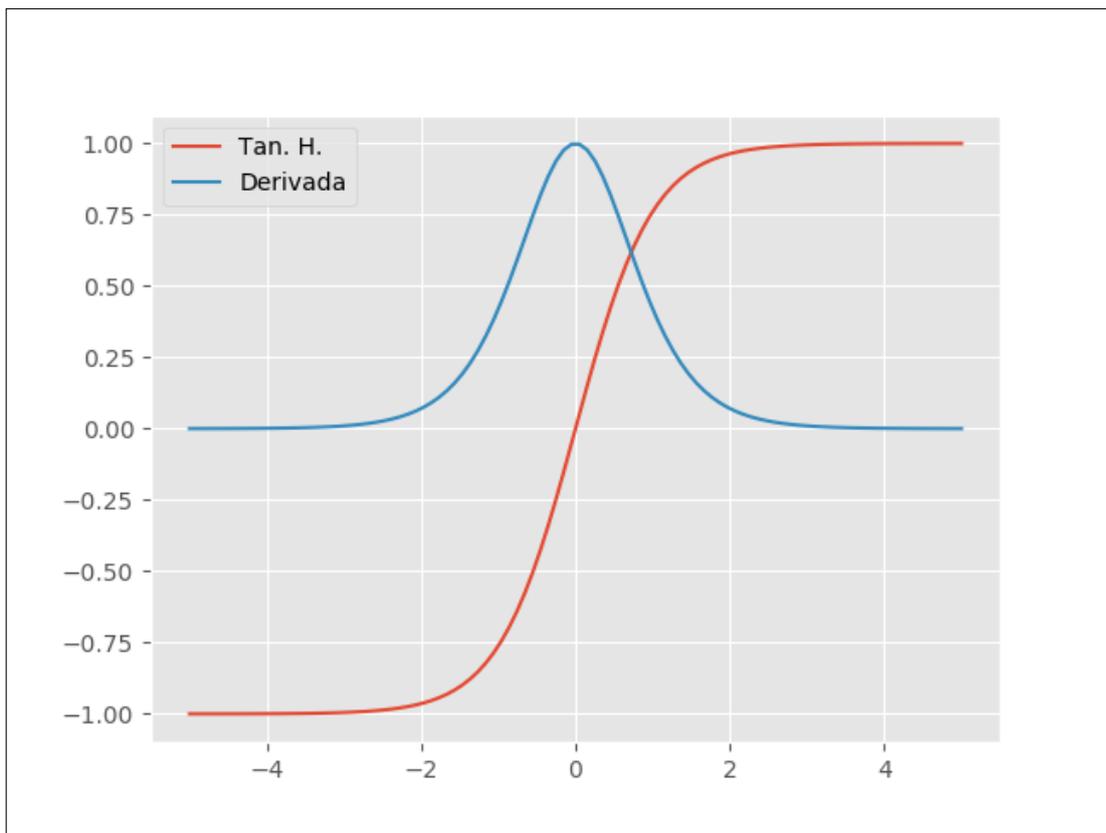


Figura 3.4 – Curva característica da função tangente hiperbólica e sua derivada. Com a derivada da função tanh podemos observar a evolução da convergência do aprendizado. Nesse caso, ocorre a estagnação em valores maiores de 2 e menores que -2 e a partir desses valores não ocorre a convergência.

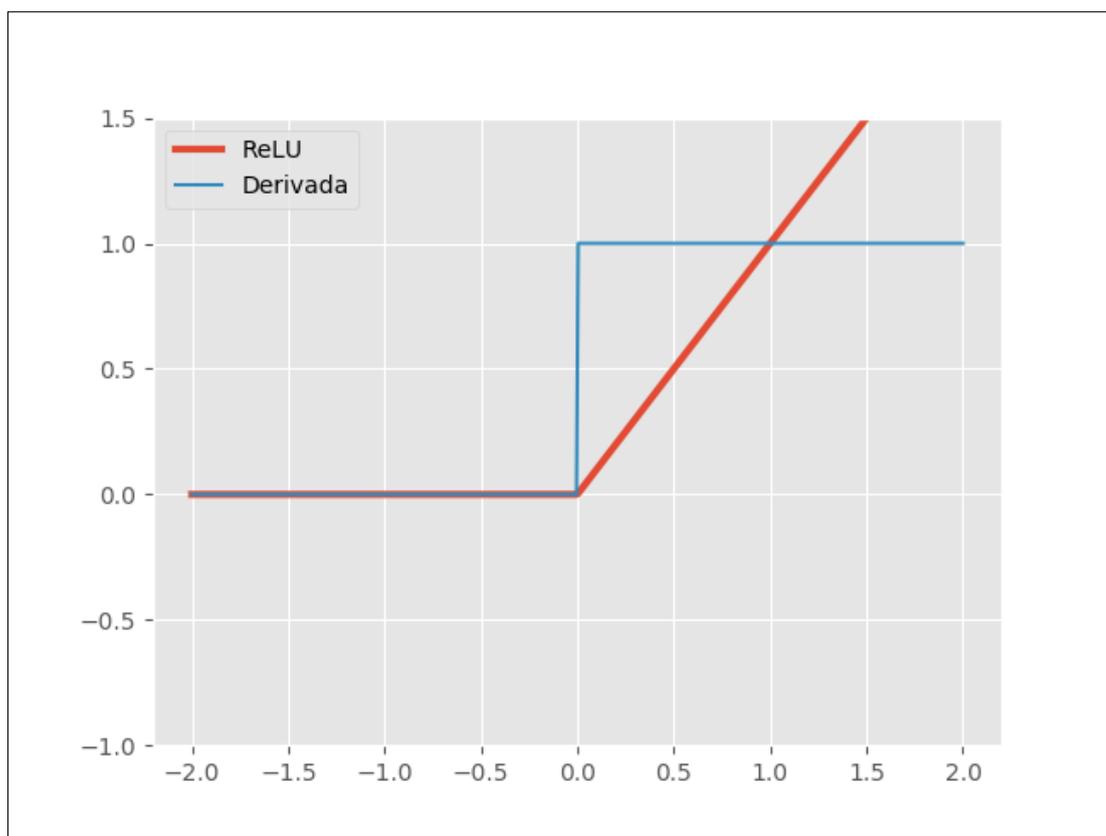


Figura 3.5 – Curva de função relu e sua derivada. Podemos observar o comportamento da convergência pela derivada, por isso a convergência utilizando a função de ativação relu não estagna antes de 1, não tende a 0 em nenhum valor de x maior que 0.

3.2 Aprendizado de Máquina

O tipo particular de aprendizagem praticado [87], é estabelecido pela forma determinada como escolhemos os ajustes realizados nos parâmetros da rede. A compreensão sobre o ajuste do modelo é importante para entender o quão preciso seu modelo está para a aplicação desejada. Esse entendimento irá nortear como podemos determinar se o modelo está subajustado ou sobreajustado [88]. A rede faz subajuste ou como conhecido pelo termo em inglês *underfitting*, quando trabalha de modo insatisfatório nos dados durante a aprendizagem e não consegue capturar a relação entre os exemplos de entrada e os valores de saída. O desempenho insatisfatório pode ocorrer caso o modelo seja muito simples e os dados de entrada utilizados não são suficientemente em quantidade e não são expressivos para descrever a saída [89]. O modelo de rede pode apresentar um comportamento de sobreajuste ou como conhecido pelo termo em inglês *overfitting*, quando trabalha de modo satisfatório durante o aprendizado, mas não consegue generalizar seus resultados diante de novas entradas. O modelo só aprendeu com os dados reconhecidos, mas não consegue fazer a generalização nos exemplos não vistos durante a aprendizagem [90]. Por isso, é necessário definir muito bem o tipo de aprendizagem de acordo com a aplicação e os dados disponíveis. Diferentes métodos de aprendizagem de RNA foram desenvolvidos, esses poderiam ser agrupados em 3 tipos de paradigmas conhecidos como supervisionado, não supervisionado e de reforço [87]. O esquema comparativo entre os paradigmas de aprendizado é apresentado na figura 3.6

No aprendizado supervisionado um agente externo atua como um supervisor, indicando à rede neural a resposta desejada para o padrão de entrada [91]. O processo de aprendizado é baseado em um conjunto de exemplos de estímulos e suas respostas ou alvos, são referentes ao comportamento que deve ser apresentado pela rede neural [91]. Esse conjunto geralmente é separado em dados de treino e validação. O agente externo supervisor é responsável por rotular os dados de entrada com nome, classe pertencente e o alvo, gerando um vetor bidimensional de características de entradas e respostas esperadas [92]. Nesse caso, a RNA é apresentada aos pesos e com isso a rede responde e então verifica o erro que é a diferença entre o alvo e a resposta obtida dos lotes de dados de treino. Com a influência do erro, os pesos da rede são ajustados. O ajuste é executado de forma que aprendizagem é transferida para a rede através do treinamento até que não haja mais aprendizado e com isso os pesos sinápticos não são mais alterados [92]. A rede pode chegar a uma saída com erro zero ou chegar bem próximo a isso, ao processar todo o conjunto de dados. O processo de atuação da rede sobre o conjunto de dados é chamado de interação. Para evitar o sobreajuste, geralmente são utilizados limitadores de número de interações e são realizadas validações periódica do aprendizado durante o processo, utilizando os dados do lote de validação.

O processo de aprendizagem não supervisionado é autônomo. A rede organiza os va-

lores de entrada com base em propriedades estatísticas, definindo uma classificação dos dados de entrada [92, 93]. Nessa forma de treinamento, somente padrões de entrada são apresentados a rede e são definidos pelos padrões correlacionados associados estatisticamente as informações [93] de entrada agrupando conforme a suas similaridades. Suas funções internas procuram padrões sem estímulos exteriores [92]

O aprendizado por reforço é considerado um tipo de aprendizado intermediário, quer dizer entre o aprendizado supervisionado e não supervisionado. Ele é definido pelo critério numérico externo que avalia a resposta do aprendizado fornecida pela RNA em instantes espaçados de tempo, de forma positiva ou negativa sinalizando o comportamento durante uma ação através de interações do tipo tentativa e erro [94]. Tem principais aplicação em jogos e robótica. O ambiente pode ser, por exemplo, uma representação de labirinto. A idéia principal é percorrer o labirinto no melhor caminho possível, com menor número de ações. As ações são os movimentos que levarão em direção a saída. Um sinal de reforço primário, é recebido e convertido do estado atual do ambiente em sinal de melhor qualidade, ensinando o sistema enquanto ele avalia uma sequência temporal das características do sinal, que pode gerar uma saída com maior precisão [94]. Isso quer dizer que a cada dado de entrada a rede recebe uma indicação do estado atual do ambiente e toma uma ação, que no exemplo do labirinto, é escolher um movimento, ponderado pela avaliação em relação ao mapeamento do ambiente, sinalizando durante um momento no tempo pré definido, o tipo de recompensa pela tentativa. Caso a recompensa seja positiva, reforça a ação tomada. Com isso, desenvolve a habilidade de aprender a desempenhar uma tarefa, somente com base nas saídas da sua própria aprendizagem, resultante da interação com o ambiente [94].

Apesar de existirem três principais tipos de aprendizado de máquina, atualmente os métodos de aprendizados supervisionados ganharam destaque na aplicação de reconhecimento de padrões em imagens. Esse tipo de aprendizagem apresenta um fluxo de dados que caminha em dois sentidos.

3.2.1 Fluxo de Dados em Redes Supervisionadas

Para redes neurais em gerais o desafio de aprendizagem está relacionado a determinação de uma combinação de parâmetros, para que uma função se aproxime dos padrões da entrada ao máximo. Geralmente, as RNA possuem o fluxo da rede que se move da camada de entrada x através das camadas ocultas para a camada de saída z conhecido como propagação direta ou adiante ou pelo termo inglês *forward-pass*. Essa propagação é responsável pelo fluxo do sinal funcional, que é chamado dessa forma, porque em cada neurônio da rede pelo qual o sinal passa, o sinal é calculado como uma função representada pelas entradas e pelos pesos associados ao neurônio. Em redes supervisionadas, foco

desse trabalho, a decisão da camada de saída é medida em relação às saídas esperadas y , conforme apresentado no esquema da figura 3.6.

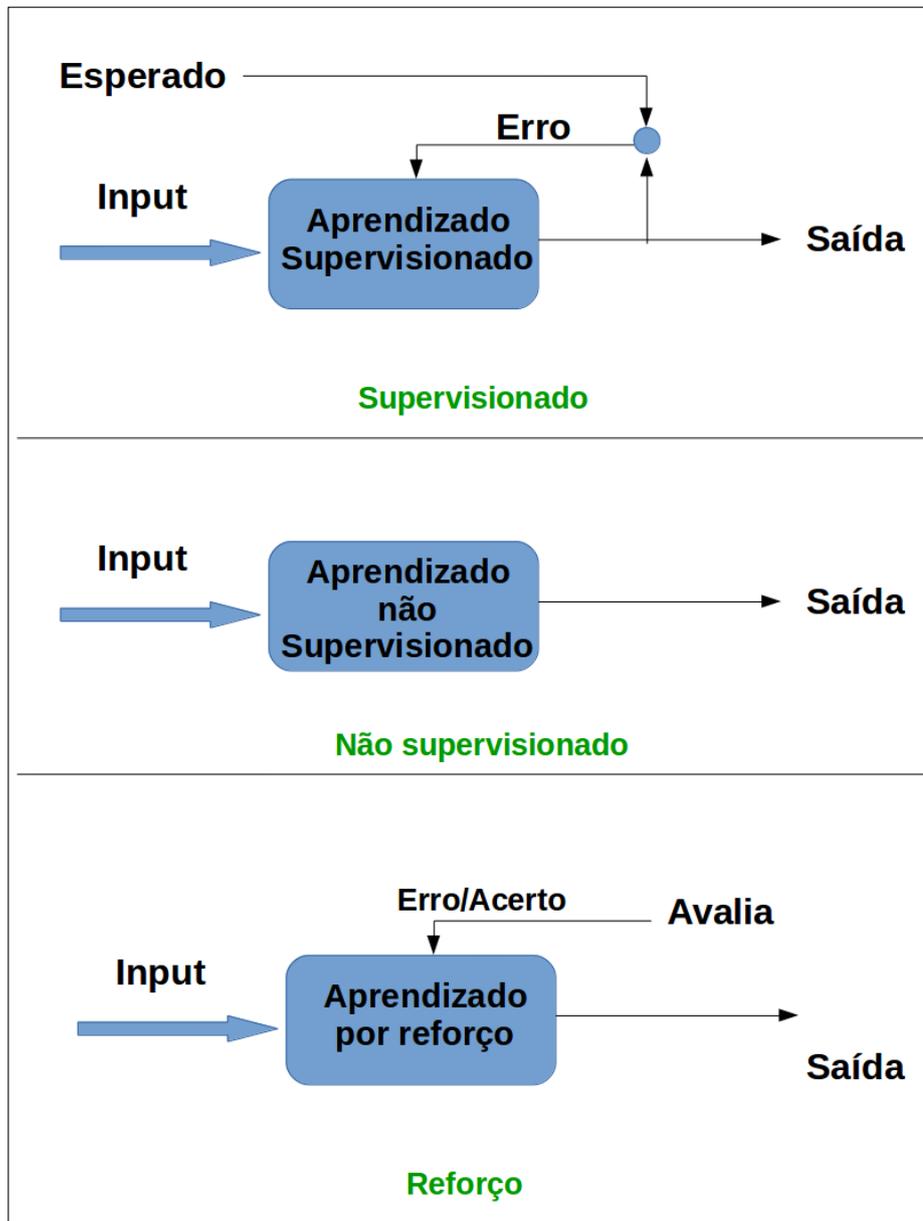


Figura 3.6 – Esquema comparativo dos principais tipos de aprendizagem

Para exemplificação, considerando uma rede com entrada e saída representadas respectivamente por vetores,

$$x = \{x_1, x_2, x_3, \dots, x_m\}, \quad (3.4)$$

$$z = \{z_1, z_2, z_3, \dots, z_n\}, \quad (3.5)$$

onde m e n determinam a dimensão dos vetores, temos um conjunto com p pares de ordenados C_{ea} de entrada e alvo,

$$C_{ea} = \{(x^1, y^1), (x^2, y^2), \dots, (x^q, y^q)\}, \quad (3.6)$$

assim quando iniciamos a rede, os pesos e os vieses iniciais podem ser determinados de maneira aleatória ou não. Dessa forma, haverá dados de entrada de treinamento x^j que após a passagem pelas camadas ocultas produzirá uma saída z^j inicialmente diferente do desejado y^j . Em cada camada oculta são alocados uma quantidade de neurônios, considerando a arquitetura necessária para o tipo de aplicação desejada.

Na retropropagação o que o método de aprendizagem supervisionado [95] da rede neural busca é aproximar os resultados da tabela verdade y^j dos resultados z^j obtidos pela rede. Nessa retropropagação, as derivadas parciais da função conhecida como função custo, calculam o erro em função dos parâmetros pesos [95] e vieses aplicados à entrada x e retropropaga através rede, conforme visto no esquema da figura 3.8. Com esse erro são calculados os gradientes, ao longo do qual os parâmetros, pesos e vieses, podem ser ajustados para minimizar o erro [95]. Por isso, o sinal na retropropagação é conhecido como sinal de erro, pois se propaga de trás para o início da rede camada por camada.

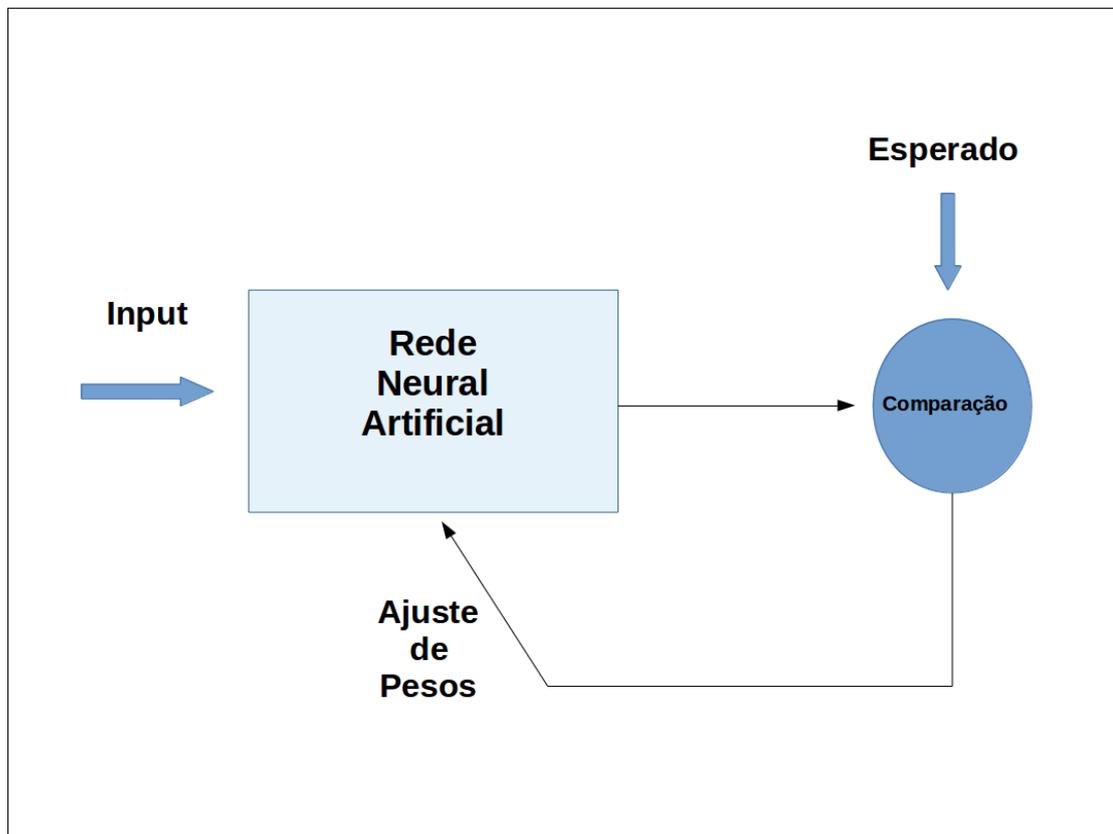


Figura 3.7 – Esquema geral de retropropagação de uma Rede Neural Artificial, dados de entrada de treinamento x^j , após a passagem pelas camadas ocultas produzirá uma saída z^j diferente do desejado y^j . O erro é calculado em função dos parâmetros pesos e vieses aplicados à entrada x e retropropaga através da rede.

A saída da função custo ou conhecida pelo termo em inglês *Loss* (\mathcal{L}) é uma escalar responsável por dizer quão longe o desempenho está da resposta ideal. É calculada no

final da propagação direta, expressa por,

$$\mathcal{L} = \frac{1}{2} \sum_{j=1}^p \| z^j(w, b) - y^j \|^2, \quad (3.7)$$

compara a saída da tabela verdade y com a saída da rede z , conforme esquema da figura 3.8. Como a saída z está em função dos parâmetros w_1, w_2, \dots, w_m e viés b_1, b_2, \dots, b_m utilizados durante a propagação direta, retorna durante a retropropagação os valores de sua derivada, i. e. gradiente

$$\nabla \mathcal{L}_m = \left(\frac{\partial \mathcal{L}(w_{m-1}, b_{m-1})}{\partial w_m} \right), \quad (3.8)$$

que tem objetivo de otimizar os resultados das camadas, construídas com as entradas ponderada com os parâmetros, assim controla o aprendizado através dos pesos e dos vieses. Os novos parâmetros são atualizados durante a retropropagação. O gradiente calculado com a derivada da função Custo L num ponto, indica a taxa de variação dessa função em relação ao argumento w_m da própria função, assim,

$$\Delta w = w_m - w_{m-1}, \quad (3.9)$$

e

$$\Delta w = -\eta \frac{\partial \mathcal{L}_m}{\partial w_m}, \quad (3.10)$$

onde a derivada da função Custo L representa o coeficiente angular da reta tangente ao gráfico em cada ponto onde a deriva exista. Como é desejado que o custo diminua, a busca é feita pelo argumento mínimo da função L , conforme apresentado exemplo no gráfico característico com apenas um mínimo local e em função apenas de w na figura 3.9. A taxa de variação η , conhecida como taxa de aprendizagem em RNA, geralmente é pré definida, e altera o comportamento da função, representando o tamanho do passo dado em direção ao argumento mínimo de L . A escolha da taxa de aprendizado deve ser realizada de forma criteriosa, pois com passos muito próximos à zero, há uma grande possibilidade de chegar ao ponto de mínimo da função, porém talvez sejam necessárias mais iterações, tornando o algoritmo lento. Para um η com valor grande o passo poderá pular o ponto de mínimo e chegará em um ponto na função custo mais alto do que o de onde começou. Em geral, passos pequenos são utilizados para evitar situações onde o ocorra o salto do ponto onde o ponto mínimo pode estar, conforme exemplificado na figura 3.10. Utilizando esse método de otimização, de gradiente descendente, a rede continua até que o erro não possa mais ser reduzido atingindo um estado de convergência.

As RNA além de apresentarem um tipo de método de aprendizagem distinto e de otimização, também podem possuir características diferenciadas quanto a sua arquitetura.

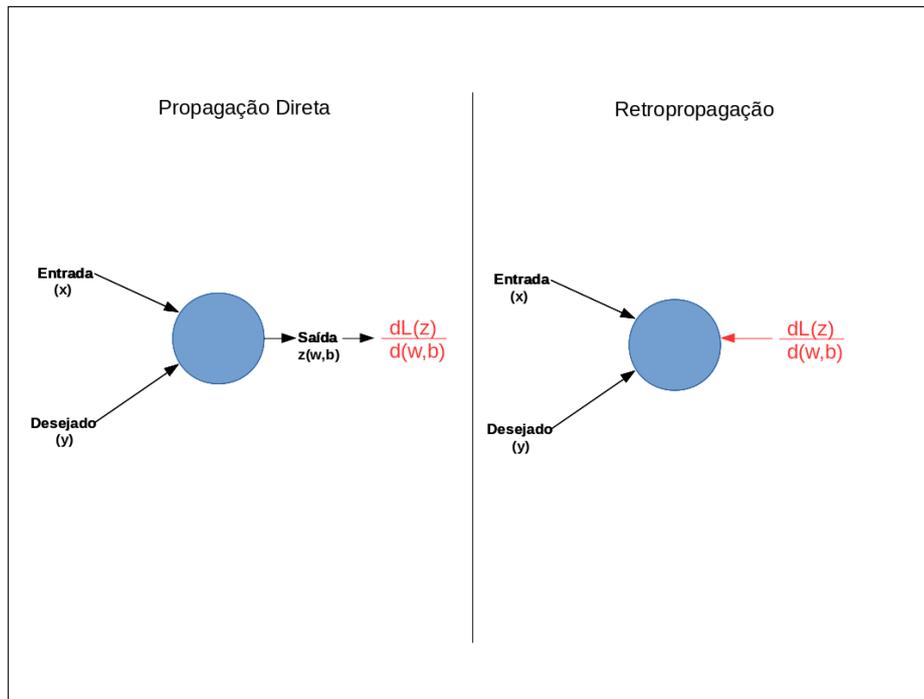


Figura 3.8 – Esquema função custo durante o processo de propagação adiante ou direta e retropropagação da rede. Com esse erro são calculados os gradientes, ao longo do qual os parâmetros, pesos e vieses, podem ser ajustados para minimizar o erro.

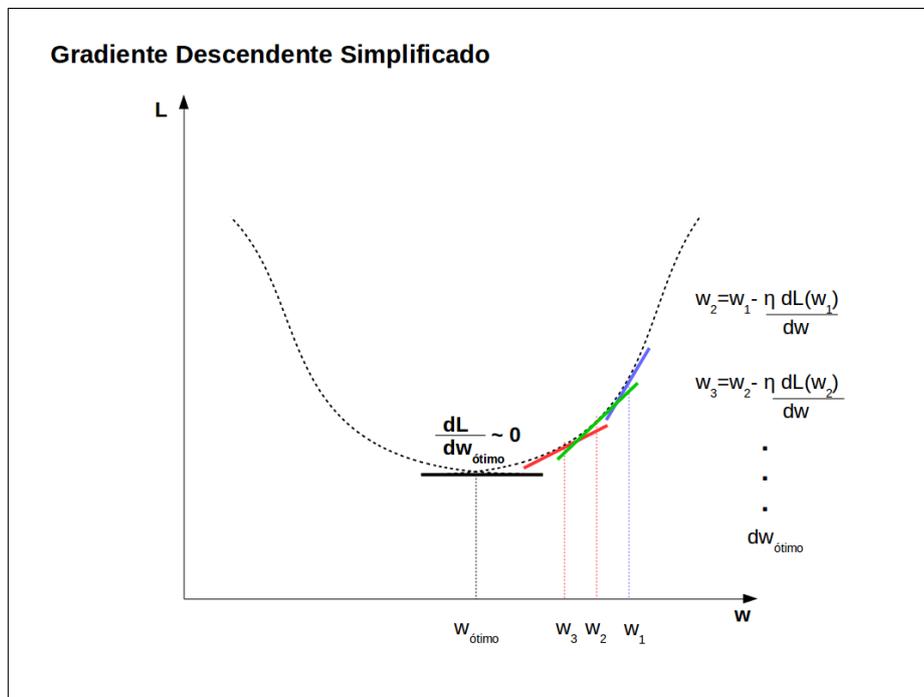


Figura 3.9 – Gráfico característico simplificado da função custo em função apenas dos pesos. O conceito é baseado na busca do argumento mínimo da função, que nesse caso é a função custo L , limitado pela taxa de variação η , que determina o passo da descida em direção ao mínimo.

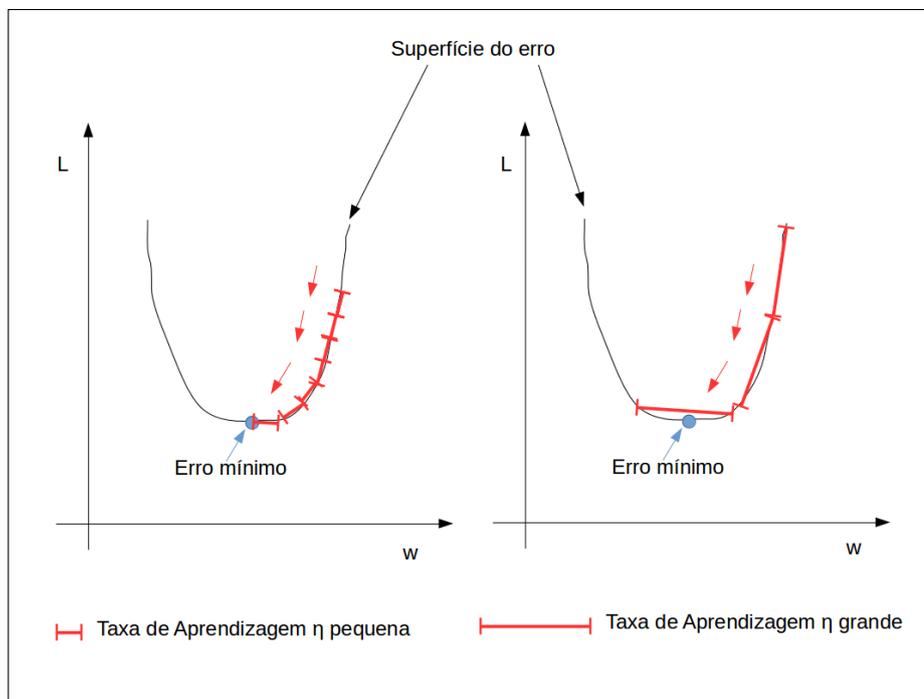


Figura 3.10 – Esquema exemplo comparativo de taxas de aprendizagem, com variação de η pequena e grande, indicando que pequenas taxa tem uma chama maior de não saltar o erro mínimo.

3.3 Tipos de Arquiteturas RNA

As arquiteturas de RNA podem ser definidas por três tipos principais de configurações de interconexão de neurônios. Além disso, cada arquitetura pode variar com o número de camadas ocultas, número de neurônios por camada e tipos de função de ativação. Redes Neurais de Alimentação Direta, Redes Neurais Recorrentes e Redes Neurais Reticuladas são os tipos mais comuns de configurações de alimentação de camadas relacionadas à disposição dos neurônios e o padrão de conexão entre elas. Essas redes possuem diferentes métodos para definir a convergência do aprendizado. Na figura 3.11 apresentamos um esquema comparativo das principais tipos de configurações de arquiteturas detalhadas a seguir.

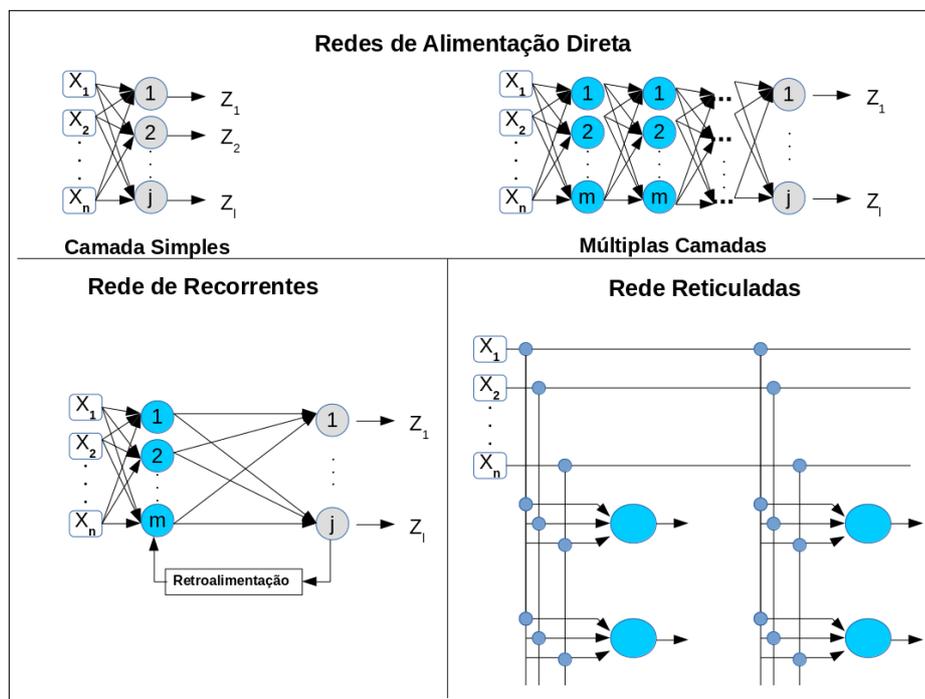


Figura 3.11 – Esquema comparativo dos principais tipos de interconexão de neurônios. Os retângulos brancos X_1 até X_n , representam os neurônios da camada de entrada, já os m círculos azuis representam os neurônios das camadas ocultas e os j círculos cinzas representam os neurônios da camada de saída, onde Z_1 a Z_l representam os resultados da rede.

Redes neurais de alimentação direta ou acíclicas possuem uma arquitetura que pode utilizar arranjo de camada simples e de múltiplas camadas. Os dois tipos propagam a informação em um único sentido. Arranjos de camada simples são bastante usadas em classificação de padrões e filtragem linear [96, 97, 98, 99]. Configurações com múltiplas camadas são mais utilizadas na aproximação de funções não lineares, classificação de padrões, identificação de sistemas, otimização, robótica, controle de processos e filtragem não linear [98, 100, 101, 102].

As redes de alimentação direta de camada simples são arquiteturas que possuem apenas

a camada de entrada, os neurônios e a camada de saída. A primeira é composta pelos estímulos de entrada, e a segunda é referente ao neurônio processador, responsável pela transmissão do estímulo da entrada para a camada de saída. Essa arquitetura é capaz de resolver problemas linearmente separáveis. Essas redes são conhecidas como redes Perceptrons ou redes Adalines, criadas em 1958 [103] composta de apenas um neurônio, utilizado para classificar dados em duas classes linearmente separáveis um exemplo dessa aplicação foi apresentado na seção 3.1.1, onde os vieses tem papel importante na inclinação da função.

As redes alimentadas diretamente de múltiplas camadas são versões mais complexas das RNAs de camada simples, nessa arquitetura, possuem além da camada de entradas e de saídas, as camadas de neurônios ocultas, também conhecidas como camadas intermediárias de neurônios. O fluxo de informações segue das entradas passando pelas camadas ocultas seguindo para a camada de saídas. Permitem obter soluções aproximadas para problemas de classificadores [98, 100]. Os neurônios dessa arquitetura não fazem conexão com outros neurônios da mesma camada, mas são conectados a todos os neurônios da camada seguinte. São redes conhecidas como Perceptrons de Múltiplas Camadas ou pelo termo em inglês *Multilayer Perceptrons* foram muito populares em 1980 e recentemente voltaram a ganhar destaque principalmente na aplicação de detecção de objetos e pessoas em imagens [98, 100], e por isso é a configuração que iremos focar nesse trabalho. Contudo iremos introduzir os conceitos de outros tipos de configurações existentes como as redes recorrentes e competitivas.

As redes neurais recorrentes ou realimentadas são recomendáveis em aplicações de problemas de séries temporais para controle de processos [104]. A saída de alguns neurônios alimentam neurônios da mesma camada, inclusive ele mesmo, ou neurônios de camadas anteriores, conforme apresentado no esquema 3.11. O dado percorre a rede em duas direções, utiliza memória dinâmica e dispõe de capacidade de representação em processos dinâmicos [104, 105]. Os dados são transmitidos de neurônio para neurônio gerando uma memória internas, auxiliando no aprendizado sequencial, possibilitando um processamento dinâmico de informações [106]. Algumas redes conhecidas são a Perceptrons de Múltiplas Camadas com retroalimentação, onde nos algoritmos de aprendizagem são utilizadas funções de minimização na retroalimentação.

As redes neurais reticuladas são ditas competitivas pois, a cada apresentação de um padrão de entrada, os neurônios competem pela classificação desse padrão, e o neurônio mais ativo, passa a representar a classe a qual pertence o padrão de entrada. Assim, os pesos das conexões são modificados de acordo com essa classificação, dessa forma, são ajustados os pesos para que as entradas similares ativem o mesmo neurônio [107]. Todos os neurônios recebem informações iguais na entrada, utilizando os pesos, os neurônios da rede computam os seus respectivos valores em resposta a uma função discriminante de padrões. Esta função define as bases para a competição entre os neurônios, declarando

o neurônio vencedor, o qual mais se aproxima do peso sináptico. A posição do neurônio vencedor nessa estrutura da rede interfere nos novos valores dos pesos [108]. A estrutura dessa rede é representada conforme o esquema da figura 3.11 e é chamado de mapa bidimensional de neurônios. Esse mapa representa uma estrutura progressiva, configura uma única camada computacional, ou mapa topologicamente ordenado, consistindo de neurônios arranjados em linhas e colunas. Cada neurônio nesse mapa é conectado a todos os nós, representados pelos pequenos círculos azul, relacionados a cada camada de entrada [109]. Este tipo de rede, por exemplo, em aplicações tais como problemas de agrupamento de dados ou do termo em inglês *clustering*, utilizando, em treinamento não supervisionado. Para essa aplicação, classifica padrões apresentados em sua entrada a partir das similaridades encontradas entre estes padrões, definindo automaticamente classes que agregam padrões similares [110]. Um exemplo desse tipo de rede é conhecida como Rede de Kohonen [107, 110] que utiliza algoritmo de aprendizagem competitiva devido o processo de competição entre os neurônios.

Os conceitos básicos de uma RNA, os tipos de aprendizados e principais configuração camadas de neurônios, foram incorporando novas redes é preciso compreender a relação do número de camadas ocultas em uma arquitetura de rede.

Muitas topologias de redes neurais foram construídas, treinadas e testadas para uma determinada tarefa, uma rede neural é especificada, principalmente pela sua topologia.

3.4 Topologias das Redes Neurais

A topologia de uma rede neural se refere às diferentes composições estruturais possíveis, com diferentes quantidades de neurônios nas camadas de entrada, intermediária e de saída da rede [111]. De maneira intuitiva poderíamos pensar que um número muito grande de camadas ocultas em uma rede neural artificial, poderia obter sempre um melhor desempenho comparado à redes com poucas camadas ocultas. Contudo, a quantidade de camadas ocultas utilizadas, o número de neurônios, assim como o método de aprendizagem e otimização, dependem do tipo de tarefa desejada e da capacidade de processamento computacional [111]. Para a maioria dos problemas, uma camada intermediária é suficiente, pois consegue calcular uma função arbitrária qualquer a partir de dados fornecidos [80]. Para isso, considerando n número de neurônios da camada de entrada, o número de neurônios da camada intermediária precisa ter em torno de $2(n+1)$ [111]. Entretanto, há casos onde uma camada escondida adicional pode ajudar e dependendo da configuração da rede, é possível aproximar qualquer função [80, 112]. Nesse caso, considerando m neurônios de saída a primeira camada intermediária precisa de aproximadamente $m(n+1)$ neurônios e a segunda camada intermediária precisa ter número próximo de $2m$. Em geral, as redes com até duas camadas ocultas alimentadas com um fluxo de dados pela rede em vetores, são conhecidas tipicamente como arquiteturas rasas, ou do termo em inglês *shallow*

architectures.

Redes caracterizadas por apresentarem arquiteturas rasas, representadas pelo esquema da figura 3.12, são algoritmos destinados a realizar a classificações binárias, ou seja, prevendo se o dado de entrada pertence ou não à uma determinada categoria de interesse. Essas redes possuem até duas camadas ocultas, utilizando em seu algoritmo um método de aprendizagem específico para cada aplicação [113].

No entanto, embora as arquiteturas rasas, tenham a capacidade de modelar e armazenar informações, são limitadas pelo seu pequeno número de camadas, o que dificulta resolver problemas que envolvam comportamentos não-lineares de classificação de padrões em vários etapas utilizando operações matemática entre blocos de dados. Apesar de não ser possível afirmar que arquiteturas com mais camadas sejam mais eficientes que as rasas em todos os problemas [113, 114], existem evidências que redes com mais camadas ocultas podem ser mais eficientes, quando a tarefa é extrair padrões que exigem vários níveis de reconhecimento e necessitam de dados de entradas suficiente para extrair esses padrões [115]. Além disso, as arquitetura com maior número de camadas podem aprender com menor envolvimento humano durante a construção do modelo de aprendizagem, pois podem integrar as etapas de extrair características e classificar padrões, de um número variado de classes em um único modelo [115]. Essas arquiteturas são conhecidas como profundas ou pelo termo em inglês *deep architectures*, apresentando diferenças da arquiteturas rasas conforme visto em 3.13.

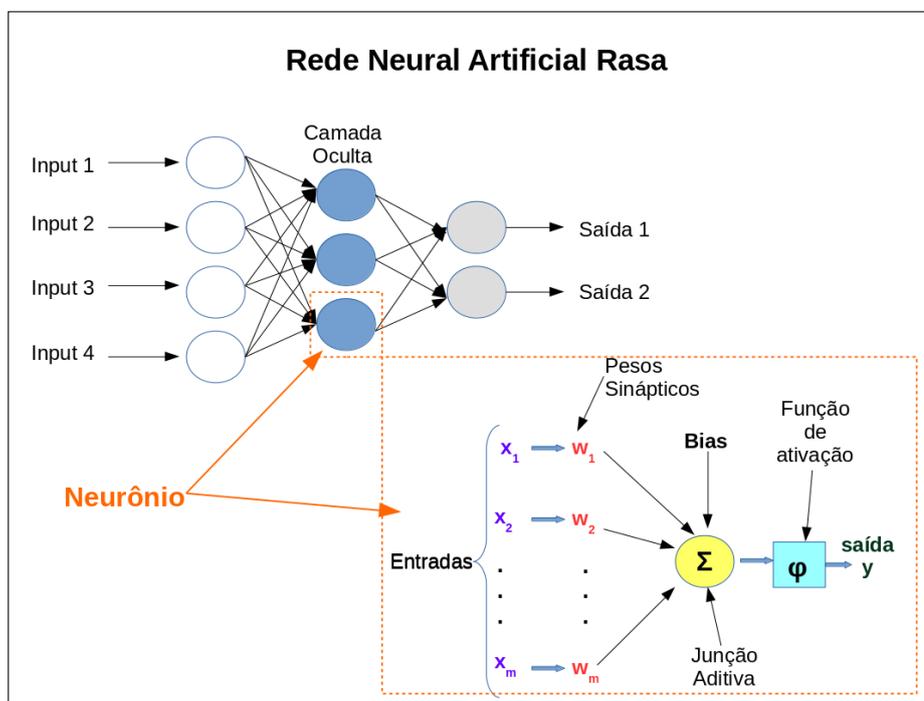


Figura 3.12 – Esquema rede neural artificial rasa com uma camada oculta. Apresentamos relação entre um único neurônio artificial da camada oculta com sua representação.

As principais aplicações das arquiteturas profundas, estão em geral, voltadas para o

reconhecimento de padrões em vídeos, sons e imagens que exigem modelos matemáticos mais sofisticados, redes que processem os dados em profundos blocos de matrizes, um número maior de camadas ocultas e treinamento que envolvam uma quantidade maior de parâmetros. O treinamento dessas redes profundas em geral, utilizam os métodos de aprendizagem supervisionado ou não supervisionado. Contudo, como o foco desse trabalho são redes supervisionadas, nas seções seguintes, iremos concentrar nesse tipo de aprendizado de máquina voltado para reconhecimento de padrões em imagens.

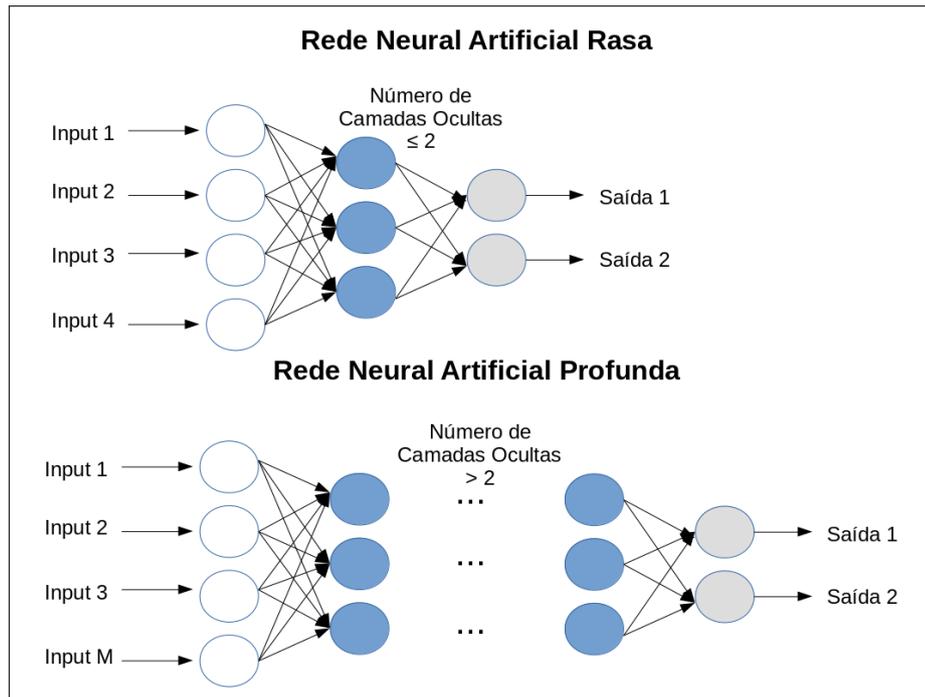


Figura 3.13 – Esquema comparativo entre redes neurais rasa e redes neurais profundas. Redes profundas possuem um número de camadas acima de 2.

3.4.1 Aprendizagem Profunda

A aprendizagem profunda ou conhecida pelo termo em inglês *Deep Learning* é utilizado para caracterizar o aprendizado de máquina de redes com arquiteturas profundas. Os métodos que utilizam aprendizagem supervisionada em arquiteturas rasas para classificação de padrões em imagens, buscam descobrir um modelo de predição utilizando um conjunto de dados de exemplos para treinamento [116]. Em geral, as entradas são vetores de atributos da imagem. O método supervisionado é utilizado para guiar o aprendizado do modelo a partir desses exemplos. Ao final do processo de aprendizagem obtém uma função capaz de receber por entrada outros dados e fornecer como saída, uma representação adequada da entrada [117]. No entanto, utilizando a aprendizagem supervisionada em arquiteturas profundas é possível utilizar bloco de imagens de entrada da rede, eliminando a necessidade de uma pré extração e seleção de características da imagem [20].

Essa abordagem tem recentes resultados eficazes na resolução de processos automatizados, superando em alguns aspectos o desempenho da interação humana em determinadas áreas [118]. Tarefa como o reconhecimento de características e padrões em grandes volumes de imagens consumia muito tempo e estava sujeito a interpretação de cada profissional. Porém, há necessidade de um grande número de dados para treinamento [20, 22].

A aprendizagem profunda AP é um método de AM que utiliza RNA para aprender características, onde cada camada da rede aprende um nível de representação das características dos dados de forma hierarquizadas [20, 119]. As representações são compostas de níveis de características altos abstratos e não lineares e esses níveis altos são formados de subníveis mais baixos de características mais básicas [120, 121].

Essas características formam mapas, que são definidos entre as camadas da rede de forma automática, representando os níveis hierárquicos das características dos dados [78, 122]. Os mapas de características e a estrutura das informações são estabelecidos através dos dados de aprendizagem analisados com o mínimo possível de intervenção do operador [123, 124].

O aprendizado profundo quando aplicado na detecção de um objeto de uma determinada classe em uma imagem, utiliza a rede para aprender com imagens digitalizadas diversas, contendo vários objetos pertencentes à mesma classe [125, 126, 127]. Durante o aprendizado supervisionado se a rede não distinguir precisamente o padrão característico, os parâmetros são ajustado pelo algoritmo. A finalidade da etapa de aprendizagem é capacitar a rede para que identifique de forma consistente características comuns em um lote de imagens [128]. Esse tipo de rede é composto de diversas camadas de neurônios virtuais diferentes, arquitetada através de recursos computacionais e matemáticos. A primeira camada desse tipo de rede aprende características como as bordas por exemplo [129, 130]. Uma forma usual de fazer isso é utilizar filtros de bordas que identifiquem essas características [131]. A rede, dessa forma, consegue aprender à reconhecer atributos mais complexos associando os conhecimentos das camadas iniciais, como uma região específicas na imagem. O procedimento segue camada após camada, até que a rede reconheça com convicção o objeto.

3.4.2 Redes Neurais Convolucionais

As redes neurais convolucionais conhecidas como CNN da sigla em inglês *Convolutional Neural Network* [132], são baseadas no sistema biológico, relacionado ao córtex visual. O processo começa quando um estímulo visual alcança à retina e equivale a um sinal que atravessa regiões específicas no cérebro. Essas regiões são responsáveis pelo reconhecimento de cada uma dessas características correspondentes [132, 133]. Os neurônios biológicos das primeiras regiões respondem pela identificação de formatos geométricos primários, enquanto neurônios das camadas finais têm a função de detectar características mais complexas, formadas pelas formas simples anteriormente reconhecidas [133, 134]. As

características como padrões muito específicos do objeto, são estabelecidas depois que o procedimento se repete. De forma análoga, a CNN decompõe a tarefa de reconhecimento de um objeto em subtarefas. Para isso, divide em subníveis a representação de características na aprendizagem e reconhecer posteriormente novas amostras de mesma classe [20, 134]. Como as CNNs são formadas por camadas sequenciadas cada uma destas possui uma função específica na propagação do dado de entrada. Essas redes podem ser construídas por diferentes arquiteturas conhecidas pelo termo *Convnet*, elaboradas de acordo com a aplicação. A primeira arquitetura CNN foi criada por Yan Lecun [135] em 1998, treinada com retropropagação, para reconhecimento de caracteres de caligrafia, como interpretador de códigos postais. A figura 3.14 apresenta a arquitetura geral de uma CNN e seus três principais tipos de camadas [136, 137], conhecidas como convolucionais + FA, subamostragem e totalmente conectadas. As camadas convolucionais + FA são responsáveis por extrair particularidades dos volumes de entradas [22, 138]. As camadas de subamostragem reduzem a dimensionalidade do volume após as camadas convolucionais e tornam a representação do padrão invariante a pequenas mudanças na entrada [20, 139]. As camadas totalmente conectadas recebem vetores e propagam padrões por meio da multiplicação ponto a ponto [23, 140]. Na aplicação de classificação de imagem, o resultado da camada de saída de uma CNN é uma nota entre 0 e 1, referente a imagem de entrada pertencer a uma das classes as quais a rede aprendeu a identificar [21]. As principais vantagens da CNN são a invariância a escala e a necessidade mínima de pré-processamento, tornando o processo mais automatizado possível [141, 142].

3.4.2.1 Camada convolucional + Função de ativação

Cada camada de convolução possui um conjunto de parâmetros conhecidos como filtros ou do termo em inglês *kernels*. A saída de uma camada de convolução conhecida como mapa de características é representada por neurônios e é resultado de uma operação entre uma região da imagem I_i de entrada e um filtro qualquer W^k . Como em uma rede convolucional a posição do pixel da imagem é importante, pois é baseada em um subconjunto das dimensões da entrada, o filtro varre a imagem como uma janela deslizante [132, 143]. Uma função de ativação *FA* pré definida é calculada para os resultados das operações com o mesmo filtro em diferentes regiões da imagem. Por isso, cada neurônio de saída, é definido pelo resultado da função de ativação em relação ao cálculo de produto escalar entre I_i e um dos filtros utilizados do conjunto W_k [144] e os vieses b_k representado por:

$$MC = FA(W^k * I_i + b^k), \quad (3.11)$$

onde o conjunto de todas as saídas formando um mapa de característica MC referentes a um filtro W_k é conhecido pelo termo em inglês *feature map k*. Se uma entrada I_i de dimensões $d1 \times d1 \times d3$ o filtro W_k terá dimensões $d1' \times d1' \times d3$ e o resultado computado apresentará dimensões $d1 \times d1 \times d3$ [145]. Os filtros são parâmetros ajustados durante

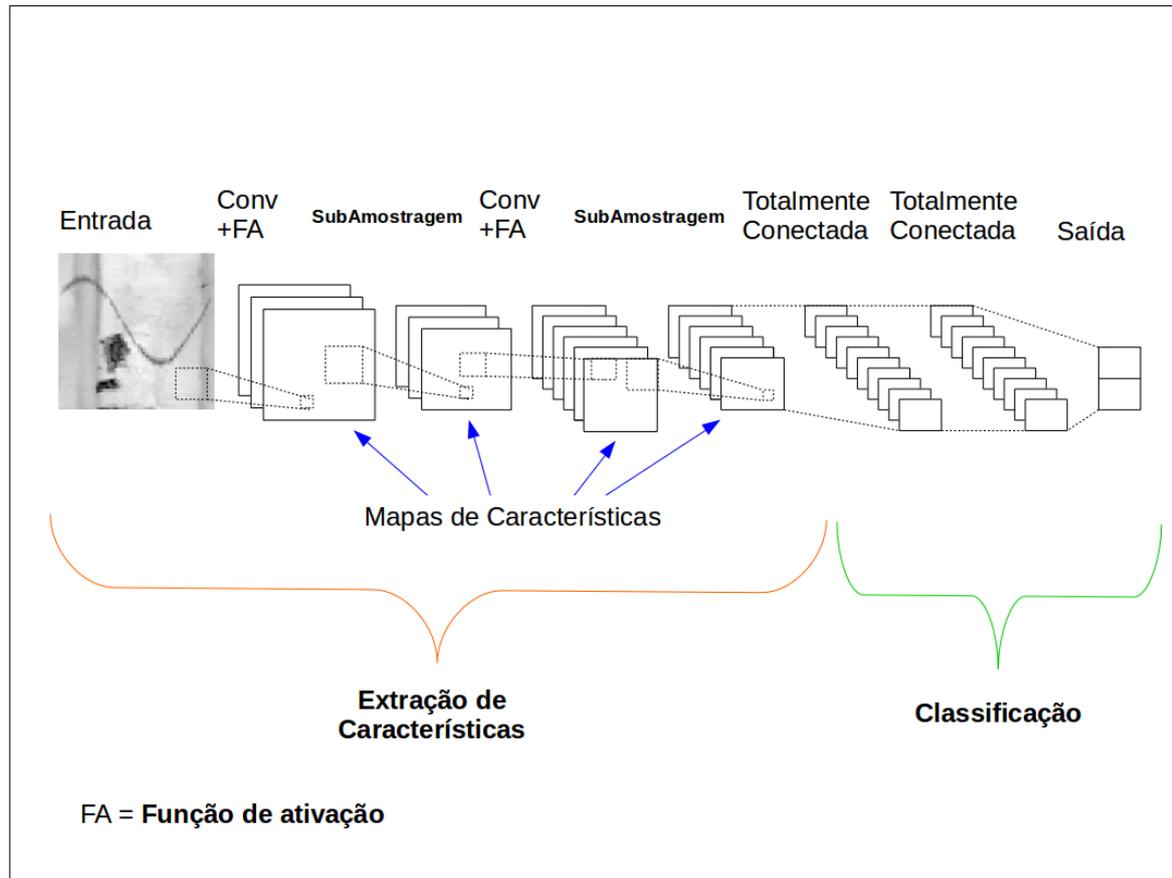


Figura 3.14 – Esquema exemplo de arquitetura de rede neural convolucional (CNN) adaptado de [142]. Os exemplos de mapas de características são referentes as principais camadas da CNN. A imagem de entrada é dividida em campos que alimentam uma camada convolucional que, em seguida, extrai os recursos da imagem de entrada. A próxima camada é a de subamostragem, que reduz a dimensionalidade dos recursos extraídos, pois retém as informações mais importantes. A saída da rede alimenta um perceptron multicamada totalmente conectada. Essas redes, geralmente aprendem usando retropropagação. Nesse exemplo, a rede busca duas classes, logo duas notas são apresentadas na saída.

o treinamento, que aprende quais conseguiram estabelecer relações de vizinhança entre pixels, funcionam como um extrator de características específicas de imagens [132]. Cada imagem após o filtro é um mapa de características. Em uma CNN, cada camada oculta é composta de diversos mapas de características, onde cada unidade desse mapa é resultado da mesma operação utilizando o mesmos parâmetros de filtro e viés depois de treinada, em uma região sobre a imagem de entrada. Essas unidades dentro de um mapa de característica são interpretados como detectores de uma mesma característica, mas cada uma delas está conectada a uma região diferente da imagem de entrada [146]. Apresentamos na figura 3.15 um exemplo onde o filtro consegue destacar os pixels das bordas, através da operação de convolução. A CNN soma um viés e aplica a função de ativação, com por exemplo a Relu que computa como zero os pixels com valores negativo [147].

Alguns parâmetros predefinidos controlam as operações realizadas nas camadas con-

volucionais. A profundidade é o primeiro parâmetro que define o volume resultante é referente ao número de filtros utilizados, essa etapa exige capacidade computacional, que demanda tempo e maior capacidade de memória da máquina [148]. A altura do volume resultante o segundo parâmetro que depende do salto ou passo durante a operação de convolução. Por exemplo, o passo igual a 1, fará com que o filtro deslize somente uma posição a cada vez. No entanto sendo o passo igual a 2, o filtro desliza e pula à cada duas posições [149]. O tamanho do passo definirá a altura e largura do volume resultante. Contudo se o salto for muito grande características essenciais desapareceriam, dessa forma valores maiores que 2 não são comuns. O terceiro parâmetro é utilizar ou não preenchimento com zeros ou conhecido pelo termo em inglês *zero-padding*. Ao definir esse preenchimento, realiza uma operação que adiciona zeros nas bordas do volume do dado de entrada [150]. Dessa forma, dependendo do tamanho do passo é possível manter a altura e largura do volume resultante com valores iguais do volume de entrada conforme podemos observar no esquema da figura 3.16. Nesse esquema, o passo definido para a realizar a operação de convolução foi igual a 1. No esquema completo apresentado na figura 3.17, apresentamos a convolução entre as matrizes de entrada e os filtros, somados aos vieses. As matrizes à direita representam os canais da imagem de entrada com dimensões $5 \times 5 \times 3$, onde foram adicionados os preenchimentos com zeros, alterando suas dimensões para $7 \times 7 \times 3$, para manter as saídas com a mesma dimensão da entrada, após a convolução com passo 1.

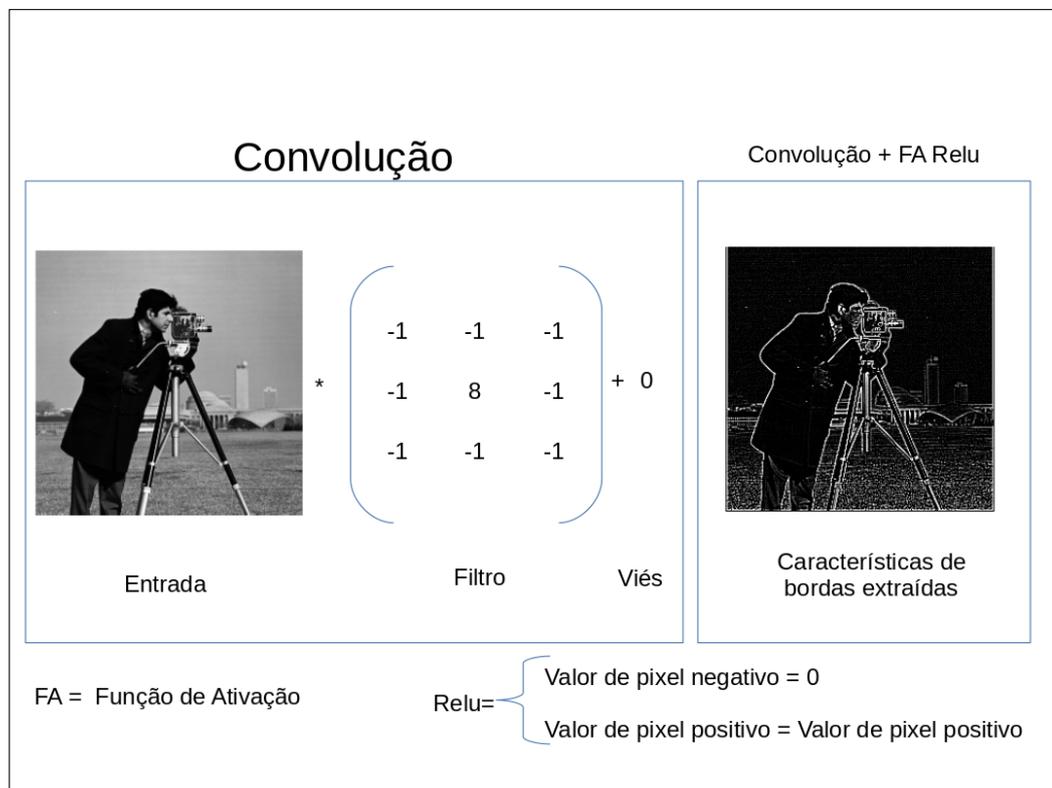


Figura 3.15 – Esquema exemplo de extração de características utilizando convolução. No exemplo podemos identificar o filtro, o viés e a função de ativação, utilizados para extrair características da imagem destacando os pixels das bordas.

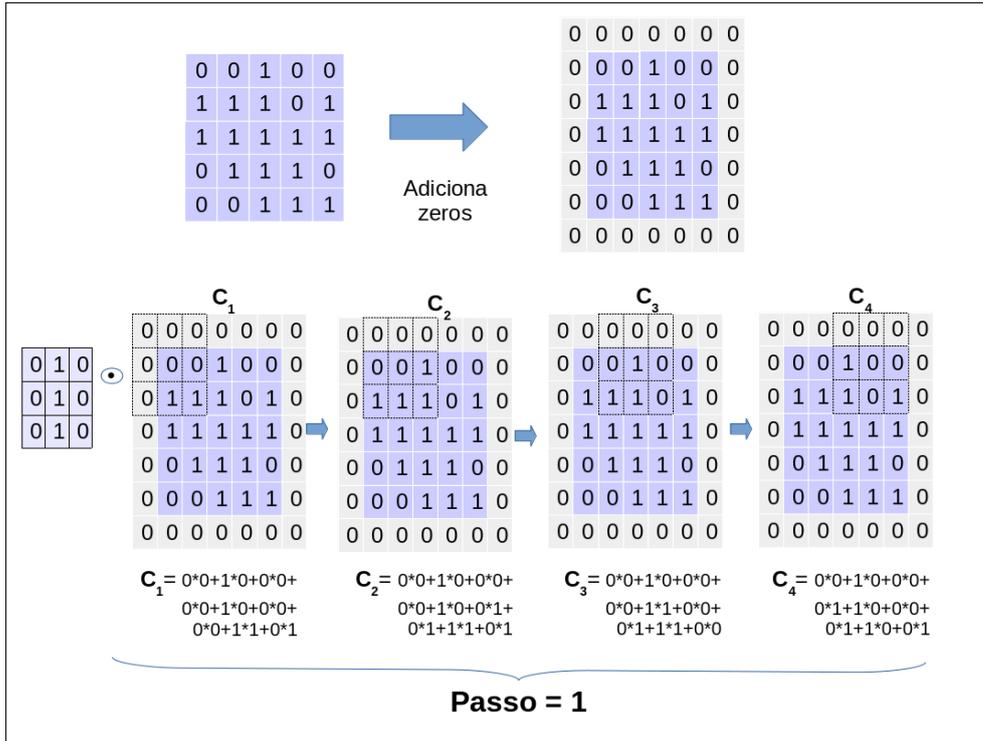


Figura 3.16 – Esquema de preenchimento de zeros e detalhes de operação de convolução usando filtro deslizante, onde cada C é referente ao cálculo de cada etapa da convolução sem o somatório do viés, para fins de simplificação na apresentação.

3.4.2.2 Camadas de Subamostragem

As camadas de subamostragem ou do termo em inglês *max pooling* são responsáveis por reduzir a dimensionalidade do volume resultante após as camadas convolucionais e ajudam a tornar a representação invariável a pequenas mudanças na entrada MC, em uma operação conhecida como subamostragem [132, 138]. Isso significa maior possibilidade de encontrar padrões mesmo em posições variantes. Em processamento de imagens, a subamostragem de uma imagem envolve reduzir a sua resolução, sem no entanto alterar significativamente o seu aspecto [132, 151]. No contexto de uma CNN, a subamostragem reduz a dimensionalidade de um mapa de característica fornecido como entrada e produz outro mapa de característica, uma espécie de síntese do primeiro [139]. A forma mais comum de subamostragem consiste em substituir os valores de uma região pelo valor máximo, como mostra 3.18, onde uma entrada 4×4 utilizando um filtro 2×2 , reduziu o tamanho da saída, conseqüentemente reduziu o processamento e auxiliou nos casos de invariâncias locais. Essa operação é conhecida como máxima subamostragem ou pelo termo em inglês *max pooling* e é útil para eliminar valores desprezíveis, reduzindo a dimensão da representação dos dados e acelerando a tempo de processamento necessário para as próximas camadas, além de gerar uma invariância a pequenas mudanças e distorções locais[138]. Outras funções de subamostragem também podem utilizar a média, a norma e a média ponderada [151].

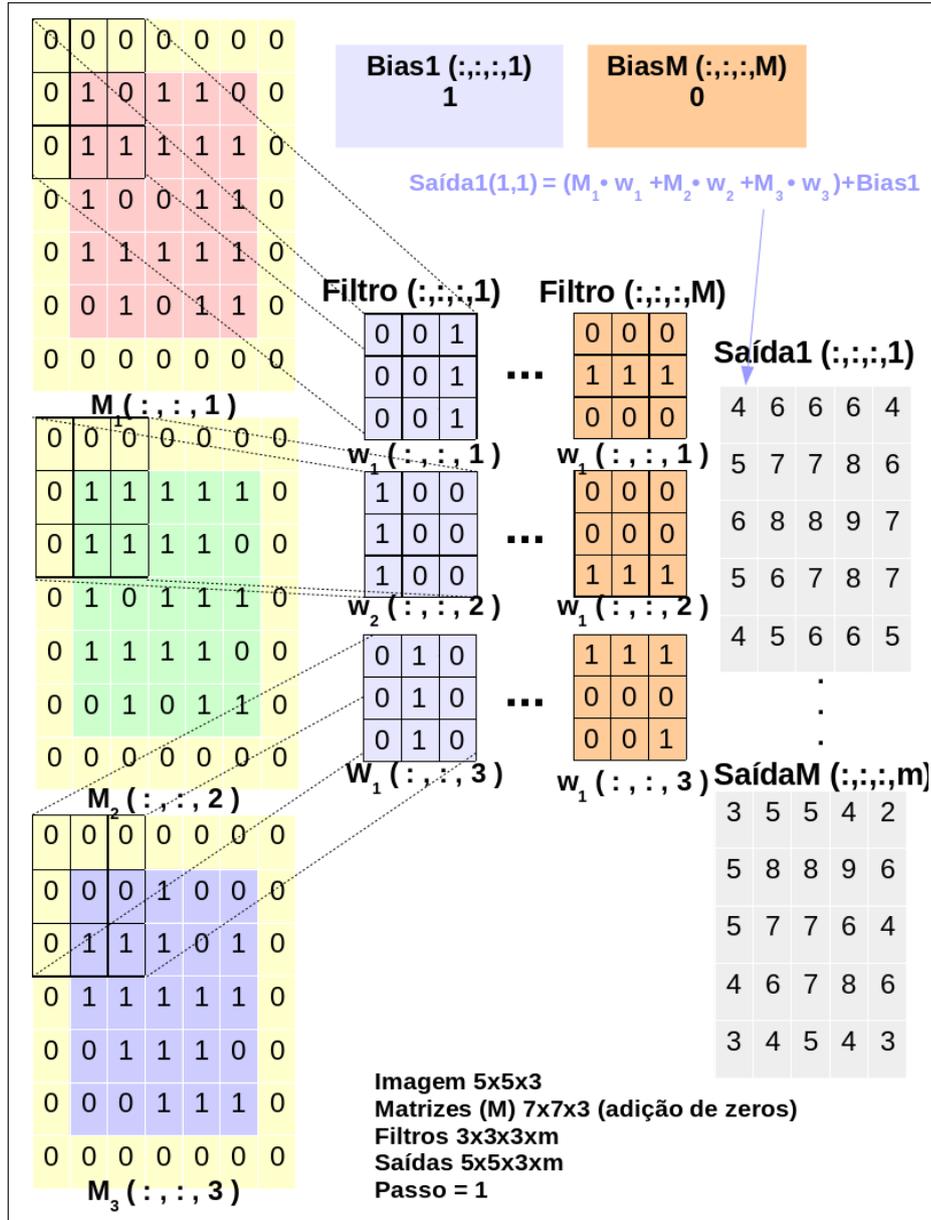


Figura 3.17 – Convolução entre as matrizes de entrada e de filtros, somados ao bias. Esquema adaptado de [133]. As matrizes à direita representam os canais da imagem de entrada com dimensões $5 \times 5 \times 3$, onde foram adicionados os preenchimentos com zeros, alterando para $7 \times 7 \times 3$, para manter a saída com a mesma dimensão da imagem, após a convolução.

3.4.2.3 Camadas totalmente conectadas

A saída das camadas convolucionais e de subamostragem representam as características extraídas da imagem de entrada [132, 140]. O objetivo das camadas totalmente conectadas FL do termo em inglês *fully connected layer* é utilizar essas características para realizar regressão de algum parâmetro ou classificar no final a imagem em uma classe pré-determinada, como na figura 3.14. As camadas totalmente conectadas são as mesmas que as de uma rede neural artificial convencional que usa um função de ativação generalizada, tal como a função sigmóide conhecida pelo termo em inglês *softmax* na última camada

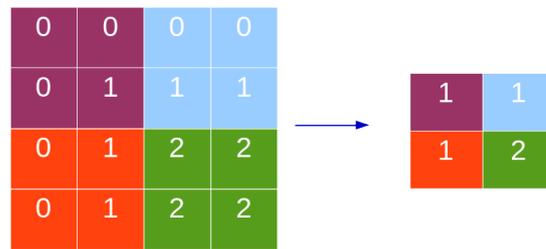


Figura 3.18 – Esquema camada de pooling ou subamostragem. Apresentamos a subamostragem em uma imagem 4x4 utilizando um filtro 2x2. Além de reduzir o tamanho da imagem, consequentemente reduz o processamento para as próximas camadas, essa técnica também auxilia no tratamento de invariâncias locais [132, 139].

de saída [141, 143]. Essa função permite que cada saída tenha valores entre 0 e 1 considerando todas as saídas necessárias. As FL são formadas por unidades de processamento i.e. neurônio, e o termo totalmente conectado significa que todos os neurônios da camada anterior estão conectados a todos os neurônios da camada seguinte [143]. A função *softmax* recebe um vetor de valores como entrada e produz a distribuição probabilística da imagem de entrada pertencer a cada umas das classes na qual a rede foi treinada [145] definida por,

$$Softmax = P(classes/input) = \frac{e^{h^k}}{\sum_{j=1}^C e^{h^j}}, \mathbf{k}=1, \dots, \mathbf{K} \quad (3.12)$$

onde h é o vetor de saída, k é o neurônio ou elemento do vetor de saída, K é o número total de saídas, C é o número total de saídas igual ao número de classes e j o índice de cada elemento, sendo a soma de todas as probabilidades igual a 1 para cada análise[141]. Uma técnica conhecida como pelo termo em inglês *dropout* também é bastante utilizada entre as camadas totalmente conectadas para reduzir o tempo de treinamento e evitar que o classificador tende a se adaptar demais a detalhes específicos de treinamento, o que pode causar uma redução da taxa de acerto [152]. Essa técnica de *dropout* consiste em remover, aleatoriamente a cada iteração durante o aprendizado da rede, uma determinada porcentagem dos neurônios de uma camada, readicionando na iteração seguinte [132]. Isso confere à rede a habilidade de aprender atributos mais robustos, uma vez que um neurônio não pode depender da presença específica de outros neurônios [152].

3.4.3 Treinamento da Rede Neural Convolutacional

O treinamento de uma rede necessita que os dados sejam divididos em lotes que possibilitem a validação da rede. O intuito dessa validação é avaliar o modelo em lotes

de dados diferentes do utilizado para adaptar os pesos sinápticos, evitando que ocorra o treinamento excessivo de redes profundas. Nesse caso, sem essa atenção, a rede pode não adquirir a capacidade de generalizar o seu aprendizado. A distinção e o cuidado durante a escolha da separação entre esses grupos é importante para estabelecer de forma eficiente o aprendizado e a robustez do modelo nas futuras aplicações. Dentro do grupo de dados é preciso estabelecer três lotes. Uma escolha comum [139, 153] é utilizar um percentual em torno de 60% de todo o dado atribuído para lote de treino, 30% para o lote de validação e 10% para o lote de teste para avaliar o modelo, porém essas taxas podem variar de acordo com a rede. O treinamento da CNN precisa de um conjunto de dados de treino suficientemente grande, caso contrário o algoritmo de treinamento é capaz de sobreajustar todos os exemplos do conjunto, sem produzir um modelo capaz de generalização. Na aprendizagem supervisionada durante a propagação adiante os dados de treinamento são propagados na direção das camadas até a saída onde é comparado com a saída desejada. Na retropropagação os filtros são ajustados e a partir de um número de iterações pré definidas que são equivalente à uma época, é realizado uma avaliação com imagens do lote de validação para evitar o excesso de ajuste dos filtros. A propagação adiante e a retropropagação foram exemplificados na figura 3.19. Durante uma época de treinamento são processadas todas as imagens do conjunto de treinamento, até que o erro obtido pela rede durante a época atual seja menor que um liminar pré definido ou um número máximo de interações seja atingido. E no final de todas as épocas com a rede treinada é aplicado a segunda avaliação com o lote de teste. Em alguns trabalhos com [153, 154] um modelo de CNN já treinado pode ser utilizado para treinar uma nova rede utilizando sua arquitetura, e os filtros já treinados, como parâmetros iniciais para adaptação no novo treino com outros dados. Esse tipo de estratégia é conhecida como transferência de conhecimento ou pelo termo em inglês *Transfer Learning*.

O treinamento dessas redes se tornou viável, devido a produção de hardware para a indústria de games. As fabricantes como a *NVIDIA*[®] investiram massivamente em GPU's (Unidade de Processamento Gráfico ou do termo em inglês *Graphics Processing Unit*), utilizadas na renderização de imagens [134, 155]. Com o tempo perceberam que, esse artifícios poderiam ser usados para resolver problemas matemáticos gerais que requerem computação pesada. Dessa forma, desenvolveram a GPU com a linguagem de arquitetura de computação de dispositivo unificado conhecida como CUDA do sigla do termo em inglês *Compute Unified Device Architecture* [156], responsável por processadores gráficos capazes de processar um grande volume de dados, assim códigos podem interagir diretamente com o desempenho da GPU. Com a utilização CUDA a memória e núcleos ou do termo em inglês *cores* da GPU são gerenciáveis de maneira eficiente com códigos aproveitando a performance da GPU utilizando o ambiente para desenvolvimento em C e C++ conhecido como CUDA Toolkit. Essa ferramenta dispõe de compilador, bibliotecas matemáticas, ferramentas de depuração e otimização de performance em ambiente de desenvolvimento

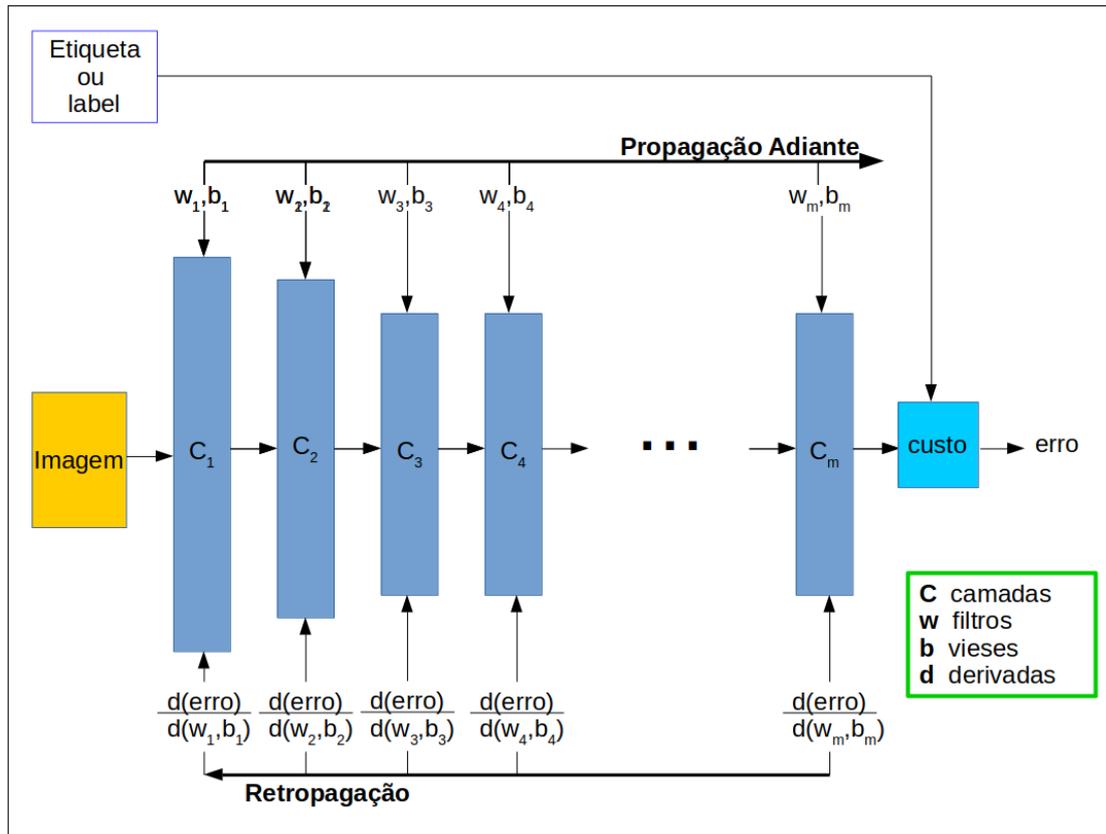


Figura 3.19 – Esquema propagação direta e retropropagação.

para aplicativos de processamento massivo que utilizam a CPU e GPU de forma gratuita ¹.

A Unidade de Processamento Central CPU do termo em inglês *Central Processing Units* possui poucos núcleos de processamento, otimizados para processamento serial sequencial. A GPU possui uma arquitetura paralelizada consistindo em até milhares de núcleos mais eficientes e menores para tratar tarefas múltiplas simultâneas. O fluxo de processamento em CUDA começa quando os dados são copiados da memória principal para a unidade de processamento gráfico. Em seguida, o processador aloca o processo para a GPU, executando as tarefas de forma simultânea em seus núcleos ou cores. Depois, a saída retorna da memória da GPU para a memória principal. O desempenho da placa gráfica depende da quantidade de núcleos. A computação acelerada por GPU trabalha em conjunto com as CPU. A GPU recebe do código de aplicação a indicação da necessidade de maior uso computacional, como ilustrado na figura 5.1. A desvantagem é o nível de complexidade necessário para desenvolver algoritmos de redes neurais artificiais com programação sofisticada utilizando GPU. A maior vantagem para a execução de um algoritmo de aprendizagem profunda é que utilizando CPU pode levar da ordem de semanas de processamento, a medida que processos semelhantes podem durar da ordem de horas em uma GPU.

¹ Download em <https://developer.nvidia.com/cuda-toolkit>

O treinamento da CNN é acelerado utilizando GPU, contudo técnicas de otimização utilizando gradiente foram melhoradas para aumentar a performance durante o treino.

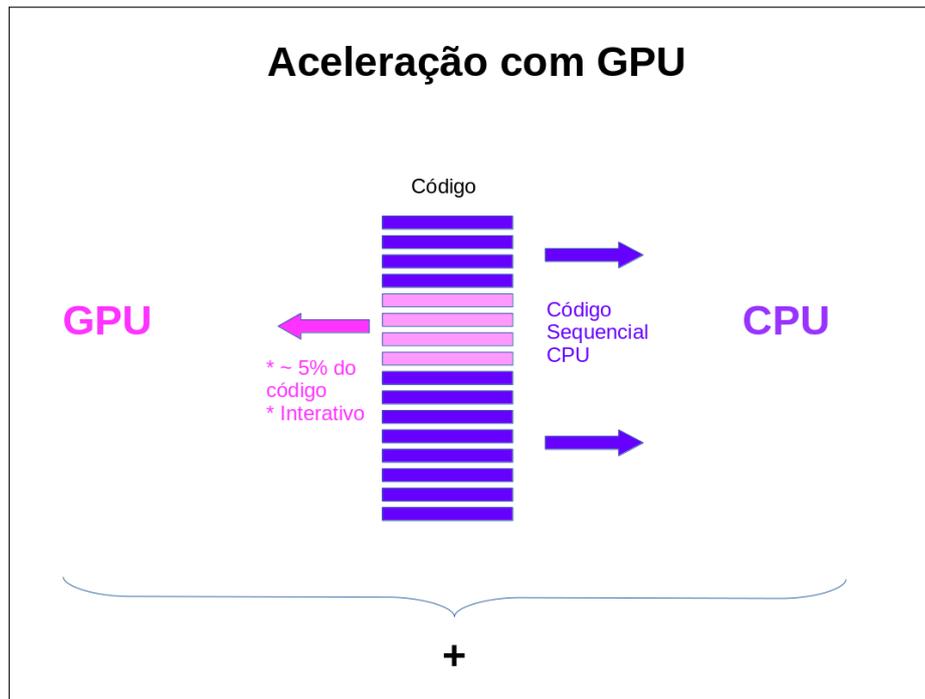


Figura 3.20 – Esquema de aceleração de trabalhos usando GPU. A maior parte do código é executada pela CPU, a GPU desempenha funções que necessitam alto processamento matemático.

3.4.3.1 Gradiente descendente Estocástico com Momento

Para otimizar os filtros até $w_{ótimo}$, considerando o gradiente em função de w e b , de acordo com as equações 3.8, 3.9 e 3.10, cada filtro w_m é atualizado através do cálculo,

$$w_m = w_{m-1} - \eta \frac{\partial \mathcal{L}(w_{m-1}, b_{m-1})}{\partial w_m}, \quad (3.13)$$

sendo a taxa η de aprendizado entre 0 e 1 pré definida [22, 138, 139].

O gráfico 2D visto na figura 3.9 referente gradiente descendente é calculado apenas em função de w , porém como devemos considerar também o viés b , os gráficos característicos são conforme apresentados na figura 3.21.

Nesses gráficos, podemos verificar que o caminho da descida do gradiente pode não ser tão simples como apresentado no gráfico (2) com um mínimo que é global e local ao mesmo tempo. O processo de otimização pode encontrar mínimos locais como no gráfico (1) que não são globais demorando muito para atingir ou não atingindo um único mínimo global. Essa condição pode acontecer, quanto utiliza todos os dados disponíveis sem necessidade. Com isso, o método conhecido como gradiente de descida estocástico (*GDE*) é utilizado pelos algoritmo de redes neurais treinadas com grandes volumes de dados, considerando a aprendizagem por amostragem de dados. Dessa forma, a chance de encontrar o mínimo

global aumenta de forma considerável. Contudo, a cada ciclo de cálculo ou interação os parâmetros b e w são responsáveis por um passo em direção ao mínimo da função L , logo o tamanho desse passo será o valor do gradiente naquele ponto multiplicado pela constante η . Podemos observar em relação ao gráfico (2) na figura 3.21, que quanto mais próximos do ponto de mínimo, menor é a inclinação da função L , com isso menor é o gradiente, logo menor é o passo necessário em direção ao mínimo.

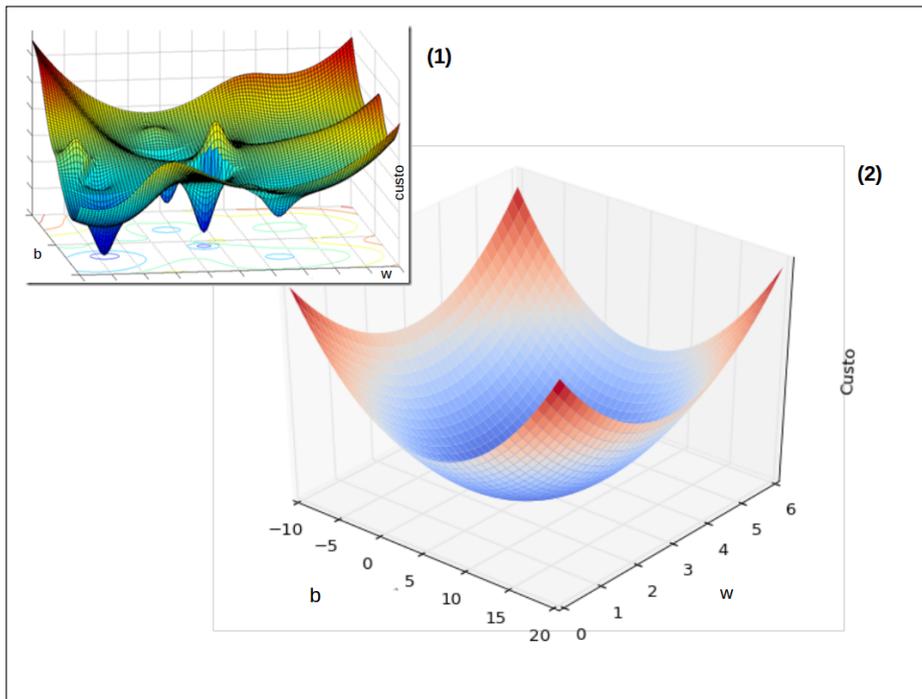


Figura 3.21 – Esquema gráficos custo em função dos filtros e vieses. Nesses gráficos, podemos verificar que o caminho da descida do gradiente pode não ser tão simples como apresentado no gráfico (2) com um mínimo que é global e local ao mesmo tempo. O processo de otimização pode encontrar mínimos locais como no gráfico (1) que não são globais demorando muito para atingir ou não atingindo um único mínimo global

No entanto, controlar o passo não impede que ocorram desvios durante o caminho de descida. Por isso, há várias extensões de técnicas com o propósito de minimizar as flutuações e acelerar o aprendizado. Uma das técnicas é incorporar um termo de momento (γ), que leva em consideração o efeito das mudanças anteriores dos pesos na direção do movimento atual dos pesos. Dessa forma, a equação 3.10 considerando a taxa de momento (γ) fica,

$$\Delta w_m = \gamma \Delta w_{m-1} - \eta \frac{\partial \mathcal{L}(w_{m-1}, b_{m-1})}{w_m}, \quad (3.14)$$

onde o comportamento da superfície de L pode ser comparada na figura 3.22, podemos verificar os caminhos até mínimo global em quatro situações. Os caminhos apresentados em (3) e (4) utilizando o termo momento são os caminhos mais rápidos e com menor flutuações.

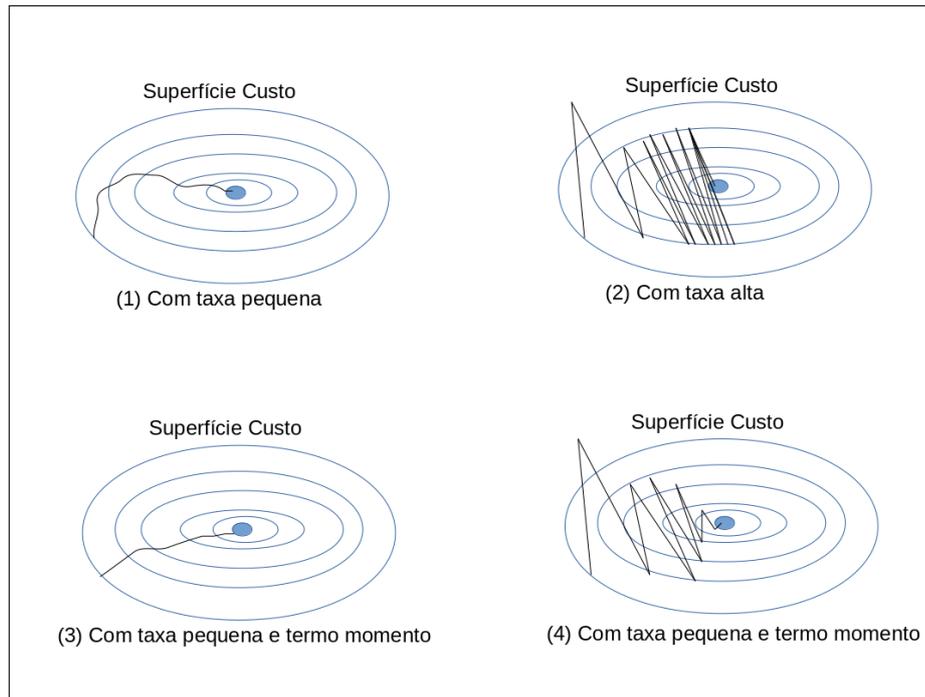


Figura 3.22 – Esquema comparativo de caminhos de descida na superfície de custo utilizando termo de momento. Observamos que os caminhos (1) até o mínimo global possui menos flutuações em relação ao (2) porém é muito lento. Nos casos (3) e (4) foram utilizados os termos de momento, porém em (4) a velocidade de processamento é maior em relação ao (3).

3.4.4 Rede Convolutiva baseada em Propostas de Regiões de Imagem

Algumas redes convolucionais desenvolvidas para detecção de objetos, pessoas e animais em imagens, utilizam algoritmos baseados em propostas de regiões e são conhecidas como RCNN, sigla do termo em inglês *Region-based Convolutional Neural Network*. Essas propostas de regiões das imagens tem o intuito de hipotetizar locais de objetos, já que as CNN convencionais não localizam cada objeto na imagem [134, 137]. Uma RCNN é uma rede convolutiva que prevê, simultaneamente, limites e pontuação de objeto em cada posição para cada região. Esse tipo de rede é treinada de ponta a ponta para gerar classificação de regiões propostas, que são utilizados para detecção [22, 134]. Para cada região proposta extraída da imagem de entrada, é aplicada a CNN conforme detalhada na seção 3.4.2. A vantagem desse tipo de rede é que ela é capaz de reconhecer mais objetos em uma imagem, isso torna a detecção de múltiplas fraturas e desmoronamentos possível. Porém, como desvantagem, nesse tipo de rede são processadas todas as etapas de CNN para cada região proposta, acarretando no aumento do tempo de processamento [22, 139, 156].

3.4.4.1 Regiões propostas

Para gerar as regiões propostas (RP) para RCNN existem vários métodos. O mais comum é o algoritmo de pesquisa seletiva ou conhecido pelo termo em inglês *Selective Search* por Uijlings [157]. Essa função gera regiões definida por caixas de vários tamanhos. O tamanho mínimo dessas caixas é definido pelo usuário. Durante nossos testes foi necessário avaliar as dimensões mínimas do objeto se que deseja detectar. O tempo para encontrar todas as regiões candidatas depende do tamanho da imagem e da dimensão mínima definida para as menores caixas delimitadoras. O código gera as menores caixas com todos os objetos previamente encontrados e aumenta progressivamente o tamanho das demais caixas delimitadoras de acordo com a similaridade entre esses primeiros objetos por textura, cor ou fechamento. A identificação utilizando fechamento envolve um processo de reconhecimento de um objeto mesmo que seus limites não estejam bem definidos, conforme podemos verificar nos exemplos da figura 3.23, onde podemos identificar as formas de um triângulo, quadrado e um cubo. A identificação dos objeto na imagem

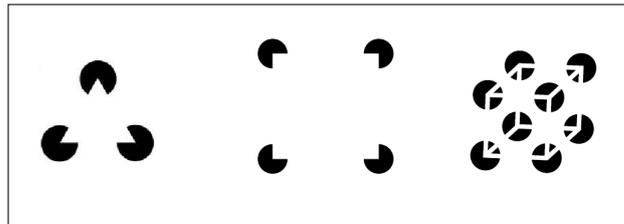


Figura 3.23 – Esquema exemplo de identificação por fechamento. As formas de triângulo, quadrado e cubo conseguem ser identificadas, mesmo sem os limites bem definidos.

é feita por segmentação após a aplicação de filtro. O filtro utilizado é o gaussiano, utilizado para pré processamento, realizando a suavização da imagem contribuindo para a diminuição de ruídos originados da aquisição. Através de uma máscara correspondente à uma janela deslizante pela imagem computa os novos valores dos pixels, através da operação de convolução. Para encontrar a máscara para o filtro gaussiano em 1D, para uma distribuição de valores, é utilizada a equação,

$$Gauss(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad (3.15)$$

onde x é a distribuição de valores e σ é o desvio padrão da distribuição. No caso da imagem, como é uma distribuição de valores em duas dimensões, o filtro é obtido utilizando a gaussiana 2D com:

$$Gauss(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (3.16)$$

onde x é a distribuição de valores de pixels na direção horizontal e y valores da direção vertical. Uma técnica que acelera o processamento é ao invés de realizar a convolução em 2D, é fazer a convolução 1D na direção x e depois usar a mesma máscara 1D da gaussiana na direção y da imagem [158].

Após a filtragem realiza segmentação utilizando o método de Felzenszwalb e Huttenlocher (MFH) baseado em grafos [158, 159]. Esse algoritmo de segmentação, tem como base a identificação dos limites dos objetos, utilizando o conceito de grafo para representar a imagem. A ideia de transformar a imagem na forma de grafo, é baseado na representação por um conjunto não vazio de nós chamado de vértices e um conjunto de arcos ou arestas, de tal forma que cada aresta conecta dois vértices. A definição de grafo não direcionado é definido por,

$$\text{Grafos} = (V, A), \quad (3.17)$$

onde V é o conjunto dos vértices, sendo o conjunto de arestas A , localizados pelas coordenadas x_i e y_j , entre os pares de vértices v_x e v_y . Considerando a aplicação de grafos em imagens, os vértices são os valores dos pixels da imagem, elementos do conjunto V . Os valores de A são as medidas do quão similares são os pares de pixels ligados pela aresta. Para o método FH, a segmentação S dos objetos de uma imagem é representada por um conjunto de componentes C , ou seja os pixels com algum tipo de similaridade, sendo $C \in S$. Sendo cada objeto segmentado representado por uma componente definida por um subconjunto de A e V representado por um grafo do objeto,

$$\text{Grafos}' = (V', A'), \quad (3.18)$$

sendo $A' \subseteq A$ e $V' \subseteq V$. Isso significa, que é esperado que arestas que conectem vértices do mesmo objeto ou componente apresentem valores maiores como peso, ao passo que valores menores de peso representem a dissimilaridade, i. e. a falta de similaridade em relação aos valores dos vértice ou pixels. Como podemos verificar, no exemplo de alguns pixels de imagem apresentado em grafos na figura 3.25. Os valores em vermelho são os menores e representam as arestas com maior dissimilaridade com o vértice vizinho. Os valores em preto representam os maiores valores e a maior similaridade entre os vértices ou pixels.

O cálculo dos pesos das arestas em imagens, na sua representação matricial, é feito através do cálculo da distância euclidiana por:

$$d_{ij} = \sqrt{\sum_{f=1}^p (x_{if} - x_{jf})^2}, \quad (3.19)$$

onde p é o número total de colunas da matriz, x_{if} e x_{jf} são os valores dos pixels ou vértices comparados em relação a coluna f . Os valores de pixels similares correspondem á um mesmo objeto. Os pesos com valores mínimos definem o limite do objeto. A escolha do tamanho das arestas, pode ser incrementado. O algoritmo pode aumentar as regiões, definindo novas bordas para propor novas regiões de objetos.

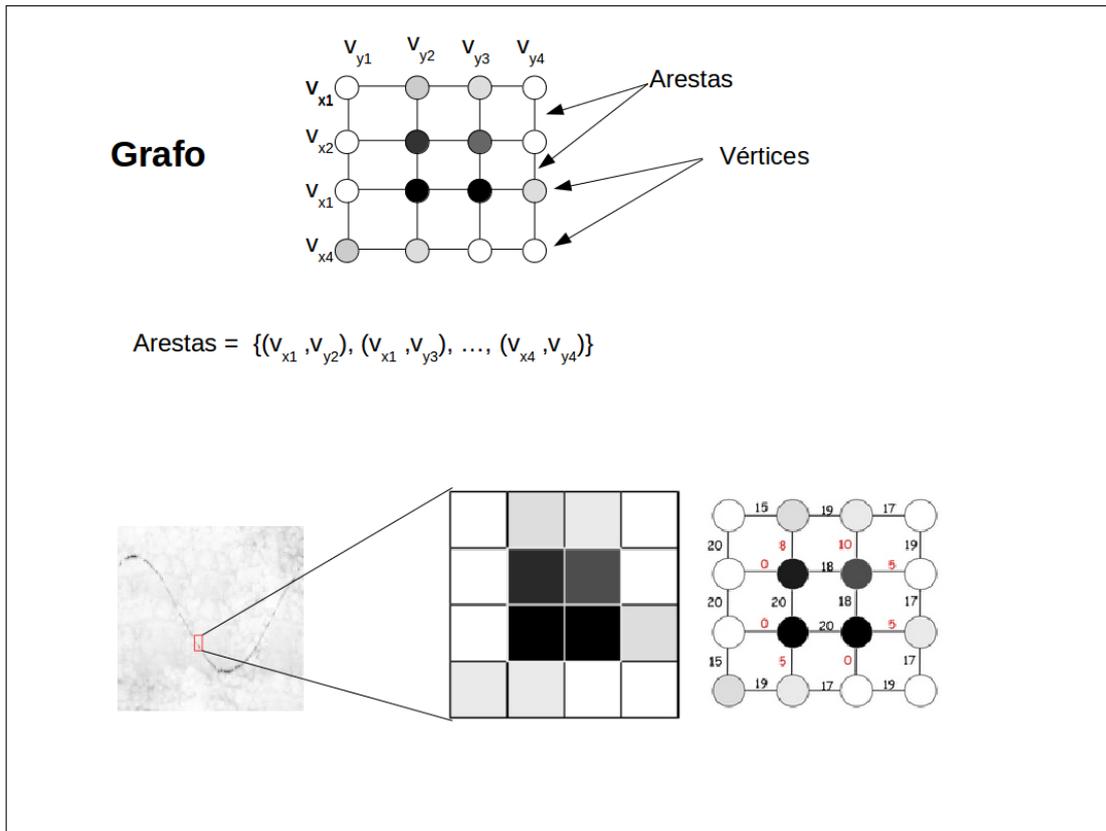


Figura 3.24 – Esquema com exemplo de trecho de imagem representada em grafos, utilizado para segmentação. Os valores de pesos menores em números vermelhos indicam uma diferença de região nos 4 pixels centrais.

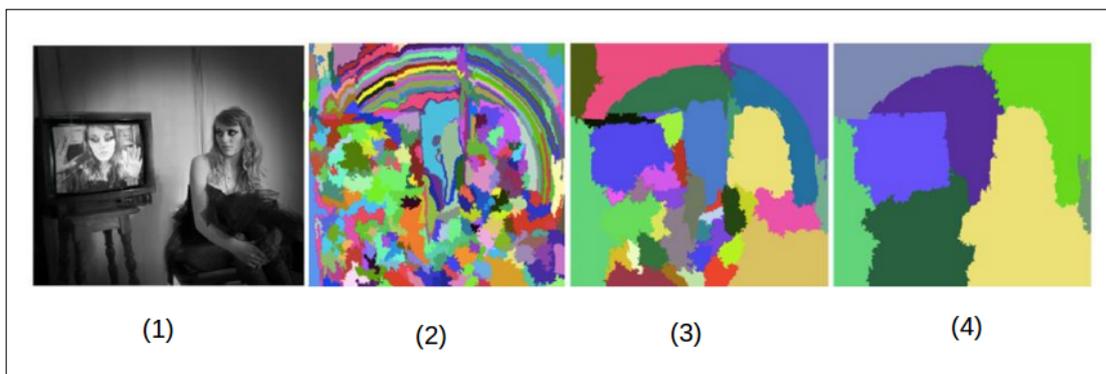


Figura 3.25 – Exemplo com figura adaptada de [159] resultado da segmentação utilizando o método Felzenszwalb e Huttenlocher. Em 1 apresentamos a imagem original, em 2, 3 e 4 os resultados considerando novas arestas.

3.5 Rede Neural Convolutiva Rápida baseada em regiões de imagem

A Rede Neural Convolutiva Rápida baseada em regiões ou RCNN rápida (FRCNN) da sigla em inglês *Fast Region-based Convolutional Neural Network* é uma rede que compartilha camadas convolucionais na parte da arquitetura dedicada a detecção, definindo as

RP de maneira diferenciada das RCNNs anteriores. A rede FRCNN considera como uma única entrada uma imagem inteira, as coordenadas das regiões verdadeiras, a etiqueta de classe de cada coordenada e um conjunto de *RP* de objetos [22, 160]. A rede primeiro processa de uma única vez em toda imagem um conjunto de 5 camadas convolucionais, produzindo mapas de características [22]. Propondo regiões de interesse (RoI) da sigla em inglês *Region of Interest* do extrator de características no final do quinto bloco de camadas, referente as *RP* geradas da imagem, gerando um vetor de características de comprimento fixo [22].

O esquema apresentado em 3.26, mostra uma visão geral do funcionamento de uma CNN baseada em propostas de regiões. No exemplo dessa figura, a região da fratura foi projetada do mapa de características na saída da quinta camada de máxima subamostragem. Essa projeção é conhecida como Região de Interesse ROI do termo em inglês *Region of Interest*. Essa ROI é representado por um vetor de características que alimenta uma sequência de camadas totalmente conectadas e funcionam como RNA convencional que definem na saída a probabilidade de detecção de cada classe e realizando a regressão linear entre as coordenadas das regiões propostas e as regiões detectadas no mapa de características refinando a localização dos objetos [22].

3.5.1 Arquitetura

A arquitetura da FRCNN conforme apresentada na figura 3.28 apresenta inicialmente 5 blocos de camadas convolucionais que utilizam como função de ativação a *Relu* e são seguidas das camadas de subamostragem. Esses blocos são baseados nas camadas convolucionais do modelo VGG16 da sigla em inglês de *Visual Geometry Group16* [23]. Essa rede utiliza arquitetura profunda, configurada com interconexão direta de neurônios e modelo de aprendizado supervisionado.

Todos os filtros utilizados nas camadas convolucionais dessa rede possuem dimensões fixas estipulados de acordo com a saída desejada. A saída de cada camada é um volume ou matriz multidimensional conhecida como tensor, formado pelo enfileiramento dos mapas de características.

Assim podemos considerar uma imagem I com dimensões $HI \times WI \times 3$, onde HI é a altura, WI a largura e 3 é a profundidade referente aos canais de cores, como entrada no primeiro bloco composto por duas camadas convolucionais e uma de máxima subamostragem ou *max pooling*. Nesse bloco inicial, são extraídas as características ligadas à traços. A camada *Conv11* é responsável pela primeira convolução e utiliza 64×3 filtros, pois a entrada tem 3 canais. Os mapas de características, MC, de saída da *Conv11* são definidos por:

$$MC11 = \text{Relu}(\text{Convolução}(I * W11) + b11), \quad (3.20)$$

sendo o volume do filtro $W11(3 \times 3 \times 3 \times 64)$ e as dimensões do tensor de saída do

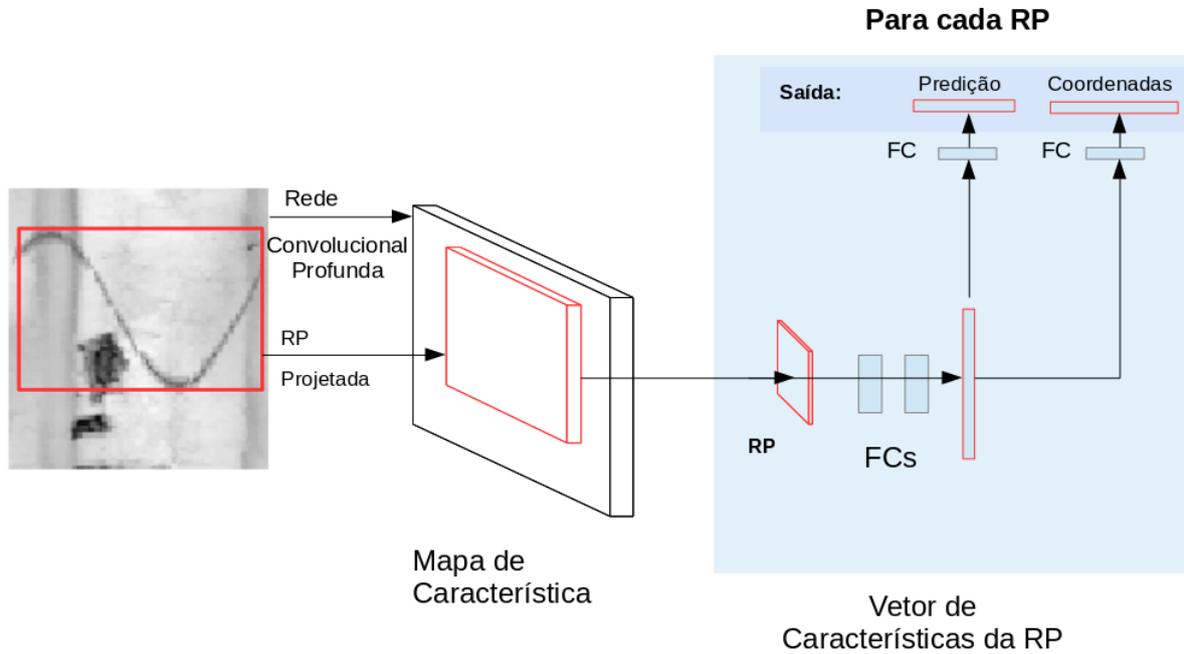


Figura 3.26 – Esquema Geral FRCNN adaptado de [22]. Exemplo onde uma região proposta RP da imagem com fratura foi projetada do mapa de características da quinta camada de máxima subamostragem definindo a Região de Interesse ROI do termo em inglês *Region of Interest*. Essa ROI é vetorizada e alimenta uma sequência de camadas totalmente conectadas estimando na saída a classe e as coordenadas dos objetos [22]

$MC11(HI \times WI \times 64)$, onde 64 é o número de MC, cada um referente ao resultado de um filtro e seu respectivo valor de viés $b11$. Uma nova etapa de convolução *Conv12* aplica 64×64 filtros. Assim,

$$MC12 = \text{Relu}(\text{Convolução}(MC11 * W12) + b12), \quad (3.21)$$

onde o filtro tem volume $W12(3 \times 3 \times 64 \times 64)$, $b12$ o viés. O volume de saída $MC12(HI \times WI \times 64)$ é a entrada para a primeira camada de máxima subamostragem (MP1), definida por:

$$MP1 = \text{Argumento Máximo}(MC12), \quad (3.22)$$

sendo utilizado um volume $mp1(2 \times 2 \times 64)$ para computar os valores máximos. Dessa forma, as dimensões do dado de saída dessa camada são reduzidos para $MP1(HI/2 \times WI/2 \times 64)$, resultando em um volume que correspondem a metade da altura e largura de entrada.

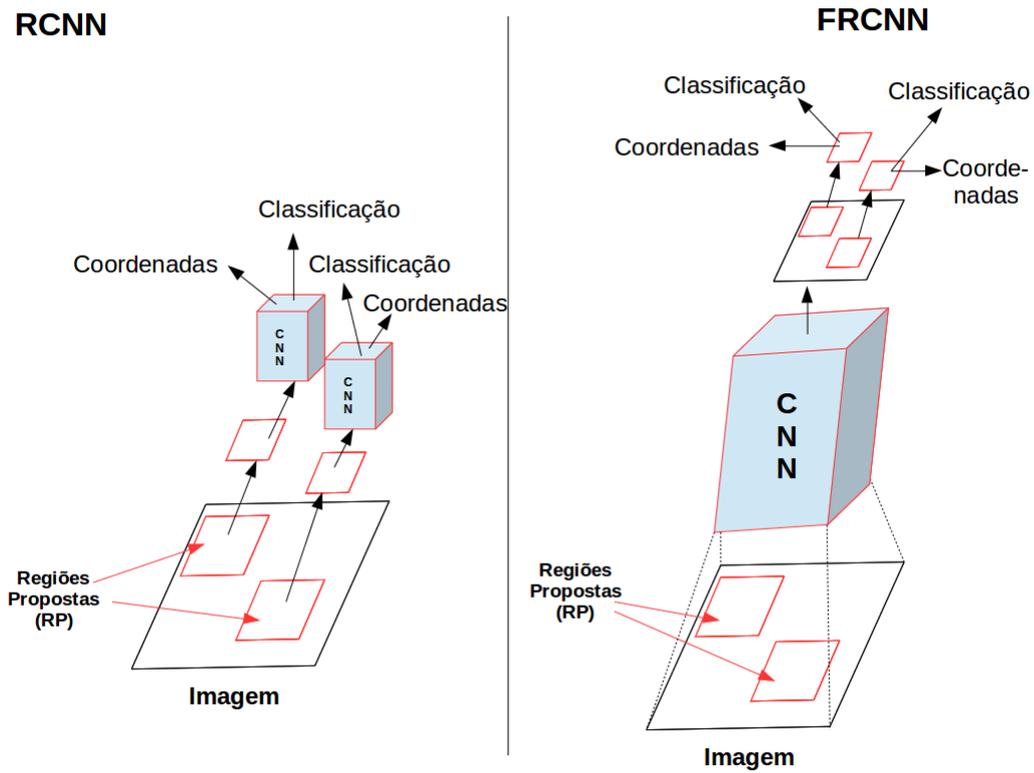


Figura 3.27 – Comparativo RCNN com FRCNN adaptado de [22]. A RCNN rápida (FRCNN) da sigla em inglês *Fast Region-based Convolutional Neural Network* é uma rede que compartilha a CNN em todas as regiões propostas RP, diferente das RCNNs comuns que aplicam a CNN em cada uma separadamente [22, 160].

No segundo bloco, são extraídas as características referentes à curvas. O volume de MP1 é entrada. As camadas desse bloco *Conv21* e *Conv21* são definidas por:

$$MC21 = \text{Relu}(\text{Convolução}(MP1 * W21) + b21) \quad (3.23)$$

e

$$MC22 = \text{Relu}(\text{Convolução}(MC21 * W22) + b22) \quad (3.24)$$

onde os filtros de cada camadas tem o volume representado respectivamente, $W21(3 \times 3 \times 64 \times 128)$ e $W22(3 \times 3 \times 128 \times 128)$. O tensor de saída de MC21($HI/2 \times WI/2 \times 128$) é igual a MC22 em volume. A camada de máxima subamostragem que segue as camadas convolucionais do bloco 2, recebe o volume de MC22, sendo representada por:

$$MP2 = \text{Argumento Máximo}(MC22), \quad (3.25)$$

sendo o volume que computa o argumento máximo definido como $mp2(2 \times 2 \times 128)$. A saída contribui para uma nova redução, apresentando $MP2(HI/4 \times WI/4 \times 128)$ de dimensões, que é a entrada do próximo bloco.

O terceiro bloco de camadas convolucionais é responsável por extrair características ligadas a textura e possui 3 camadas convolucionais e os mapas são definidas por:

$$MC31 = \text{Relu}(\text{Convolução}(MP2 * W31) + b31), \quad (3.26)$$

$$MC32 = \text{Relu}(\text{Convolução}(MC31 * W32) + b32) \quad (3.27)$$

e

$$MC33 = \text{Relu}(\text{Convolução}(MC32 * W33) + b33) \quad (3.28)$$

onde a entrada em MC31 é a saída de MP2 e os filtros associados à cada camada respectivamente são, $W31(3 \times 3 \times 128 \times 256)$, $W32(3 \times 3 \times 256 \times 256)$ e $W33(3 \times 3 \times 256 \times 256)$ e volume de saída das três camadas MC31, MC32 e MC33 possuem as mesmas dimensões com $HI/4 \times WI/4 \times 128$.

Ao fim desse bloco temos a terceira camada de máxima subamostragem que segue as camadas convolucionais representada por:

$$MP3 = \text{Argumento Máximo}(M33), \quad (3.29)$$

o volume que desliza pela entrada e computa o valor máximo tem dimensões $mp3(2 \times 2 \times 256)$, onde o volume de saída da camada é novamente reduzido para saída $MP3(HI/8 \times WI/8 \times 256)$.

O quarto bloco de camadas convolucionais já pode ser capaz de extrair características referentes à parte do objeto composto por características já obtidas das camadas anteriores. Uma parte de um rosto, por exemplo, pois já reconhece a forma de semicírculo referente à um traço e uma curva extraídas das duas primeiras camadas e a textura que caracteriza a pele do rosto obtida da terceira camada. Suas saídas são definidas por:

$$MC41 = \text{Relu}(\text{Convolução}(MP3 * W41) + b41), \quad (3.30)$$

$$MC42 = \text{Relu}(\text{Convolução}(MC41 * W42) + b42) \quad (3.31)$$

e

$$MC43 = \text{Relu}(\text{Convolução}(MC42 * W43) + b43) \quad (3.32)$$

onde os volumes dos filtros correspondem respectivamente, $W41(3 \times 3 \times 256 \times 512)$, $W42(3 \times 3 \times 512 \times 512)$ e $W43(3 \times 3 \times 512 \times 512)$ e a saída das três camadas possuem as mesmas dimensões com $HI/8 \times WI/8 \times 256$. Ao fim desse bloco temos a quarta camada de máxima subamostragem que segue as camadas convolucionais representada por:

$$MP4 = \text{Argumento Máximo}(M43), \quad (3.33)$$

onde o volume deslizante que computa o argumento máximo é $mp4(2 \times 2 \times 512)$ e o volume de saída da camada $MP4(HI/16 \times WI/16 \times 512)$.

O quinto bloco, consegue realizar a extração de características de todo objeto, ou quase todas, com o auxílio das partes que foi extraídas nas camadas antecessoras. Utilizando o exemplo anterior, onde obtive características de várias partes do rosto humano e com o

novo aprendizado os novos filtros conseguem extrair características referentes à um rosto completo.

$$MC51 = \text{Relu}(\text{Convolução}(MP4 * W51) + b51), \quad (3.34)$$

$$MC52 = \text{Relu}(\text{Convolução}(MC51 * W52) + b52) \quad (3.35)$$

e

$$MC53 = \text{Relu}(\text{Convolução}(MC52 * W53) + b53) \quad (3.36)$$

o volume dos filtros são respectivamente, $W51(3 \times 3 \times 512 \times 512)$, $W52(3 \times 3 \times 512 \times 512)$ e $W53(3 \times 3 \times 512 \times 512)$ e o de saída das três camadas possuem as mesmas dimensões com $HI/16 \times WI/16 \times 512$.

Ao fim desse bloco temos a quinta camada de máxima subamostragem. Como podemos verificar na figura 3.28, essa camada é diferente das camadas anteriores de subamostragem. Nessa etapa da rede em diante a FRCNN é diferente do modelo original VGG16. A quinta camada de máxima subamostragem recebe como entrada além da saída MC53 as informações para definir os limites das regiões de interesse (*ROIs*) da saída da última camada convolutiva referentes às regiões propostas (RP). Essa informação vem com as coordenadas referentes à RP e o índice referente a imagem a que pertence. Cada RP tem coordenadas ajustadas para as novas posições referente as dimensões de MC53. Depois, suas dimensões de largura e altura são redimensionadas para corresponder à uma saída 7×7 , conforme apresentamos um esquema na figura 3.30. Essa camada é definida por:

$$MP5 = \text{Argumento Máximo}(\text{Setorização}(MC53(RP))), \quad (3.37)$$

onde a saída é definida em etapas. A primeira dividi a proposta de região projetada de MC53 em setores, assim o número delas é igual a da dimensão da saída. No segundo momento são obtidos os valores máximo de cada setor. Como algumas dimensões de proposta não tem o número de pixels que permita que todos os setores tenham um número igual de pixels, apresentamos um exemplo simplificado desse caso, na figura 3.29. Nesse exemplo, utilizando a subamostragem para definir a ROI com saída reduzida 2×2 . Assim, a proposta do exemplo com dimensões 7×5 utilizando setores, resulta em uma saída 2×2 . O resultado é que, a partir de uma lista de retângulos delimitadores referentes às RP com tamanhos diferentes, podemos obter rapidamente uma lista de mapas de recursos de ROI correspondentes, com um tamanho fixo. A dimensão da saída de agrupamento da ROI não depende, na verdade, do tamanho do mapa de recursos de entrada nem do tamanho das propostas de região. É determinado apenas pelo número de seções em que dividimos a proposta. O volume que sai da MP5 corresponde a $MP5(7 \times 7 \times 512 \times (\text{Número de RP}))$.

Então para cada ROI como entrada na primeira FCL terá um volume $7 \times 7 \times 512$ referente à uma RP, que utiliza um filtro com volume $FL1(7 \times 7 \times 512)$ e definido por:

$$FC11 = \text{Relu}(FC11 * W11) + b11 \quad (3.38)$$

Com isso, se considerarmos todas das RP da imagem, o resultado do volume de características é $7 \times 7 \times 512 \times (\text{Número de RP})$ que precisa ser reduzido para a próxima camada, para gerar uma saída vetorizada.

Arquitetura FRCNN (VGG16 Modificada)

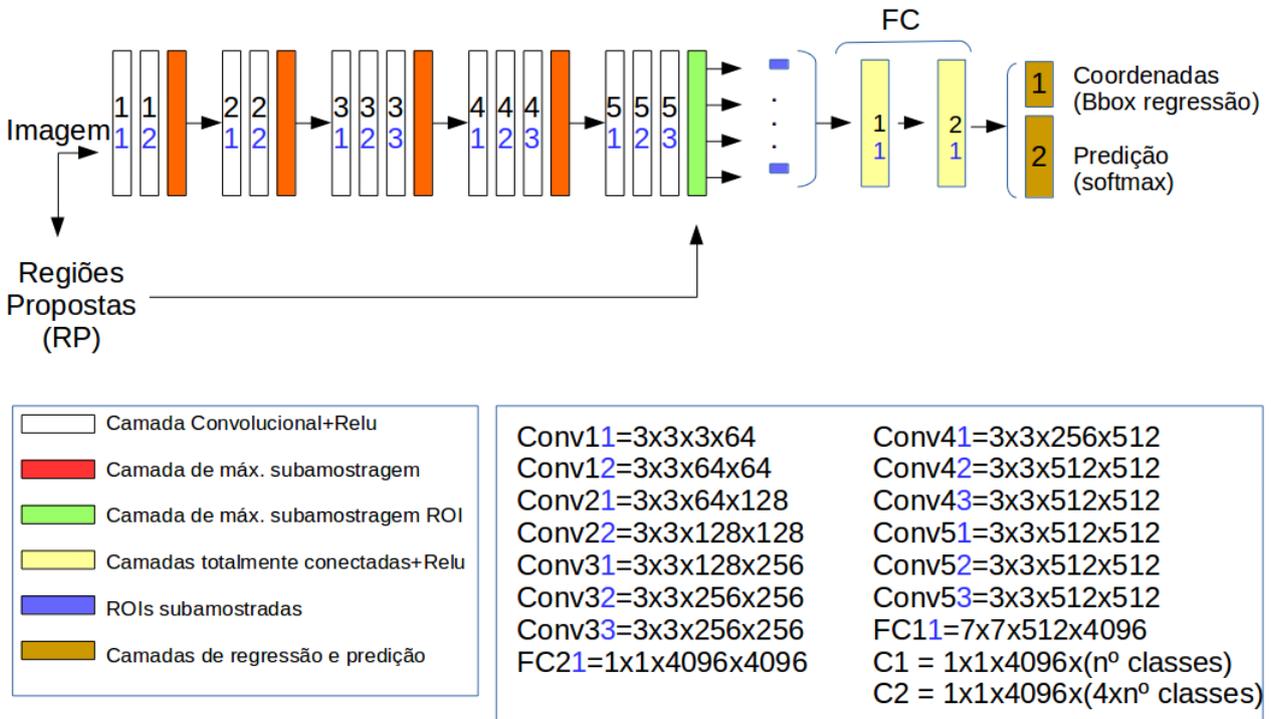


Figura 3.28 – Detalhes das camadas da arquitetura FRCNN. Cada camada é definida pela legenda de cores. Na tabela estão as respectivas dimensões do filtro utilizadas em cada camada.

3.5.1.1 Decomposição do Valor Singular Truncado

Como o volume do Mapa de Características transmitidas pela rede, requer muito processamento, e com a saída da camada ROI de subamostragem a quantidade de dados aumenta com o número de regiões propostas. Logo, com o intuito de otimizar esse processo é realizada uma redução das características [161] definidas pela primeira *Fully Connected Layer* FCL ou camada densa, buscando manter suas características principais, utilizando a decomposição de valores singulares truncado (SVDT) conhecido pelo

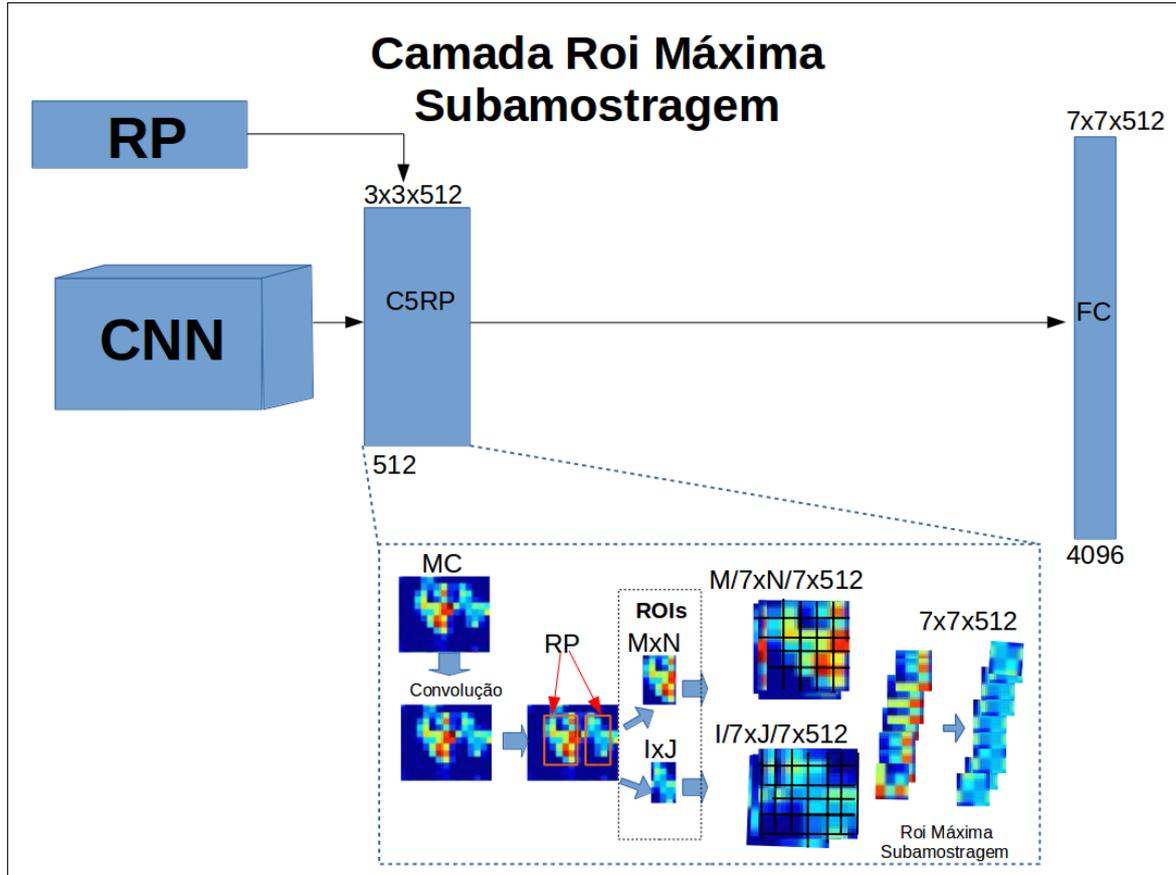


Figura 3.29 – Esquema referente à camada de subamostragem [22]. Essa camada é diferente das camadas anteriores de subamostragem. Dessa etapa da rede em diante a FRCNN é diferente do modelo original VGG16. A quinta camada de máxima subamostragem recebe informações das RP. Cada RP depois de redimensionar, tem suas dimensões de largura e altura divididas por setores, exemplificada na figura como uma grade. O número de setores é determinado pela dimensão de largura e altura da saída, nesse caso 7×7 . Assim, dessa forma realiza a subamostragem em cada setor, obtendo os valores máximos.

termo em inglês *singular values decomposition truncated* conforme esquematizado na figura 3.31. Considerando um único Mapa de Características 2D, representado por uma matriz com seus valores de pixels. Aplicar a técnica de SDVT nessa matriz determinada por $A \in \mathbb{R}^{m \times n}$ consiste em fatorar essa matriz em um produto de matrizes USV^T , onde $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{n \times n}$ são matrizes ortogonais [162]. Sendo $V = [v_1, \dots, v_n]$, $U = [u_1, \dots, u_m]$, $S \in \mathbb{R}^{m \times n}$ em que $S = [s_{ij}]$, com $s_{ij} = 0$ se $i \neq j$, sendo s as raízes dos autovalores de A . É possível utilizar a SVDT para calcular o posto p da matriz A . Posto de uma matriz p é equivalente ao número de linhas não-nulas quando a mesma está escrita na forma reduzida escalonada. Então, obtendo uma matriz A_k que se aproxime o máximo possível de A porém $A_k \neq A$, para tal dado $A_k = \sum_{i=1}^{k_g} s_i u_i v_i^T$, $V = [v_1, \dots, v_k]$, $U = [u_1, \dots, u_k]$ e $S_k = \text{diagonal}[s_1, \dots, s_k, 0, \dots, 0]$, resultando em $A_k = U_k S_k V_k^T$, então $\min_k \|A - A_k\|_2 = s_{k+1}$, onde $k = \text{posto } p \text{ de } A$. Assim, A_k , onde $k < p$ é considerado uma boa aproximação para A e a matriz resultado será a máxima aproximação da matriz

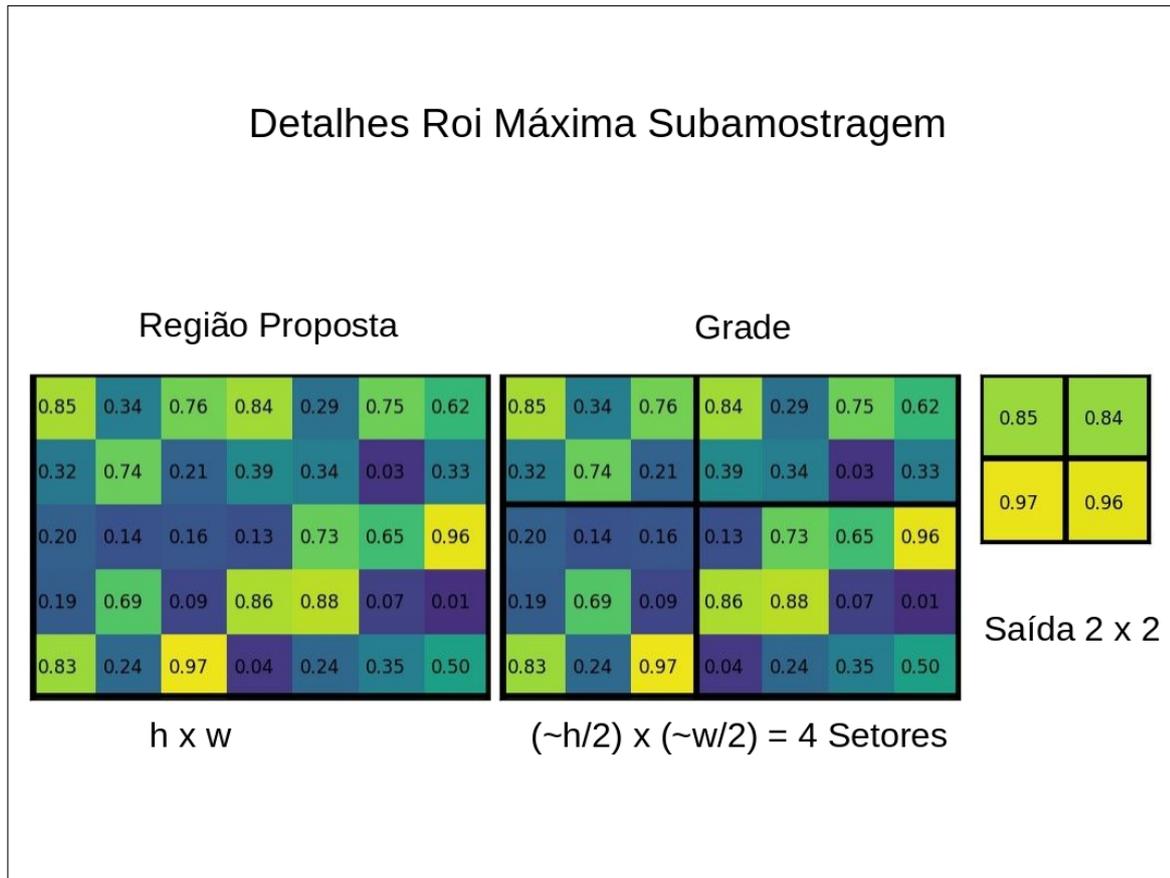


Figura 3.30 – Esquema referente à ROI subamostragem [22]. Como algumas dimensões de proposta não tem o número de pixels que permita que todos os setores tenham um número igual de pixels, apresentamos um exemplo simplificado desse caso, onde é utilizada a subamostragem para definir a ROI com saída reduzida 2×2 . A proposta do exemplo com dimensões 7×5 utilizando setores, resulta em uma saída 2×2 .

original [163]. Um exemplo é apresentado na figura 3.32, utilizando uma matriz $A \in \mathbb{R}^{m \times n}$ onde $m = 3$ e $n = 2$ o resultado decomposição valores singulares é apresentado com $A = USV^T$. O posto de A é 2, então para $k < p$, podemos considera $k = 1$, logo A_1 é uma boa aproximação para A . A aproximação A_1 é igual ao produto (primeira coluna da primeira matriz U) * (primeiro elemento da segunda matriz S ou s_1) * (primeira linha da terceira matriz referente a V^T). Os parâmetros utilizados são definidos na saída da primeira FLC por sua equivalente $S_k V_k^T$ na segunda FCL por sua equivalente U_k . Dessa forma, considerando uma RP o volume de características de entrada das FCL com $7 \times 7 \times 512 = 25088$ utilizando esse recurso é reduzido para um vetor de características com $1 \times 1 \times 4096 = 4096$

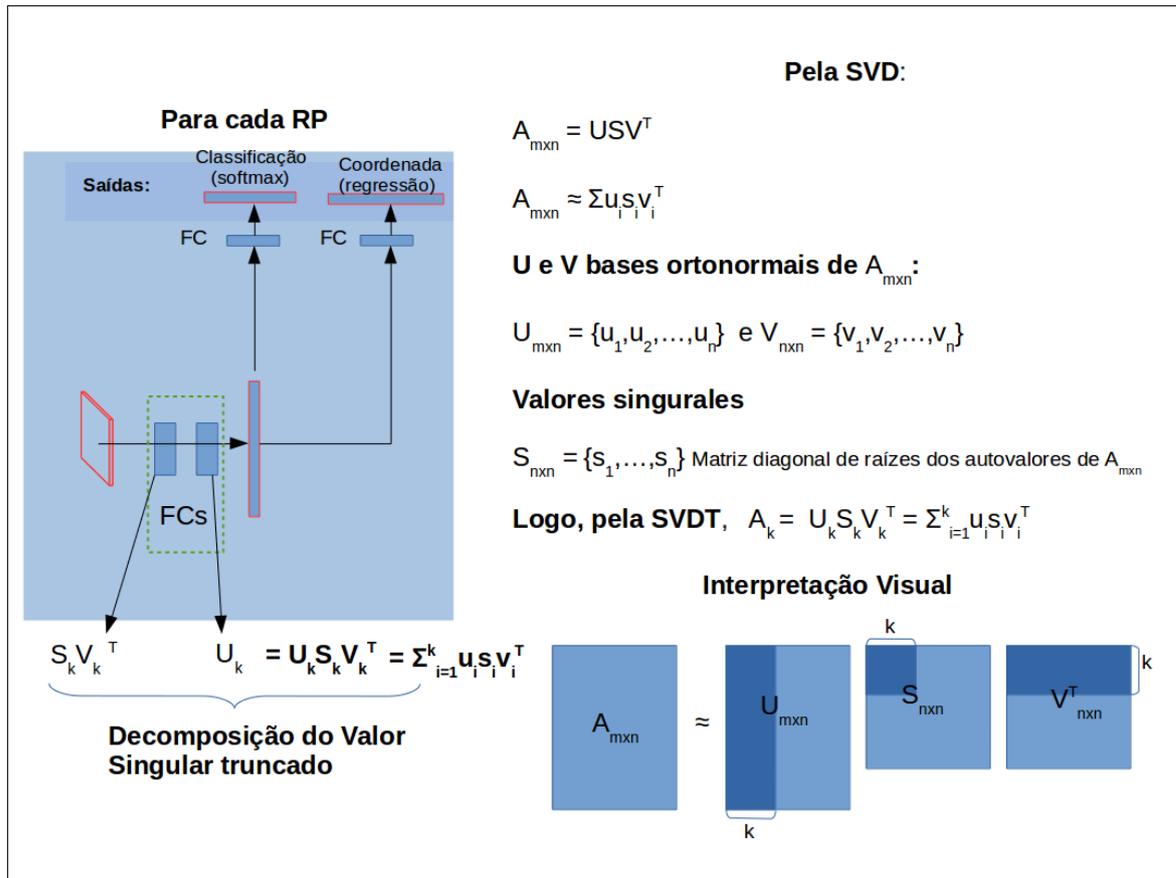


Figura 3.31 – Esquema de Decomposição do Valor Singular Truncado utilizado nas camadas FCL da arquitetura apresentada em [22]. Recurso utilizado para redução das características.

3.5.1.2 Escala e rotação invariante

Detectar objetos em diferentes tamanhos com a FRCNN treinada é possível mesmo que as imagens de treinamento tenham objetos com outra dimensão [164]. Isso porque, a FRCNN utiliza técnicas diferentes durante o treinamento e a detecção para permitir detectar o objeto independente da escala. Antes do treinamento todas as imagens de entrada são redimensionadas com escala de 600 pixels para o lado menor da imagem e com 1000 para o maior logo a escala das imagens e das RP são mantidas fixas. Porém, durante a etapa posterior de teste de detecção, que equivale a uma propagação direta, cada imagem de entrada de qualquer dimensão é escalonada em dimensões menores finalizando em 224^2 pixels de área. Esse recurso é chamado de multiescala. Na detecção a camada de convolução 5 fixa as dimensões de saída e a subamostragem ROI pooling faz a subamostragem das Rois de várias dimensões. Então objetos pertencentes à mesma classe, pequenos ou grandes são generalizados e são possíveis de detecção em novos dados utilizando a rede treinada. Os autores [22, 164] testaram o método de multiescalas onde são utilizadas algumas variações de escala durante o treinamento. Mas, conforme os testes de desenvolvimento da técnica de multiescalas no treinamento, o tempo de processamento

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow AA^T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}; \quad A^T A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \Rightarrow$$

Autovalores e autovetores para AA^T : $(1-\lambda)[(2-\lambda)(1-\lambda)-1]-(1-\lambda)=(3-\lambda)(1-\lambda)\lambda=0 \Rightarrow$

$$\lambda_1=3; \quad u_1 = \begin{bmatrix} 1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}; \quad \lambda_2=1; \quad u_2 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix}; \quad \lambda_3=0; \quad u_3 = \begin{bmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix};$$

Autovalores e autovetores para $A^T A$: $(2-\lambda)^2-1=(3-\lambda)(1-\lambda)=0 \Rightarrow$

$$\lambda_1=3; \quad v_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}; \quad \lambda_2=1; \quad v_2 = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}; \Rightarrow$$

$$Av_i = \sigma_i u_i; \quad i=1,2 \Rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \sigma_1 \begin{bmatrix} 1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix} \Rightarrow \sigma_1 = \sqrt{3}; \quad \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \sigma_2 \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix} \Rightarrow \sigma_2 = 1$$

$$\Rightarrow \quad A = \underbrace{\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}}_U \underbrace{\text{diag}\{\sigma_1, \sigma_2\}}_D \underbrace{\begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix}}_{V^T} = \begin{bmatrix} 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & \sqrt{3} & 0 \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} & 0 & 1 \\ 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Sendo $k=1$, $A_k = \text{produto}$

Figura 3.32 – Esquema exemplo de decomposição do valor singular truncado. Utilizando uma matriz $A \in \mathbb{R}^{m \times n}$ onde $m = 3$ e $n = 2$ o resultado DVS é apresentado por $A=USV^T$. O posto de A é 2, então para $k < p$, podemos considera $k = 1$, logo A_1 é uma boa aproximação para A . A aproximação A_1 é igual ao produto (primeira coluna da primeira matriz U) * (primeiro elemento da segunda matriz S ou s_1) * (primeira linha da terceira matriz referente a V^T)

aumentou consideravelmente e não obteve um ganho de performance considerável. Assim como, perceberam que a retropropagação do erro nas camadas convolucionais ficou prejudicada utilizando multiescala.

Considerando a invariância com relação a rotação, esta é obtida através da técnica de amostragem em algumas imagens. A imagem de entrada pode ser escolhida durante o treinamento para sofrer rotação com probabilidade de 50% [22, 165].

3.5.1.3 Classificação

Classificação é o processo de, no contexto da FRCNN apresentada, determinar a classe ou categoria que a imagem de entrada pertence. A FRCNN é proposta para classificação de forma eficiente de objetos usando rede convolucional profunda. A proposta é um algoritmo de treinamento que aprende classificar propostas de regiões de objetos e refinar sua localização espacial utilizando técnica de regressão linear, conforme apresentado na figura 3.34, onde o somatório das notas de classificação p são considerados igual a 1 [22].

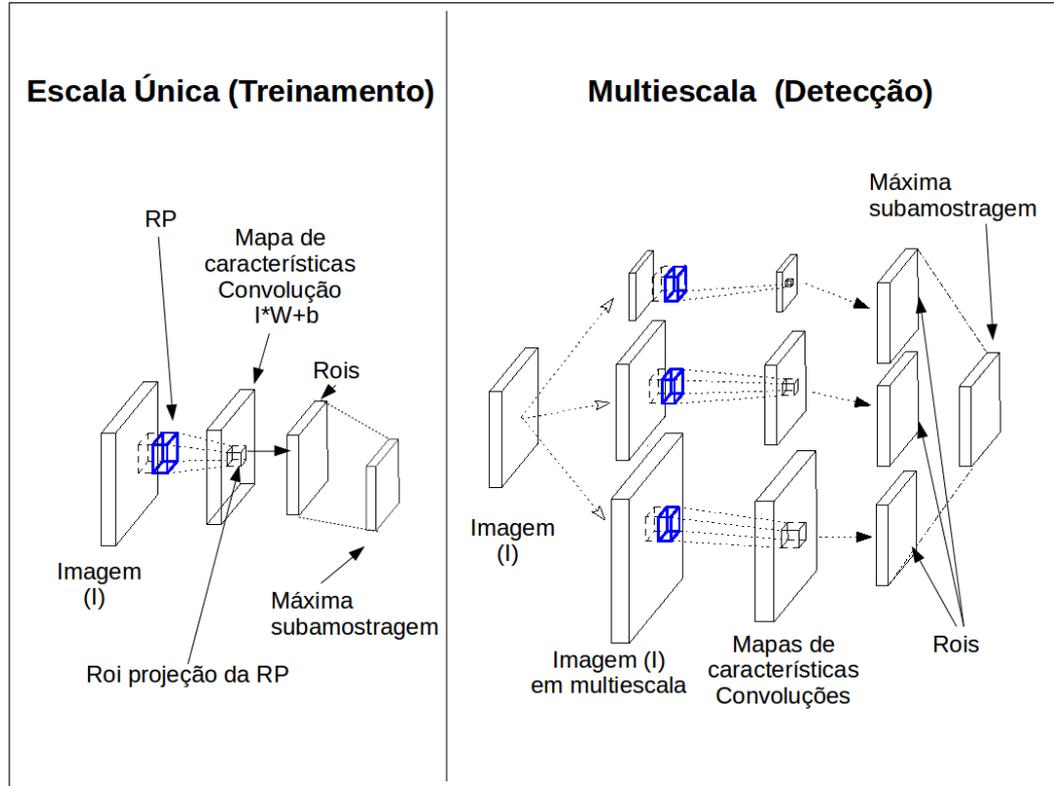


Figura 3.33 – Esquema representação multiescala e escala única. A técnica de escala única durante o treinamento, além de permitir a retropropagação do erro nas camadas convolucionais. A técnica de multiescala durante a detecção permite detectar objetos e confere a rede a capacidade de generalização com relação a escala.

A função custo calculada durante o treinamento da FRCNN é diferenciada, pois analisa o resultado por classe e por localização da caixa delimitadora da região do objeto. Para cada RP, a função custo é,

$$L_{reg}(p_i, t_i^R, t_i^v) = \sum_{i=1}^{N_{reg}} L_1^{smooth}(t_i^R - t_i^v) \quad (3.39)$$

onde o p_i é a nota referente à RP, N_{reg} é o número de RP definida por imagem, t^v é a coordenada referente à região verdade definida por:

$$t^v = [(x^v - x^p)/w^p, (y^v - y^p)/w^p, \log w^v/w^p, \log h^R/h^p] \quad (3.40)$$

e t^R coordenada referente à região predita pela rede definida por:

$$t^R = [(x^R - x^p)/w^p, (y^R - y^p)/w^p, \log w^R/w^p, \log h^v/h^p] \quad (3.41)$$

sendo x^R e y^R , x^v e y^v e x^p e y^p são respectivamente as coordenadas do centro da região predita, da região verdade e região proposta em x e y. As variáveis w^R e w^R , w^v e w^v e w^p e w^p são respectivamente as coordenadas da largura da região predita, da região verdade e região proposta. Os termos h^R e h^R , h^v e h^v e h^p e h^p são respectivamente as coordenadas da altura da região predição, da região verdade e região proposta, como apresentada na figura 3.34.

A idéia é baseada na utilização de L_1^{smooth} [22] que é um método de estatístico de regularização utilizado para penalizar coeficientes grandes, baseado na norma 1, também chamada de penalização da norma L1 definida por:

$$L_1^{smooth}(t_i^R - t_i^v) = \begin{cases} 0.5(t_i^R - t_i^v)^2, & \text{se } |(t_i^R - t_i^v)| < 1 \\ |(t_i^R - t_i^v)| - 0.5, & \text{senão} \end{cases}, \quad (3.42)$$

Esse recurso é utilizado em problemas de classificação que buscam reduzir a diferença entre o estimado e o real, onde são necessários o controle da função custo em relação à utilização de parâmetros adicionais de localização, relevante para a capacidade de generalização [166].

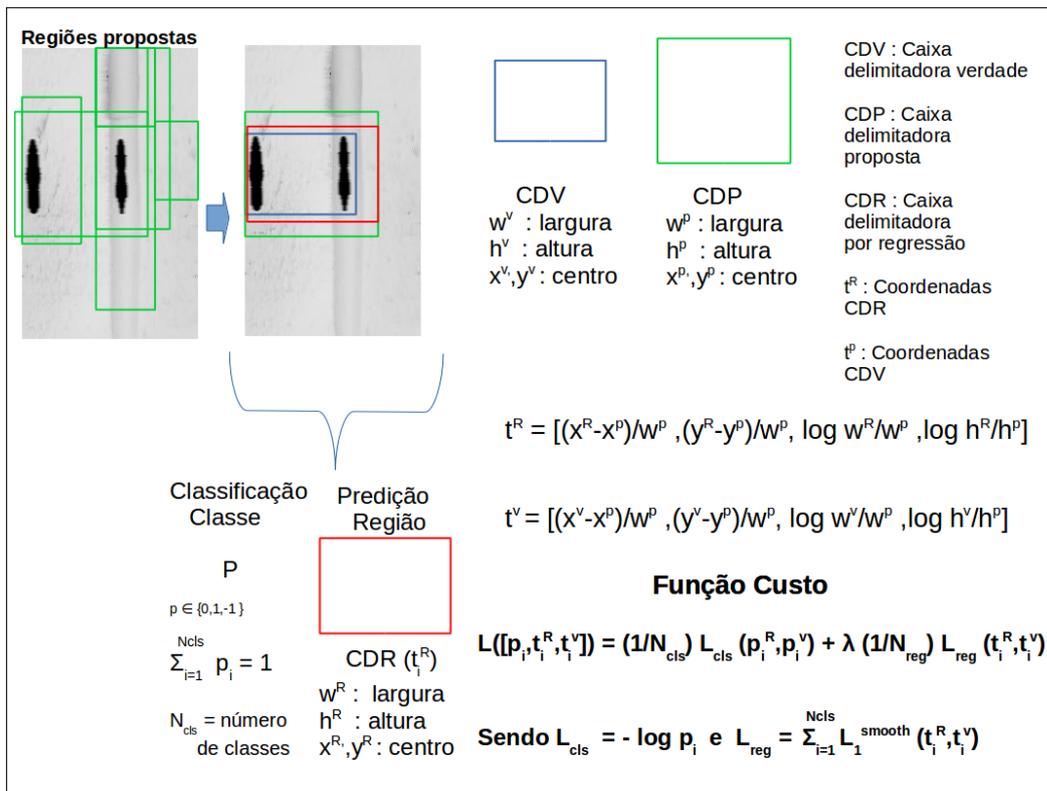


Figura 3.34 – Esquema de regressão de regiões de detecção. A função custo calculada durante o treinamento da FRCNN é diferenciada, pois analisa o resultado por classe e a localização da caixa delimitadora da região do objeto.

No geral, método FRCNN apresenta algumas vantagens, uma delas é a maior qualidade de detecção em relação a RCNN, e a forma de utilização das RP [22]. O treinamento é de um único estágio, isso significa que é o processo é realizado na imagem inteira, usando controle através da função custo definida para prever notas e regiões de detecção, atualizando todas as camadas de rede e armazenando as características durante o processo.

São incorporados à rede ajustes finos para limitação das áreas de detecção no final da rede. Treinar todos os pesos da rede com a última propagação é uma capacidade importante, além da vantagem de compartilhamento de recursos de CPU e GPU durante o treinamento.

Nessa rede há duas camadas de saída como visto na figura 3.35. A primeira produz uma distribuição de notas discretas de cada detecção por classes, utilizando a função *softmax*, detalhada na seção 3.4.2.3. A segunda determina as coordenadas resultados da detecções.

Utilizando o treinamento supervisionado com os dados de entrada, ajusta os erros na retropropagação. A diferença importante do treinamento de uma FRCNN em relação a CNN é que uma das camadas será a Roi máxima subamostragem, logo os filtros ajustados serão baseados na saída dessa camada.

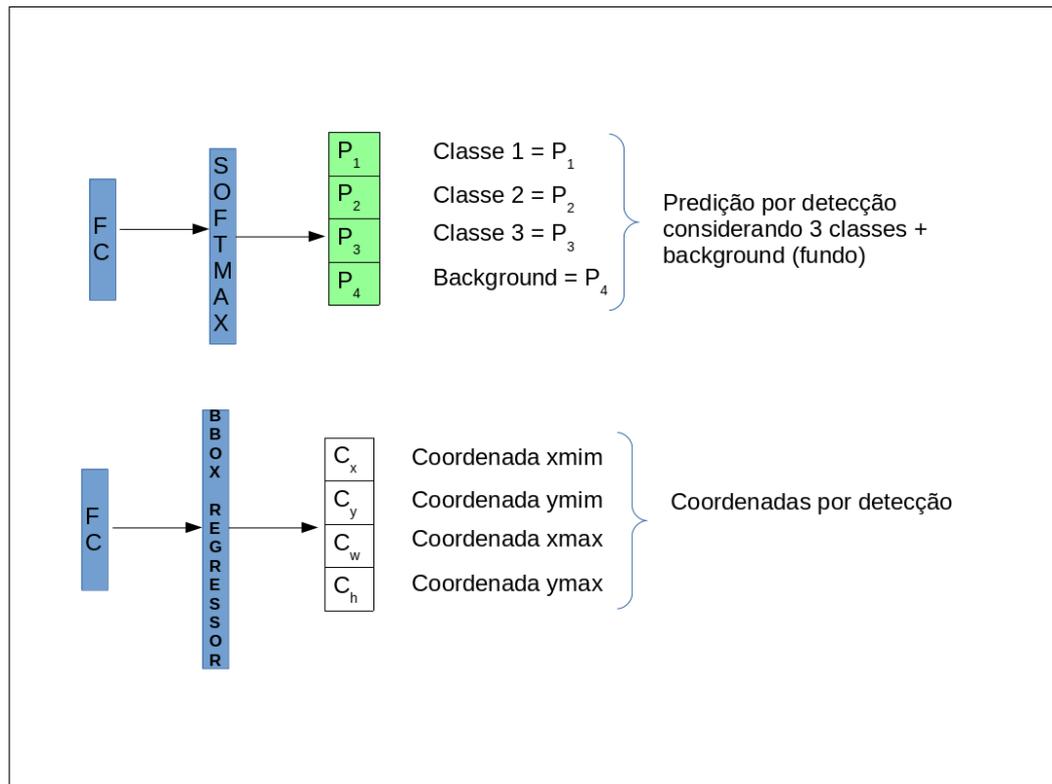


Figura 3.35 – Esquema de saídas da FRCNN considerando 3 classes. A primeira produz uma distribuição de notas discretas de cada detecção por classes, utilizando a função *softmax*, e a segunda realiza a regressão para determinar o ajuste da região de detecção, definindo as coordenadas dos resultados.

3.5.2 Aferição em Novos Dados

A aferição em novos dados ou detecção cega é a aplicação do modelo de rede, ou seja, a rede já treinada, em novos dados de origem diferente dos dados de treinamento, sendo dados onde a rede não enxerga a pré-classificação das ocorrências. Diferente da validação cruzada utilizando os lotes de validação e teste dos dados de treinamento que verifica a qualidade do treino, esse novo teste na fase posterior o modelo da rede já é capaz de realizar predições em novos dados de qualquer origem sem pré classificação. O intuito dessa nova avaliação é analisar a generalização, abstração e tolerância a falhas, características inerentes a um sistema de redes neurais artificiais eficiente. Para isso, o

conjunto de treinamento deve ser abundante e variado o suficiente, para que a rede seja capaz de classificar corretamente novas imagens que não estavam presentes no processo de treinamento.

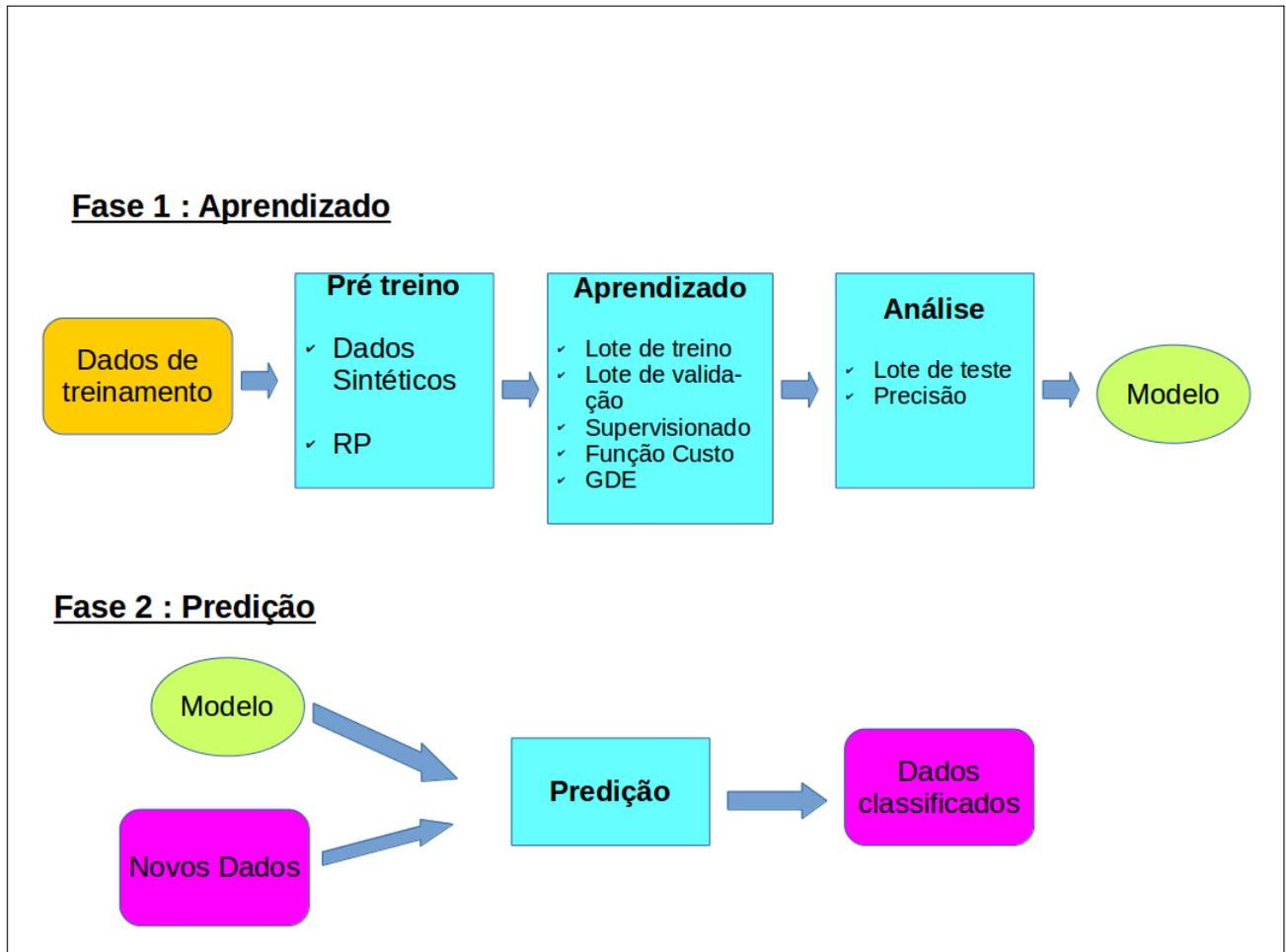


Figura 3.36 – Diferença entre a fase de treinamento e fase utilização do modelo de predição em novos dados. O intuito dessa nova avaliação é analisar a generalização, abstração e tolerância a falhas, características inerentes a um sistema de redes neurais artificiais eficiente.

CAPÍTULO 4

MÉTODOS CLÁSSICOS DE DETECÇÃO AUTOMATIZADA DE ESTRUTURAS

A possibilidade de identificar, realizar estimativas e caracterizar utilizando imagens gerou a demanda de muitos tipos de recursos computacionais para essas tarefas. Ainda assim, muita das vezes parte ou mesmo a totalidade do processo de classificação ainda exija um operador humano. O sistema visual humano possui notável capacidade de reconhecer padrões. Entretanto, ele dificilmente é capaz de processar o enorme volume de detalhes presente numa imagem. No entanto, processos cada vez mais automatizados e eficazes surgem com o objetivo de tornar mais produtivo o trabalho de profissionais de diversas áreas. Nesse capítulo apresentamos algumas das técnicas automáticas clássicas comumente utilizadas para identificar objetos, que empregamos afim de localizar fraturas e desmoronamentos em imagens acústicas de poço, baseados em estudos anteriores de [5, 11, 12]. Entre os modelos aplicados utilizamos técnicas de Processamento digital de imagens PDI utilizando o *software Matlab*[®] em todas as etapas.

4.1 Processamento Digital de Imagens

Um sistema de processamento digital de imagens (PDI) é utilizado com o objetivo de reconhecimento e caracterização de imagens, a partir da manipulação através de operações matemáticas afim de fornecer a identificação e a extração de fraturas naturais e desmoronamentos contidos nas imagens, para posterior interpretação. Isso porque vários tipos de degradações provenientes dos processos de aquisição, transmissão de dados e visualização, podem ser encontrados na imagem acústica. Logo a necessidade de técnicas mais automatizadas. A cadeia de PDI proposta por esse estudo é composta de várias etapas como apresentado na figura 4.1. As etapas começam com a aquisição da imagem, depois a digitalização da imagem, seguida de pré-processamento que é uma preparação prévia para a

segmentação, utilizada para separar os objetos. Após isso, é aplicado o pós-processamento onde são utilizados recursos computacionais para destacar as partes pertencentes ao objeto de interesse. Por fim, é realizada a extração de atributos dos objetos que contribuem para a etapa final de classificação e reconhecimento dos mesmos.

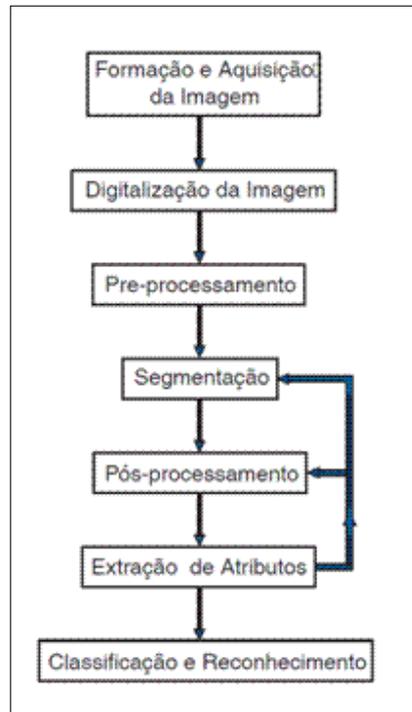


Figura 4.1 – As etapas do processamento digital de imagens PDI propostas nesse estudo é definida desde a aquisição até a classificação e reconhecimento de fraturas e desmoronamento na imagem.

4.1.1 Pré-processamento

No pré-processamento de uma imagem o objetivo é corrigir distorções, minimizar ruído proveniente dos processos de aquisição e destacar sua forma para auxiliar a tarefa de segmentação. Para isso, utilizamos filtros de realce de contraste. O contraste entre dois objetos é definido pelo limite que os separa entre dois níveis da escala de cor utilizada. Quando dois objetos possuem seus níveis de cor muito próximos é difícil identificar seus contornos. Dessa forma, aumentar o contraste é deixar esses objetos bem distintos a ponto de facilmente identificar suas fronteiras. Quando a imagem é muito grande, há a necessidade de definir áreas específicas para aplicação de filtros na imagem, por isso regiões podem ser definidas com a finalidade de refinar o tratamento e a busca por objetos de interesse.

4.1.1.1 Regiões de Interesse

A imagem acústica real é composta de milhares de linhas de pixel quando digitalizada. Separar regiões de possíveis ocorrências contribui no processo de filtragem e diminui o tempo de processamento durante as demais etapas. Como cada material contido no poço e cada ocorrência geológica podem ser representados na matriz rochosa devido as mudanças de intensidade das suas cores, podemos descrever os elementos contidos em uma imagem como diferentes tipos de texturas. Para realizar a análise textural e identificar melhor as Regiões de Interesse (ROIs), utilizamos um processo que obtêm através da imagem as suas matrizes de co-ocorrências de níveis de cinza (GLCM) da sigla do termo em inglês *Gray Level Cooccurrence Matriz* [167, 168]. Essas matrizes M são uma espécie de tabela de probabilidades p de combinações diferentes de valores de intensidade dos pixels (i.e. níveis de cinza) ocorrem em uma imagem. Dessa forma, a matriz de co-ocorrência pode caracterizar uma imagem por texturas utilizando a estatística considerando a relação entre dois pixels (i, j) por vez, um chamado de pixel referência i e o outro de pixel vizinho j considerando quaisquer distâncias e em qualquer direção, tais como a direita, a esquerda, acima, abaixo, ou diagonais. Através dessas matrizes podemos calcular os 16 parâmetros de Haralick [169, 170]. Para cada duas linhas de pixels da imagem, a partir de equações descritas no apêndice A à fim de caracterizar grande parte das texturas encontradas nas imagens. Assim, buscando definir um atributo que melhor identificasse regiões de fraturas e desmoronamentos e verificamos a partir da avaliação visual que o parâmetro de contraste poderia identificar fraturas e o parâmetro agrupamento proeminente (AP) poderia localizar desmoronamentos, através das equações

$$\text{Contraste} = \sum_{n=0}^{N_g-1} n^2 \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right)_{|i-j|=n}, \quad (4.1)$$

$$\text{AP} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - M_x + j - M_y)^4 \cdot p(i, j), \quad (4.2)$$

sendo $p(i, j)$ a probabilidade p definida na matriz M GLCM de de ordem $N_g \times N_g$ de que dois valores de intensidade de cinza (i, j) estejam envolvidos pela relação espacial entre eles como a distância e o ângulo. Essas e as demais equações de atributos de Haralick foram detalhadas no apêndice A.

Os picos nos valores de contraste encontrados acima do valor médio da curva apresentada na figura 4.2 são consideradas possíveis fraturas naturais e contribuem para localizar regiões que possuem fraturas em imagens com muitas linhas de pixels. Assim como, desmoronamentos correspondem a regiões com picos de valores CP, também acima do valor médio da imagem [167, 170]. Nos valores que definem os dois gráficos, são realizadas buscas por trechos de curva que possuam valores acima de 0 e que sejam maiores que 10 linhas de pixel.

Os valores calculados com as equações apresentadas em A.2 e A.26, identificam regiões onde o objeto se destaca respectivamente por contraste ou por ser um aglomerado proeminente de pixels com valores de tons de cinza próximos. Contudo, a aplicação dessa etapa só faz sentido em imagem com uma considerável quantidade de linhas de pixels, já que esse processo de definição de pequenas regiões candidatas facilita o processamento em uma imagem inteira de poço.

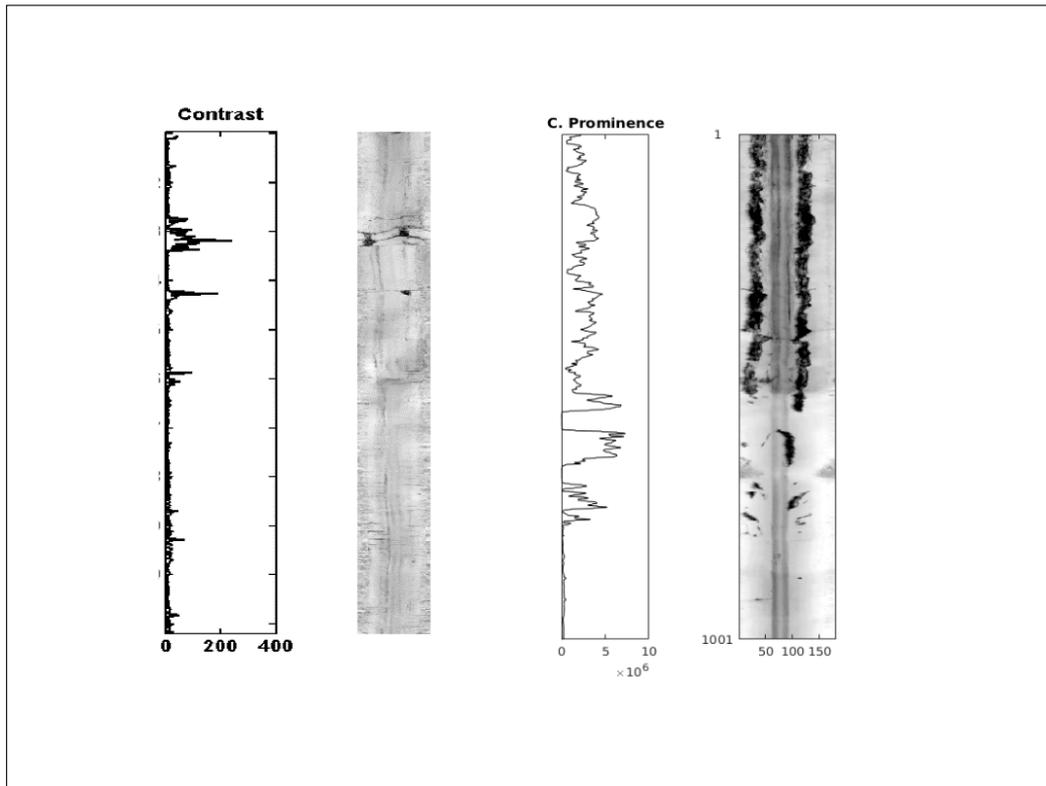


Figura 4.2 – Exemplo de gráficos com as curvas de atributo de Haralick contraste, do termo em inglês *Contrast* e aglomerado proeminente do termo em inglês *cluster prominence*. Os picos nos valores de contraste encontrados acima do valor médio da curva são consideradas possíveis fraturas naturais e contribuem para localizar regiões que possuem fraturas em imagens com muitas linhas de pixels. Assim como, desmoronamentos correspondem a regiões com picos de valores CP, também acima do valor médio da imagem.

4.1.1.2 Filtro Baseado em Correlação

Como partes pertencente à fratura podem ser visíveis na matriz rochosa, buscamos em primeiro lugar maximizar o contraste da fratura em relação ao fundo da imagem, isso contribui para a qualidade da segmentação [171, 172]. Utilizamos uma técnica de filtragem para reduzir o ruído aleatório, acentuar as bordas e retificar problemas de aquisição [173, 174]. Dessa forma, a idéia principal é realçar a fratura ou segmentos dela do restante da imagem. O procedimento é realizado através da filtragem da imagem utilizando a multiplicação de cada pixel da imagem com o valor referente à soma dos valores máximos

(MaxCorr) e mínimos de correlação (MinCorr) [175, 176] da ROI. A função do *Matlab*[®] `imfilter` foi utilizada para realizar a operação de multiplicação entre a escalar resultado da soma por todos os pixels [171, 172, 177]. Dessa forma, valores de pixels mínimos, e.g. regiões escuras mais prováveis como fraturas, continuaram mínimos [177]. No entanto, pixels com valores máximos, e.g. regiões mais claras, consistentes com a matriz rochosa, ficam ainda maiores e mais claras [176, 177].

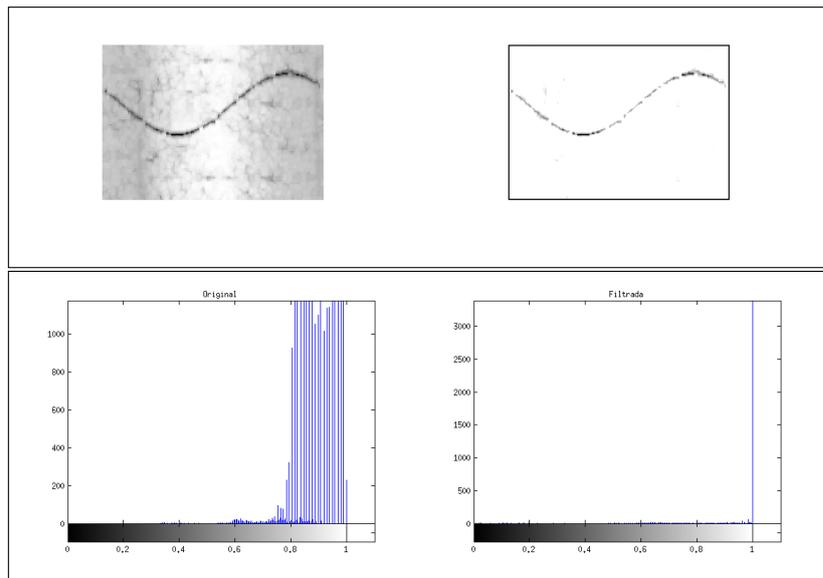


Figura 4.3 – Resultado de imagem utilizando filtro baseado em valores de correlação. A imagem da esquerda é a imagem original, à direita apresentamos a imagem após filtro utilizando correlação. Embaixo de cada imagem seu respectivo histograma, observamos que após o filtro as intensidades distribuídas foram filtradas sobrando apenas um pico bem definido de frequência para os valores de pixels da fratura. Nesse caso em particular, o filtro evitou problemas de disparidades no fundo, facilitando o trabalho dos algoritmos de segmentação. Uma borda foi introduzida propositalmente à imagem filtrada com a finalidade de especificar melhor os limites da imagem, para avaliação através da observação visual.

4.1.2 Segmentação

O processo de segmentação tem o objetivo de separar objetos do fundo da imagem, identificando e destacando-os do fundo, e.g. [178, 179, 180]. Como resultado, podemos obter um conjunto de objetos ou bordas extraídas da imagem. Os pixels componentes da mesma região são similares em alguma característica de cor, intensidade, textura ou continuidade. Desse modo, regiões diferentes devem possuir diferenças significativas nesses parâmetros.

Entretanto, diversos fatores podem aumentar o erro na definição de fronteiras de um objeto em uma imagem como, por exemplo, defeitos oriundos da aquisição da imagem [171, 172] ou indefinição de bordas devido à níveis de cor muito próximos, i.e. pouco contraste. Por isso, é comum o uso de um pré-processamento e um ou vários processos

de filtragem, antes da segmentação e.g. [171, 181]. Aliada à essas dificuldade existe a ausência de uma técnica única de segmentação que atenda a todos os tipos de imagens. Por isso, existem vários algoritmos e técnicas que devem ser avaliados e testados para atender a aplicação desejada [172, 182, 183]. À seguir, apresentamos algumas das técnicas que testamos nas imagens de fraturas e desmoronamentos [184] e que são diferentes opções no algoritmo de detecção de desmoronamentos e fraturas por processamento digital de imagens.

4.1.2.1 Gradiente Direcional

Considerando uma imagem definida como $f(x,y)$, o gradiente direcional é definido por:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (4.3)$$

sendo $\frac{\partial f(x,y)}{\partial x}$ e $\frac{\partial f(x,y)}{\partial y}$ as derivadas parciais em relação a x e y . Assim, respectivamente, os gradientes direcionais G_x e G_y correspondem as variações no sentido horizontal e vertical, logo realçam bordas com variação horizontais e verticais [11, 185]. Com os novos valores de pixels computados, as bordas passam a ter os valores máximos. Dessa forma, é possível obter apenas os pixels com os valores máximos no direcional desejado facilitando a sua segmentação [185, 186]. Testamos visualmente e optamos por utilizar o gradiente direcional em y , pois o padrão senoidal maximiza o gradiente nessa direção.

4.1.2.2 K-médias

O K-médias ou do termo em inglês *K-means*, é um dos algoritmo desenvolvidos para agrupamento de dados [181, 187, 188, 189, 190, 191]. O princípio básico do algoritmo é apresentar uma classificação baseada na análise e distinção entre os seus valores numéricos, i.e., neste caso, de intensidade dos pixels. O algoritmo inicializa definindo aleatoriamente os centróides K e associa cada pixel i a uma dessas classes. Como critério para pertencer a classe são definidas as menores distâncias euclidianas dos centróides. Após esse procedimento ele define novos centróides ou médias de cada classe ou agrupamento (i.e. *cluster*) e repete o processo de redefinição das classes por um máximo de N interações ou quando nenhum ponto for associado a uma classe diferente da iteração anterior.

O algoritmo assume que os atributos dos pontos da imagem formam um espaço vetorial, minimizando a variância dos pontos que estão dentro de uma determinada classe [191]. Dessa maneira equivale a minimizar o somatório do erro quadrático (SEQ_K), referente a perspectiva de otimização do cálculo da variância intra-cluster definindo,

$$SEQ_K = \sum_{i=1}^N \sum_{c=1}^K \mu_{ic} |x_i - \bar{x}_c|^2 \quad (4.4)$$

Sendo $\sum_{c=1}^K \mu_{ic} = 1$ e $\mu_{ic} \in \{0, 1\}$.

A qualidade da solução final depende do número de classes K pré-definida [191], como podemos verificar em 4.4 e 4.5 onde temos os resultados da segmentação da fratura utilizando 2 e 3 agrupamentos. Nessas imagens podemos observar que o melhor resultado foi alcançado utilizando 3 clustes, com o resultado para objetos cluster3. A necessidade de 3 agrupamentos é vista observando a imagem original, onde é possível identificar que o fundo possui 2 tipos de tons de cinza e a fratura um tom mais escuro que os anteriores.

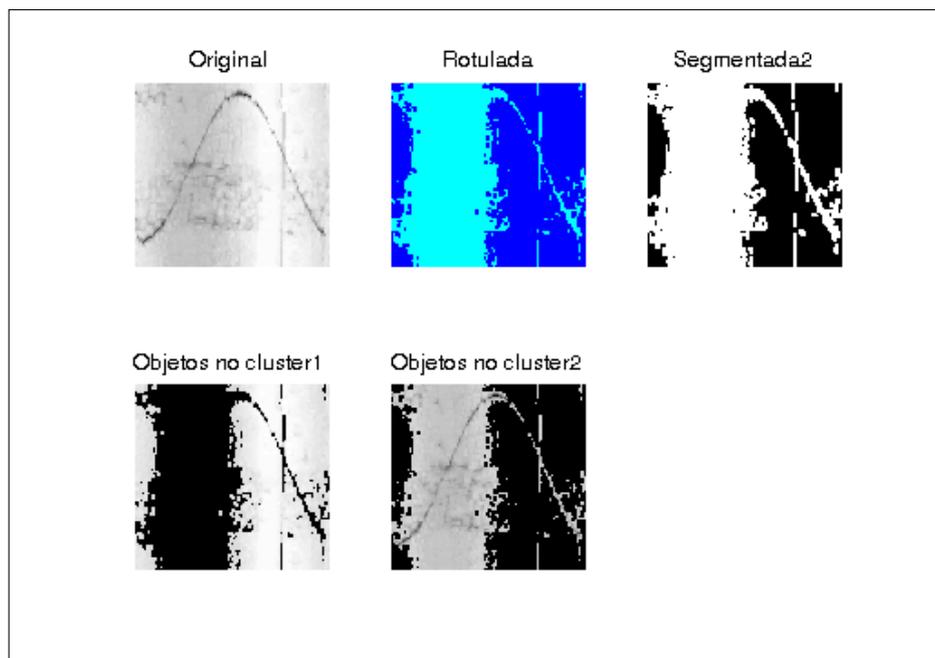


Figura 4.4 – Imagens resultados de segmentação utilizando K-médias para 2 clusters. Resultado de segmentação para 2 classes (clusters). Nas 3 imagens de cima da esquerda para direita temos respectivamente, o trecho de imagem de entrada, o mesmo trecho rotulado pelas 2 classes por cores diferentes e resultado da escolha do cluster 2 segmentado. Abaixo as duas classes separadas. Dessa forma, para 2 classes não seria a melhor para separar a fratura do fundo da imagem.

4.1.2.3 Otsu

Otsu é um processo de limiarização bimodal considerado como sendo o particionamento dos pixels de uma imagem com N níveis [192] em duas classes 0 e 1, i.e. binarização, que representam respectivamente componentes do fundo e do objeto [181, 192]. Para se obter um limiar entre as duas classes, é calculado um valor de corte conhecido como thresholding T . Essa medida é determinada a partir do histograma da imagem, como apresentado no exemplo da figura 4.10, onde as frequências dos valores dos pixels do objeto e fundo da imagem são separado por um corte. O T é *thresholding*, o valor encontrado

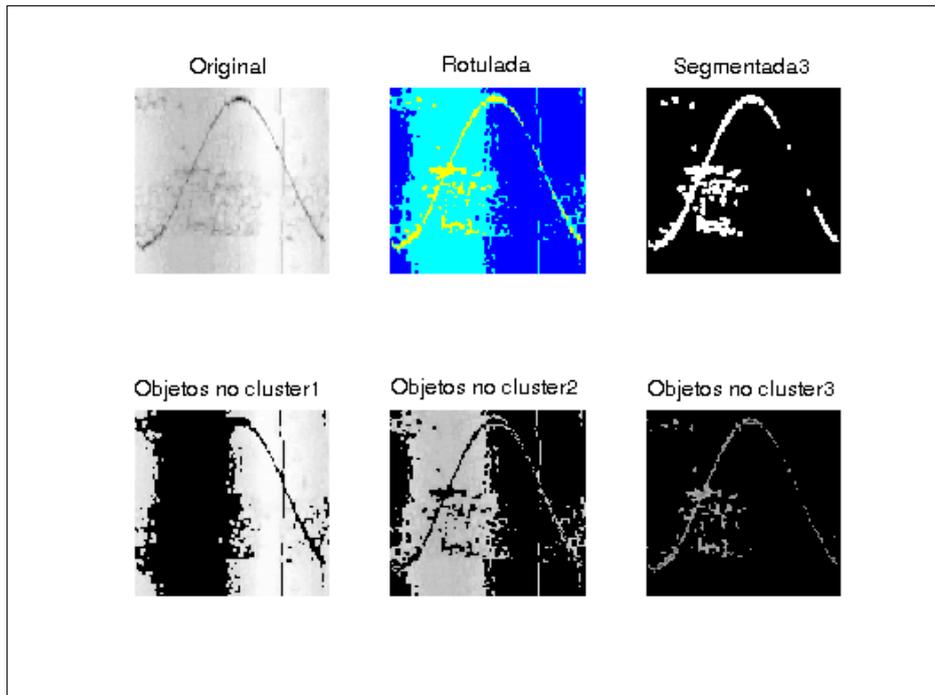


Figura 4.5 – Imagens resultados de segmentação utilizando K-médias para 3 clusters. Resultado da mesma imagem segmentada na figura 4.4, porém dessa vez para 3 classes (clusters). Nas 3 imagens de cima da esquerda para direita, o trecho de imagem sintética de entrada, o mesmo trecho rotulado pelas 3 classes por cores diferentes e resultado da escolha do cluster 3 segmentado. Abaixo as três classes separada. Dessa forma, para 3 classes o resultado desejado consiste na melhor opção de número de classes, já que a segmentação alcançou uma separação maior do fundo.

para corte de separação. Na figura 4.7 apresentamos um exemplo com imagem e seus respectivos histogramas com corte definido pelo processo Otsu. A imagem da esquerda apresenta abaixo o seu histograma com o corte em vermelho feito pelo processo. A imagem binarizada, em preto e branco, é o resultado da segmentação utilizando o corte. Em imagens onde o objeto e o fundo são bem definidos o corte é realizado com sucesso.

4.1.2.4 Niblack

O método de limiarização local proposto por Niblack [193], cuja idéia central é definir um limite ou thresholding T definido por:

$$T = \mu + (c\sigma), \quad (4.5)$$

onde μ é a média em uma janela de vizinhança c é uma constante empírica de valor -0.2 definida pelo autor [194], i.e. representa o peso do desvio e σ um desvio padrão local. O método de Niblack é sensível à definição do tamanho da vizinhança [195] permitindo uma análise local. As dimensões escolhida para a janela deslizantes [11] devem ser pequenas suficiente para preservar detalhes locais evitando problemas de aquisição de ou-

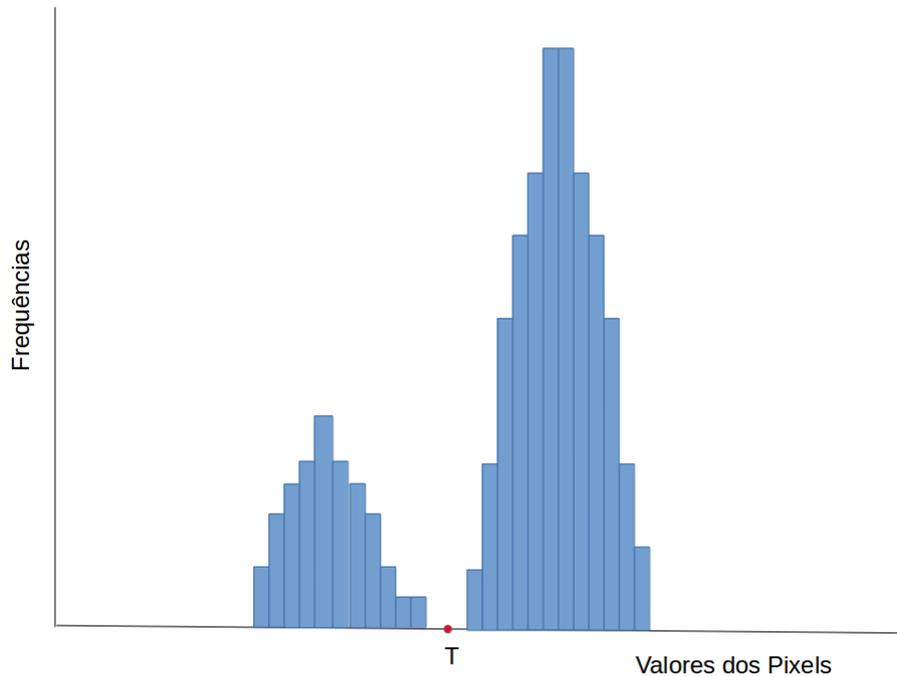


Figura 4.6 – Exemplo de histograma característico utilizado pelo método Otsu para determinar objeto e fundo da imagem. O T é *thresholding*, o valor encontrado para corte de separação.

tras partes da imagem, porém suficiente grande para não produzir mais ruídos [196, 197]. Após alguns testes, definimos uma janela 15×15 para as imagens utilizadas nesse trabalho.

4.1.2.5 Entropia Local

A entropia de um sistema [181, 198] é definida como uma função de probabilidade de diferentes estados. Em um sistema com x estados diferentes com probabilidades de evento p_i desses estados, onde $i = 1, 2, 3, \dots, x$ e $\sum_{i=1}^x p_i = 1$, o evento i apresenta um rendimento de informação indicado por:

$$\Delta I = -\log_2 p_i. \quad (4.6)$$

Segundo Shannon [198], a entropia de um sistema é expresso por:

$$H = -\sum_{i=1}^x p_i \log_2 p_i. \quad (4.7)$$

Esse conceito foi aplicado em sistemas de informações e assim na extração de informação de imagem, relacionando os diversos valores de pixels de uma imagem com os diferentes estados de um sistema. A entropia de uma imagem está relacionada ao grau de informações obtidas por meio de histograma da imagem ou definida localmente através

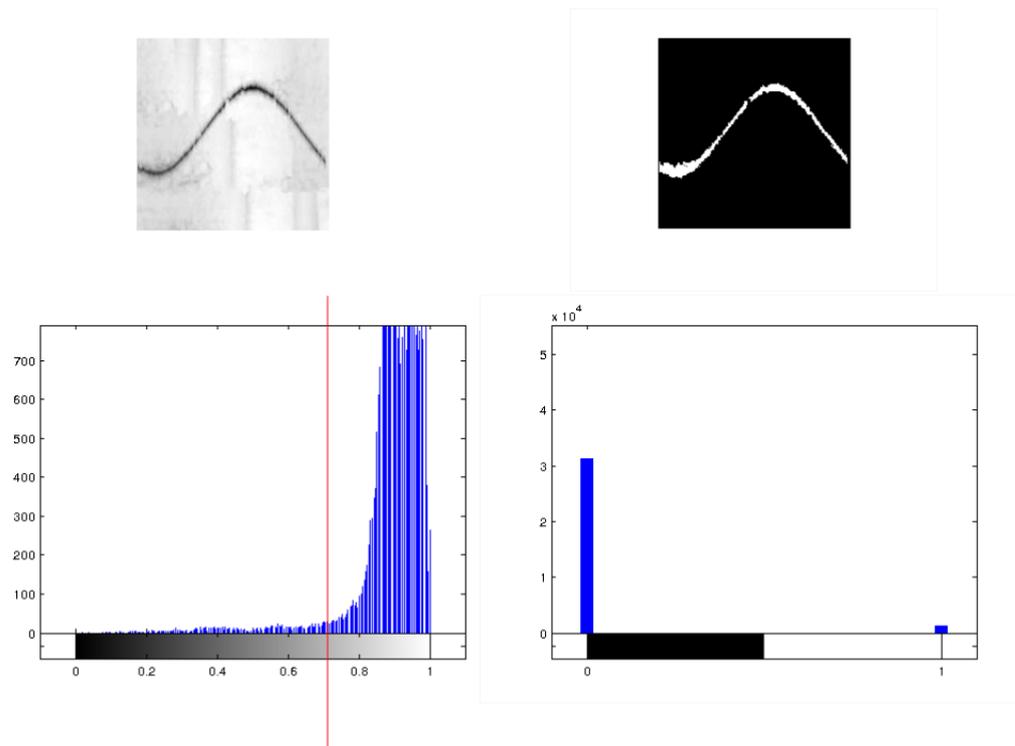


Figura 4.7 – Exemplo de imagem e seu respectivo histograma com corte definido pelo processo Otsu. A imagem da esquerda apresenta abaixo o seu histograma com o corte feito pelo processo Otsu. A imagem binarizada, em preto e branco, é o resultado da segmentação utilizando o corte. Em imagens onde o objeto e o fundo são bem definidos o corte é realizado com sucesso.

de varredura utilizando janela deslizante, calculando o valor de cada pixel associando a entropia da sua vizinhança definida pela janela [199, 200]. Esse processo local realiza uma filtragem por filtro de entropia local. A utilização do conceito de entropia e a maximização da entropia tem sido amplamente utilizado para separar fundo e objeto, ou diferentes fases de uma imagem (e.g. [181]). O filtro de entropia local é utilizado para detecção de bordas e segmentação, e.g. [199, 200, 201]. Este filtro realça regiões da imagem com maior entropia, i.e. com maior variação de valores de pixel na região da janela, tipicamente são as regiões de transição entre duas texturas. Como se trata de um processo localizado, este filtro se destaca por ser pouco sensível a certas fontes de erros sistemáticos na aquisição da imagem tal como de iluminação da cena em que a imagem é gerada e ainda é possível compensar não uniformidades nas condições de iluminação como sombras ou reflexos locais. Após a filtragem é possível localizar os pontos da imagem com os valores máximos, referentes aos possíveis pixels componentes de fraturas. Após alguns testes visuais, utilizamos uma janela 9×9 .

4.1.3 Pós-processamento

Após a etapa de segmentação, o resultado é uma imagem onde o objeto está apresentado em um fundo preto composto de pixels branco, e.g. uma imagem binária [202]. Durante a fase de pós-processamento são aplicadas operações morfológicas de dilatação de regiões com pixels brancos e retirada de áreas menores que 50 pixels.

A dilatação em uma imagem binarizada é um processo utilizado para unir partes de objetos. Considerando uma imagem A como uma matriz com elementos 0 e 1 com dimensões 6×6 , e B uma matriz quadrada 3×3 com todos elementos de valor 1, a dilatação da imagem é realizada pela operação lógica *xor* entre A e B, definida por:

$$\text{Dilatação (Imagens Binárias)} = A \oplus B \quad (4.8)$$

onde a matriz B é considerada uma janela deslizante ou conhecida como elemento estruturante. Assim, quando B desliza sobre A os elementos de A são comparados com os de B e cada elemento de A é alterado de acordo com a relação da tabela 1.

Tabela 1 – Tabela relação xor.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Uma função conhecida como *imdilate* do *software Matlab*[®] foi utilizada, para realizar a dilatação, para unir partes de objetos em imagens binarizadas, onde aplicamos para compensar as perdas das bordas ocorridas durante a segmentação [189, 203].

A expansão da área de pixels segmentados é realizada de acordo com o tamanho da janela deslizante que analisa a vizinhança de cada pixel recalculando o seu valor caso necessário aumentando assim os limites dos objetos, como visto em 4.8 e 4.9.

4.1.4 Extração de Atributos

Na etapa de extração de atributos, a função *regionprops* em *Matlab*[®] foi utilizada para rotular cada objeto segmentado como visto na figura 4.9. Para cada objetos são calculados diversos atributos como área, posição do centro de massa, coordenadas dos pixels e orientação [204]. O principal objetivo da extração de atributos é caracterizar os objetos para serem reconhecidos através de medidas sobre a imagem que possam indicar algum padrão comum entre objetos pertencentes à mesma classe [205]. Com esses atributos é possível retirar dessa imagem informações que associem as características dos objetos segmentados para a classificação como fratura ou desmoronamentos.

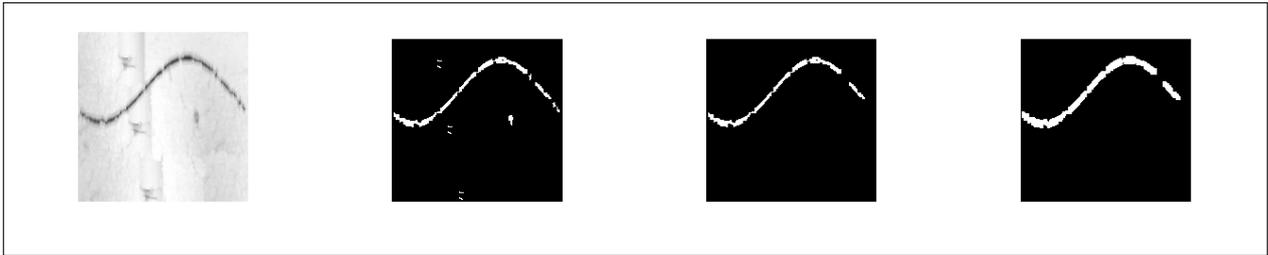


Figura 4.8 – Imagens de resultados das etapas de PDI, onde da esquerda para direita apresentamos a imagem original com fratura sintética, a imagem binária resultado da segmentação, a imagem após retirada de áreas menores de 50 pixels e a imagem após a dilatação utilizando janela 3×3 pixels.

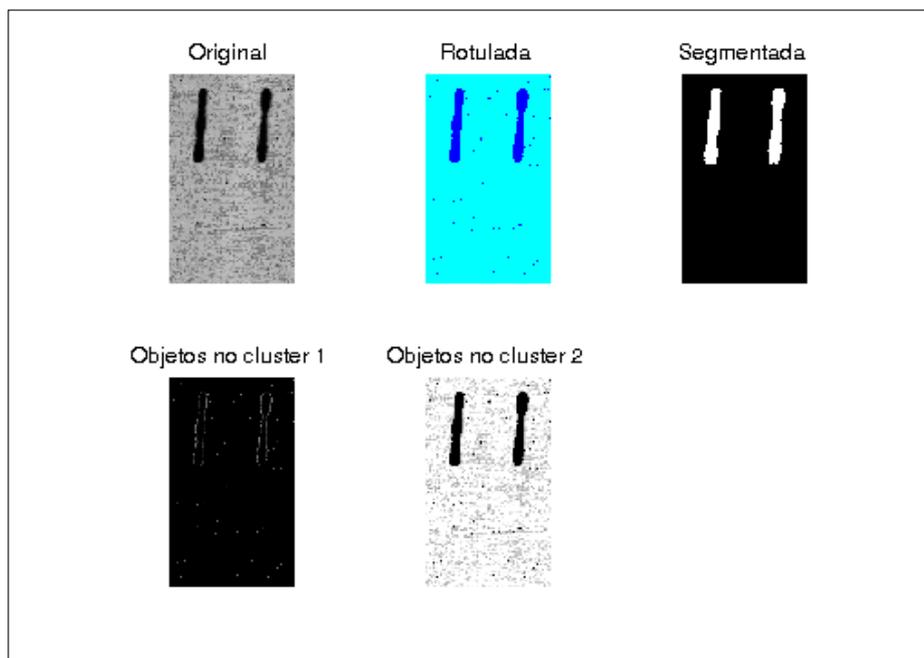


Figura 4.9 – Imagem com fundo muito ruidoso resultado do processo de aquisição da imagem, seguido de resultados do algoritmo de detecção de desmoronamentos, utilizando etapas de pré-processamento detalhada em 4.1.1, método de segmentação K-means descrito em 4.1.2.2 escolhido de forma automática para 2 clusters e pós-processamento detalhado em 4.1.3, eliminado pequenas áreas e utilizando dilatação.

4.1.5 Classificação e Reconhecimento

A classificação é o processo de visão computacional de extração de informações em imagens onde são utilizados dados que identificam o quão similar é o objeto ao padrão de interesse [206]. Parte da premissa de que a similaridade entre objetos implica que eles possuam características compatíveis, correspondente a mesma classe ou agrupamento. Geralmente o resultado da classificação é um valor percentual ou nota indicando a probabilidade da ocorrência do objeto pertencer a uma determinada classe [207]. Uma fratura por exemplo, apesar de possuir um padrão similar à uma senóide, pode conter descon-

tinuidades causadas pela aquisição, por isso a nota que define a sua classificação seria baixa. De outra maneira, podemos considerar que a fase de classificação consiste em reconhecer um objeto em uma imagem [208]. Três metodologias baseadas em PDI foram adotadas para classificação e reconhecimento de fraturas nomeadas como Método de Classificação por Polinômio, Método Transformada de Radon e Método Transformada de Hough. E o método de encontrar desmoronamentos é nomeado como Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).

4.1.5.1 Método de Classificação por Polinômio

Para classificar os objetos segmentados na etapa anterior, utilizamos nessa abordagem, o ajuste por um polinômio do quarto grau chamado de Método de Classificação por Polinômio (*MCP*). Esse tipo de abordagem, é análogo a desenvolvida por Ruela [11]. Segundo a autora, em quaisquer situação, o padrão da fratura é correlacionado à uma curva polinomial de 4^o grau. No entanto, nesse trabalho desenvolvemos um algoritmo que antes analisa os atributos dos objetos segmentados quanto à largura e simetria de pontas e compara os pixels com a melhor curva de ajuste de um polinômio do quarto grau. Considerando a imagem segmentada uma matriz com dimensões $M \times N$, o Método de Classificação por Polinômio (*MCP*) primeiro avalia a largura dos objetos segmentados [184]. Caso um deles ultrapasse à 50 % da largura N , considera uma possível fratura, eliminando outros objetos. O método depois procura objetos que possuam pixels localizados na laterais da imagem em profundidades iguais unindo esses objetos.

Obter uma função matemática que ajuste os pontos baseado no método de mínimos quadrados, permiti ajustar medidas à uma curva ou reta. Para a análise de dados, ajustar pontos significa encontrar a curva ou reta que passa pela maior quantidade de pontos avaliando cada aproximação comparando os resíduos estatísticos. De acordo com a autora Ruela [11], a curva que melhor se adapta ao padrão das fraturas são ajustados por mínimos quadrados de um polinômio de 4^o grau.

Segundo Ruela, considerando a geometria do poço, uma metodologia baseada no ajuste polinomial é justificada, pois atende a ocorrências de fraturas de poço em condições cilíndricas ou ovalizadas geradas pela perfuração. No aspecto matemático, uma senoide pode possuir um ou no máximo dois pontos de inflexão, então polinômios de quarto atendem esse critério. Além disso, o *MCP* utilizou em média 0,03 segundo de processamento computacional por imagem.

Cada objeto candidato a fratura apontado pela técnicas *MCP* têm as posições de seus pixels ajustados pelo polinômio, apresentando uma curva resultante do ajuste dos dados por mínimos quadrado. Através do coeficiente de correlação de Pearson R entre as posições do pixels dessa curva e dos pixels do objeto, consideramos os percentuais ou notas que classificam o objeto como fratura. O coeficiente de correlação R entre as distribuições

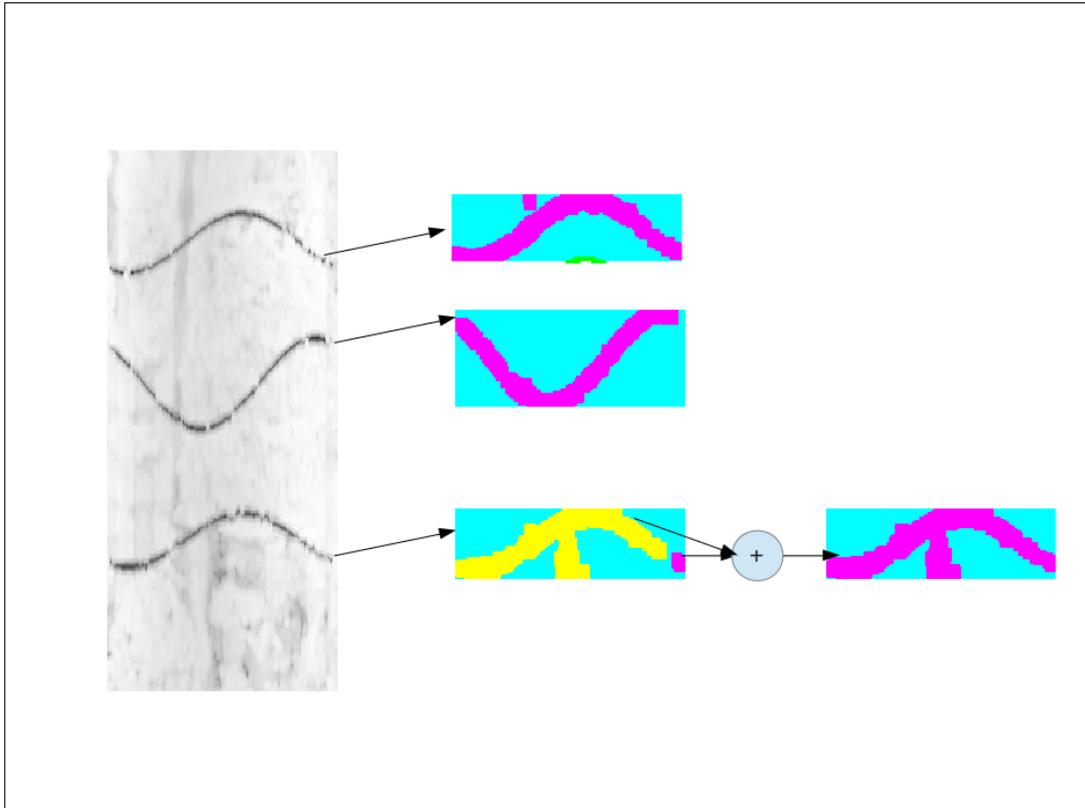


Figura 4.10 – Resultado da utilização dos critérios aplicados pelo Método de Classificação por Polinômio, as duas primeiras fraturas foram segmentadas por completo. Porém uma pequena região da última fratura, foi classificada primeiramente pertencente à outro objeto, mas após a verificação dos critérios os dois objetos formaram apenas um.

de valores x_1, x_2, \dots, x_n referente aos pontos dos pixels encontrados e y_1, y_2, \dots, y_n definidos pelos pontos interpolados com o ajuste polinomial são obtidas por:

$$\mathbf{R} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (4.9)$$

onde \bar{x} e \bar{y} são os valores médios das distribuições. No exemplo apresentado na figura 4.11 a caixa delimitadora em verde determina a região do pixels classificados como fratura e na caixa azul a respectiva nota final determinada pelo coeficiente de correlação R . Dessa forma, como informação de saída em um arquivo formato ASCII, o código apresenta a localização de uma caixa delimitadora contendo as dimensões da fratura, como os quatro números ($x_{min}, y_{min}, x_{max}, y_{max}$) representando as coordenadas da caixa, assim como o valor a sua nota atribuída a classificação do objeto como fratura pelo ajuste do polinômio.

4.1.5.2 Método Transformada de Radon

O Método Transformada de Radon (MTR) proposto utiliza a transformada de Radon das imagens para classificar como fraturas, objetos segmentados e binarizados. Segundo Cornet [12], é possível encontrar curvas identificando senóides e classificar como fraturas

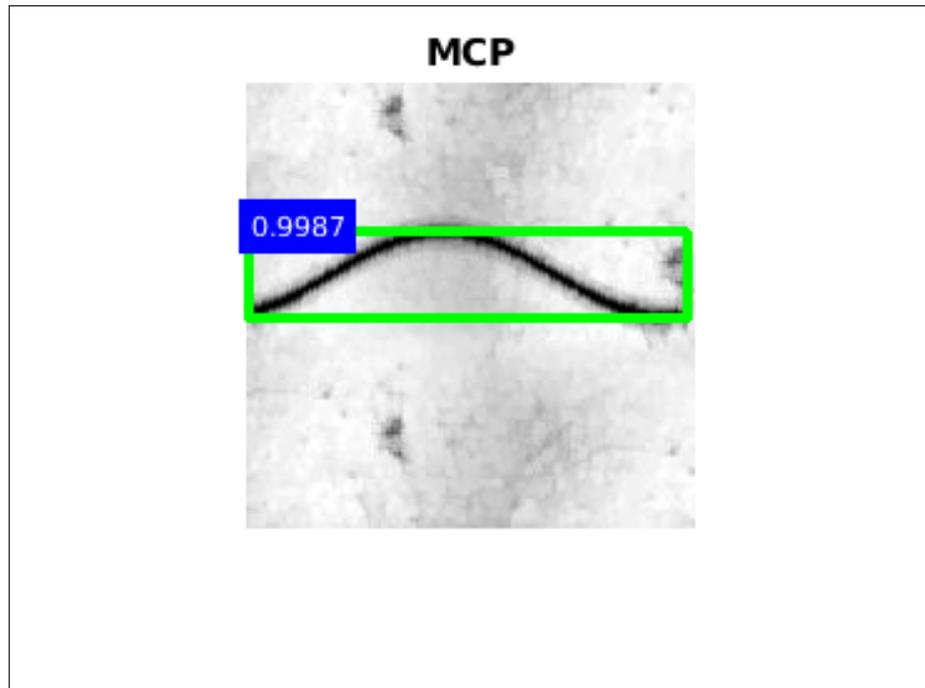


Figura 4.11 – Resultado de detecção na imagem utilizando Método de Classificação por Polinômio. A caixa delimitadora em verde determina a região dos pixels classificados como fratura e na caixa azul a respectiva nota.

naturais utilizando a Transformada de Radon. Antes de utilizar o MTR, para transformar a imagem resultado da segmentação em uma curva mais refinada como apresentada na figura 4.12 o algoritmo utilizado calcula o gradiente da imagem em relação a direção Y . Esse procedimento é importante para facilitar a identificação da curva primária que determina a senóide e a sua orientação.

Então considerando uma curva no plano (x,y) , o conceito de Transformada de Radon (TR) [209] consiste na integral de linhas das retas que interceptam a curva $f(x,y)$ como $L1$ e $L2$, conforme visto em 4.12. Assim, $L1$ e $L2$ são as distâncias das retas que interceptam a curva nos pontos $P1$ e $P2$ até a origem e G é o ângulo que as retas fazem com a horizontal (x) [210]. A TR pode projetar retas, curvas, círculos e o conjunto dos pontos da Transformada de Radon, para diferentes P e G geram a respectiva representação no novo plano resultado da parametrização em coordenadas polares. As integrais descrevem as características do objeto, da mesma forma que uma seção do corpo é reconstituída pelo computador matematicamente, a partir de infinitas partes de suas projeções em um aparelho de tomografia [209]. Então considerando uma imagem com uma curva [209, 210] como um conjunto de valores de pixels 0 em preto e 1 em branco, conforme apresentado na figura 4.13, é realizada o MTR [211, 212]. Assim, seja (x,y) a localização em coordenadas cartesianas dos pixels brancos referente a fratura segmentada e binarizada, a transformada de Radon converte os pares de coordenadas cartesianas desses pixels (x_i, y_i) em coordenadas polares (P, G) conforme figura 4.13 considerando o centro da imagem como origem [213, 214]. A integral funciona como uma acumuladora, computando a quantidade

de vezes que cada combinação (P, G) .

Com isso, pixels que compõe curva na imagem, geram picos com intensidades de cores mais claras nas novas coordenadas obtidas com a transformada de Radon, como visto na figura 4.14, permitindo identificar esses pixels e os seus respectivos ângulos da curva. De posse dessas informações o algoritmo pode verificar se o objeto é uma curva senoidal. Calcula os parâmetros para reconstruir senóides com no mínimo 3 pontos baseadas na equação y_{ij} de construção de uma senóide.

Uma vantagem ao utilizar o MTR é relacionado ao tempo de processamento, apresentando medidas 10 vezes menor que utilizando a transformada de Hough, em média de processamento em 0,06 segundos por imagem 180×180 pixels.

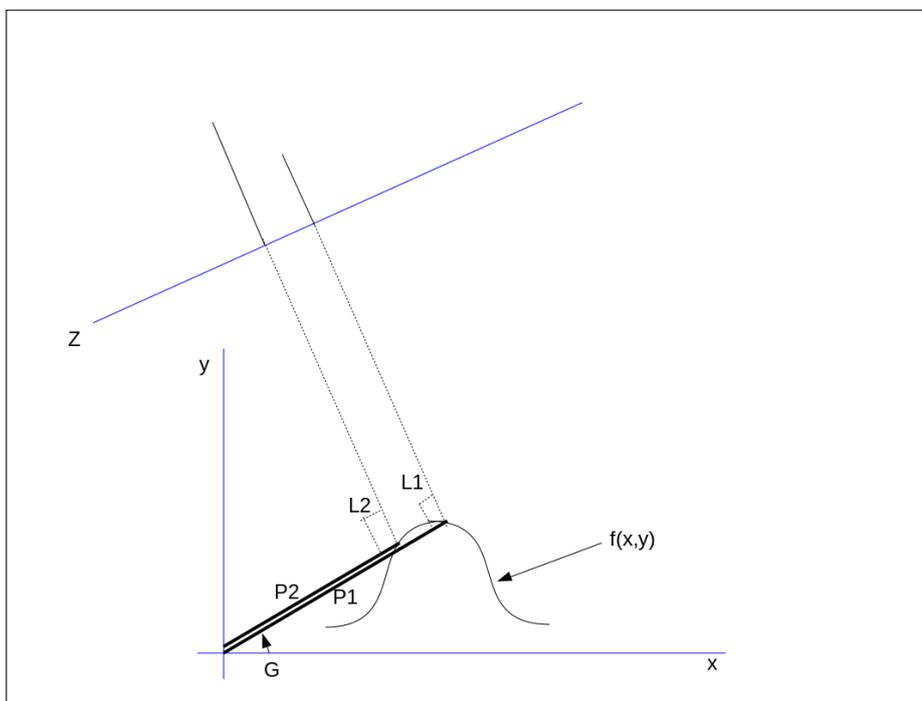


Figura 4.12 – Esquema com a forma representativa de uma curva pela transformada de Radon.

Após projeção da curva e reconstrução dos pixels formando senóide a saída do algoritmo apresenta as coordenadas referentes ao objeto detectado como fratura e o respectivo percentual de classificação. Esse percentual é referente ao coeficiente de correlação R entre as distribuições de valores x_1, x_1, \dots, x_n referente aos pontos dos pixels da fratura e y_1, y_2, \dots, y_n definidos pelos pontos interpolados com o ajuste senoidal.

Dessa forma, quanto mais pixels reconhecidos como componentes do objeto, maior é a probabilidade que o algoritmo tem que de identificar e classificar a fratura de forma correta. A saída do código é um arquivo com formato ASCII contendo as informações referentes a cada detecção como o nome das imagens, as coordenadas das fraturas encontradas e a nota da classificação como apresentada na figura 4.15.

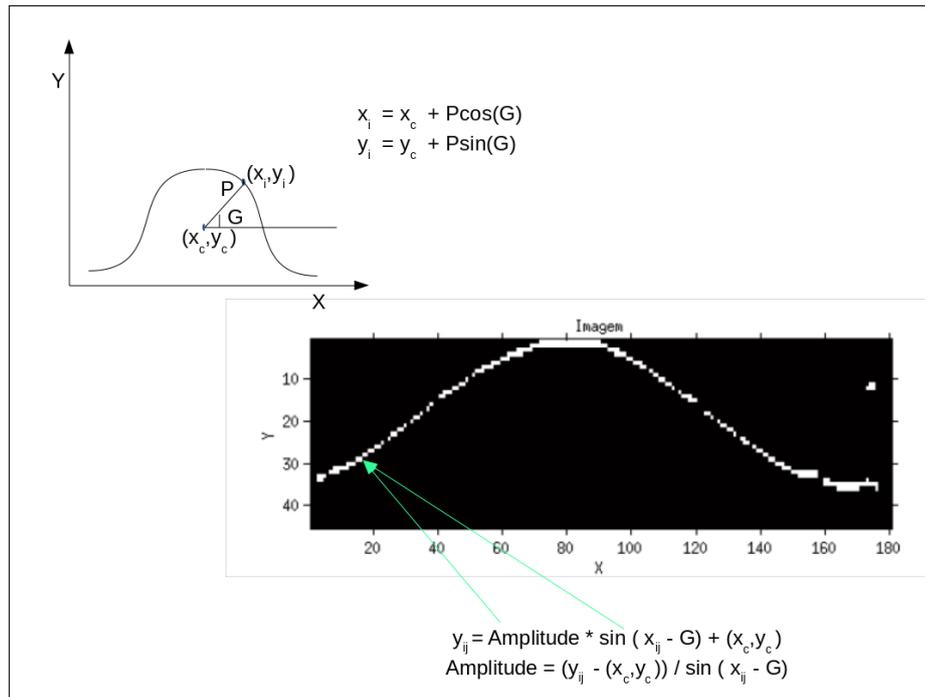


Figura 4.13 – Parametrização do curva Senoidal, onde x_{ij} e y_{ij} são as coordenada dos pixels da imagem referente a curva e x_c e y_c são as coordenadas do centro da curva.

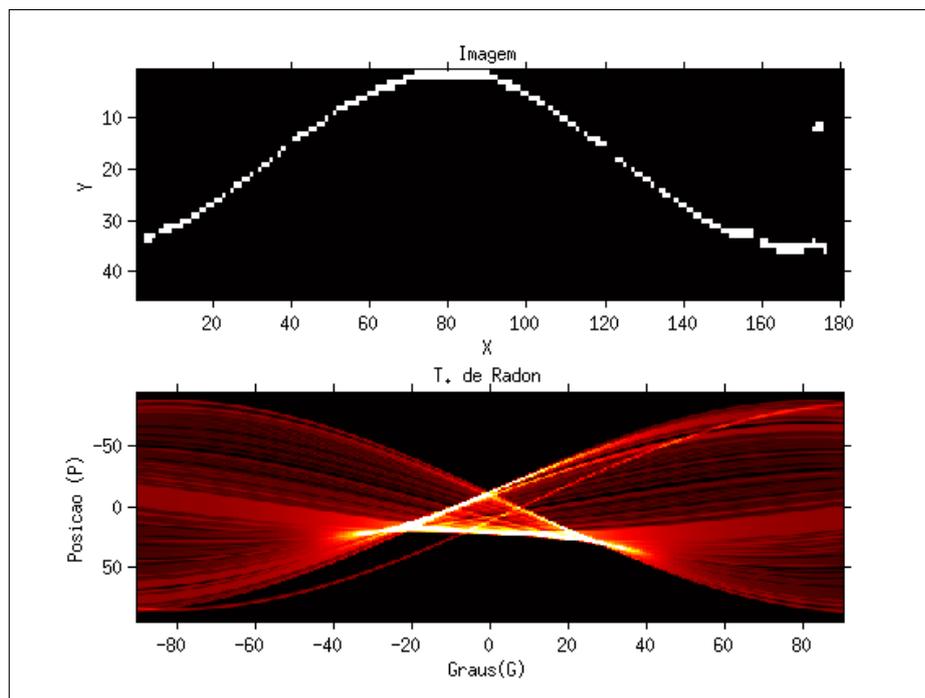


Figura 4.14 – Imagem de fratura segmentada com respectivo espaço de parâmetro obtido por Método Transformada de Radon com intervalo em graus (G) de $[0,180]$. O espaço de parâmetro possibilita a análise do objeto no espaço representado pela posição dos pixels por variações de graus.

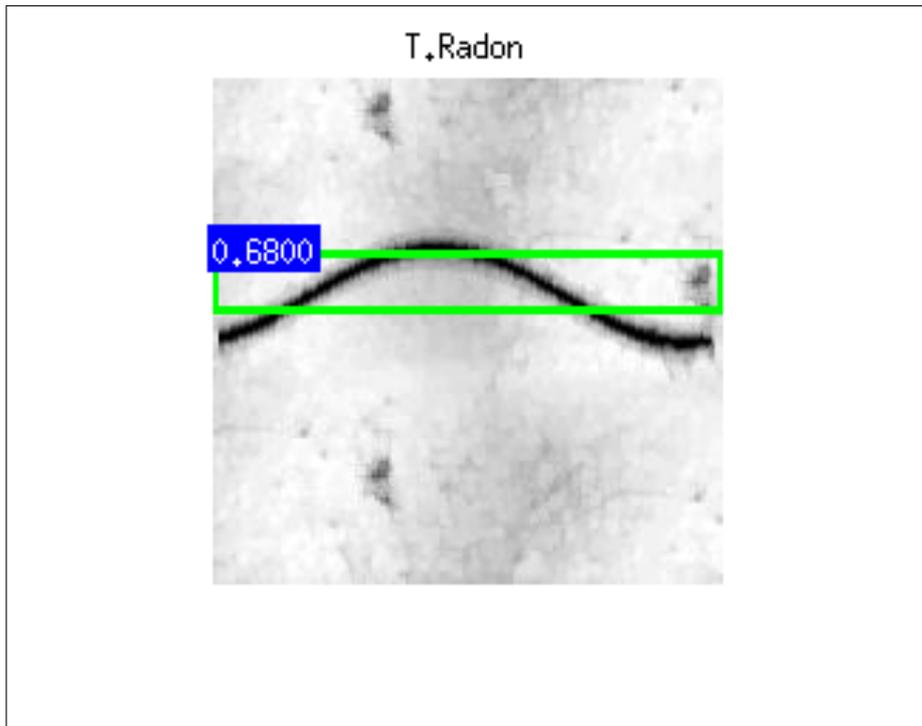


Figura 4.15 – Resultado imagem de detecção de fratura utilizando Transformada de Radon, com a respectiva nota de classificação. Nesse exemplo, o detector identificou parte da fratura dentro da caixa delimitadora em verde.

4.1.5.3 Método Transformada de Hough

Uma técnica bastante utilizada para detecção de linhas em imagens é a transformada de Hough [215, 216].

A Transformada de Hough (TH) patenteada em 1962 por Hough [217, 218], foi adaptada para detecção de retas, porém pode ser generalizada, e uma curva qualquer em (x, y) é parametrizada para uma curva em (P, G) , sendo P o vetor de coordenadas e G o vetor de parâmetros associado à angulação da curva [216, 219].

A metodologia adotada na utilização desse método é muito semelhante a utilizada pelo Método Transformada de Radon [220]. A Transformada de Hough consiste em mapear cada pixel da imagem no espaço de Hough através dos parâmetros P e G , definidos na figura 4.13, que é organizado em forma de um acumulador m -dimensional, onde m corresponde ao número de parâmetros. Assim analisando cada pixel da imagem (x_{ij}, y_{ij}) , são identificadas todas as curvas senóides que passam por este ponto, no mapeamento feito para o espaço de parâmetros (P, G) , a posição do acumulador correspondente ao resultado do mapeamento que é incrementado [221]. Dessa forma, é utilizado um limiar que estabelece um valor mínimo de 3 pontos para decidir se o valor acumulado corresponde a uma fratura [222]. Encontrando no mínimo 3 pontos várias senóides são definidas pela função y_{ij} em 4.13, aceitando a senóide que utiliza o maior número de pontos [12, 222]. A desvantagem desse método é o tempo de processamento médio de 0,30 segundo por imagem, 5 vezes maior que o apresentado pelo Método Transformada de Radon. Esse

fato ocorre pois exige muito processamento, já que para cada coordenada da imagem, são construídas senóides encontrando novas coordenadas de P variando o G de 0 a 180 graus.

A saída do código é um arquivo com formato ASCII, assim como detalhado nos métodos anteriores. Assim como utilizado no Método Transformada de Radon, a nota registrada na saída do código é associada a confiança na detecção, referente ao coeficiente de correlação R entre os pontos da fratura detectada e seus pontos após ajuste senoidal. Dessa forma, quanto mais pixels o algoritmo detecte como candidatos à curva, melhor é o ajuste e maior a certeza ou nota da detecção como apresentada na figura 4.16, localizada próxima a caixa delimitadora em verde.

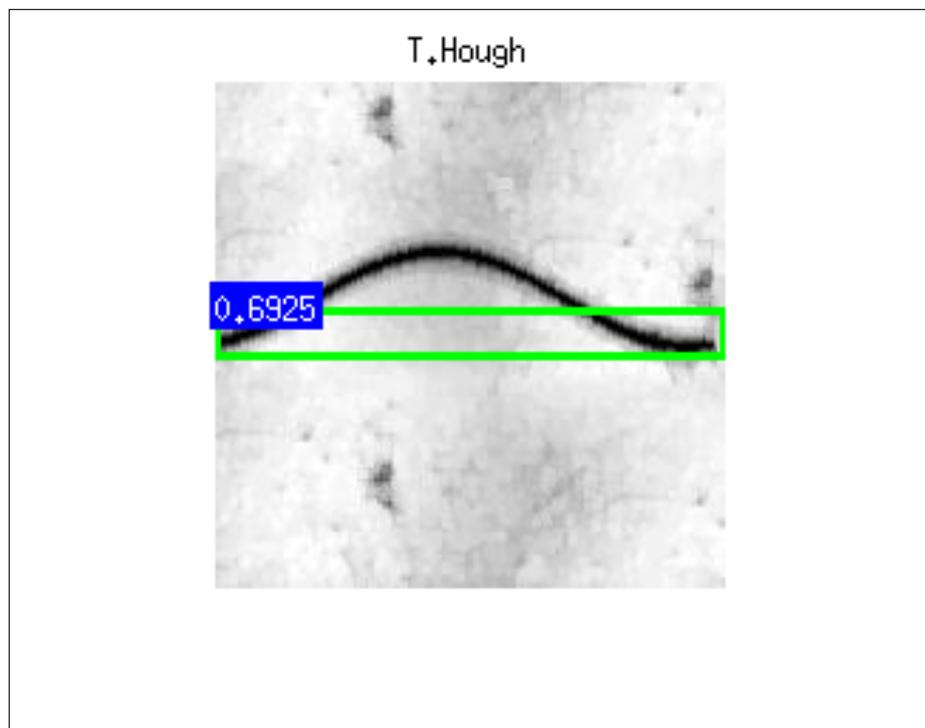


Figura 4.16 – Resultado da detecção de fratura utilizando Método Transformada de Hough, com a delimitação da fratura encontrada e a respectiva nota de classificação.

4.1.5.4 Método de Detecção de Desmoronamentos por Reconhecimento de Padrões

Os desmoronamentos apresentam um padrão diferente das fraturas, por isso o Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), possui critério de classificação e reconhecimento diferenciado.

Durante o pré-processamento detalhado em 4.1, ao utilizar imagens acima de 400 pixels de linha o algoritmo varre a imagem procurando regiões de objetos detectados pelos cálculos de aglomerado proeminente através da equação A.26 apresentada no apêndice A. Em imagens com número inferior de linhas, essa etapa não é realizada e são aplicadas os demais passos de processamento digital de imagem apresentadas em 4.1. Nós utilizamos a etapa de segmentação com K-médias e como resultado os objetos da imagem estavam

em branco com pixels equivalente a 1 e o fundo preto com pixels referentes a 0. No pós-processamento são aplicados índices ou etiquetas em cada objeto para obter respectivamente atributos de cada um. Dessa forma, critérios são verificados para identificar padrões atribuídos as desmorações na imagem acústica.

Nós utilizamos critérios baseados nos atributos dos objetos para identificar os desmorações. Os atributos analisados foram: Posições de cada pixel, posição do centro de massa e orientação de cada objeto. Cada objeto rotulado é verificado, e se houver dois objetos, A e B , tocando a borda da imagem equidistantes na posição vertical, o algoritmo considera que são partes do mesmo objeto que foram divididos pela projeção do poço em uma imagem 2D. Essa verificação é feita analisando as posição dos pixels de A e B e se pelo menos um pixel compartilhar a mesma posição vertical, são considerados um único objeto. Caso isso ocorra, ambos são unidos e os atributos recalculados. No exemplo da figura 4.17 pelo menos um dos desmorações foi separado, mas as duas partes foram computadas como uma única após a verificação. Assim, os pares de desmorações foram devidamente detectados, incluindo as partes que antes estavam separadas.

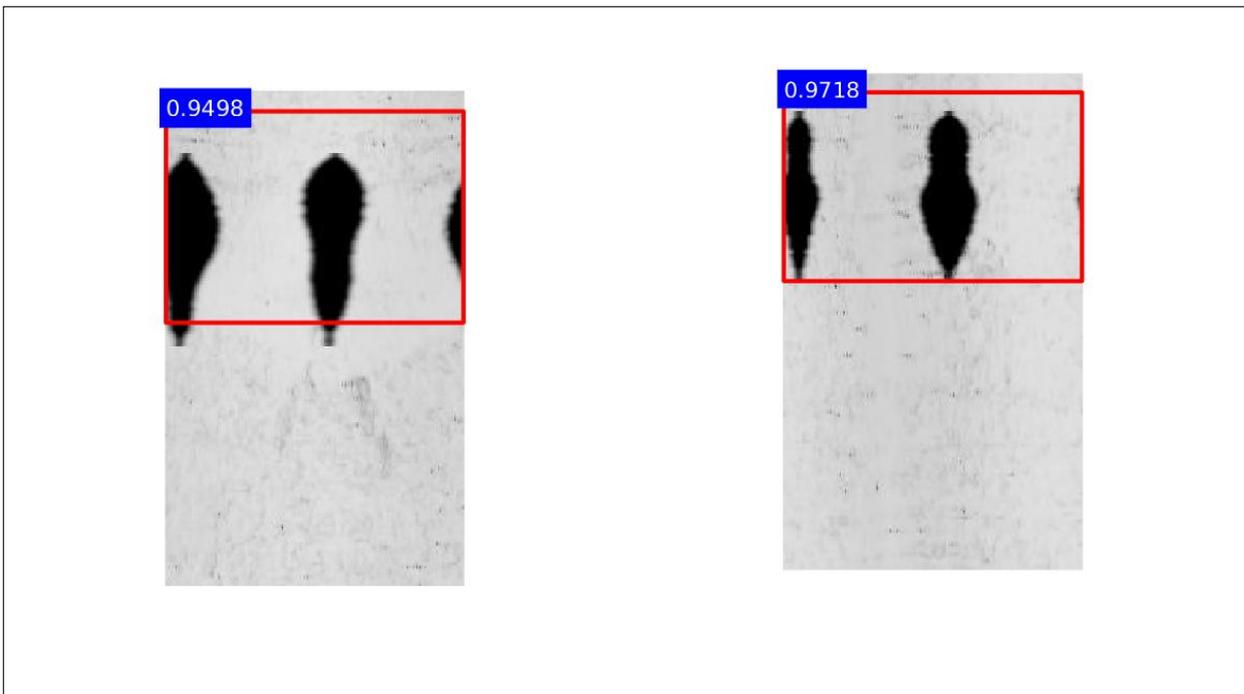


Figura 4.17 – Resultados da detecção da desmoração utilizando o Método de Detecção de Desmorações por Reconhecimento de Padrões (DRP). Nesses exemplos, pelo menos um dos desmorações foi separado no processo de projeção da imagem 2D, mas as duas partes foram computadas como uma única após a verificação. Assim, os pares de desmorações foram devidamente detectados, incluindo as partes que antes estava separadas.

Em seguida, os objetos A e B são analisados quanto a diferença da posição vertical de centro de massa C_A^v e C_B^v , distância horizontal de centro de massa C_A^h e C_B^h e diferença de orientação O_A e O_B . A posição horizontal e vertical do centro de massa de cada objeto

foi calculado por:

$$C^h = \frac{\sum_{i=1}^n x_i}{n}, \quad (4.10)$$

e

$$C^v = \frac{\sum_{i=1}^n y_i}{n}, \quad (4.11)$$

onde x_i equivale as posições horizontais dos pixels, y_i as posições verticais dos pixels e n é número de pixels do objeto. A orientação do objeto foi calculado através do ângulo ϕ entre o eixo maior de uma elipse circunscrita no objeto e o eixo horizontal, como apresentado no exemplo da figura 4.18.

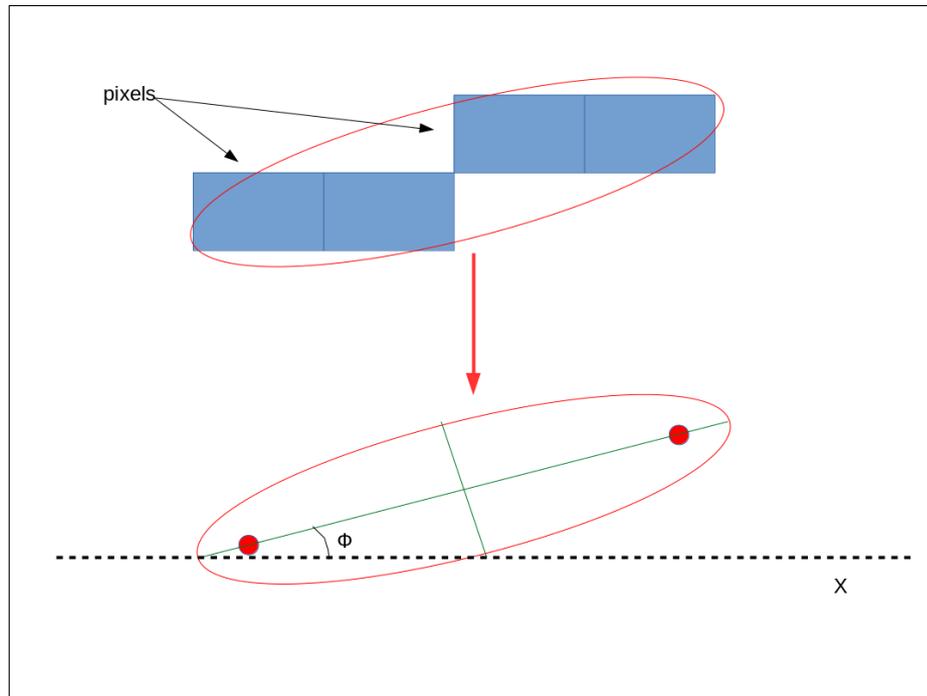


Figura 4.18 – Esquema adaptado de [223] apresenta a definição de orientação do objeto. Uma elipse é circunscrita nos pixels do objeto e o ângulo do seu eixo maior em relação ao eixo horizontal da imagem é determinada como o ângulo ϕ do objeto.

A imagem possui 180 pixels de largura referentes aos 360 graus da circunferência da parede do poço, conforme apresentado no exemplo 4.19. Dessa forma, cada pixel da imagem representa as medidas de 2 graus da circunferência da lateral do poço. Os desmoronamentos possuem um padrão de distância entre o par de 180 graus, causado pela variação da tensão da rocha durante a perfuração, então seus centros de massas devem possuir 90 pixels de distância. Assim, a classificação percentual da detecção do par de desmoronamento é determinada pela diferença entre 1 e o erro percentual entre distância do par detectado comparada com a distância esperada igual a 90 pixels. Utilizando as coordenadas do par de desmoronamento, são gerados as caixas delimitados das fraturas e a sua respectiva nota.

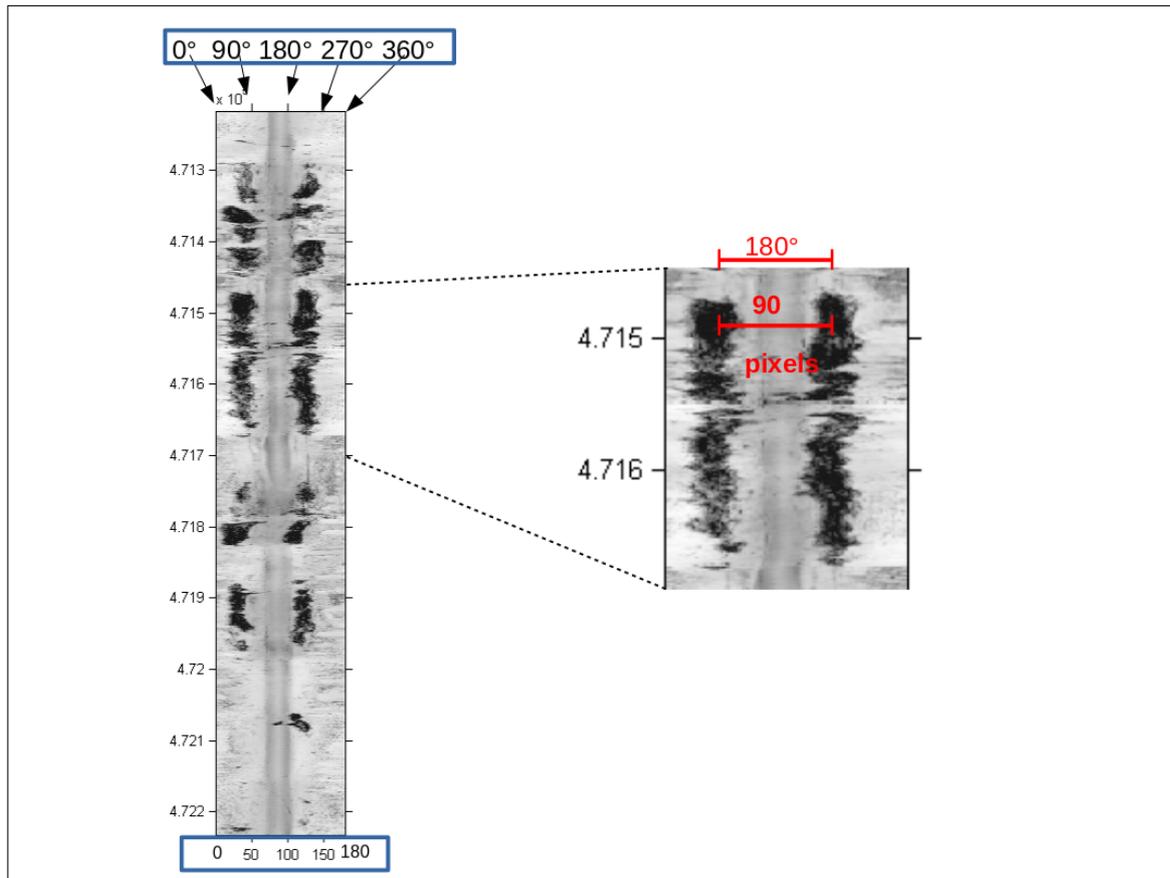


Figura 4.19 – A imagem possui 180 pixels de largura referentes aos 360 graus da circunferência da parede do poço. Os desmoronamentos possuem um padrão de distância entre o par de 180 graus, causado pela variação da tensão da rocha durante a perfuração.

4.2 Imagem Acústica Sintética

Para avaliar a qualidade dos métodos de busca automatizada de fraturas e desmoronamento em poço são necessárias tabelas verdade, isto é, imagens previamente classificadas nas quais seja possível determinar se um determinado método foi eficaz e acurado em detectar a ocorrência. A disponibilidade deste tipo de informação é limitada pelos dados disponíveis na literatura [23, 22] e obtidos junto a Petrobrás e a ANP a poucas dezenas de ocorrências. Como os principais métodos avaliados nesta dissertação dependem de um processo de treinamento supervisionado são necessárias, pelo menos, milhares de imagens, o que também nos permite estabelecer a completeza. Para obter um número significativo de dados optamos por gerar imagens simuladas de poço contendo fraturas e desmoronamentos. A vantagem desse tipo de abordagem é a reprodutibilidade, uma vez que, dois ou mais intérpretes de perfis de imagens podem divergir sobre a existência ou não de uma fratura ou desmoronamento em um dado real podendo, assim, gerar um viés na análise.

Os dados sintéticos são provenientes de 3 fontes distintas. As imagens sintéticas utilizadas pelos métodos de IA no treinamento foram fornecido pela *Petrobrás*[®] e iremos nomear no decorrer desse trabalho como Dado 1. O Dado 1 contém imagens não libe-

radas para publicação, por isso foram utilizadas apenas para treino. A imagem do poço fornecida pela ANP liberada para nossa publicação, foi base para construção dos dados sintéticos para avaliação dos classificadores, e são nomeados durante esse trabalho como Dado 2. As imagens reais que serviram de teste adicional foram obtidas de dados abertos ao público apresentados pelo estudo de Cornet [12] e nomeamos no decorrer desse estudo como Dado 3.

O Dado 2 foi solicitado pelo nosso grupo de pesquisa à ANP e as imagens acústicas de amplitudes são pertencentes ao poço 1 – *IMET – 8MG* com informações descritas no apêndice B, coordenadas datum geográfico SAD69 MC:-45° zona:23S, compreendido na bacia de São Francisco, apresentando uma área de 350.000 km^2 , cobrindo áreas pertencentes aos estados de Minas Gerais e Bahia e pequenas partes de Goiás, Tocantins e Distrito Federal. Em especial, o poço 1 – *IMET – 8MG* está localizado em Minas Gerais e faz parte da formação Grupo Paranoá caracterizado por uma ampla variação de fácies. Seu preenchimento sedimentar é representado principalmente por rochas clásticas e carbonáticas. A bacia despertou interesse exploratório de hidrocarbonetos desde que foram descobertos indício de gás natural ou em poços rasos perfurados para água. Quatro poços exploratórios foram perfurados na bacia, dois apresentaram vazão sub-comercial de gás [224, 225].

Antes de introduzir as fraturas sintéticas precisávamos de diferentes fundo de imagens acústicas para verificação da rede neural proposta e detalhada na seção 4. Para tal finalidade, utilizamos a função Image Quilting [226] no programa *Matlab*® [223] para gerar imagens sintéticas à partir das imagens do Dado 1. Essa função constrói uma nova imagem à partir de um ou mais trechos da imagem base conforme apresentado na figure 4.20, onde o trecho B da imagem de entrada é utilizado para confeccionar a nova imagem, cobrindo e dispondo esses trechos de maneira sobreposta como o item a, em diferentes ângulos de forma aleatória, analisando o melhor ponto de corte representado pela linha ondulada no item b, entre as partes definida pelos valores mínimos de desvio padrão entre as bordas das partes sobrepostas. Então, a partir de um trecho de imagem conforme apresentado na figura 4.21 a esquerda, podemos simular outro trecho de imagem como exemplificado à direita. As diferentes características de texturas são redistribuídas, mantendo a mesma escala de valores de amplitudes acústicas.

4.2.1 Fraturas e Desmoronamentos Sintetizados

Para simular as fraturas, utilizamos uma curva senoidal onde a posição dos pixel componentes é baseada na equação:

$$y_i = A * \sin(x_i - G) + D, \quad (4.12)$$

onde y_i é a posição da coordenada vertical da imagem, x_i é a posição da coordenada horizontal, A é a amplitude, G é a fase e D é a profundidade ou posição. Definimos

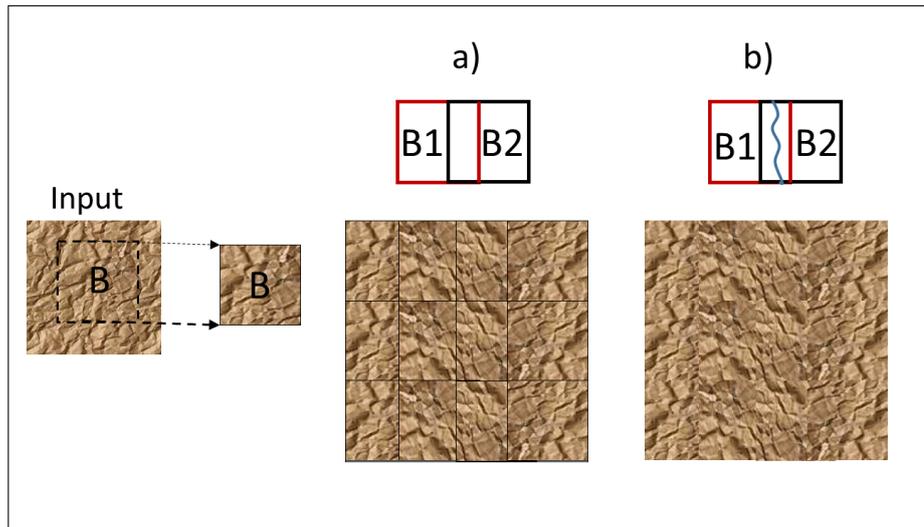


Figura 4.20 – Esquema Image Quilting. Com essa função é possível construir uma nova imagem à partir de um ou mais trechos da imagem base, onde o trecho B da imagem de entrada é utilizado para confeccionar a nova imagem, cobrindo e dispendo esses trechos de maneira sobreposta como o item a, em diferentes ângulos de forma aleatória, analisando o melhor ponto de corte entre as partes, representado pela linha ondulada no item b, definida pelos valores mínimos de desvio padrão entre as bordas das partes sobrepostas. Então a partir de uma única imagem podemos ter outra imagem com características e dimensões diferentes.

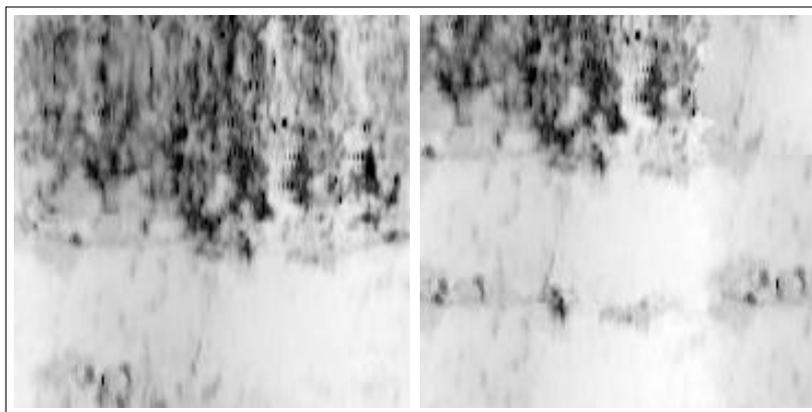


Figura 4.21 – Esquema representativo da formação de imagem de textura simulada, realizado pela função Image Quilting [226]. Trecho à direita é a imagem base original. Trecho à esquerda é resultado da simulação.

aleatoriamente um valor entre os intervalos para os 5 parâmetros de forma e localização, apresentados na tabela 2, respectivamente, a amplitude, a abertura (i.e. espessura), o período, a fase e a posição para cada senóide na imagem. Introduzimos descontinuidades aplicando falhas durante a construção na curva com pixels de valores 0 representando a cor preta, conforme apresentamos na figura 4.23 a senóide com nível máximo e mínimo de descontinuidade. Dessa forma, o nível de descontinuidade é a largura da descontinuidade em pixels. Mas, a frequência de introdução é escolhida aleatoriamente, alternando dentro de um intervalo de 7 a 15 pixels durante a construção da senóide. Alteramos o foco con-

trolando a nitidez da fratura utilizando um filtro gaussiano local calculado com a equação 3.16, no conjunto de pixels de linhas x e colunas y , conforme apresentamos resultado na figura 4.24. A máscara gerada à partir na equação é aplicada em um percentual da senóide determinada pela intensidade definida para o foco. Adicionamos ruído baseado na textura do entorno da fratura, incorporando áreas sombreadas nas bordas da curva, conforme apresentado na figura 4.25. O valor mínimo dos pixels da imagem diferente de zero, do fundo da imagem é computado próximo a borda superior e inferior da fratura. A proximidade é definida pelo número de pixels da abertura dividido por 2. nível de ruído controla o percentual do valor aplicado na borda. Por exemplo, considerando um pixel da borda da senóide com novo valor de 0,4, caso o nível definido para o ruído seja 0 esse valor permanece, caso seja 0,4 o valor resultante do pixel será 0,2. Redefinimos o nível de contraste aplicando ou não ajuste na nitidez de toda imagem com a fratura, como no exemplo da figura 4.26, onde podemos verificar que utilizando essa etapa o contraste da fratura com o fundo fica prejudicado. O nível de contraste igual a 1, nesse caso, é definido pela diminuição da abertura da fratura em alguns pontos com maiores valores de pixel, contribuindo para um menor contraste dessas partes com o fundo da imagem. Esse efeito foi possível utilizando a função imdilate, que aplica a operação de dilatação em uma imagem $N \times M$ com valores de pixels entre 0 e 1 em tons de cinza, onde A_i é a região da imagem 3×3 varrida por B um elemento estruturante formado por uma matriz de 3×3 composta de valores 1. Esse elemento estruturante é uma matriz que funciona como uma janela deslizante sobre a imagem realizando a operação de dilatação sobre a região da imagem A_i , sendo $i = 1, 2, \dots, N \times M$. Dessa forma,

$$Dilatação(A, B) = \max(A), \quad (4.13)$$

Para a saída da imagem resultado da dilatação possuir as dimensões iguais da imagem inicial, antes é aplicado o preenchimento de zeros em torno dos pixels. A figura 4.22 apresenta um exemplo de trecho de imagem 6×6 em tons de cinza, utilizando elemento estruturante B 3×3 para dilatação. A janela B ao deslizar por cada região A_i até A_{36} varre toda a imagem com preenchimentos de zeros, computando um novo valor de pixel referente ao maior valor de cada A_i .

A dilatação aplicada nessa etapa utilizando imagem em tons de cinza é diferente da dilatação apresentada antes na equação 4.8, aplicada em imagens binarizadas. Um exemplo de fratura sintética com a definição dos parâmetros utilizado é apresentado na figura 4.27. Nesse exemplo foram atribuídos uma escolha aleatória entre os parâmetros, apresentados na tabela 2.

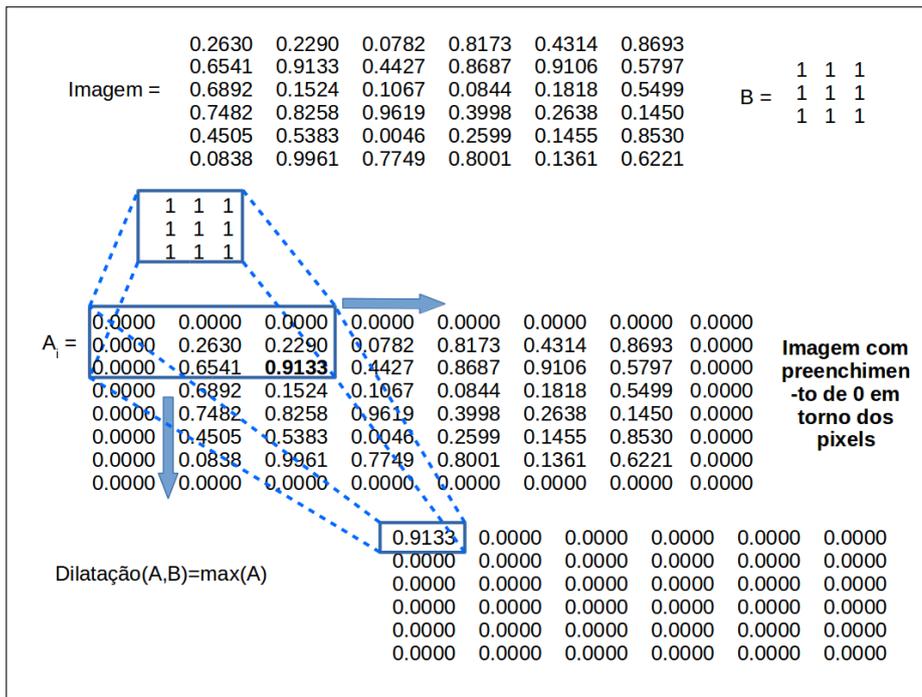


Figura 4.22 – Com função imdilate também é possível realizar a dilatação em imagens de tons de cinza, onde A_i é a região da imagem 3×3 varrida por B um elemento estruturante formado por uma matriz de 3×3 composta de valores 1. Um novo valor é computado no resultado, referente ao maior valor da região varrida.

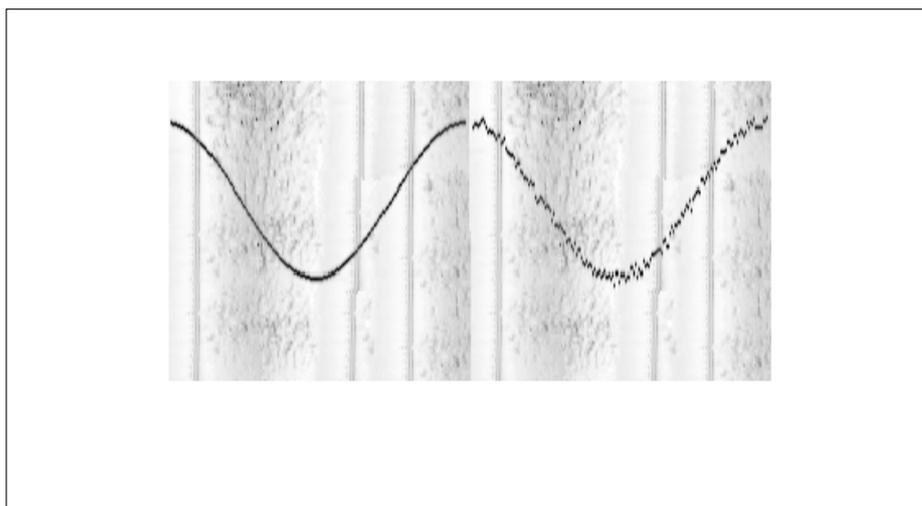


Figura 4.23 – Comparativo fratura simulada com parâmetro de descontinuidade. Esse atributo possui valores entre 0 e 10. A imagem à esquerda apresenta a curva senoidal com valor 0 de descontinuidade, à direita a curva com valor 10. A frequência de introdução é escolhida aleatoriamente, alternando dentro de um intervalo de 7 a 15 pixels durante a construção da senóide.

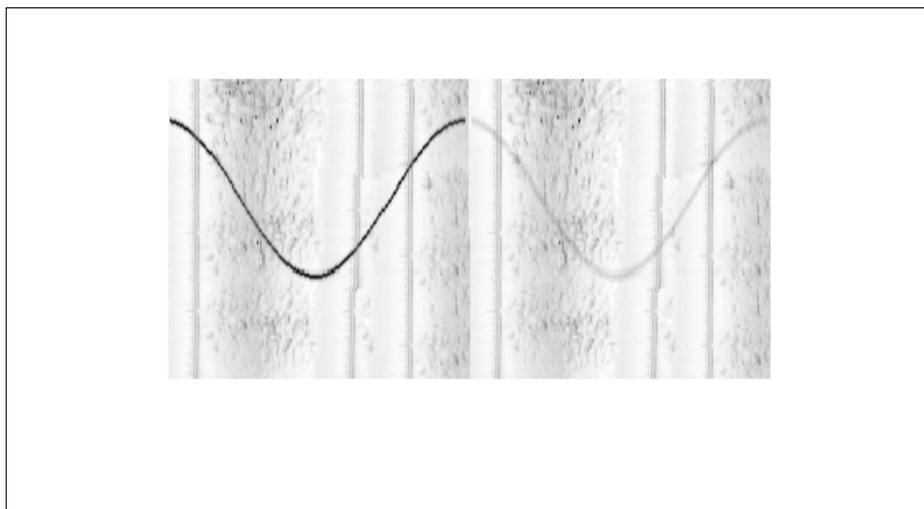


Figura 4.24 – Comparativo fratura simulada com parâmetro de foco. Podem ser utilizados valores entre 0 e 0,9, a imagem à esquerda apresenta valor 0,9 de foco, à direita com o valor 0.

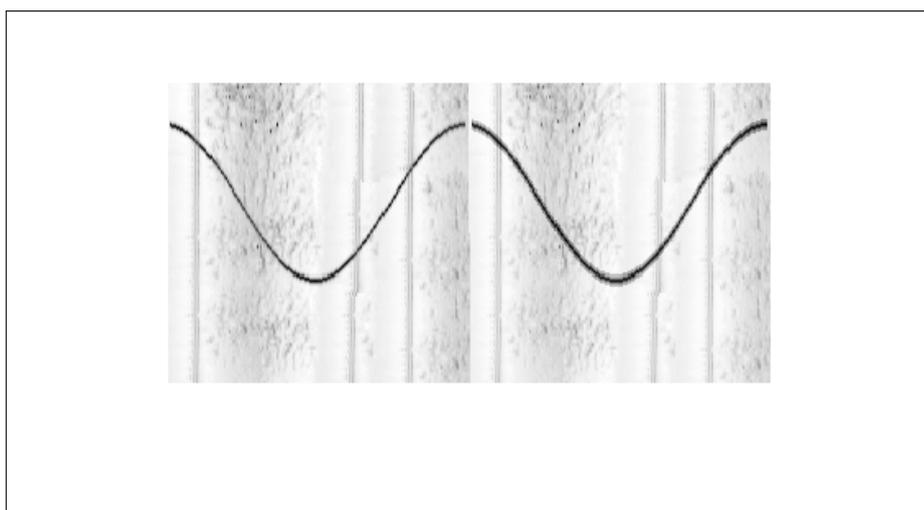


Figura 4.25 – Comparativo fratura simulada com parâmetro de ruído. O ruído pode ser controlado dentro de um intervalo que vai de 0 a 0,9, onde a imagem à esquerda apresenta 0 de ruído e à direita 0,9.

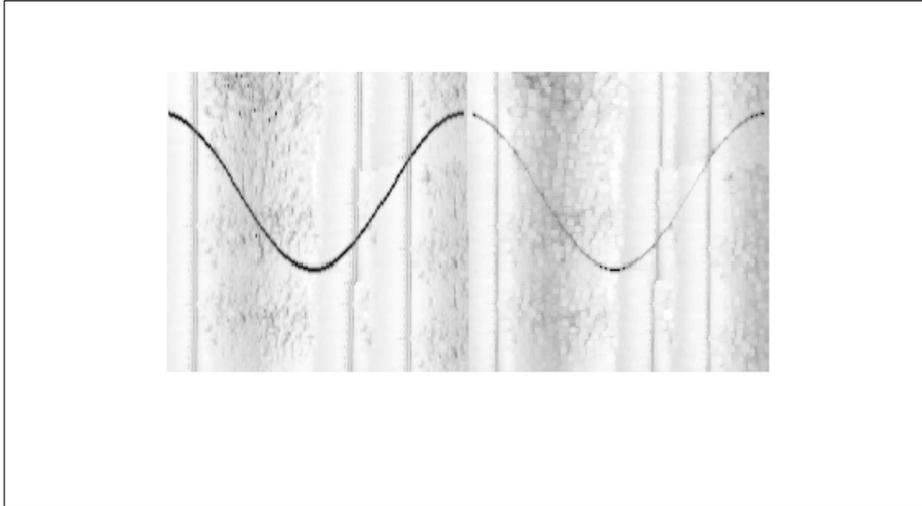


Figura 4.26 – Comparativo fratura simulada com parâmetro de contraste. O intervalo que define o contraste vai de 0 a 0,9, a imagem à esquerda apresenta o resultado da curva senoidal com valor 0,9, à direita o valor 0.

Tabela 2 – Tabela de intervalo de parâmetros das fraturas sintéticas.

Atributos	Intervalo	Medida
Amplitude	[10,70]	Medida em pixels
Abertura	[2,9]	Medida em pixels
Fase	$[-5/2\pi, +5/2\pi]$	Medida em pixels
Período	+1 ou -1	Medida em pixels
Localização	[50,150]	Medida em pixels
Descontinuidade	[0,10]	Medida em pixels
Foco	[0,0.9]	Percentual
Ruído	[0,0.9]	Percentual
Contraste	0 ou 1	Percentual

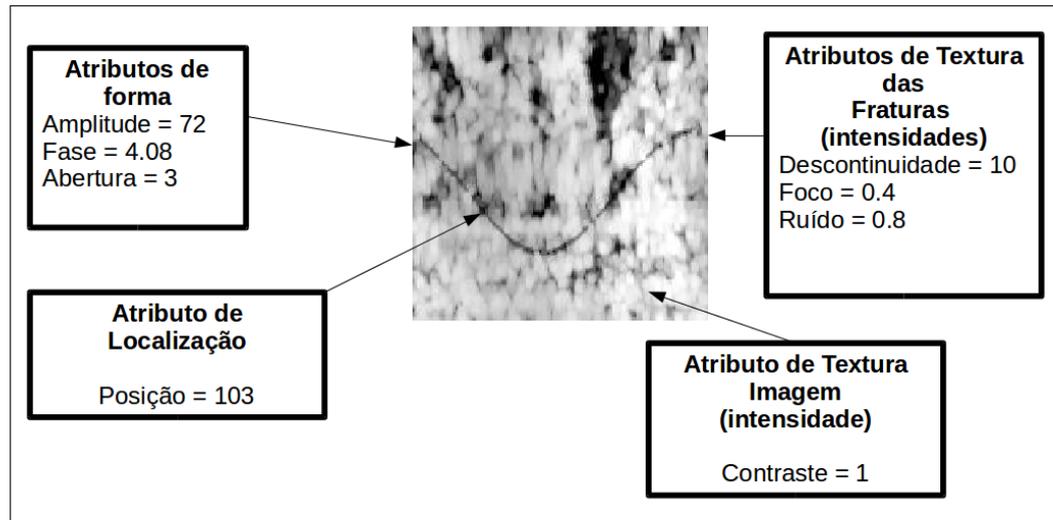


Figura 4.27 – Exemplo de trecho de imagem acústica com fratura sintética e suas respectivas atribuições de forma e localização (distância em pixels). Da mesma forma, as intensidades dos atributos de textura da fratura e da imagem no todo. Todas as medidas foram escolhidas aleatoriamente dentro dos intervalos apresentados em 2 pelo programa que gera as ocorrências.

Com a finalidade de reprodução das ocorrências dos desmoronamentos conhecidos como *breakouts*, utilizamos um código gerador. O algoritmo busca representar o principal padrão de disposição entre um par de desmoronamento reais de 90 pixels de distância entre os seus centros geométricos conforme detalhado no capítulo 2.0.6. Os centros geométricos dos desmoronamentos são localizados no ponto onde é definido aleatoriamente a profundidade dos desmoronamentos. Em cada imagem sintética, é aplicado uma elipse com eixo maior na vertical com tamanho escolhido aleatoriamente entre a dimensão vertical da imagem considerando a profundidade inicial escolhida também aleatoriamente. Na figura 4.28 apresentamos um esquema ilustrativo de apenas um dos pares de desmoronamento, considerando a imagem com pixels em tons de cinza representando o fundo da imagem e os pixels pretos o desmoronamento. O eixo horizontal é equivalente ao eixo menor da elipse e é gerado por linhas em pixel preto com larguras diferenciadas, escolhidas aleatoriamente dentro do conjunto de valores de pixels de acordo com a dimensão horizontal da imagem, limitada pela metade da distância entre os centros geométricos, produzindo uma elipse de contornos irregulares. No entanto, as primeira e as últimas linhas de pixels são os menores valores. Aplicamos alterações de orientação escolhidas de forma aleatória dentro do intervalo de 0 a 6 graus, para tornar mais próximo do real. Por fim, foi aplicado filtro gaussiano nos pixels próximos a borda, com máscara definida pela equação 3.16, para desfocar os contornos, diminuindo o contraste com o fundo da imagem.

A utilização de imagens sintética permitiu construir um grande número de dados. Assim, foi possível utilizar no treinamento dos métodos de IA dados de origem diferente dos dados utilizados na avaliação final de todos os métodos na etapa de teste de generalização dos modelos.

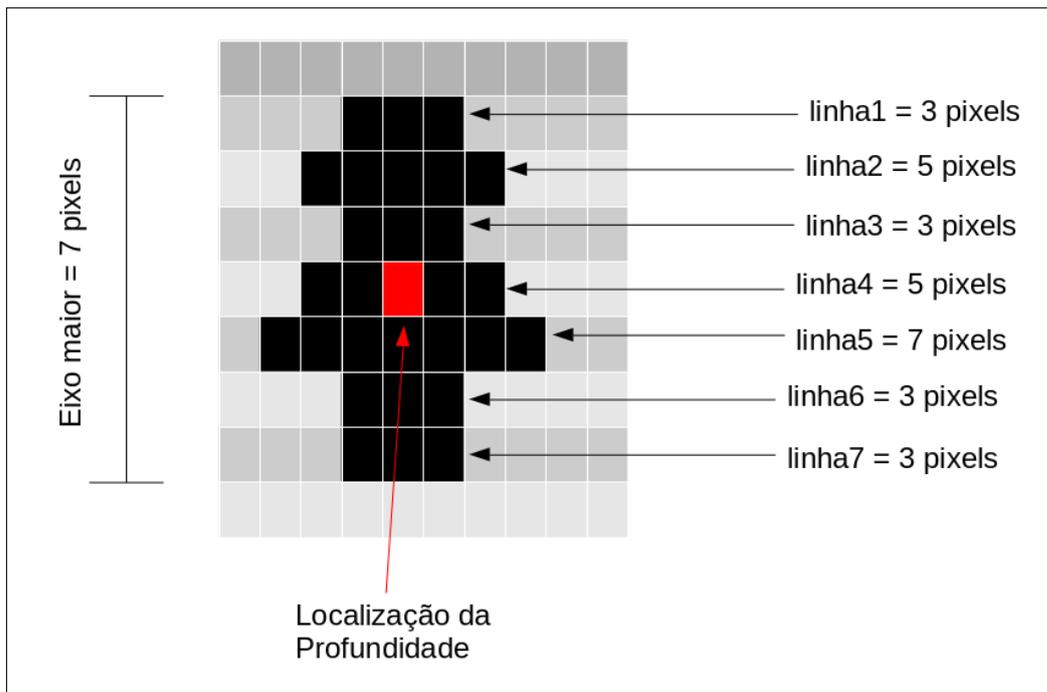


Figura 4.28 – Esquema de gerador de um único desmoronamento. Primeiro a localização da profundidade é definida, depois o eixo maior da elipse, assim linhas de pixels pretos de tamanhos aleatórios são empilhadas até completar a dimensão vertical. O pixel em vermelho apenas foi utilizado nessa figura para identificação da posição da localização da profundidade ou centro geométrico do desmoronamento. Na construção final o pixel introduzido é preto.

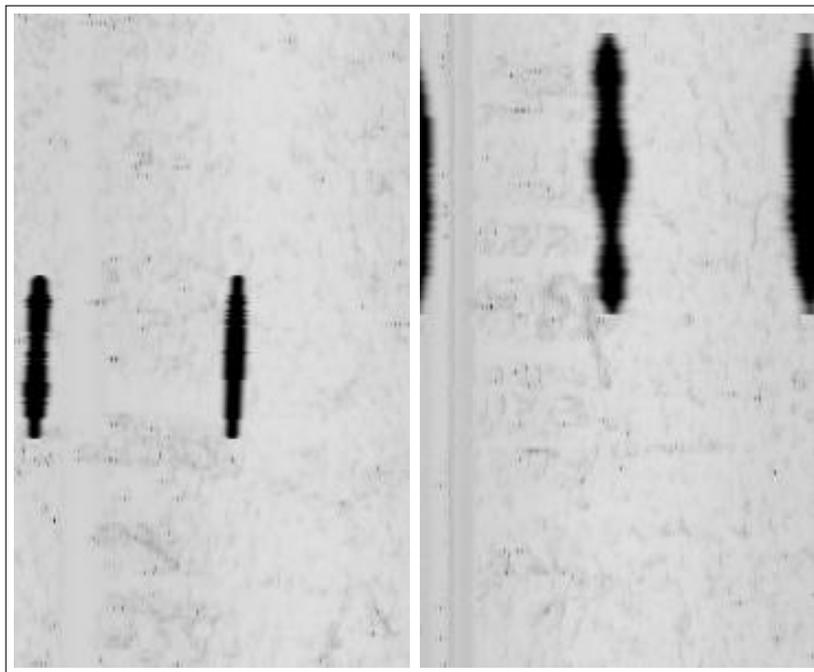


Figura 4.29 – Dois exemplos de desmoronamentos sintéticos desenvolvidos através do código gerador de desmoronamentos sintéticos desenvolvido por [169].

4.3 Característica de Operação do Receptor

A curva característica de operação do receptor conhecida como curva ROC da sigla em inglês *Receiver Operating Characteristic* foi desenvolvida por engenheiros durante a Segunda Guerra Mundial e utilizada com o intuito de avaliar radares na detecção de inimigos durante os confrontos [227, 228]. Passou a ser implementada em estudos de psicologia para detecção percentual de estímulos. Atualmente foi estabelecida no estudo científico da aprendizagem de máquina sendo utilizada para representar graficamente o desempenho ou performance de um sistema classificador [229, 230] como apresentado na figura 4.30.

Os valores computados no eixo vertical são relacionado a TPR que é a proporção de positivos verdadeiros do total de classificados como positivos definida como,

$$TPR = \frac{TP}{TP+FN}, \quad (4.14)$$

sendo TP conhecido como positivo verdadeiro representando os acertos e FN é conhecido como falso negativo representa as detecções indevidas, como por exemplo a não detecção de fratura (ou desmoronamento) em regiões onde há fratura (ou desmoronamento). No eixo horizontal do gráfico ROC temos o FPR que é a proporção de falsos positivos do total de negativos definida como,

$$FPR = \frac{FP}{FP+TN}, \quad (4.15)$$

sendo FP conhecido como falso positivo representando os exemplos classificados como positivo e na realidade não são ocorrência do objeto. A curva ROC é descrita por $TPR = f(FPR)$.

Quanto mais a curva se aproxima do canto superior esquerdo, representado na figura 4.30 com a letra A , melhor é considerado o desempenho do classificador, pois indicar um alto TPR e um baixo FPR . É possível verificar que a curva exemplo apresentada com a letra B representa um bom classificador. A reta C representa um classificador ruim, com performance aleatória, logo quanto mais próximo a curva se aproxima de C , menor é o TPR e maior é o FPR .

A medida da área abaixo da curva, AUC , é uma medida de qualidade do método, sendo que a área abaixo de uma curva qualquer é obtida pela integral definida com,

$$AUC = \text{Área} = \int_a^b f(x)d(x), \quad (4.16)$$

sendo $f(x)$ a função que descreve a curva, a e b os pontos iniciais e finais no eixo x , como visto na figura 4.31.

Cada classificador considera como acerto detecções com percentual de confiança acima de 0. Utilizando imagens sintética é possível obter regiões verdadeiras, previamente classificadas com fraturas e desmoronamentos simulados. Essas simulações contribuíram para

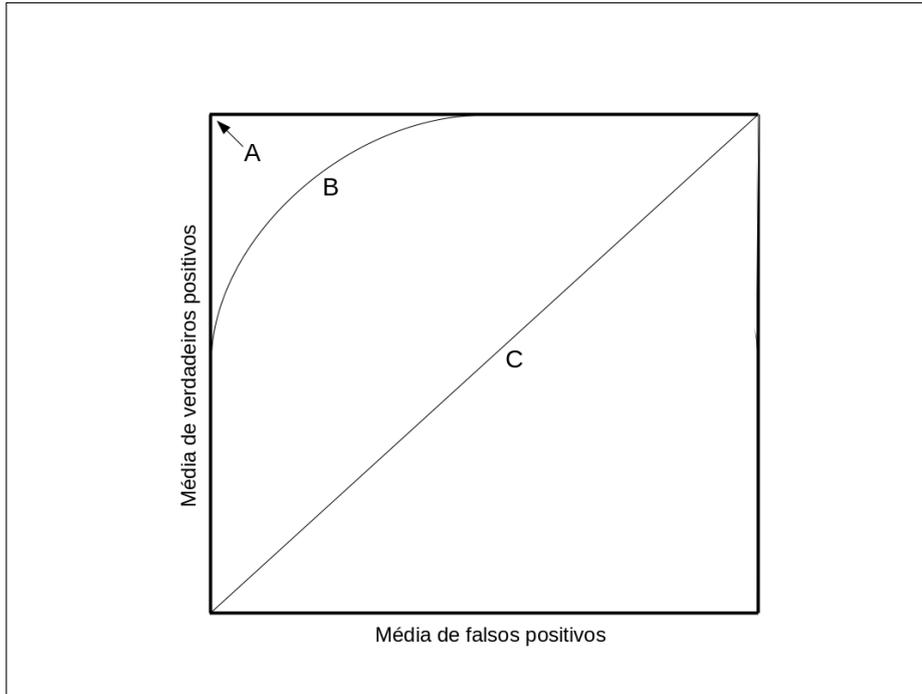


Figura 4.30 – Esquema ROC. A curva exemplo, apresentada com a letra *B* representa um bom classificador. A reta *C* representa um classificador ruim, com performance aleatória, logo quanto mais próximo a curva se aproxima de *C*, menor é o RPR e maior é o FPR.

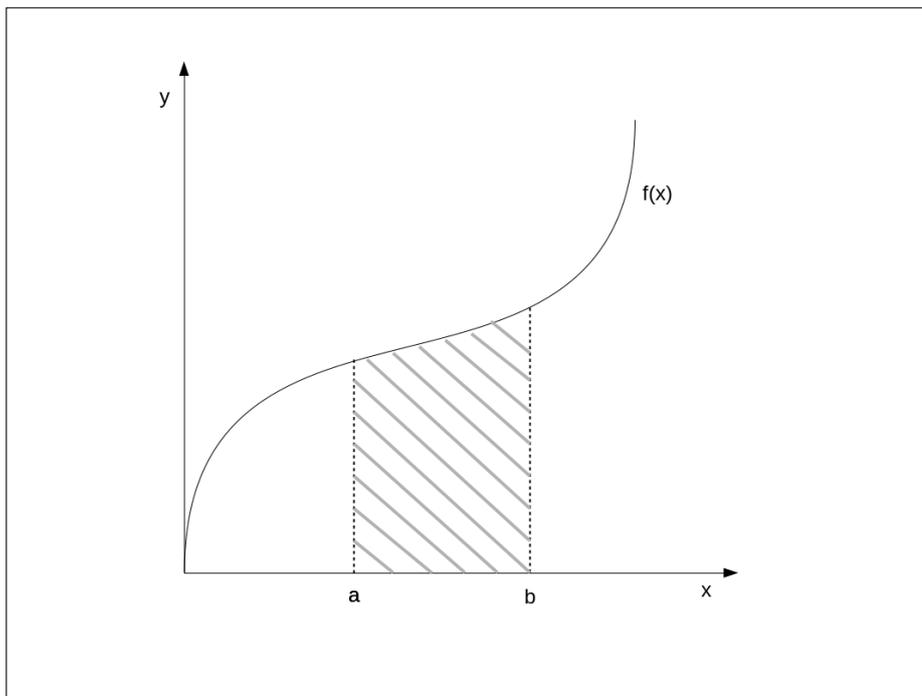


Figura 4.31 – Esquema área sob a curva com integral definida entre a e b.

avaliação controlada utilizando o gabarito das fraturas e desmoronamentos. Para construção da ROC para avaliação dos classificadores, foram considerados alguns critérios de definição de casos positivos e negativos. Cada região de detecção é comparada com

as regiões verdadeiras, em relação à distância dos seus centros. Nós consideramos casos verdadeiros positivos para detecção de fraturas a distância máxima de 30 pixels entre os centros das regiões e para desmoronamento utilizamos 150 pixels. Regiões de detecção que compreendessem mais de 4 vezes a região verdadeira foram desconsideradas, pois poderiam representar uma falha ao classificar a região do objeto.

No caso das RNA rasas que classificam a imagem não regiões, não são considerado critério de distância e dimensões comparativas com as regiões verdadeiras.

4.4 Avaliação Detecção de Fraturas

Nós utilizamos avaliação *ROC* com a finalidade de comparar as técnicas de classificação de fraturas *MCP*, *MTR* e *MTH*, apresentados no capítulo 4.1, resumido conforme diagrama geral de trabalho em 4.32. Com essa finalidade foram geradas 3000 imagens, proveniente do Dado 2, sendo 1000 com dimensões 180×180 pixels, com uma fratura sintética simulada, dentro dos parâmetros definidos na tabela 2, utilizando como base o Dado 1. Lembrando que as imagens utilizadas nessa avaliação não passaram pela etapa de determinação de região de interesse, que utiliza o atributo de Haralick para prevê regiões, devido as dimensões reduzidas dessas imagens. Entre os métodos, o que destacou foi o *MCP* que possui uma área abaixo da curva *AUC* da *ROC* de 0,85, conforme apresentamos na figura 4.33 e registrada na tabela 3. Esse método apresentou detecção de fraturas, conforme apresentamos alguns casos na figura 4.34, onde apresentamos detecções com nota superior à 0,5%. No entanto, não detectou fraturas sintéticas com níveis de contraste reduzido com relação ao fundo da imagem. Utilizando o *MTR* e o método *MTH* a classificação não foi eficiente e mostrou imprecisão principalmente quando as fraturas sintéticas apresentavam contraste reduzido, e dessa forma a segmentação não separou perfeitamente as ocorrências do fundo da imagem. Esse fato impediu que esses buscadores de curvas não identificassem os objetos segmentados como curvas. Os resultados apontaram áreas abaixo de 0,5 da curva *ROC*, conforme tabela 3, devido à notas inferiores de verdadeiros positivos apresentados nos dois métodos *MTR* e *MTH* e o baixo número de acertos, conforme apresentamos na tabela 4. Esses números indicam que a performance dos classificadores *MTR* e *MTH* é de caráter aleatório para a detecção de fraturas em imagens acústicas.

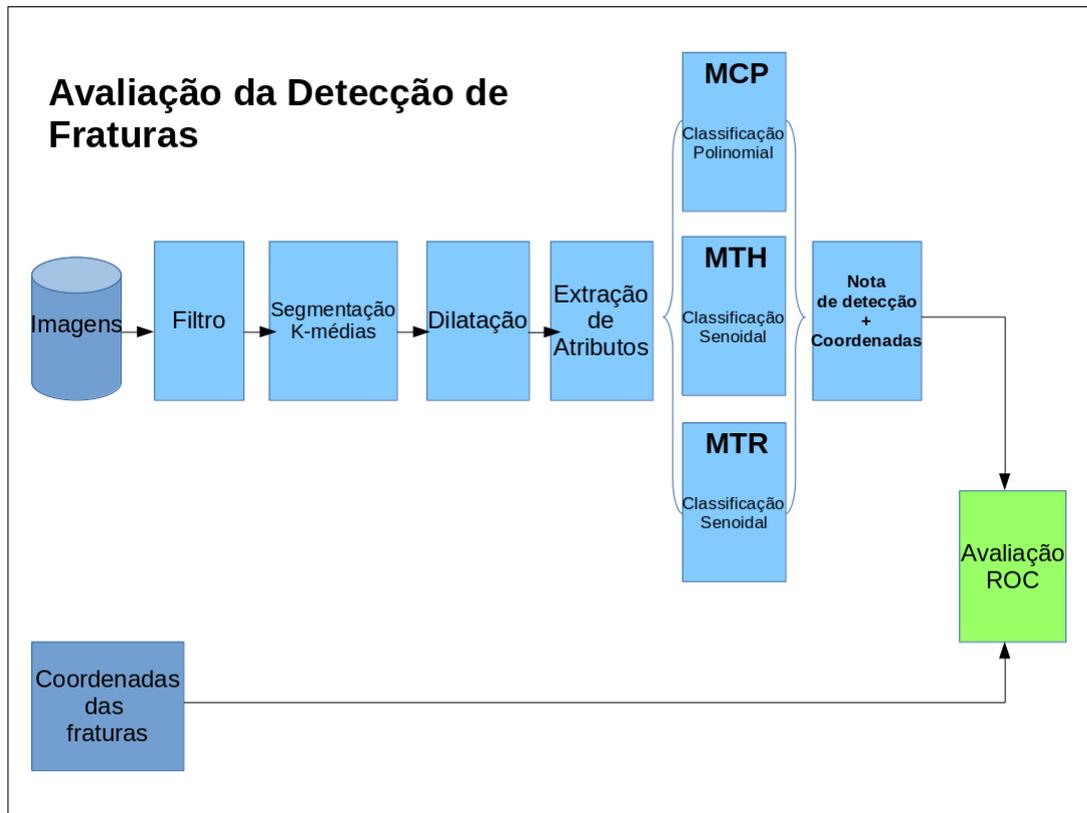


Figura 4.32 – Diagrama com o resumo das etapas do processamento digital de imagens PDI até avaliação da detecção de fraturas.

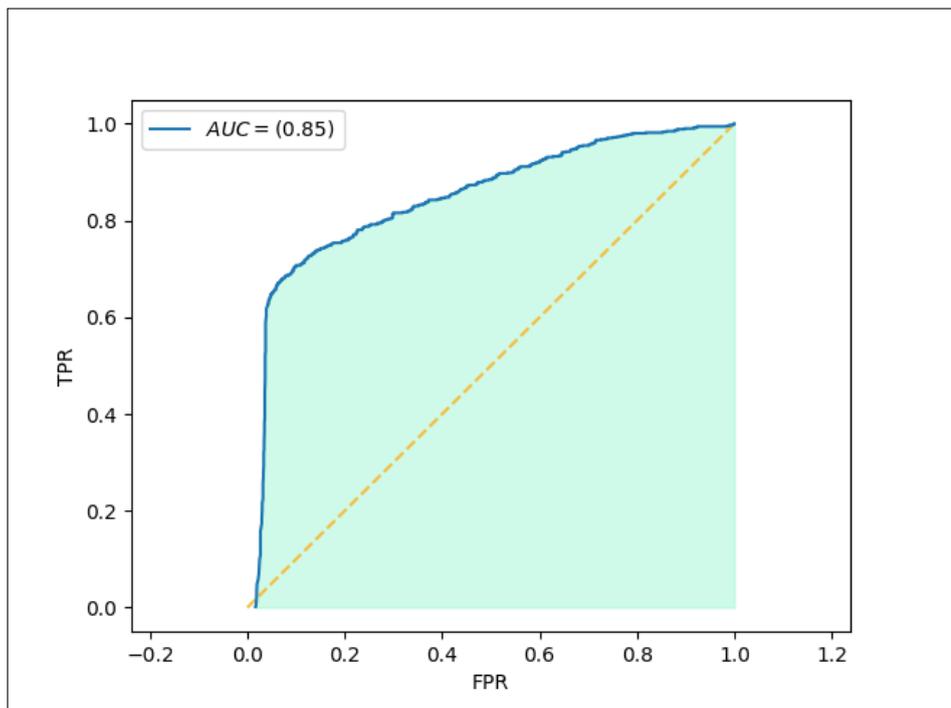


Figura 4.33 – ROC de detecção de fraturas utilizando Método de Classificação por Polinômio (MCP), com área abaixo da curva (AUC) de 85%.

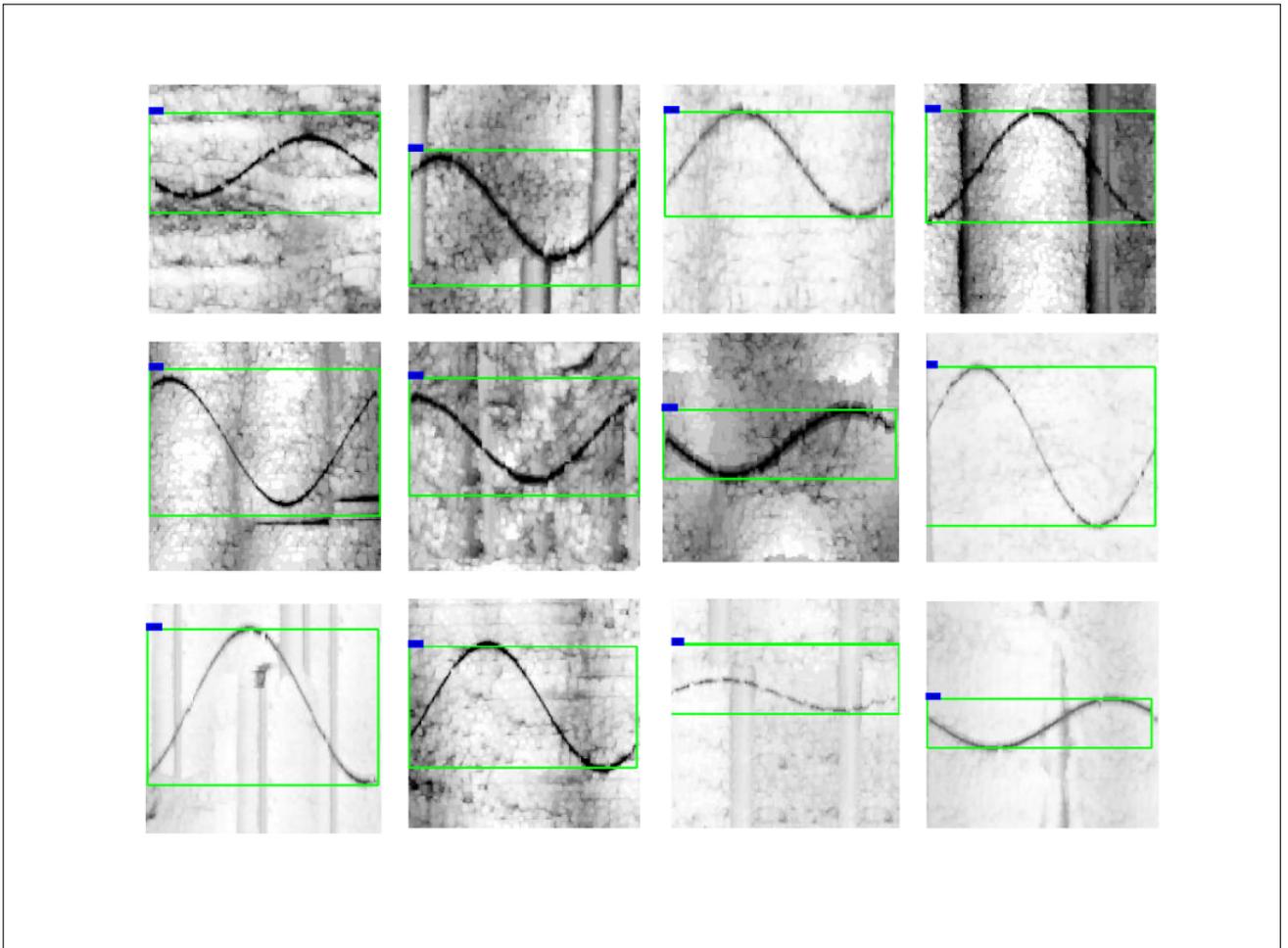


Figura 4.34 – Casos de verdadeiros positivos de detecção fraturas sintéticas método Método de Classificação por Polinômio (MCP). Apresentamos detecções com nota superior à 0,5.

Tabela 3 – Tabela de percentuais de área abaixo da curva AUC do métodos clássicos de processamento digital de imagens (PDI), obtida utilizando imagens sintéticas.

Métodos	AUC
MCP	85 %
MTH	50 %
MTR	43 %
DRP	73 %

Tabela 4 – Tabela de detecções de fraturas com Método Transformada de Hough (MTH) e Método Transformada de Radon (MTR).

Métodos	Total de Detecções	Verdadeiros Positivos
MTH	3827	630 (20 %)
MTR	679	142 (21 %)

4.5 Avaliação da Detecção de Desmoronamentos

No intuito de avaliar o poder de detecção do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), aplicamos a técnica em 3000 imagens sintéticas proveniente do Dado 2, sendo 1000 imagens de dimensões 300×180 pixels contendo desmoronamentos, conforme apresentamos no diagrama da figura 4.35. Como as imagens sintéticas utilizadas nessa avaliação possuem dimensões reduzidas, não foi utilizado o recurso de busca de regiões de possíveis desmoronamento com o atributo de Haralick aglomerado proeminente, por isso esse passo não conta desse diagrama. Como apresentado na figura 4.36 e na tabela 3, avaliamos e apresentamos a curva *ROC* que indicou uma área de 0,73 abaixo da curva. Apresentamos exemplos de sucessos na detecção de desmoronamentos utilizando a técnica (DRP) na figura 4.37, considerando casos com notas superiores à 0,5.

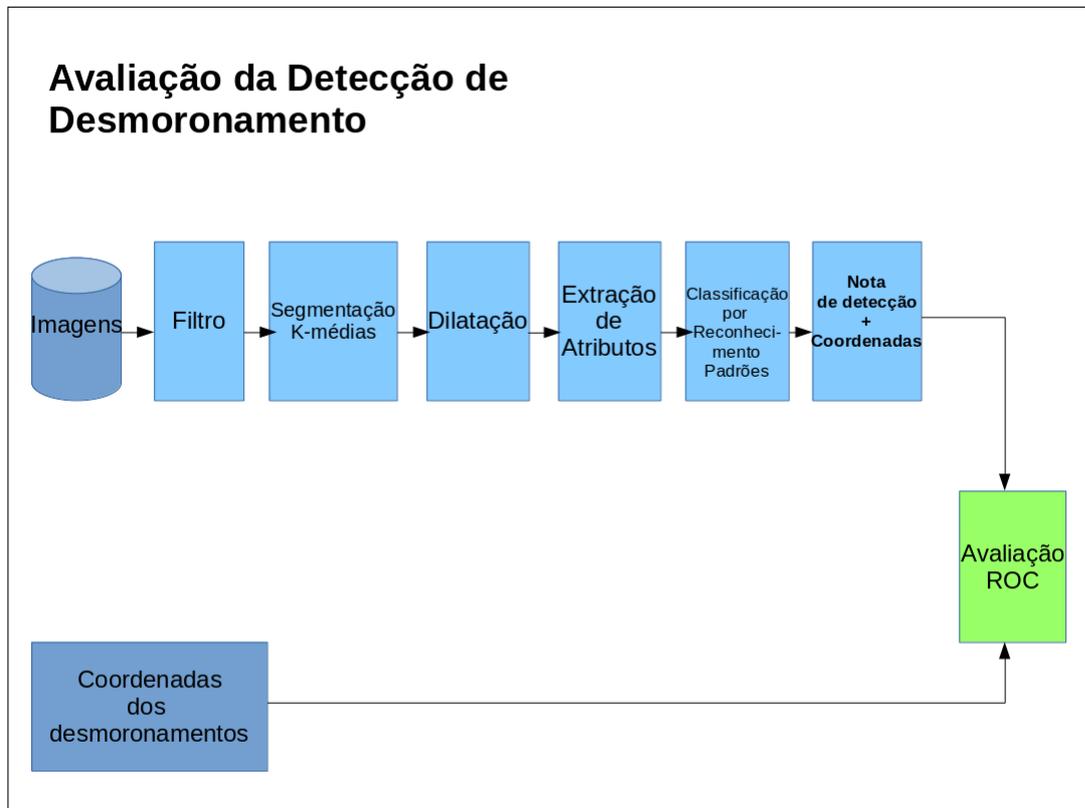


Figura 4.35 – Diagrama com o resumo das etapas do processamento digital de imagens do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP).

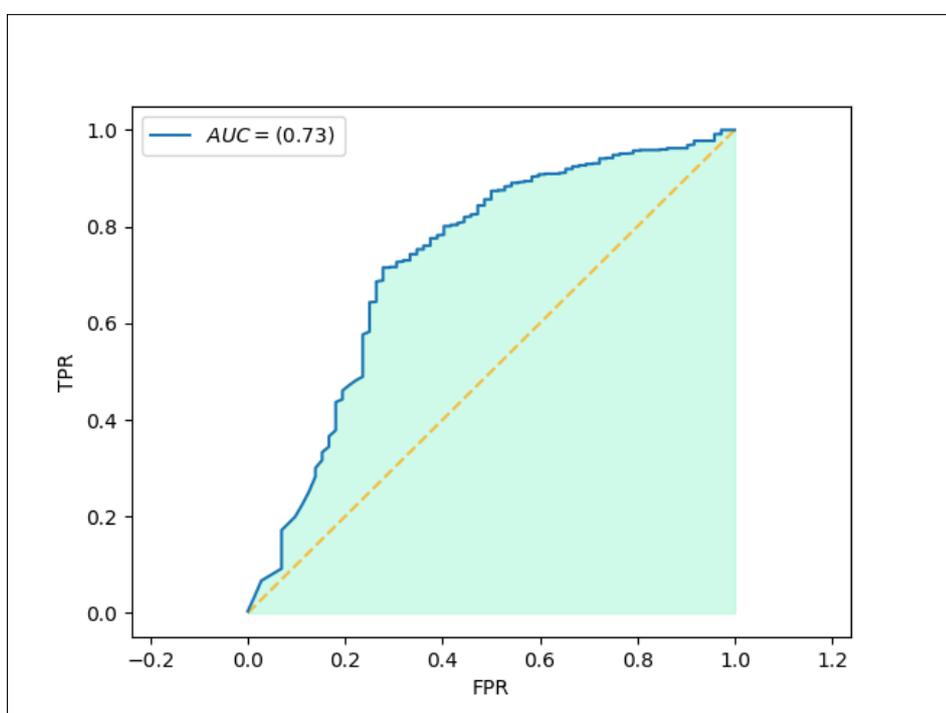


Figura 4.36 – ROC detecção do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), com área abaixo da curva AUC de 73%.

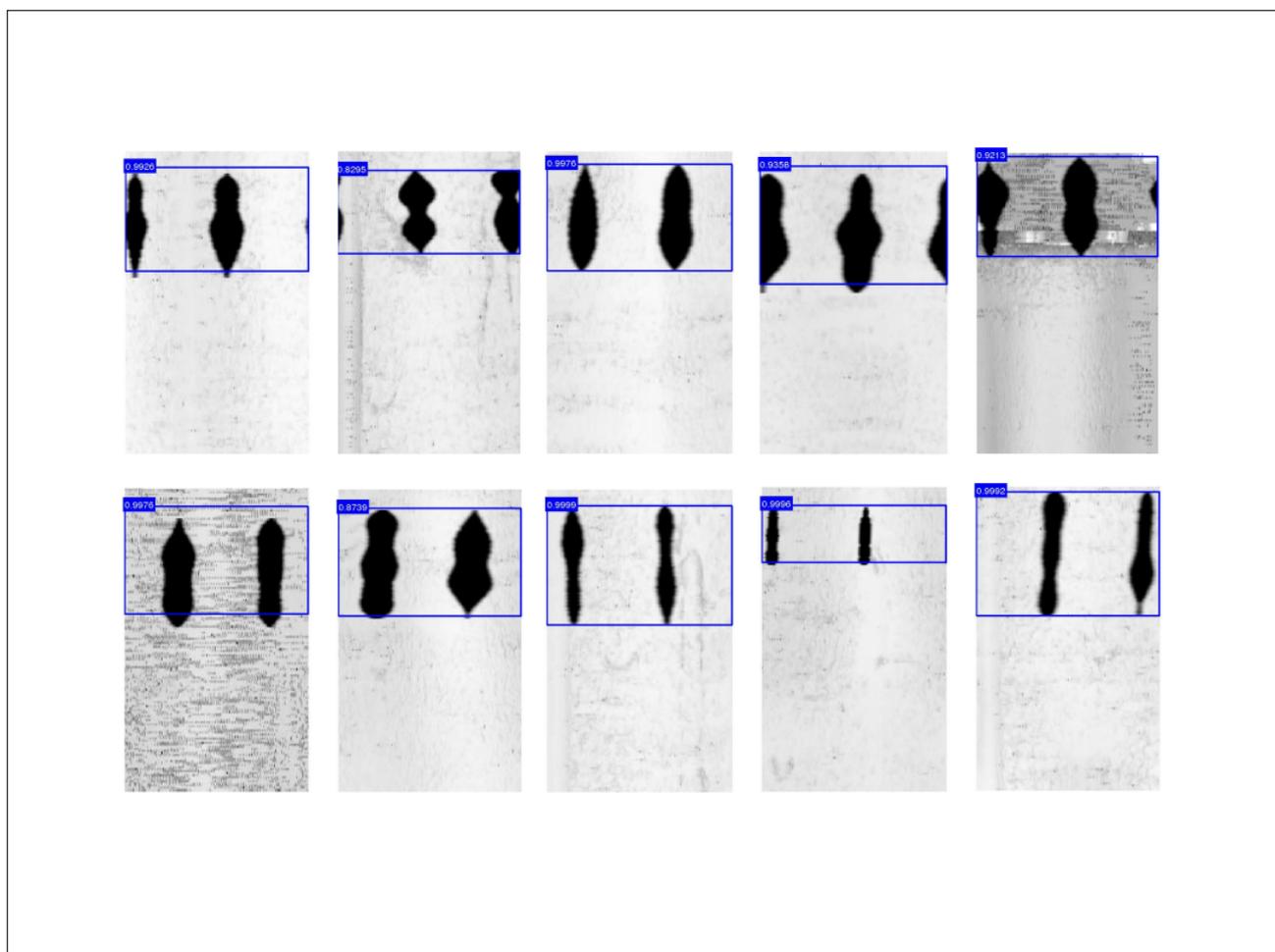


Figura 4.37 – Casos de Sucesso na detecção de desmoronamento sintéticos com Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), com notas acima de 0,5.

CAPÍTULO 5

DETECÇÃO AUTOMATIZADA UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Redes Neurais Artificiais são ferramentas vastamente usadas para determinação de padrões em imagens [231]. O objetivo deste estudo foi avaliar o uso de redes convolucionais profundas baseadas em regiões (FRCNN) como técnica para detectar fraturas e desmoronamentos e comparar seus resultados com os obtidos pelos métodos baseados em rede neural artificial (RNA) rasa e técnicas de detecção fundamentadas em processamento digital de imagem (PDI) e reconhecimento de padrões. Todas as redes utilizadas foram implementadas em *Matlab*[®] com tempos de processamento apresentados na tabela 5. A FRCNN utilizou placa GPU com suporte a CUDA. Os métodos baseados em redes neurais supervisionadas necessitam para a sua caracterização, a definição de três grupos de dados. Os dados foram divididos em conjunto de treinamento, conjunto de validação e um conjunto de teste a ser aplicado sobre a arquitetura após a fase de treinamento. As imagens do conjunto de teste possuem origem diferente das imagens de treinamento e validação e foram utilizadas para verificar a capacidade de generalização e adaptação da rede. Com o intuito de obter uma quantidade significativa de dados, imagens sintéticas [232, 152] foram construídos com simulador de fraturas e desmoronamentos, com detalhes apresentados na seção 4.2. Esse recurso contribuiu para eficiência no aprendizado de máquina [233] na primeira fase, também verificou a generalização, através da produção de um maior número e distinto de dados [234]. O estudo comparativo tem a intenção de avaliar ferramentas de detecção de ocorrências observadas em perfil de imagens acústicas de poço. Dessa forma, utilizamos um modelo estatístico para avaliar a classificação dos métodos aplicados nesse estudo utilizando curva de avaliação do classificador.

Tabela 5 – Tabela de tempo de processamento da Rede Neural Artificial (RNA) e da Rede Neural Convolutacional Rápida baseada em regiões de imagens (FRCNN). A RNA processou 2 vezes 3000 imagens, uma para cada classe.

Métodos	Fase	Tempo (hs)	Imagens	Saída
RNA	Treino + validação	4	6000	Notas
RNA	Teste	4	6000	Notas
FRCNN	Treino + validação	18	30000	Notas e coordenadas
FRCNN	Teste	3	3000	Notas e coordenadas

5.1 Modelo Proposto de Rede Neural Rasa

Utilizando a ferramenta de rede neural artificial rasa do *Matlab*[®] [105], para identificar se no trecho de imagem existem fraturas ou desmoronamentos, porém essa rede não identifica os limites das ocorrências. Essa rede, utilizou uma arquitetura com duas camadas ocultas e com 200 neurônios cada, com funções de ativação respectivas logarítmica e tangente hiperbólica, com retropropagação utilizando gradiente descendente com momento. Treinamos uma rede alimentada com vetores de valores extraídos de cada imagem a partir do cálculo de 16 atributos conhecidos como descritores de características Haralick [167, 168, 170]. Esses atributos são descritores de textura da imagem. Os descritores de contraste e aglomerado proeminente, como já apresentado, conseguem indicar possíveis fraturas e desmoronamentos. Por isso, a utilização de mais atributos contribuiu para avaliar a capacidade de detecção dessas ocorrências. A cada duas linhas da imagem foram calculadas as 16 equações de Haralick apresentadas no apêndice A, através de matrizes de co-ocorrência com dimensões 2×180 pixel. Uma matriz de co-ocorrência que chamamos de GLCM considera a relação entre dois pixels por vez. Um deles é considerado o pixel de referência e o outro de vizinho. O pixel vizinho escolhido pode, ser vizinho da direita ou 0° , da esquerda 180° , de acima 90° , de abaixo, ou vizinhos das diagonais, ou seja à 45° , 135° , 225° , 315° em relação à cada pixel referência. A co-ocorrência, pode ser especificada por uma matriz de frequências relativas levando em consideração os valores de pixel em tom de cinza que variam de 0 a 1. O de acordo com a orientação em relação ao pixel de referência, sendo computado em inicialmente no canto superior esquerdo e procedendo até o inferior direito. No final a matriz de co-ocorrência resultante são os valores médios de todas as GLCM.

As médias de cada imagem foram computadas para cada atributo, resultando em uma matriz com 16 linhas referentes aos descritores pelo número de colunas de acordo com a quantidade de imagens analisadas. Treinamos a rede com 3000 imagem sintéticas sendo escolhidas aleatoriamente 2100 para treino, 450 para validação e 450 de teste. A figura 5.2 apresenta a arquitetura da rede com a disposição das camadas de neurônios. A camada de entrada possui 16 neurônios referentes ao número de atributos calculados por imagem,

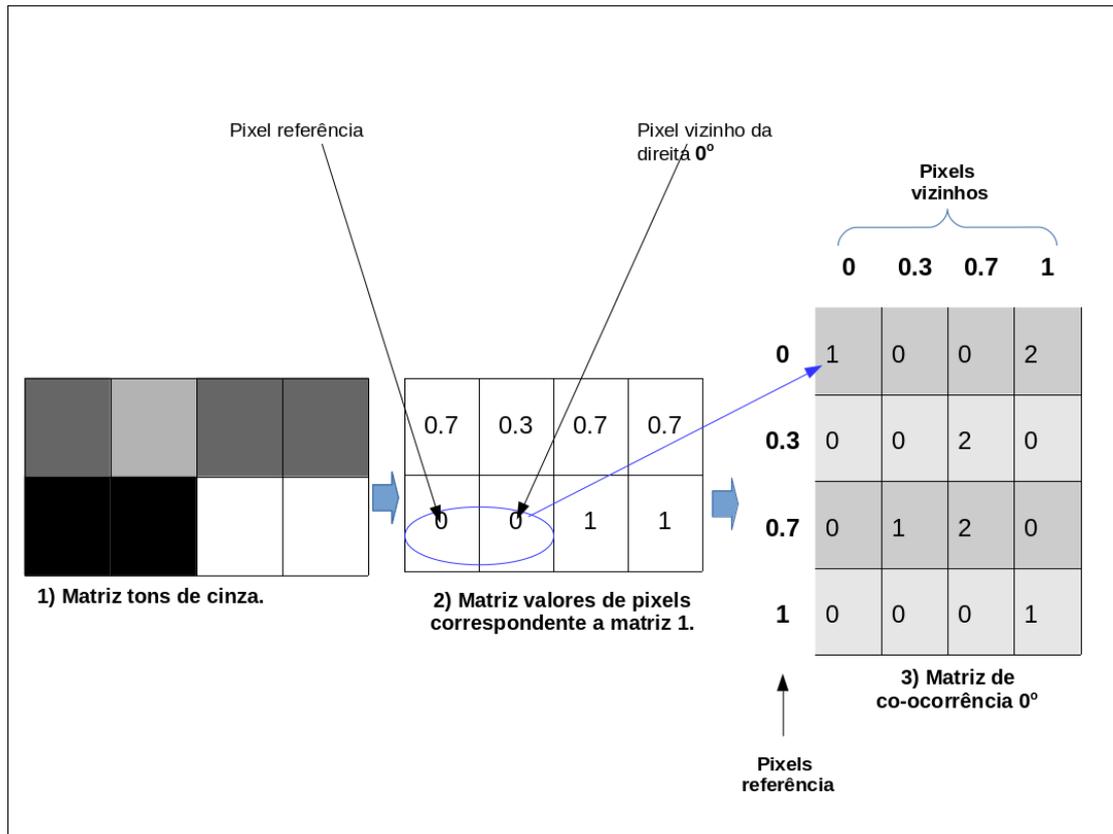


Figura 5.1 – Esquema matriz de co-ocorrência, exemplo da GLCM com frequências referência aos valores de pixels vizinhos à direita em 0° .

cada camada oculta foi definida com 200 neurônios cada e a camada de saída composta de 1 neurônio responsável pela saída da rede com uma nota de classificação. A rede foi treinada 2 vezes, a primeira para classificar fraturas e a segunda para desmoronamentos. Aplicamos a rede pronta em 3000 imagens sintéticas para avaliar a qualidade do classificador para detectar as imagens com fraturas e desmoronamento a partir dos atributos desse novo lote de teste.

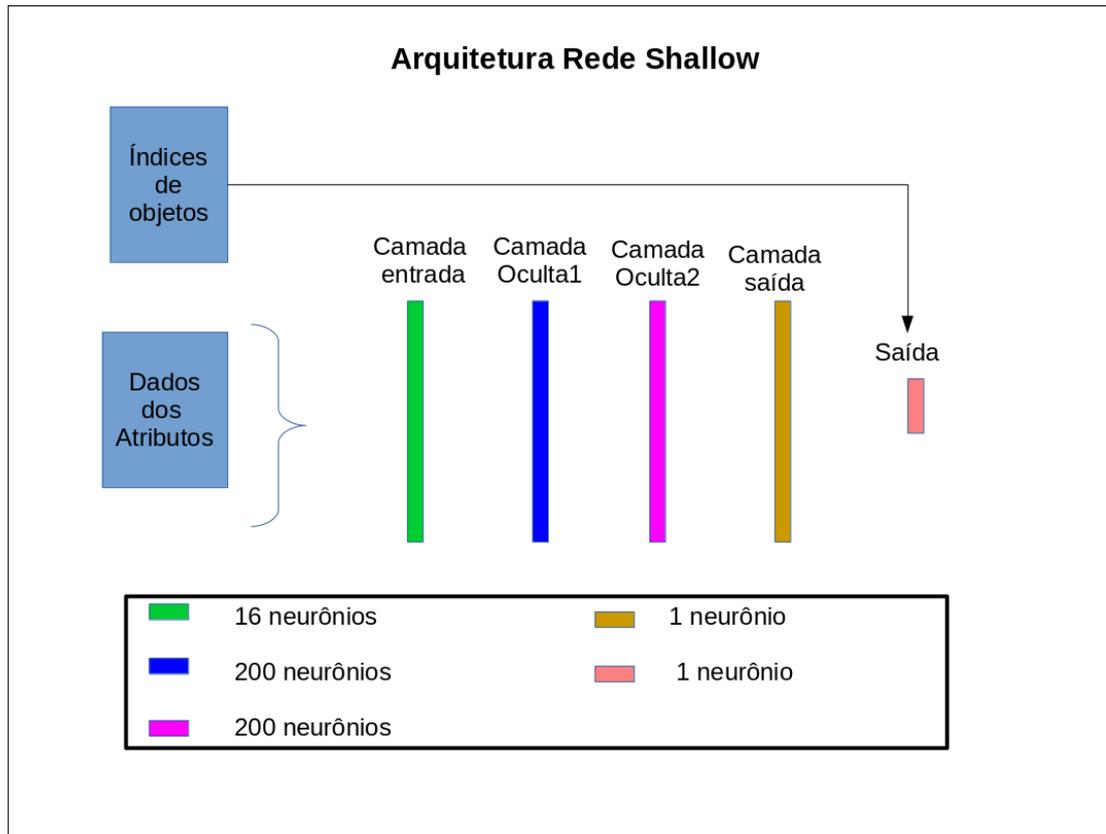


Figura 5.2 – Esquema arquitetura rede neural rasa utilizada para detectar imagens com fraturas e desmoronamentos. A camada de entrada possui 16 neurônios, cada camada oculta foi definida com 200 neurônios e a camada de saída composta de 1 neurônio responsável pela saída da rede com uma nota de classificação.

5.2 Modelo de Rede Neural Convolutacional com Proposta de Regiões de Imagem

A estratégia de transferência de conhecimento foi empregada em nosso estudo, utilizando a Rede Neural Convolutacional Rápida baseada em regiões de imagens (FRCNN) com os filtros já treinados, utilizados para detecção de objetos, animais e pessoas para inicializar a rede. Esse recurso foi utilizado durante o treinamento para iniciar todos os parâmetros, assim realizar a calibração durante o novo treinamento para um outro modelo com objetivo de detectar fraturas e desmoronamentos.

O algoritmo utilizado possui várias funções responsáveis pelo gerenciamento dos dados de entrada para treinamento. Com essas funções é possível desenvolver um código principal que lê um arquivo ASCII, contendo em cada linha o diretório e o nome das imagem de entrada, as coordenadas que definem o *bounding box*, i.e caixas que limitam a região do objeto ($x_{min}, y_{min}, x_{max}, y_{max}$) pré classificados e a respectiva classe dos objetos. Com essas informações, outras funções são chamadas para gerar as regiões propostas *RP* pelo processo detalhado em 3.4.4.1 e os catálogos de acesso das informações por imagem como um banco de dados que o algoritmo acessa enquanto treina a rede.

Durante a construção dos lotes utilizados para aprendizado, são selecionadas aleatoriamente as imagens de treino, validação, teste do treinamento. O lote de treino conhecido como *training set* é utilizado para computar o gradiente e atualizar os pesos e vieses durante um percentual do número total definido de interações. O lote de validação conhecido como *validation set* é o conjunto de dados usados para monitorar os erros após o treinamento com o lote de treino entre as interações. O lote de teste, *testing set*, é um conjunto pertencente aos dados de treinamento que testa a rede no final.

Essa divisão faz sentido quando pensamos em um número maior para o lote de validação voltado para um maior aproveitamento dos ajustes dos parâmetros durante o aprendizado.

Nós construímos uma interface para gerenciar os *inputs* para treinamento, tornando mais fácil a utilização da rede. Dessa forma, ao apontar os diretórios onde estão as imagens de treinamento de cada classe e os arquivos que indicam as coordenadas dos objetos em cada uma delas, é possível treinar a rede. As imagens de treino foram geradas por simulações baseadas no Dado 1. Nesse trabalho, simulamos um único tipo de ocorrência, i.e uma fratura ou um desmoronamento, por imagem, entretanto o algoritmo simulador é capaz de realizar combinações.

A FRCNN após treinada é capaz de detectar fraturas e desmoronamento e identificar a região de cada ocorrência.

O lote de treino com 30000 imagens foi composto de 10000 imagens contendo fraturas, 10000 contendo desmoronamentos e mais 10000 sem nenhuma das duas ocorrências.

5.3 Avaliação Detecção de Fraturas e Desmoronamentos com RNA Rasa

A rede foi treinada utilizando as imagens referente ao Dado 1, representada em vetores de valores de entrada referente aos valores médios calculados com as equações de Haralick, utilizando matrizes de co-ocorrência em uma janela de varredura equivalente a duas linha da imagem. A arquitetura e as imagens utilizadas foram análogas a apresentada na avaliação de detecção de fraturas com rede rasa, porém nesse caso, a rede foi treinada para reconhecer desmoronamentos. Essa aplicação do modelo conforme apresentamos no diagrama da figura 5.8, possibilitou analisar o desempenho do classificador de fraturas e desmoronamentos com a ROC apresentada em 5.4 e 5.6, apresentando áreas abaixo da curva respectivas de 0,95 e 0,99. O uso da RNA estabelece uma dimensão fixa nos dados de treinamento, validação e teste. Esse tipo de rede não identifica a quantidade de detecções e a precisão de sua localização em uma única imagem, por isso a caixa detectora demarca a imagem inteira.

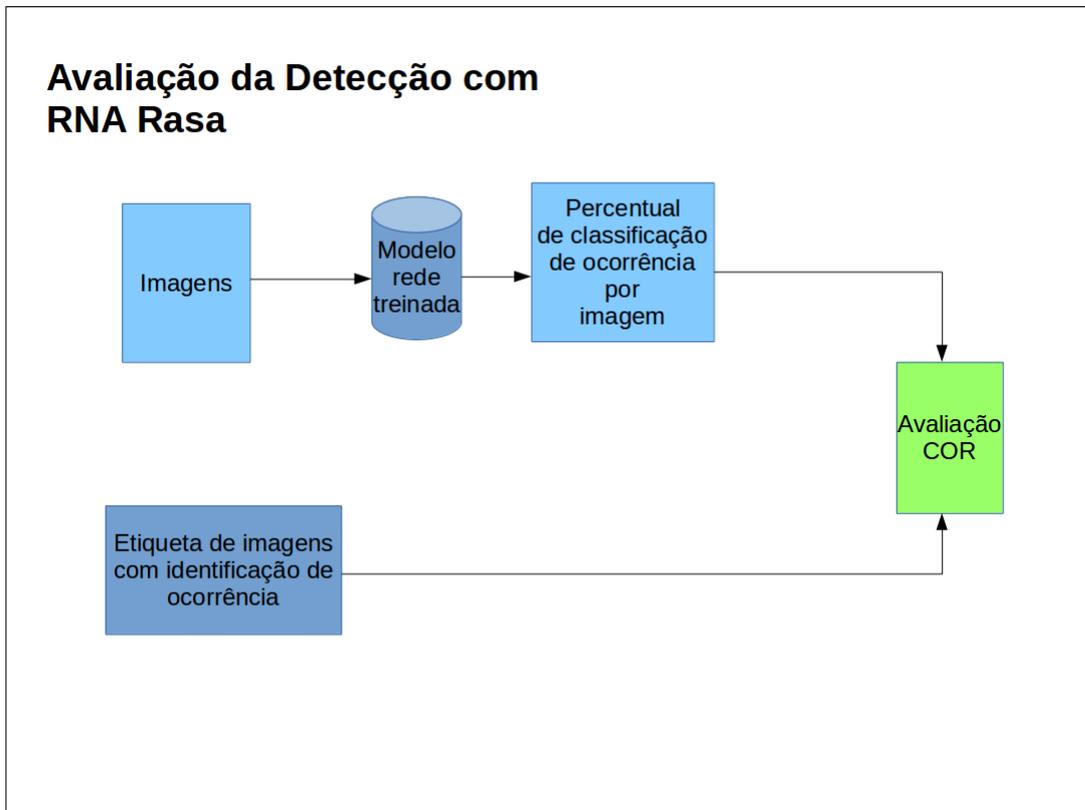


Figura 5.3 – Diagrama geral de trabalho até avaliação utilizando rede neural artificial rasa. Esse tipo de rede diferente dos demais não identifica as regiões das ocorrências.

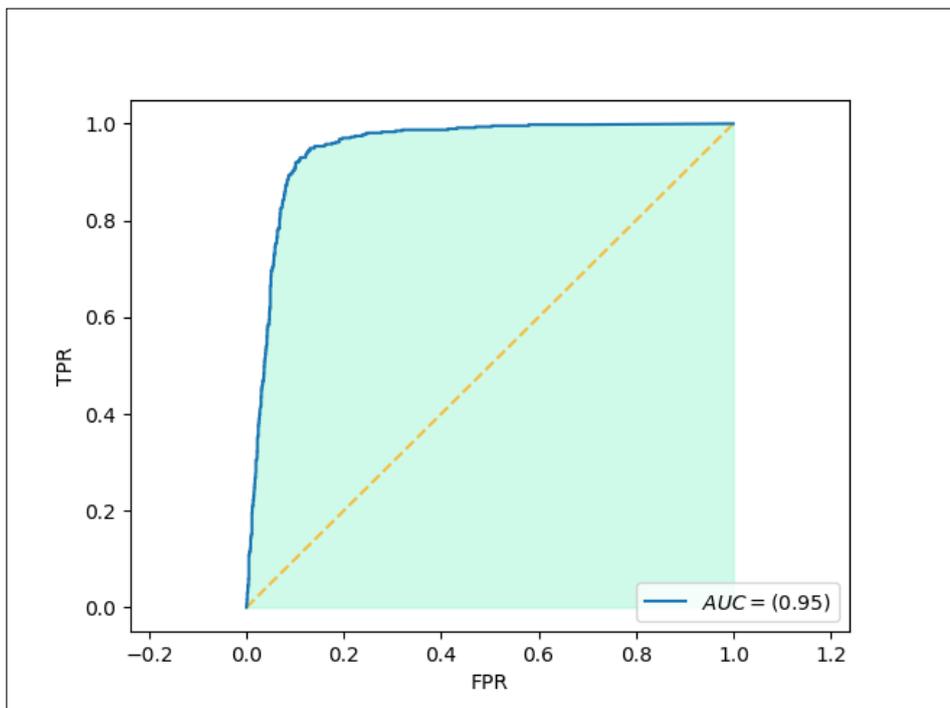


Figura 5.4 – ROC do classificador de fraturas com RNA rasa.

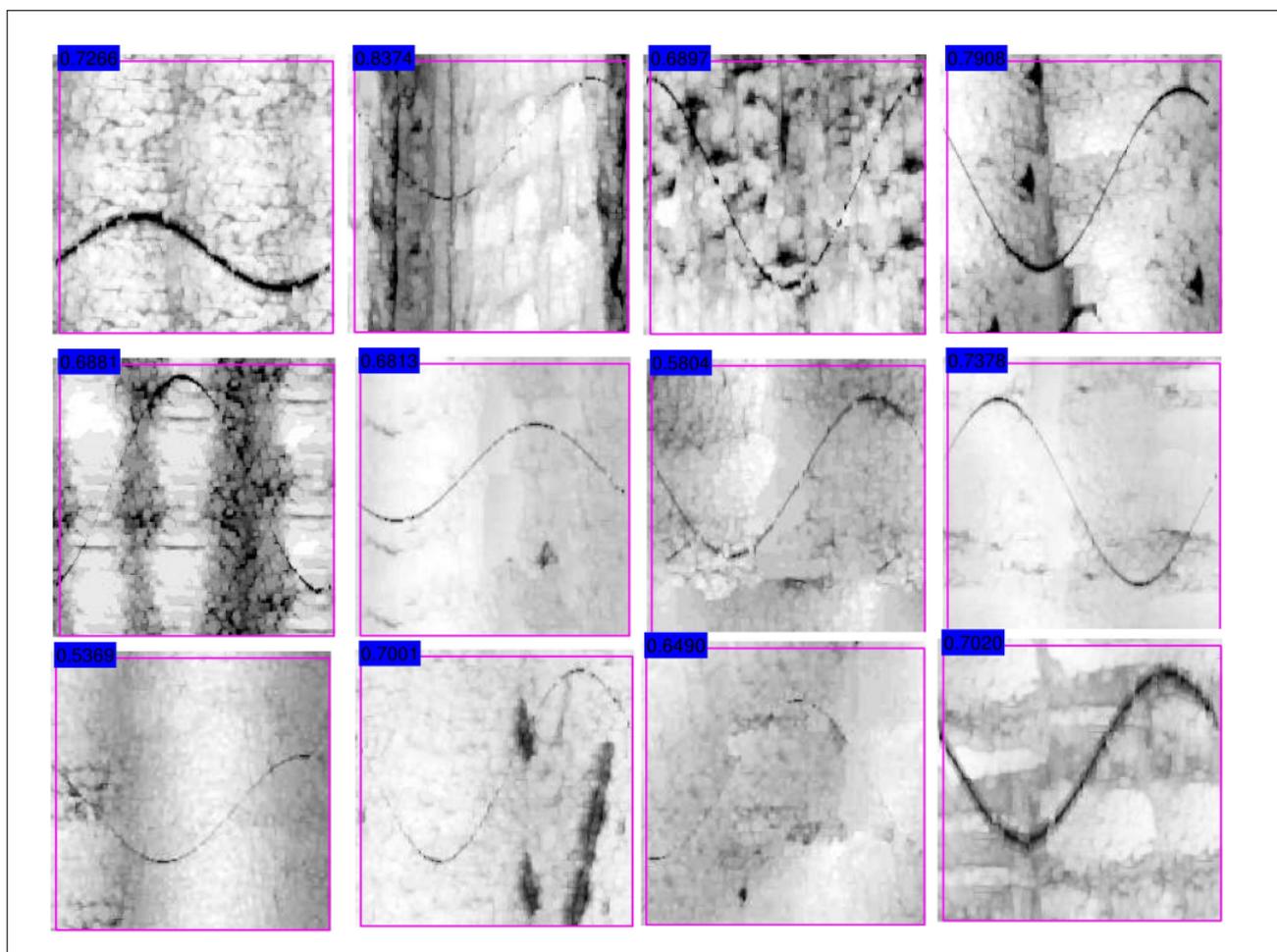


Figura 5.5 – Casos de verdadeiros positivos de detecção de fraturas em imagens sintéticas utilizando RNA rasa. Apresentamos detecções com predição superior à 50 %. Contudo, essa rede não define a quantidade e os limites precisos da detecção.

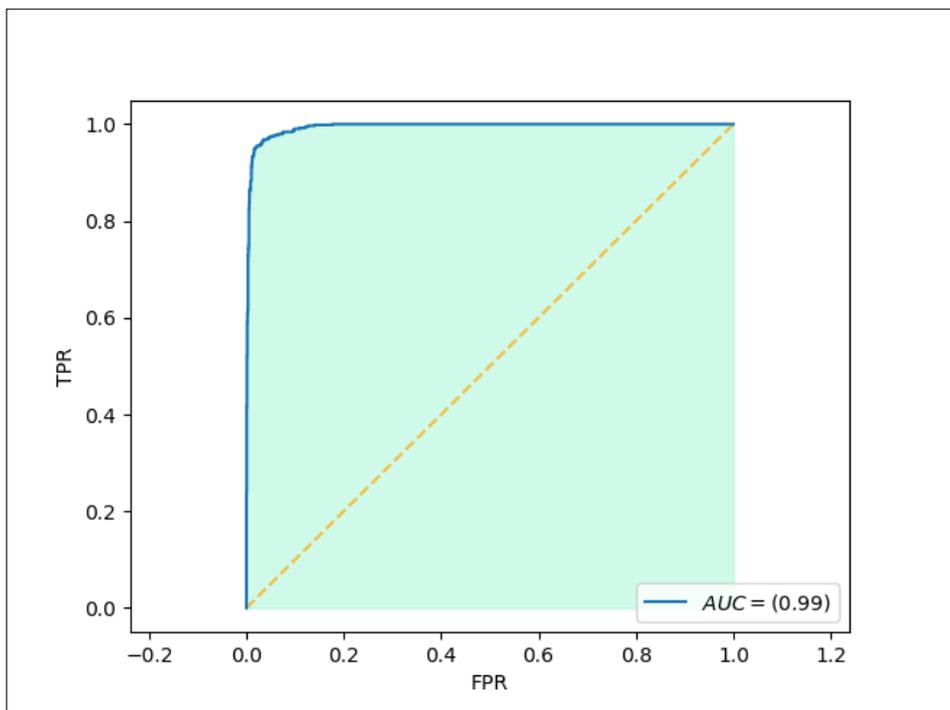


Figura 5.6 – ROC do classificador de desmoronamento com RNA rasa com duas camadas ocultas. A detecção de desmoronamento dessa rede, não identifica o número de ocorrências e a localização precisa, por isso, a caixa delimitadora demarca toda imagem.

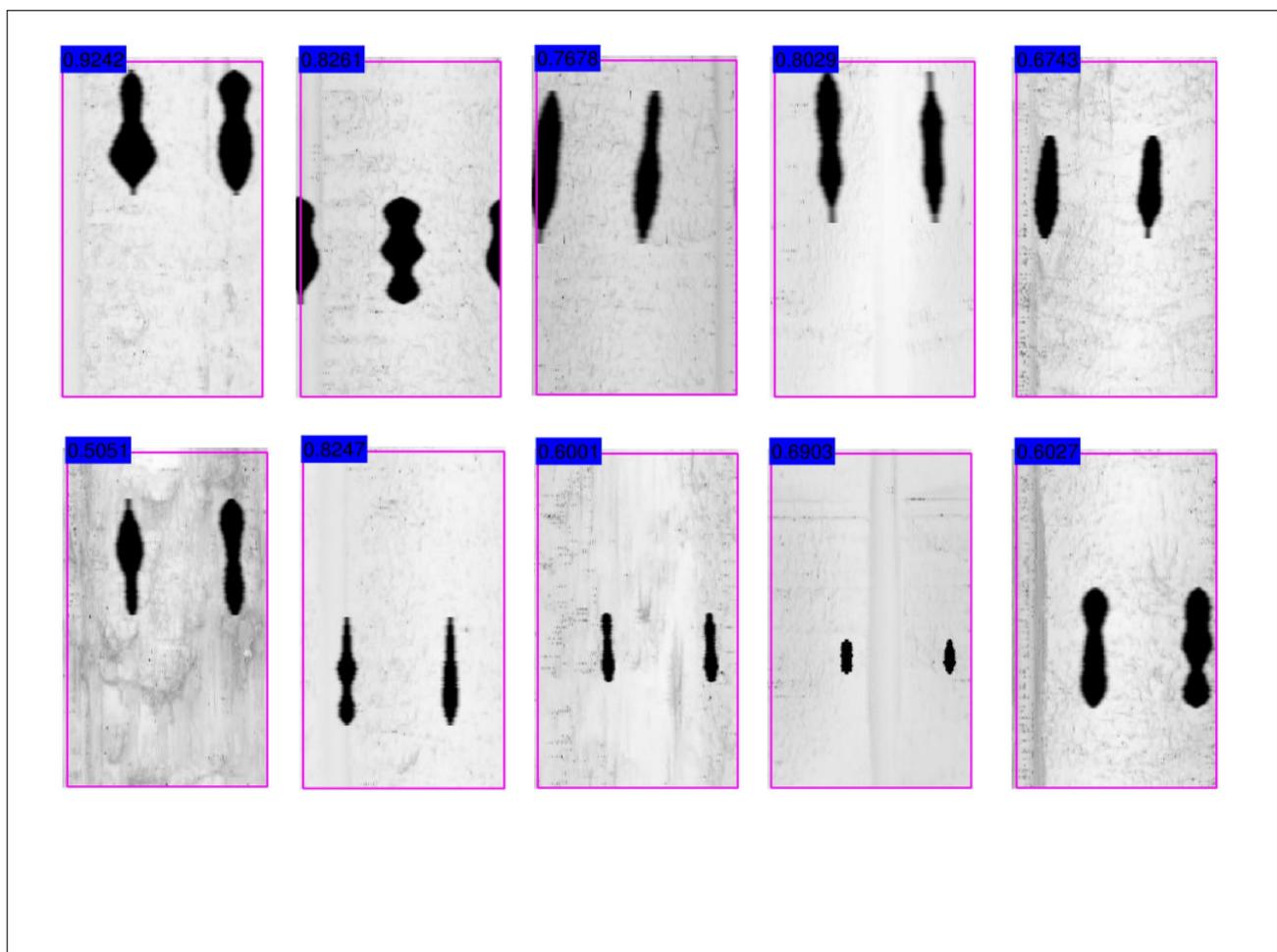


Figura 5.7 – Casos verdadeiros de detecções de desmoronamentos com rede neural rasa. O detector de desmoronamentos dessa rede não identifica o número de ocorrências e os limites mais preciso, por isso, a caixa delimitadora demarca toda imagem.

5.4 Avaliação da Detecção de Fraturas e Desmoronamentos com FRCNN

Com o modelo FRCNN, treinado para detectar fraturas e desmoronamentos, submetemos cada imagem simulada com base no Dado 2 à avaliação da rede. Observamos que a capacidade de generalização da rede foi suficiente para obter índices de áreas abaixo da curva *ROC* de 0,86 e 0,97, respectivas de detecção de fraturas e desmoronamentos como apresentado em 5.9 e 5.12. A *ROC* de fraturas apresentou um comportamento abrupto à esquerda, causada pelos critérios utilizados para definir as regiões de detecção entre todas as analisadas pela rede. Como a saída da rede apresenta uma nota para cada região proposta de entrada, o controle da sobreposição permite comparar as coordenadas das regiões verdadeiras com as coordenadas das regiões detectadas. Dessa forma, o ponto de centro geométrico de cada região verdadeira foi comparado com o centro de cada região classificada, e as que apresentaram distâncias menores de 30 pixels foram considerados acertos. Na maioria das imagens, as fraturas sintéticas são mais difíceis de identificar que os desmoronamentos. Esse fato ocorre devido a várias técnicas de degradação aplicadas durante a construção da senóide. A avaliação visual dos resultados das detecções nas imagens apresentadas, utilizou corte considerando notas acima de 0,5 e de sobreposição de regiões acima de 70%.

O detector apresentou casos onde fraturas simuladas com níveis máximos de descontinuidade e níveis mínimos de foco e contraste foram encontradas conforme apresentamos em alguns exemplos de casos positivos de detecção na figura 5.10. Os desmoronamentos sintéticos foram bem detectados, ainda que em alguns casos o modelo não detectou o par e sim dois desmoronamentos, como apresentamos alguns exemplos em 5.13, utilizando os mesmos cortes de nota e sobreposição. No entanto, ainda foram registrados casos considerados de falsos positivos para fraturas e desmoronamentos, conforme apresentamos alguns exemplos nas figura 5.11 e 5.14. A FRCNN apresenta uma nota para cada região proposta, por isso, ela apresenta um número maior de detecções comparados aos métodos de processamento digital de imagens (PDI). Os métodos utilizando PDI possuem a vantagem de utilizar na etapa de pós-processamento a exclusão de objetos com relação a forma, deixando a classificação para um número limitado de detecções. No entanto, o método FRCNN comparado aos métodos de detecção utilizando PDI, apresentou uma curva *ROC* com melhor percentual para as ocorrências de fraturas e desmoronamentos. Além disso, conseguiu detectar fraturas em meio à uma variedade de artefatos introduzidos pela simulação da imagem, algo que os demais métodos clássicos utilizando PDI não conseguiram.

O modelo FRCNN não superou o resultado *ROC* do modelo de RNA rasa como visto na tabela 6, mas possui a vantagem de utilizar como entrada a imagens, sem um pré processamento para extração de atributos. Além disso, a detecção com a FRCNN é

invariante em relação a escala, consegue determinar com precisão os limites das detecções oferecendo as coordenadas de cada ocorrência e é capaz de identificar simultaneamente múltiplas fraturas e desmoronamentos na imagem.

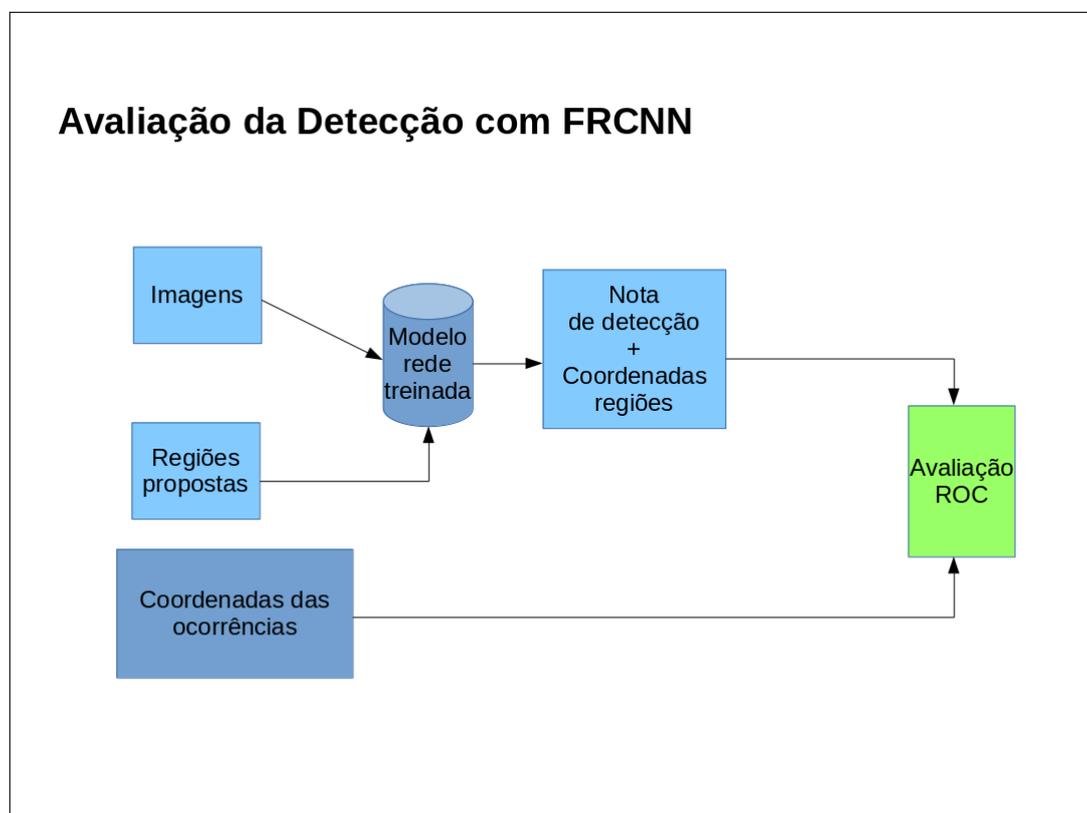


Figura 5.8 – Diagrama resumido com etapas até a avaliação de detecções utilizando Rede Neural Convolucional Rápida baseada em Regiões (FRCNN).

Tabela 6 – Tabela de percentuais de área abaixo da curva AUC da Rede Neural Convolucional Rápida baseada em Regiões (FRCNN) e Rede Neural Artificial (RNA) por classes. Apesar dos maiores índices AUC a RNA não é invariante a escala da imagem, não localiza as ocorrências e não é capaz de realizar múltiplas detecções de múltiplas classes em uma única imagem.

Redes	Classes	AUC	Saída
RNA	Fraturas	95 %	Notas
RNA	Desmoronamentos	99 %	Notas
FRCNN	Fraturas	86 %	Notas e coordenadas
FRCNN	Desmoronamentos	97 %	Notas e coordenadas

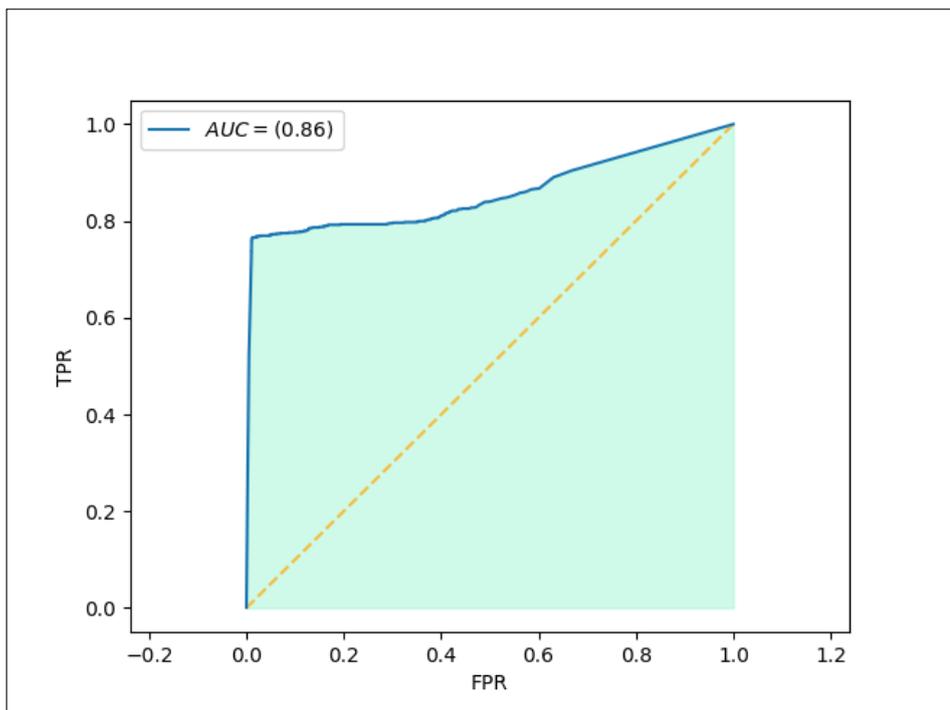


Figura 5.9 – ROC de desempenho da detecção de fraturas sintéticas utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN). O comportamento abrupto à esquerda, foi causado pelos critérios utilizados para definir as regiões de acerto entre todas as analisadas pela rede.

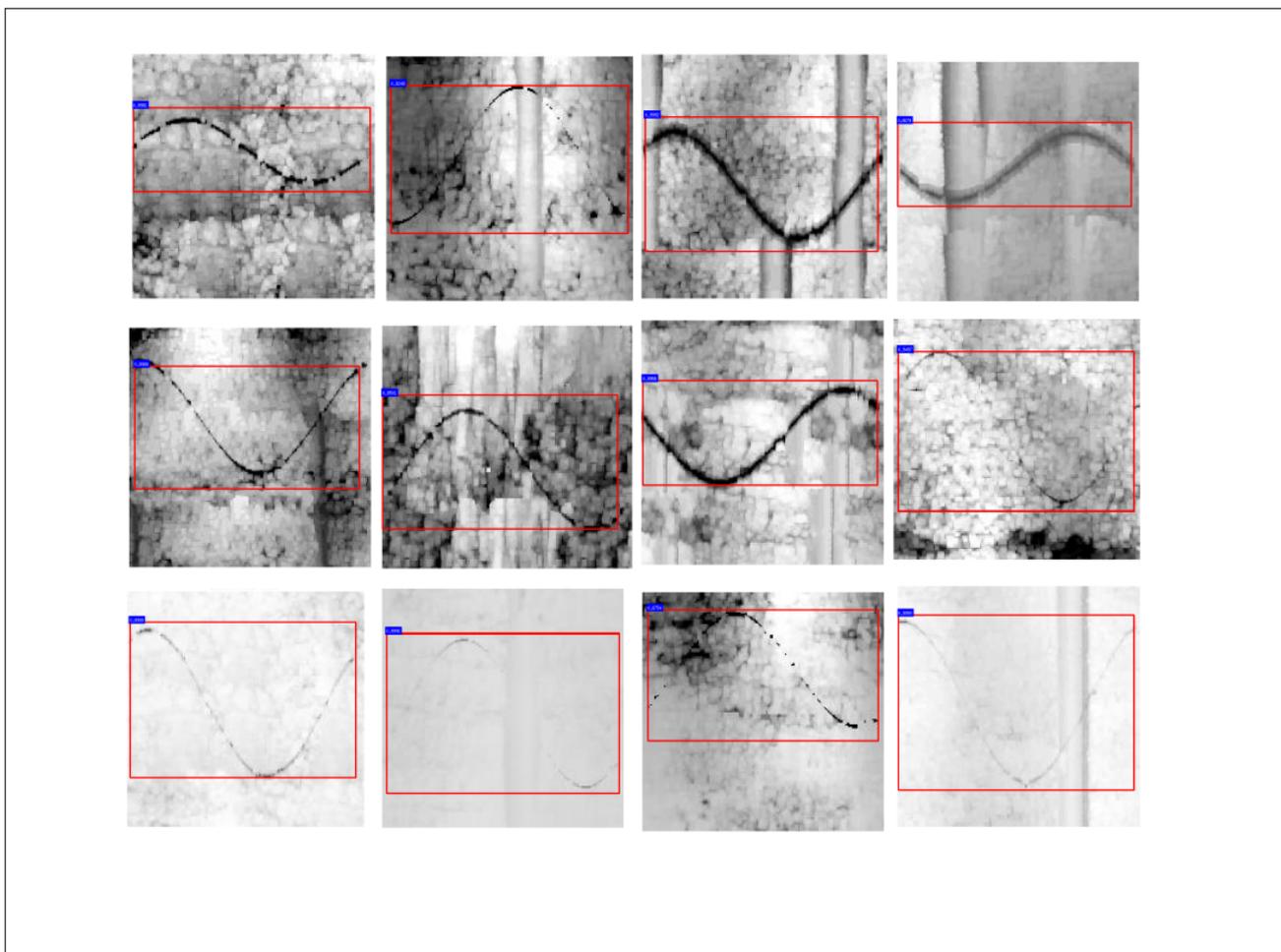


Figura 5.10 – Casos de verdadeiros positivos de detecção de regiões de fraturas sintéticas com Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN). Apresentamos regiões de detecção com classificação superior à 0,5, aceitando sobreposição de regiões acima de 70 % com regiões verdadeiras.

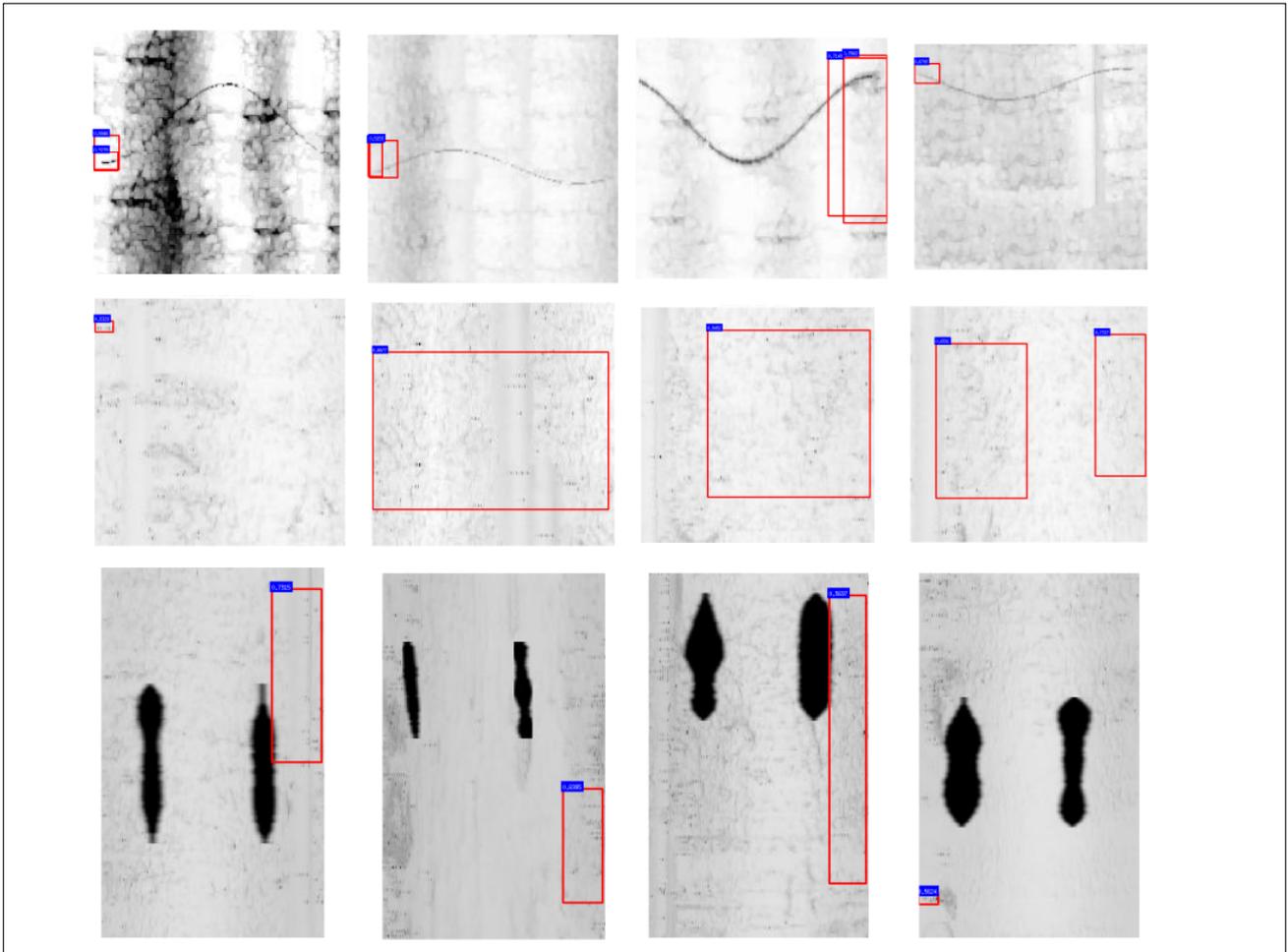


Figura 5.11 – Casos de verdadeiros positivos de detecção de regiões de fraturas sintéticas com Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN). Apresentamos regiões de detecção com nota superior à 0,5, aceitando sobreposição de regiões acima de 70% com regiões verdadeiras. As caixas delimitadoras de detecções apresentadas nas imagens, identificaram regiões sem ocorrências verdadeiras ou percentual inferior a 70% das dimensões de fraturas e desmoronamentos.

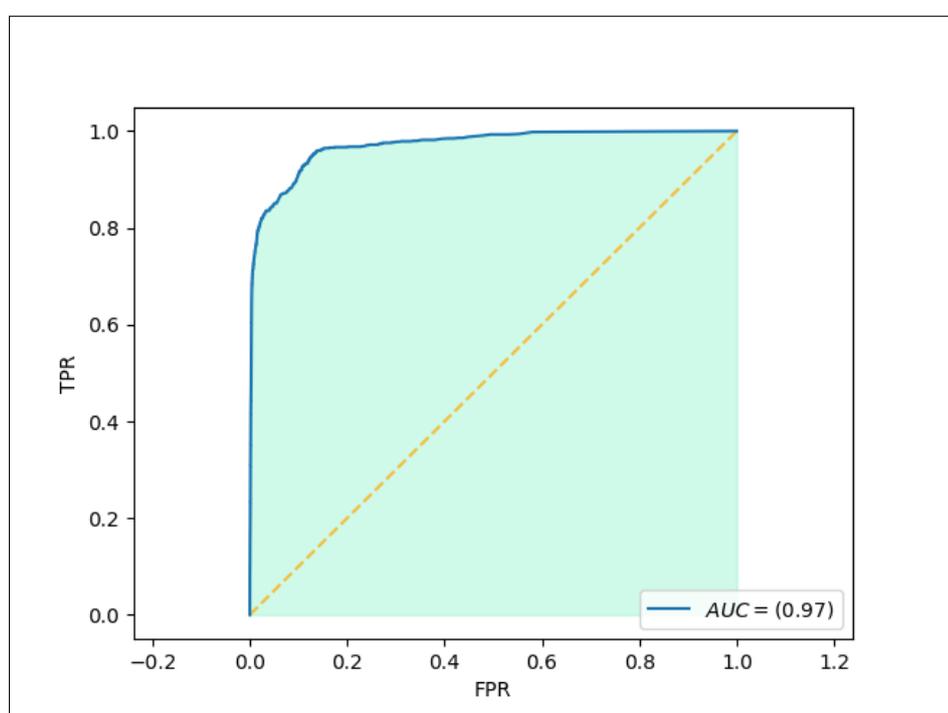


Figura 5.12 – ROC de detecções de desmoronamentos Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN).

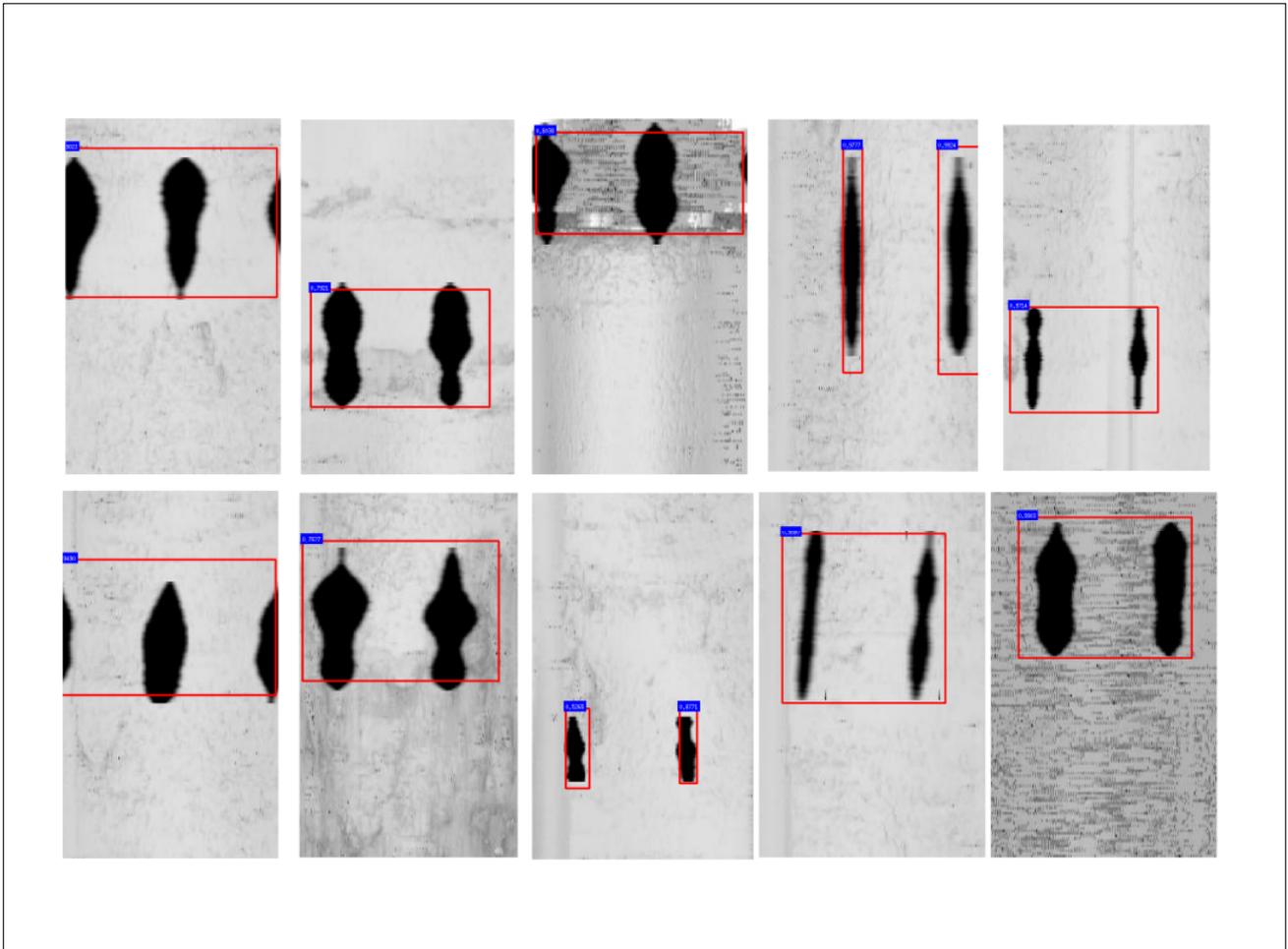


Figura 5.13 – Casos de verdadeiros positivos de desmoronamentos com regiões de detecção com classificação superior à 0,5 em imagens sintéticas utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN), aceitando sobreposição de regiões acima de 70 % com regiões verdadeiras.

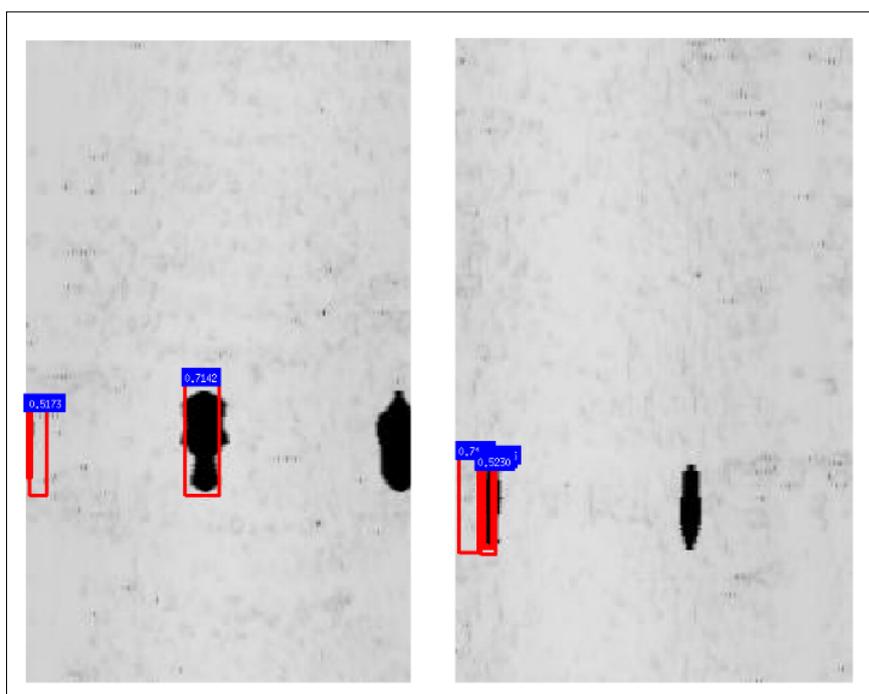


Figura 5.14 – Casos considerados como falsos positivos em imagens simuladas de desmoronamento, utilizando Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN), com regiões de detecção com classificação superior à 0,5, aceitando sobreposição de regiões acima de 70% com regiões verdadeiras.

5.5 Detecção de Fraturas e Desmoronamentos Reais

Utilizamos o Dado 3 composto de imagem acústica real disponível ao público por Cornet [12], para teste adicional de detecção de fraturas e desmoronamentos. Comparamos os modelos de detecção que determinam a coordenada das ocorrências de fraturas e desmoronamentos [235, 236]. Entre as técnicas de processamento digital de imagem utilizadas para detecção de fraturas, escolhemos o Método de Classificação por Polinômio (MCP) e para detectar desmoronamentos o Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP). Considerando os métodos de Inteligência Artificial, utilizamos para esse teste apenas a Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN), pois é a única rede que apresenta na saída as coordenadas detectando múltiplas ocorrências em uma única imagem. Assim, o MCP e a FRCNN foram utilizadas na detecção de fraturas reais e o DRP e FRCNN aplicados na detecção de desmoronamentos reais. As tabelas 7 e 8 apresentam os resultados comparativos de área abaixo da curva *AUC*, anteriormente obtidos com as imagens sintéticas e os acertos em imagens reais. As imagens reais foram classificadas pela colaboradora Juliana Coelho.

Apesar dos dados reais oferecerem um grau de complexidade maior, por apresentarem por exemplo o artefato *key seat*, foi possível obter acertos utilizando a FRCNN como nas amostras reais se considerarmos objetos com nota maior que 0,1 e admitindo regiões de detecção sobrepostas em até 70%, encontrando 10 fraturas de um total de 21 fraturas reais possíveis e confirmadas e 2 falsas ocorrências.

Os resultados utilizando MCP obtiveram classificações corretas nas amostras reais se considerarmos objetos com nota maior que 0,1, encontrando 5 fraturas de um total de 21 fraturas reais possíveis e confirmadas e 2 falsas ocorrências.

Durante a inspeção visual, observamos que o método MCP, diferente dos resultados da avaliação com imagens sintéticas, apresentou em muitos casos dificuldade no reconhecimento em fraturas reais como exemplos apresentados nas figuras 5.15.

No entanto, a detecção utilizando a rede neural profunda FRCNN treinada com imagens sintéticas, apresentou maior precisão em trechos reais de poço, como pode ser visto nos exemplos de sucesso, entretanto ocorreram casos de falsa detecção nas figuras 5.16. Em alguns casos, encontrou partes de possíveis fraturas, localizadas próximas à desmoronamentos.

As detecções de desmoronamentos utilizando a FRCNN treinada e o Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP) em imagens reais, apresentaram múltiplas detecções em um mesmo par de desmoronamento. Isso ocorreu, pois os pares de desmoronamentos estavam acompanhados de marcas de ferramenta *key seat*, ou em alguns trechos estavam unidos e isso afetou a eficiência dos classificadores. Esse fato fez com que as notas de detecção ficassem reduzidas. Por isso, normalizamos os valores e apresentamos alguns casos de verdadeiros positivos nas figuras 5.17 e 5.18, considerando como desmoronamentos notas superiores à 0,1, admitindo sobreposição entre

caixas de detecção em até 50%. As caixas em amarelo nas imagens foram introduzidas manualmente, para delimitar toda região de desmoronamentos reais. O Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP) encontrou 3 desmoronamentos de um total de 5 desmoronamentos reais e 0 falsa ocorrência. As caixas azuis, são as múltiplas detecções e a amarela a região dos desmoronamentos. Por inspeção visual, observamos que todas as caixas azuis estavam sobrepostas em relação as caixas amarelas em taxa superior à 50%, por isso consideramos todas as detecções válidas. A FRCNN encontrou 4 desmoronamentos de um total de 5 desmoronamentos reais e 0 falsa ocorrência. As caixas vermelhas são as múltiplas detecções e a amarela a região de desmoronamentos. Também observamos que todas as caixas vermelhas estavam sobrepostas em relação as caixas amarelas em taxa superior à 50%, por isso consideramos todas as detecções válidas.

Tabela 7 – Tabela comparativa de percentuais de área abaixo da curva *AUC* com imagens sintéticas, assim como, o número de acertos e falsas detecção de fraturas reais. Foram comparados o Método de Classificação por Polinômio (MCP) e a Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN) de detecção de fraturas, que são invariantes a escala e apresentam notas e coordenadas.

Modelo	AUC	Fraturas	Acertos	Falsas Detecções
MCP	85 %	21	5	2
FRCNN	86 %	21	10	2

Tabela 8 – Tabela comparativa de percentuais de área abaixo da curva *AUC* com imagens sintéticas, assim como, o número de acertos e falsas detecção de desmoronamentos reais. Foram comparados o Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP) e a Rede Neural Convolutiva Rápida baseada em Regiões (FRCNN) de detecção de desmoronamentos, que são invariantes a escala e apresentam notas e coordenadas.

Modelo	AUC	Desmoronamentos	Acertos	Falsas Detecções
DRP	73 %	5	3	0
FRCNN	97 %	5	4	0

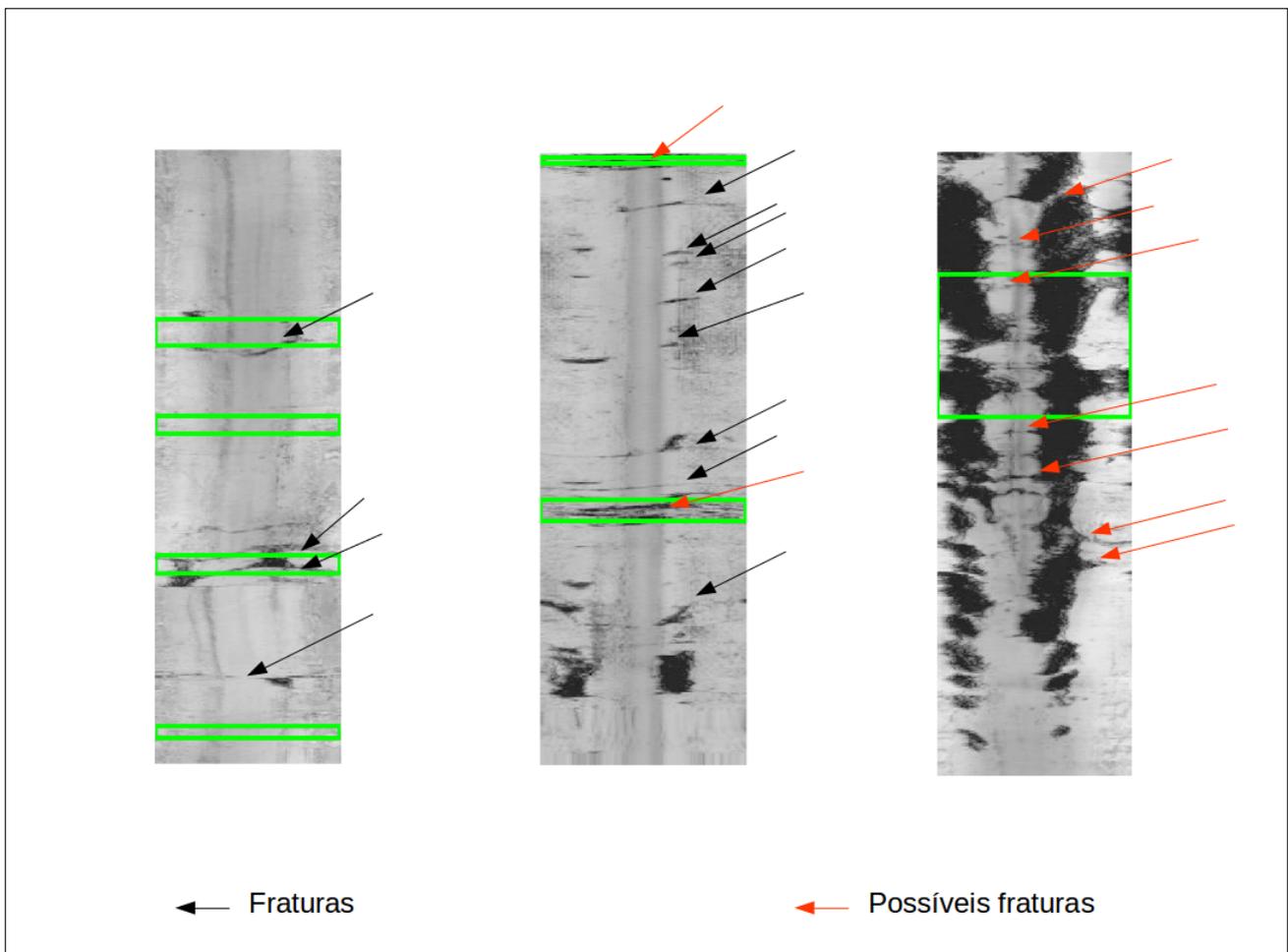


Figura 5.15 – Detecção de fraturas reais utilizando Método de Classificação por Polinômio em trecho de dado obtido em [12]. Apresentamos alguns casos de sucesso de detecção, porém com alguns falsos positivos.

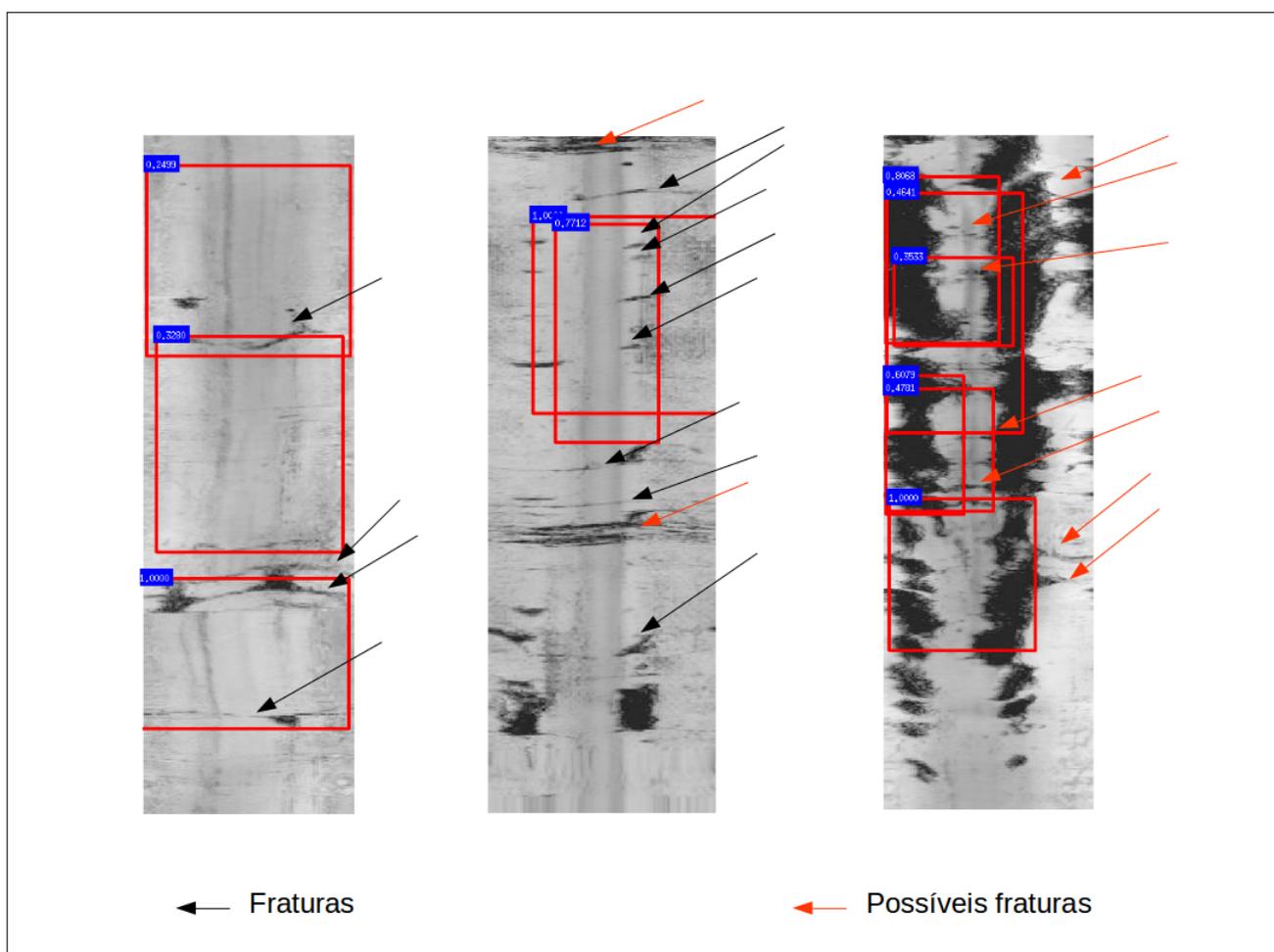


Figura 5.16 – Detecção de fraturas utilizando FRCNN em fraturas reais, de dado disponível ao público obtido em [12]. Apresentamos resultados com nota de detecção de fraturas superior à 0,1 e admitindo regiões sobrepostas em maiores que 70%.

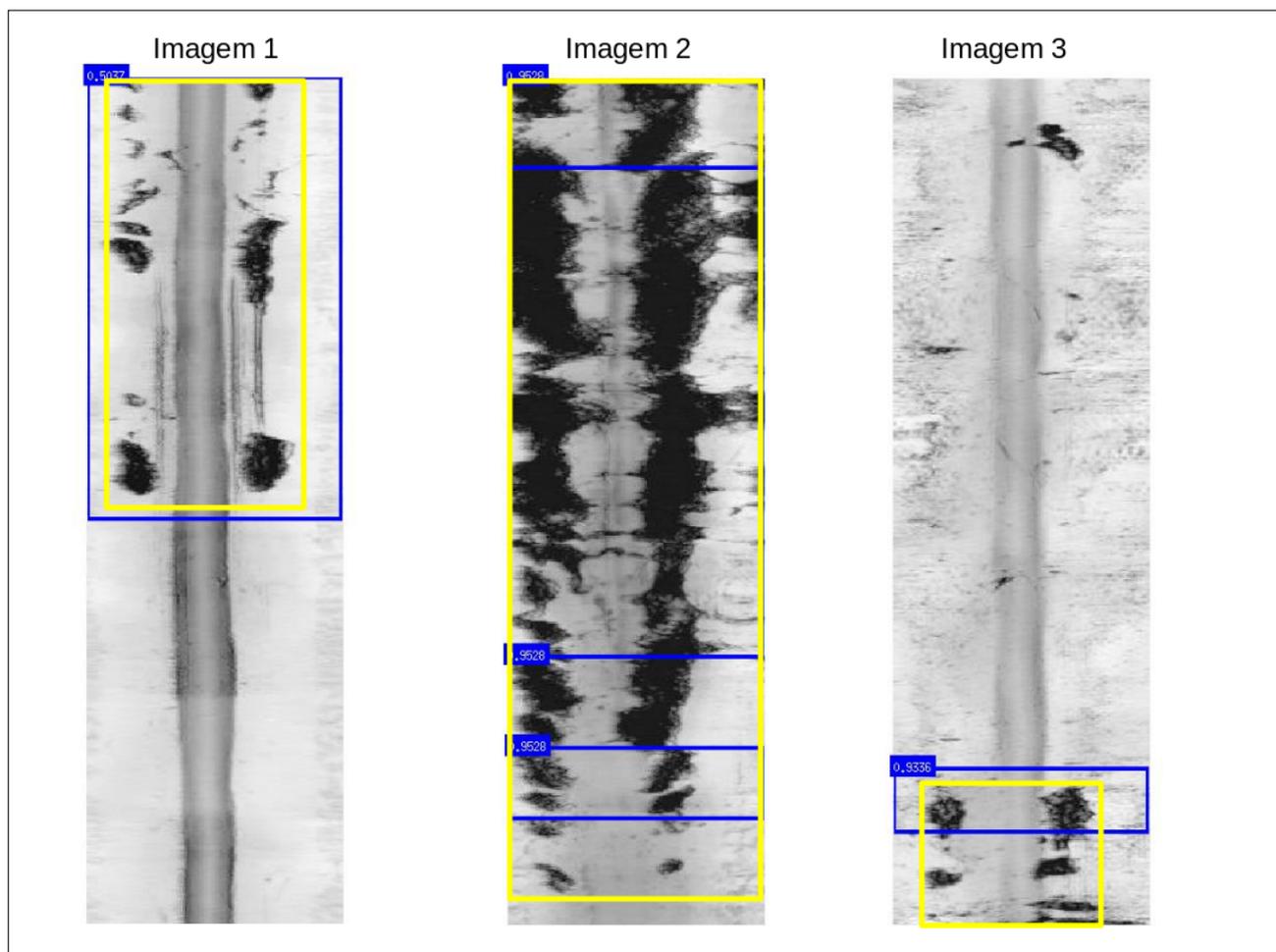


Figura 5.17 – Detecção de desmoronamentos reais utilizando Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP) em trecho de dado de acesso público obtido de [12]. O desmoronamento da direita não foi detectado com a FRCNN. Os demais encontrados pelo Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), também foram encontrados pela FRCNN.

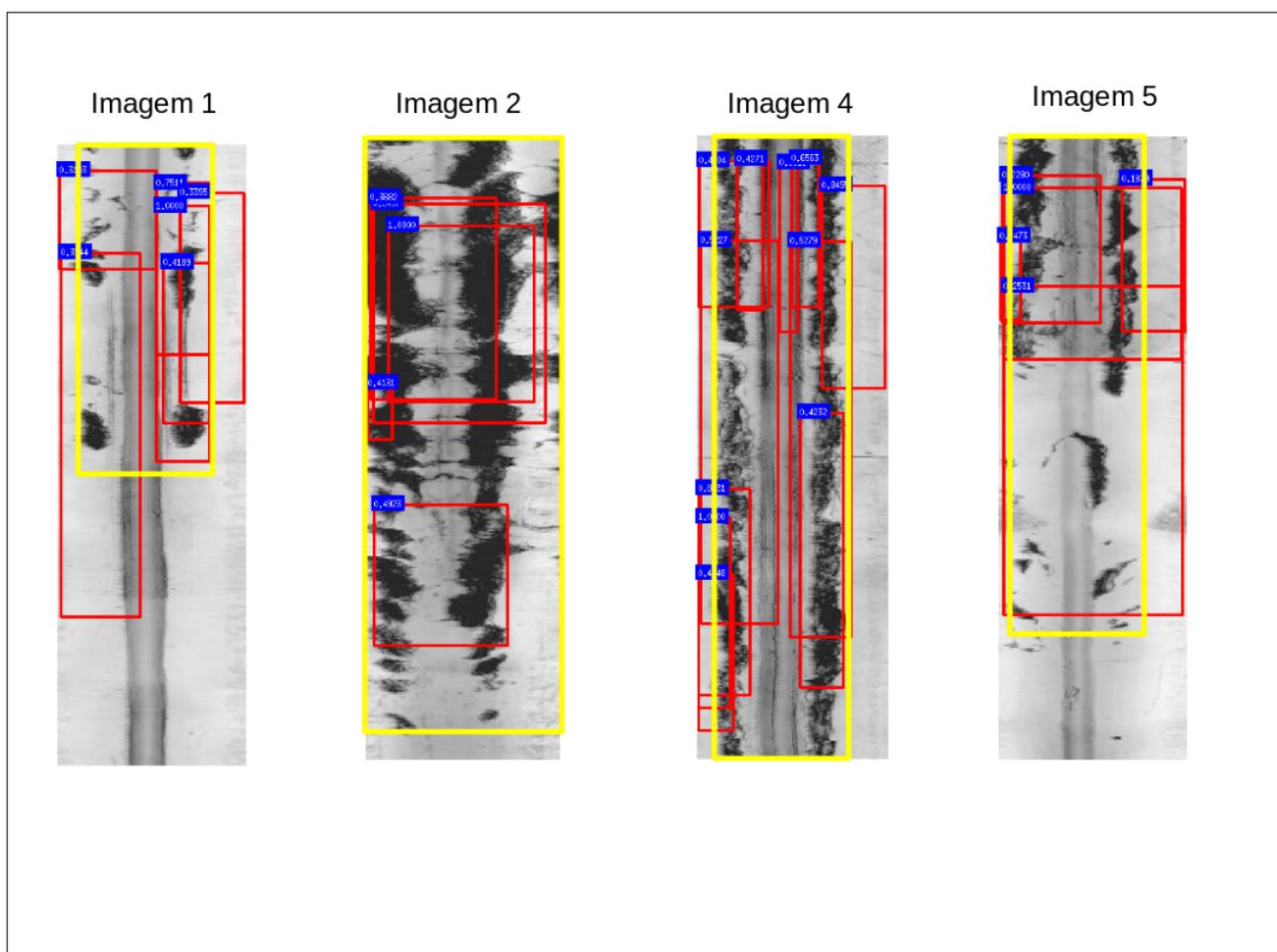


Figura 5.18 – Detecção de desmoronamento utilizando FRCNN em imagens reais, de trecho de dado obtido em [12]. Apresentamos resultados com notas normalizadas de detecção de desmoronamento, considerando como desmoronamento resultados com notas acima de 0.1, admitindo regiões sobrepostas em até 50%.

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

Neste trabalho abordamos a utilização de métodos baseados em Processamento Digital de Imagem (PDI) e Inteligência Artificial (IA) de detecção automática de fraturas e desmoronamentos em imagens acústicas de poço. Buscamos atender as necessidades de automação de detecções dessas ocorrências com a finalidade de contribuir com a objetividade, eficiência e rapidez nas atividades de geólogos e geofísicos responsáveis pela análise de dados de perfilagem de poço. A determinação dessas ocorrências tem um papel importante para a determinação do fluxo de fluidos de interesse através da caracterização de fraturas naturais, os desmoronamentos da parede do poço oferecem a direção de tensões horizontais das camadas rochosas. Vimos que o desenvolvimento de simulações dessas imagens acústicas possibilitou avaliar a qualidade dos classificadores. Contudo, também foram apresentados testes em alguns trechos de imagens reais.

Entre os métodos de PDI para detectar fraturas em imagens sintéticas, o que se destacou foi o Método de Classificação por Polinômio (MCP) baseado em ajuste polinomial, que apresentou uma curva ROC com área de 0,85. Entretanto, a técnica só encontrou ocorrências bem definidas e falhou em imagens onde a fratura possui pouco contraste com o fundo. Ocorreram muitos casos de não detecção ou detecção parcial. A vantagem desse método é o número reduzido de falsos positivos e a não necessidade de treinamento. Em imagens reais, como vimos nas imagens exemplos na figura 5.15 o método encontrou poucas fraturas, principalmente por conta das dificuldades, pois em muitos casos apresentaram características parciais de padrão senoidal e muitas estavam sobrepostas à desmoronamento e marcas de ferramenta.

Através do Método de Detecção de Desmoronamentos por Reconhecimento de Padrões (DRP), baseado em processamento digital de imagens, o classificador obteve uma área abaixo da curva ROC de 0,73, pois não conseguiu detectar com sucesso casos de desmoronamentos que apresentaram pouco contraste com o fundo da imagem. No entanto, apresenta uma vantagem que é não necessitar de treinamento. Em imagens reais, como

visto na figura 5.17, encontrou poucas ocorrências de desmoronamentos.

As redes neurais rasas com duas camadas foram utilizadas para detecção de fraturas e desmoronamentos. Diferente dos demais métodos, os dados de entrada utilizados foram os vetores com os atributos de Haralick obtidos através das imagens. Como vimos essa solução necessita treinamento. E por isso utilizamos imagens sintéticas do Dado 1 para treinar a rede. Com os modelos de redes treinadas para identificar cada ocorrência, os resultados com os novos dados de avaliação exibiram áreas abaixo da curva ROC com excelentes valores de 0,95 e 0,99 respectivamente para classificação de fraturas e desmoronamentos. Contudo, essa rede não realiza múltiplas detecções, não determina as coordenadas de cada uma delas, não consegue classificar múltiplas classes em uma única imagem em um único processo, além de só receber entradas de dimensões fixas. Essas desvantagens impossibilitam a utilização da RNA para análise em grandes trechos de imagens de poço ou até trecho de dimensões variáveis. Por isso, não foi aplicada no teste adicional e comparativo com dados reais.

Utilizamos e avaliamos a rede neural convolucional rápida baseada em regiões (FRCNN). Essa rede demonstrou maior capacidade de detecção entre os métodos MCP e DRP que apresentam além da nota de classificação, as coordenadas das ocorrências, apresentando índices de área abaixo da curva ROC de 0,86 e 0,97. Apesar da necessidade de treinamento, conseguiu identificar fraturas e desmoronamentos em novos dados com mínimo grau de contraste com o fundo da imagem que não foram identificadas pelos métodos MCP e DRP. Além disso, identificou mais ocorrências reais, em imagens apresentando artefatos concomitantes como *key seat*.

As avaliações empregadas demonstraram o potencial das redes convolucionais profundas. A possibilidade futura de treinamento com dados reais, deve permitir um salto de qualidade na detecção de cada ocorrência, identificando cada pixel pertencente a cada um dos objetos. Para isso, podemos realizar um processo de transferência de aprendizado da rede treinada com imagens sintética para treinar a rede com dados reais, inicializando assim a aprendizagem com filtros muitos mais próximos à extratores de características de fraturas e desmoronamentos. Dessa forma, será possível realizar um ajuste fino dos parâmetros da rede e determinar as coordenadas das regiões mais precisas de cada fratura e desmoronamento nas imagens acústicas reais. Outra possibilidade futura é avaliar esses novos recursos em imagens acústicas de tempo de trânsito e imagens resistivas.

Com o desenvolvimento de redes que permitem segmentação semântica [237, 238], talvez seja possível determinar as dimensões mais precisas de cada fratura e desmoronamento nas imagens acústicas de amplitudes, planejamos testar esse tipo de arquitetura em um futuro próximo. Outra possibilidade é avaliar esses novos recursos em imagens acústicas de tempo de trânsito e imagens resistivas.

REFERÊNCIAS

- [1] Peter Gerhard Tilke, Raj Banerjee, Vijaya Bhatta Halabe, Bulent Balci, Michael Thambynayagam, Jeffrey Bernard Spath, et al. Automated field development planning in the presence of subsurface uncertainty and operational risk tolerance. 2010.
- [2] RM Sok, CH Arns, MA Knackstedt, TJ Senden, AP Sheppard, Holger Averdunk, Wolf Val Pinczewski, Hiroshi Okabe, et al. Estimation of petrophysical parameters from 3d images of carbonate core. 2007.
- [3] JK Leggett. Drilling-induced structures in leg-66 cores. *Initial Reports of the Deep Sea Drilling Project*, 66(FEB):531–538, 1982.
- [4] Izabel Gioveli. Análise e simulação de padrões de fraturas geológicas. 2010.
- [5] Luciana Olivia Dias, Clécio R De Bom, Heitor Guimarães, Elisângela L Faria, P Márcio, P Marcelo, Maury D Correia, and Rodrigo Surmas. Segmentação textural de imagens de rocha por microtomografia segmentation of microtomography images of rocks using texture filter. *NOTAS TÉCNICAS*, 6(1), 2016.
- [6] Xining Li, Jinsong Shen, Zhongmin Zhu, Lei Wang, and Zhenling Li. Fracture extraction from fmi based on multiscale mathematical morphology. pages 703–707, 2016.
- [7] Geoff Dougherty. Digital image processing for medical applications. 2009.
- [8] John R Jensen and Kalmesh Lulla. Introductory digital image processing: a remote sensing perspective. 1987.
- [9] Gunther Rudzit and Otto Nogami. Geopolítica do petróleo.
- [10] CF Mora and AKH Kwan. Sphericity, shape factor, and convexity measurement of coarse aggregate for concrete using digital image processing. *Cement and concrete research*, 30(3):351–358, 2000.

-
- [11] A Ruela and C Guerra. Detecção de fraturas em imagem de perfis geofísicos de poço com morfologia matemática. *III Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, Recife, Brasil*, pages 21–30, 2010.
- [12] Jan Cornet. Fracture detection and analysis from image log raw data. 2013.
- [13] Fabiana Rodrigues Leta, Esteban Clua, Mauro Biondi, Toni Pacheco, and Maria do Socorro de Souza. An automatic process to identify features on boreholes data by image processing techniques. pages 249–262, 2013.
- [14] Rommel Anatoli Quintanilla Cruz, Diego Carriço Cacau, Renato Moraes dos Santos, Evandro Jose Ribeiro Pereira, Fabiana Rodrigues Leta, and Esteban Gonzalez Clua. Improving accuracy of automatic fracture detection in borehole images with deep learning and gpus. pages 345–350, 2017.
- [15] Mark D Zoback, Daniel Moos, Larry Mastin, and Roger N Anderson. Well bore breakouts and in situ stress. *Journal of Geophysical Research: Solid Earth*, 90(B7):5523–5530, 1985.
- [16] M Tingay, J Reinecker, and B Müller. Borehole breakout and drilling-induced fracture analysis from image logs. *World Stress Map Project*, pages 1–8, 2008.
- [17] J Reinecker, M Tingay, and B Müller. Borehole breakout analysis from four-arm caliper logs. *World stress map project*, pages 1–5, 2003.
- [18] Daniel Ask and MVS Ask. Detection of potential borehole breakouts in boreholes kfm01a and kfm01b. *SKB P-report in prep*, pages 1–5, 2006.
- [19] J Ringgaard. Mapping of borehole breakouts. processing of acoustical televiewer data from kfm01a, kfm01b, kfm02a, kfm03a, kfm03b, kfm04a, kfm05a, kfm06a and kfm07c. 2007.
- [20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.
- [22] Ross Girshick. Fast r-cnn. pages 1440–1448, 2015.
- [23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [24] Carlos Eduardo Guerra, Aldenize Ruela Xavier, and André José Neves Andrade. Watershed threshold and gray level morfology applied to object detecting in remote sensing and petrographics images. pages 146–151, 2011.

- [25] MJ Yaffe and JA Rowlands. X-ray detectors for digital radiography. *Physics in Medicine & Biology*, 42(1):1, 1997.
- [26] David C Schwartz. Image processing and analysis of individual nucleic acid molecules. February 24 1998. US Patent 5,720,928.
- [27] Clecio Roque De Bom, Martin Makler, and Marcelo Albuquerque. Método de reconhecimento de arcos gravitacionais com redes neurais utilizando a decomposição por mediatrizes. *NOTAS TÉCNICAS*, 2(1), 2012.
- [28] J Adam Donald, Tom R Bratton, et al. Advancements in acoustic techniques for evaluating open natural fractures. 2006.
- [29] Mohammed Al-Khabbaz, Hanan Abu-Hebail, Mahmood Akbar, Badruzzaman Khan, Salem Al-Sabea, Mona Rashaid, Robert Laronga, Alexander Aviantara, Sandeep Chakravorty, and Rohoullah Dashti. Complex artifacts in resistivity and acoustic image data: Recognition and use in borehole image interpretation. 2012.
- [30] XM Tang, Y Zheng, and D Patterson. Processing array acoustic-logging data to image near-borehole geologic structures. *Geophysics*, 2007.
- [31] Colleen A Barton et al. Discrimination of natural fractures from drilling-induced wellbore failures in wellbore image data—implications for reservoir permeability. 2000.
- [32] JC Lofts and LT Bourke. The recognition of artefacts from acoustic and resistivity borehole imaging devices. *Geological Society, London, Special Publications*, 159(1):59–76, 1999.
- [33] CA Barton, DA Castillo, D Moos, P Peska, and MD Zoback. Characterising the full stress tensor based on observations of drilling-induced wellbore failures in vertical and inclined boreholes leading to improved wellbore stability and permeability prediction. *The APPEA Journal*, 38(1):466–487, 1998.
- [34] Kenneth W Winkler. Azimuthal velocity variations caused by borehole stress concentrations. *Journal of Geophysical Research: Solid Earth*, 101(B4):8615–8621, 1996.
- [35] G Ooms, JM Burgerscentrum, BE Kampman-Reinhartz, et al. Influence of drillpipe rotation and eccentricity on pressure drop over borehole during drilling. 1999.
- [36] Brian E Hornby, John M Howie, and Donald W Ince. Anisotropy correction for deviated-well sonic logs: Application to seismic well tie. *Geophysics*, 68(2):464–471, 2003.

- [37] Zaki Bassiouni et al. Theory, measurement, and interpretation of well logs. 4, 1994.
- [38] Malcolm H Rider. The geological interpretation of well logs. 1986.
- [39] S Luthi. Geological well logs: Their use in reservoir modeling. 2001.
- [40] Xuanzhi Wu and Edo Nyland. Automated stratigraphic interpretation of well-log data. *Geophysics*, 52(12):1665–1676, 1987.
- [41] M Krief, J Garat, J Stellingwerff, J Ventre, et al. A petrophysical interpretation using the velocities of p and s waves (full-waveform sonic). *The Log Analyst*, 31(06), 1990.
- [42] David D Pollard and Raymond C Fletcher. Fundamentals of structural geology. 2005.
- [43] Santi Kumar Ghosh. Structural geology: fundamentals and modern developments. 2013.
- [44] Haakon Fossen. Structural geology. 2016.
- [45] Ki-Bok Min, Jonny Rutqvist, Chin-Fu Tsang, and Lanru Jing. Stress-dependent permeability of fractured rock masses: a numerical study. *International Journal of Rock Mechanics and Mining Sciences*, 41(7):1191–1210, 2004.
- [46] Peter E Gretener. Pore pressure: fundamentals, general ramifications and implications for structural geology (revised edition). 1979.
- [47] Robert L Folk. Petrology of sedimentary rocks. 1980.
- [48] Edwin Sherbon Hills. Elements of structural geology. 2012.
- [49] J. Schön. Propriedades físicas das rochas aplicadas à engenharia: Fundamentos teóricos e práticos. 2016.
- [50] JA Kessler, KK Bradbury, JP Evans, MA Pulsipher, DR Schmitt, JW Shervais, FE Rowe, and J Varriale. Geology and in situ stress of the mh-2 borehole, idaho, usa: Insights into western snake river plain structure from geothermal exploration drilling. *Lithosphere*, pages L609–1, 2017.
- [51] Colleen A Barton and Mark D Zoback. Stress perturbations associated with active faults penetrated by boreholes: Possible evidence for near-complete stress drop and a new technique for stress magnitude measurement. *Journal of Geophysical Research: Solid Earth*, 99(B5):9373–9390, 1994.
- [52] John Conrad Jaeger, Neville GW Cook, and Robert Zimmerman. Fundamentals of rock mechanics. 2009.

- [53] Martin Brudy and Halvor Kjørholt. Stress orientation on the norwegian continental shelf derived from borehole failures observed in high-resolution borehole imaging logs. *Tectonophysics*, 337(1):65–84, 2001.
- [54] HD Tjia and Mohd Idrus Ismail. Tectonic implications of well-bore breakouts in malaysian basins. *Geological Society of Malaysia Bulletin*, 36:175–186, 1994.
- [55] Mark D Zoback and John H Healy. In situ stress measurements to 3.5 km depth in the cajon pass scientific research borehole: implications for the mechanics of crustal faulting. *Journal of Geophysical Research: Solid Earth*, 97(B4):5039–5057, 1992.
- [56] Tan Chengxuan, Jin Zhijun, Zhang Mingli, Tang Liangjie, Jia Chengzao, Chen Shuping, Yang Meiling, and Zeng Lianbo. An approach to the present-day three-dimensional (3d) stress field and its application in hydrocarbon migration and accumulation in the zhangqiang depression, liaohe field, china. *Marine and petroleum geology*, 18(9):983–994, 2001.
- [57] B Haimson and A Klaetsch. Compaction bands and the formation of slot-shaped breakouts in st. peter sandstone. *Geological Society, London, Special Publications*, 284(1):89–105, 2007.
- [58] Bikash K Sinha, Jing Wang, Saad Kisra, Ji Li, Vivian Pistre, Tom Bratton, Michael Sanders, Cai Jun, et al. Estimation of formation stresses using borehole sonic data. 2008.
- [59] Martin T Hagan, Howard B Demuth, Mark H Beale, et al. Neural network design. 20, 1996.
- [60] John A Hertz. Introduction to the theory of neural computation. 2018.
- [61] A Cochocki and Rolf Unbehauen. Neural networks for optimization and signal processing. 1993.
- [62] Falah R Awwad, Mohamed Nekili, Venkatanarayana Ramachandran, and Mohamad Sawan. On modeling of parallel repeater-insertion methodologies for soc interconnects. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(1):322–335, 2008.
- [63] Camilo J Cela-Conde, Gisele Marty, Fernando Maestú, Tomás Ortiz, Enric Munar, Alberto Fernández, Miquel Roca, Jaume Rosselló, and Felipe Quesney. Activation of the prefrontal cortex in the human visual aesthetic perception. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):6321–6325, 2004.
- [64] Winfried Nöth. Imagem: cognição, semiótica, mídia. 1998.

-
- [65] Semir Zeki. The visual association cortex. *Current Opinion in Neurobiology*, 3(2):155–159, 1993.
- [66] Irving Biederman and Eric E Cooper. Priming contour-deleted images: Evidence for intermediate representations in visual object recognition. *Cognitive psychology*, 23(3):393–419, 1991.
- [67] DC Van Essen, DJ Felleman, EA DeYoe, Jaime Olavarria, and James Knierim. Modular and hierarchical organization of extrastriate visual cortex in the macaque monkey. 55:679–696, 1990.
- [68] Rudolf Arnheim. The two faces of gestalt psychology. *American Psychologist*, 41(7):820, 1986.
- [69] Robert J Schalkoff. Artificial neural networks. 1, 1997.
- [70] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.
- [71] Nirmal K Bose and P Liang. Neural network fundamentals with graphs, algorithms and applications, mcgraw-hill series in electrical and computer engineering. 1996.
- [72] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. pages 231–238, 1995.
- [73] Warren S Sarle. Neural networks and statistical models. 1994.
- [74] W Thomas Miller, Paul J Werbos, and Richard S Sutton. Neural networks for control. 1995.
- [75] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.
- [76] Adil Moujahid. A practical introduction to deep learning with caffe and python. Retrieved February, 19:2018, 2016.
- [77] Frederic Gruau et al. Neural network synthesis using cellular encoding and the genetic algorithm. 1994.
- [78] Robert R Trippi and Efraim Turban. Neural networks in finance and investing: Using artificial intelligence to improve real world performance. 1992.
- [79] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

-
- [80] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [81] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.
- [82] Zsolt László Kovács. *Redes neurais artificiais*. 2002.
- [83] Hegler C Tissot, Luiz C Camargo, and AT Pozo. Treinamento de redes neurais feedforward: comparativo dos algoritmos backpropagation e differential evolution. 2012.
- [84] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Neural networks and learning machines*. 3, 2009.
- [85] Bekir Karlik and A Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122, 2011.
- [86] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [87] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.
- [88] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.
- [89] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. 2014.
- [90] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. pages 1139–1147, 2013.
- [91] Martin Fodsslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.
- [92] Bernd Fritzke. Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural networks*, 7(9):1441–1460, 1994.
- [93] MA Andrade, P Chacon, JJ Merelo, and F Moran. Evaluation of secondary structure of proteins from uv circular dichroism spectra using an unsupervised learning neural network. *Protein Engineering, Design and Selection*, 6(4):383–390, 1993.

-
- [94] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 1(1), 1998.
- [95] Robert Hecht-Nielsen. Theory of the backpropagation neural network. pages 65–93, 1992.
- [96] Martin Kilduff, Craig Crossland, Wenpin Tsai, and David Krackhardt. Organizational network perceptions versus reality: A small world after all? *Organizational Behavior and Human Decision Processes*, 107(1):15–28, 2008.
- [97] Peter Xiaoping Liu, MQ-H Meng, Simon X Yang, Chao Hu, and Jie Sheng. Internet-based remote control by using adaline neural networks. pages 252–257, 2003.
- [98] John J Hopfield and David W Tank. “neural” computation of decisions in optimization problems. *Biological cybernetics*, 52(3):141–152, 1985.
- [99] IM Stephanakis and Stefanos Kollias. A single layer linear feedforward neural network for signal estimation in the wavelet domain. 2:487–490, 1997.
- [100] Sheng Chen, Bernard Mulgrew, and Peter M Grant. A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on neural networks*, 4(4):570–590, 1993.
- [101] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural networks*, 1(1):17–61, 1988.
- [102] Sudesh M Duggal and Paul R Popovich. Practical applications of neural networks in business. *Journal of Computer Information Systems*, 33(2):8–13, 1993.
- [103] Frank Rosenblatt. The perceptron, a perceiving and recognizing automaton project para. 1957.
- [104] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [105] Howard Demuth and Mark Beale. Neural network toolbox for use with matlab–user’s guide verion 3.0. 1993.
- [106] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806, 1993.
- [107] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [108] Bo-Suk Yang, Tao Han, and J LART An. Art–kohonen neural network for fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 18(3):645–657, 2004.

- [109] PA Aguilera, A Garrido Frenich, JA Torres, H Castro, JL Martinez Vidal, and M Canton. Application of the kohonen neural network in coastal water management: methodological development for the assessment and prediction of water quality. *Water research*, 35(17):4053–4062, 2001.
- [110] Jean-jacques Quisquater and David Samyde. Kohonen neural network. 2002.
- [111] Robert Hecht-Nielsen. Neurocomputer applications. pages 445–453, 1989.
- [112] Richard P Lippmann. An introduction to computing with neural nets. pages 36–54, 1988.
- [113] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. pages 693–700, 2010.
- [114] Alan F Murray. Pulse arithmetic in vlsi neural networks. *IEEE Micro*, 9(6):64–74, 1989.
- [115] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *Journal of machine learning research*, 10(Jan):1–40, 2009.
- [116] Zhe Li, Boqing Gong, and Tianbao Yang. Improved dropout for shallow and deep learning. pages 2523–2531, 2016.
- [117] Siyuan Ma and Mikhail Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. pages 3778–3787, 2017.
- [118] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. 1, 2016.
- [119] Hugh M Cartwright. Applications of artificial intelligence chem oxcp 11. 1997.
- [120] James Bezdek. Intelligence: Computational versus artificial. *IEEE Transactions on Neural Networks*, 4(5):737, 1993.
- [121] Benjamin W Wah and Guo-Jie Li. Computers for artificial intelligence applications. 1986.
- [122] John Henry Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. 1992.
- [123] H Van Dyke Parunak. Applications of distributed artificial intelligence in industry. *Foundations of distributed artificial intelligence*, 2, 1996.
- [124] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

-
- [125] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.
- [126] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al. Recent advances in deep learning for speech research at microsoft. pages 8604–8608, 2013.
- [127] Van B Weigel. Deep learning for a digital age: Technology’s untapped potential to enrich higher education. 2002.
- [128] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. pages 513–520, 2011.
- [129] Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. On optimization methods for deep learning. pages 265–272, 2011.
- [130] Yoshua Bengio, Ian J Goodfellow, and Aaron Courville. Deep learning. *Nature*, 521:436–444, 2015.
- [131] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. pages 184–199, 2014.
- [132] Flávio HD Araújo, Allan C Carneiro, and Romuere RV Silva. Redes neurais convolucionais com tensorflow: Teoria e prática.
- [133] Andrej Karpathy. Convolutional neural networks for visual recognition. *Dostupné na*, 2016.
- [134] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. pages 689–692, 2015.
- [135] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [136] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.
- [137] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. pages 675–678, 2014.
- [138] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. pages 806–813, 2014.

-
- [139] Tamas Roska and Leon O Chua. The cnn universal machine: an analogic array computer. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 40(3):163–173, 1993.
- [140] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Noguees, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.
- [141] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. 1(3):6, 2015.
- [142] Viet Tra, Jaeyoung Kim, Sheraz Ali Khan, and Jong-Myon Kim. Bearing fault diagnosis under variable speed using convolutional neural networks and the stochastic diagonal levenberg-marquardt algorithm. *Sensors*, 17(12):2834, 2017.
- [143] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. pages 3676–3684, 2015.
- [144] Rafael Henrique Santos Rocha. Reconhecimento de objetos por redes neurais convolutivas. 2015.
- [145] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. pages 1725–1732, 2014.
- [146] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. pages 1717–1724, 2014.
- [147] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. pages 4297–4304, 2015.
- [148] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Peppere, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.
- [149] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

-
- [150] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2016.
- [151] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Micha Andriluka, Chris Bregler, Bernt Schiele, and Christian Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. pages 3810–3818, 2015.
- [152] Philippe P Ohanian and Richard C Dubes. Performance evaluation for four classes of textural features. *Pattern recognition*, 25(8):819–833, 1992.
- [153] Xiu Li, Min Shang, Hongwei Qin, and Liansheng Chen. Fast accurate fish detection and recognition of underwater images with fast r-cnn. pages 1–5, 2015.
- [154] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. pages 4073–4082, 2015.
- [155] Endre László, Péter Szolgay, and Zoltán Nagy. Analysis of a gpu based cnn implementation. pages 1–5, 2012.
- [156] Heehoon Kim, Hyoungwook Nam, Wookeun Jung, and Jaejin Lee. Performance analysis of cnn frameworks for gpus. *Performance Analysis of Systems and Software (ISPASS)*, 2017.
- [157] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.
- [158] J-M Geusebroek, Arnold WM Smeulders, and Joost Van De Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.
- [159] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. 59(2):167–181, 2004.
- [160] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun. Object detection networks on convolutional feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1476–1481, 2017.
- [161] Sandra Nobre. A decomposição em valores singulares e suas aplicações. 2007.
- [162] Juliano Vieira de Oliveira et al. Estudo da decomposição em valores singulares e análise dos componentes principais. 2017.
- [163] Walisson Pereira Lorigiola et al. Compressão de imagens digitais via dvs. 2004.

- [164] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
- [165] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. pages 399–408, 2001.
- [166] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.
- [167] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [168] Lucas Ferrari de Oliveira, Alysson Luís Martins Narloch, Deives Mesquita Kist, Mário de Paula Soares Filho, Géri Eduardo Meneghello, Gerson Geraldo H Cavaleiro, and Maria Angela Andre Tillmann. Extração de características de forma utilizando matriz de co-ocorrência e atributos de haralick. *Curitiba, Paraná*, 2012.
- [169] Manuel Blanco Valentin, De Bom, Clecio Roque, Marcio Portes de Albuquerque, Marcelo Portes de Albuquerque, Elisangela Faria, Maury Duarte Correia, and Rodrigo Surmas. On a method for rock classification using textural features and genetic optimization. *arXiv preprint arXiv:1607.01679*, 2016.
- [170] Manuel Blanco Valentin, Clécio Roque de Bom, Márcio P de Albuquerque, Marcelo P de Albuquerque, Elisângela L Faria, and Maury D Correia. Texture classification based on spectral analysis and haralick features classificação de texturas mediante análise espectral e parâmetros de haralick. *NOTAS TÉCNICAS*, 6(1), 2016.
- [171] Chris Solomon and Toby Breckon. Fundamentals of digital image processing: A practical approach with examples in matlab. 2011.
- [172] Maurice Charbit. Digital signal and image processing using matlab. 666, 2010.
- [173] Gabriel Matos Araujo. Algoritmo para reconhecimento de características faciais baseado em filtros de correlação. *Universidade Federal do Rio de Janeiro*, 2010.
- [174] Waldir Sabino da Silva Júnior. Reconhecimento de padrões utilizando filtros de correlação com análise de componentes principais. 2010.
- [175] Alexandre P Mendonça and EAB SILVA. Filtragem discriminativa para detecção de múltiplos padrões de imagem. *Anais do XX Simpósio Brasileiro de Telecomunicações, Rio de Janeiro, RJ, Brasil*, 2003.

- [176] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. pages 856–863, 2003.
- [177] Abhijit Mahalanobis, BVK Vijaya Kumar, and David Casasent. Minimum average correlation energy filters. *Applied Optics*, 26(17):3633–3640, 1987.
- [178] André Persechino and Márcio Portes de Albuquerque. Processamento digital de imagens: conceitos fundamentais digital image processing: fundamental concepts. *MONOGRAFIAS-CBPF*, 1(4), 2015.
- [179] Gregory A Baxes. Digital image processing: principles and applications. 1994.
- [180] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. Digital image processing using matlab®. 2010.
- [181] Luciana Olivia Dias, Clécio R De Bom, Heitor Guimarães, Elisângela L Faria, Márcio P de Albuquerque, Marcelo P de Albuquerque, Maury D Correia, and Rodrigo Surmas. Segmentação textural de imagens de rocha por microtomografia segmentation of microtomography images of rocks using texture filter. *NOTAS TÉCNICAS*, 6(1), 2016.
- [182] Lucas Máximo Alves. Modelagem e simulação do campo contínuo com irregularidades: Aplicações em mecânica da fratura com rugosidade. 2011.
- [183] Waleed Al-Sit. Automatic feature detection and interpretation in borehole data. 2015.
- [184] Márcio P. Albuquerque Marcelo P. Albuquerque Elisângela L. Faria e Maury D. Correia Luciana Olivia Dias, Clécio R. Bom. Comparação de métodos de segmentação de fraturas em imagem acústica de perfilagem petrofísica. Aceita em 02-08-18:—, 2018.
- [185] Demin Wang. A multiscale gradient algorithm for image segmentation using watersheds. *Pattern recognition*, 30(12):2043–2052, 1997.
- [186] Paul R Hill, Cedric Nishan Canagarajah, and David R Bull. Texture gradient based watershed segmentation. 4:IV–3381, 2002.
- [187] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. pages 61–65, 2006.
- [188] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

- [189] Luciana Dias, Clécio Bom, Bruno Alves, Márcio Albuquerque, Marcelo Albuquerque, Elisângela Faria, and Maury Correia. Estimativa de permeabilidade absoluta com processamento de imagens utilizando distribuição de tamanho de grãos. *NOTAS TÉCNICAS*, 7(3), 2017.
- [190] Siddheswar Ray and Rose H Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. pages 137–143, 1999.
- [191] Anil Z Chitade and SK Katiyar. Colour based image segmentation using k-means clustering. *International Journal of Engineering Science and Technology*, 2(10):5319–5325, 2010.
- [192] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [193] Tapas Kanungo, Byron Dom Wayne Niblack and David Steele. A fast algorithm for mdl-based multi-band image segmentation. pages 609–616, 1994.
- [194] Carlton Wayne Niblack, Ron Barber, Will Equitz, Myron D Flickner, Eduardo H Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, and Gabriel Taubin. Qbic project: querying images by content, using color, texture, and shape. 1908:173–188, 1993.
- [195] Basilios Gatos, Ioannis Pratikakis, and Stavros J Perantonis. Adaptive degraded document image binarization. *Pattern recognition*, 39(3):317–327, 2006.
- [196] Øivind Due Trier and Torfinn Taxt. Improvement of “integrated function algorithm” for binarization of document images. *Pattern Recognition Letters*, 16(3):277–283, 1995.
- [197] T Romen Singh, Sudipta Roy, O Imocha Singh, Tejmani Sinam, Kh Singh, et al. A new local adaptive thresholding technique in binarization. *arXiv preprint arXiv:1201.5227*, 2012.
- [198] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [199] Chengxin Yan, Nong Sang, and Tianxu Zhang. Local entropy-based transition region extraction and thresholding. *Pattern Recognition Letters*, 24(16):2935–2941, 2003.
- [200] Carsten Croonenbroeck. Local entropy based image reconstruction. 2012.
- [201] M Portes de Albuquerque, IA Esquef, and AR Gesualdi Mello. Image thresholding using tsallis entropy. *Pattern Recognition Letters*, 25(9):1059–1065, 2004.

- [202] Maria Inês Meurer, Eduardo Meurer, Jorge Vicente Lopes da Silva, Ailton Santa Bárbara, Luiz Felipe Nobre, Marília Gerhardt de Oliveira, and Daniela Nascimento Silva. Aquisição e manipulação de imagens por tomografia computadorizada da região maxilofacial visando à obtenção de protótipos biomédicos. *Radiologia Brasileira*, 41(1):49–54, 2008.
- [203] Gerald Jean Francis Banon and Junior Barrera. Bases da morfologia matemática para a análise de imagens binárias. 1994.
- [204] Joelma Carla Santos. Extração de atributos de forma e seleção de atributos usando algoritmos genéticos para classificação de regiões. *Instituto Nacional de Pesquisas Espaciais (INPE)*, 2007.
- [205] Leonardo Sant’ana Bins, Guaraci José Erthal, and LEILA MARIA GARCIA FoNSECA. Um método de classificação não supervisionada por regiões. *Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, 5:65–68, 1993.
- [206] Alzir Felipe Buffara Antunes. Classificação de ambiente ciliar baseada em orientação a objeto em imagens de alta resolução espacial. 2003.
- [207] Heliana Bezerra Soares. Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte. 2008.
- [208] Joseph Kalil Khoury, Francisco de Assis de Carvalho Pinto, Nerilson Terra dos Santos Santos, Ricardo Marius Della Lúcia, and Eduardo Eiji Maeda. Análise discriminante paramétrica para reconhecimento de defeitos em tábuas de eucalipto utilizando imagens digitais. *Revista Árvore*, 29(2), 2005.
- [209] Peter Toft. The radon transform. theory and implementation. 1996.
- [210] Frantisek Matus and Jan Flusser. Image representation via a finite radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):996–1006, 1993.
- [211] Jean-Luc Starck, Emmanuel J Candès, and David L Donoho. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684, 2002.
- [212] Jin S Seo, Jaap Haitisma, Ton Kalker, and Chang D Yoo. A robust image fingerprinting system using the radon transform. *Signal Processing: Image Communication*, 19(4):325–339, 2004.
- [213] Zhi-Pei Liang and Paul C Lauterbur. Principles of magnetic resonance imaging: a signal processing perspective. 2000.
- [214] Stanley R Deans. The radon transform and some of its applications. 2007.

- [215] Paul VC Hough. Method and means for recognizing complex patterns. December 18 1962. US Patent 3,069,654.
- [216] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [217] Clecio Roque De Bom, Marcelo Portes de Albuquerque, and Martin Makler. Uso da transformada de hough para detecção de arcos. *NOTAS TÉCNICAS*, 2(1), 2012.
- [218] CR Bom, P Marcelo, and M Makler. Notas técnicas cbpf-nt-001/12 janeiro 2012. 2012.
- [219] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. pages 714–725, 1987.
- [220] Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method: randomized hough transform (rht). *Pattern recognition letters*, 11(5):331–338, 1990.
- [221] Stanley R Deans. Hough transform from the radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):185–188, 1981.
- [222] Varsha Kamat and Subramaniam Ganesan. An efficient implementation of the hough transform for detecting vehicle license plates using dsp's. pages 58–59, 1995.
- [223] Matlab optimization toolbox. 2014. The MathWorks, Natick, MA, USA.
- [224] Icília de Lima Feitosa. Litofácies, ambientes deposicionais e ciclicidade do grupo paranoá: Exemplo da região da serra de são domingos, nos municípios de buritis e formoso, mg. 2012.
- [225] Júlio César Carvalho Coelho, Marcelo Augusto Martins-Neto, and Marcelo S Marinho. Estilos estruturais e evolução tectônica da porção mineira da bacia proterozóica do são francisco. *Revista Brasileira de Geociências*, 38(2 suppl):149–165, 2008.
- [226] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. *Proceedings of the 28 annual conference on Computer graphics and interactive techniques*, -(-):341–346, 2001.
- [227] Jerome Fan, Suneel Upadhye, and Andrew Worster. Understanding receiver operating characteristic (roc) curves. *Canadian Journal of Emergency Medicine*, 8(1):19–20, 2006.
- [228] Giedrius Vanagas. Receiver operating characteristic curves and comparison of cardiac surgery risk stratification systems. *Interactive cardiovascular and thoracic surgery*, 3(2):319–322, 2004.

- [229] Yuichi Matsuki, Katsumi Nakamura, Hideyuki Watanabe, Takatoshi Aoki, Hajime Nakata, Shigehiko Katsuragawa, and Kunio Doi. Usefulness of an artificial neural network for differentiating benign from malignant pulmonary nodules on high-resolution ct: evaluation with receiver operating characteristic analysis. *American Journal of Roentgenology*, 178(3):657–663, 2002.
- [230] Yuzheng Wu, Maryellen L Giger, Kunio Doi, Carl J Vyborny, Robert A Schmidt, and Charles E Metz. Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer. *Radiology*, 187(1):81–87, 1993.
- [231] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [232] John W Harbaugh and Graeme Bonham-Carter. Computer simulation in geology. 1970.
- [233] Berthold KP Horn and Brett L Bachman. Using synthetic images to register real images with surface models. *Communications of the ACM*, 21(11):914–924, 1978.
- [234] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. pages 3234–3243, 2016.
- [235] Holly Rushmeier, Greg Ward, Christine Piatko, Phil Sanders, and Bert Rust. Comparing real and synthetic images: Some ideas about metrics. pages 82–91, 1995.
- [236] Takeo Kanade, Kazuo Oda, Masaya Tanaka, Hiroshi Kano, and Atsushi Yoshida. Method and apparatus for merging real and synthetic images. November 21 2000. US Patent 6,151,009.
- [237] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 2015.
- [238] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1605.06211*, 2016.

APÊNDICES

APÊNDICE A

DESCRITORES HARALICK

Os 13 atributos originais do Haralick, onde N_g é o número de tons do espectro de cor, podem ser calculado utilizando as equações:

$$\text{Energia (Segundo Momento)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)^2, \quad (\text{A.1})$$

$$\text{Contraste} = \sum_{n=0}^{N_g-1} n^2 \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right)_{|i-j|=n}, \quad (\text{A.2})$$

$$\text{Correlação} = \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (ij)p(i, j) - \mu_x \mu_y \right) \left(\frac{1}{\sigma_x \sigma_y} \right), \quad (\text{A.3})$$

sendo $p(i, j)$, representados pela função de probabilidade. Considerando as respectivas médias μ_x , μ_y e desvios padrão σ_x e σ_y das projeções $p(i, j)$. Deduzindo respectivamente,

$$p_x(i) = \sum_{j=1}^{N_g} p(i, j), \quad (\text{A.4})$$

$$p_y(j) = \sum_{i=1}^{N_g} p(i, j), \quad (\text{A.5})$$

$$\text{Quadrado das Somas} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu)^2 p(i, j), \quad (\text{A.6})$$

$$\text{Homogeneidade Local} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} p(i, j), \quad (\text{A.7})$$

$$\text{Soma das Médias} = \sum_{i=2}^{2N_g} [i \cdot p_{x+y}(i)], \quad (\text{A.8})$$

sendo,

$$p_{x+y}(k) = \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right)_{i+j=k}, \quad (\text{A.9})$$

$$\text{Soma das Entropias} = \text{SE} = \sum_{i=2}^{2N_g} p_{x+y}(i) \cdot \log p_{x+y}(i), \quad (\text{A.10})$$

$$\text{Soma das Variâncias} = \sum_{i=2}^{2N_g} (i - (SE))^2 p_{x+y}(i), \quad (\text{A.11})$$

$$\text{Diferenças das Variâncias} = \text{Var}(p_{x-y}), \quad (\text{A.12})$$

onde,

$$p_{x-y}(k) = \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right)_{|i-j|=k}, \quad (\text{A.13})$$

$$\text{Diferenças das Entropias} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \cdot \log [p_{x-y}(i)], \quad (\text{A.14})$$

$$\text{Informação de Correlação 1} = \left(\frac{HXY - HXY1}{\max(HX, HY)} \right), \quad (\text{A.15})$$

onde,

$$HXY = - \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \cdot \log p(i, j) \right), \quad (\text{A.16})$$

$$HXY1 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \cdot \log p_x(i) \cdot p_y(j), \quad (\text{A.17})$$

$$HX = \sum_{i=1}^{N_g} p_x(i) \cdot \log p_x(i) \quad (\text{A.18})$$

,

$$HY = \sum_{j=1}^{N_g} p_y(j) \cdot \log p_y(j), \quad (\text{A.19})$$

$$\text{Informação de Correlação 2} = \sqrt{1 - e^{-2 \cdot (HXY2 - HXY)}} \quad (\text{A.20})$$

onde,

$$HXY2 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [p_x(i) \cdot p_y(j)] \cdot \log p_x(i) \cdot p_y(j), \quad (\text{A.21})$$

Esses 3 atributos baseados em Haralick, foram propostos por [?] e foram considerado neste trabalho pode ser calculado pelas equações,

$$\text{Probabilidade Máxima} = \max p(i, j), \quad (\text{A.22})$$

$$\text{Agrupamento Sombreado} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - M_x + j - M_y)^3 \cdot p(i, j), \quad (\text{A.23})$$

sendo,

$$M_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \cdot p(i, j), \quad (\text{A.24})$$

$$M_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} j \cdot p(i, j), \quad (\text{A.25})$$

$$\text{Agrupamento Proeminente} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - M_x + j - M_y)^4 \cdot p(i, j), \quad (\text{A.26})$$

APÊNDICE B

INFORMAÇÕES GERAIS ANP POÇO

	<h3 style="margin: 0;">PERFIL COMPOSTO 1-IMET-8-MG</h3> <h4 style="margin: 0; color: red;">1CONF1MG</h4>				
DADOS GERAIS					
<p>Nome do Poço: 1-IMET-8-MG</p> <p>Operador: IMETAME ENERGIA S.A</p> <p>País: BRASIL</p> <p>Código ANP: 58275026884</p> <p>Bloco: SF-T-114</p> <p>Bacia: SÃO FRANCISCO</p> <p>Classificação: PIONEIRO</p> <p>Sonda: IME-01 (IMETAME ENERGIA S.A)</p> <p>Início Perfuração: 28/02/2013</p> <p>Término Perfuração: 12/04/2013</p> <p>Conclusão do Poço: 21/08/2013</p> <p>Situação Atual: ABANDONADO TEMPORARIAMENTE</p>	<p>Datum Geográfico: SAD69 MC: -45° Zona: 23S</p> <p>Coord. Geográficas Base</p> <p>Latitude: -17:25:22,969</p> <p>Longitude: -44:38:27,488</p> <p>Coord. Geográficas Fundo</p> <p>Latitude: -17:25:22,969</p> <p>Longitude: -44:38:27,488</p> <p>Datum Profundidade:</p> <p>Elevação / BAP: 537.6 m</p> <p>Fundo do Mar:</p> <p>Profundidade Final:</p> <p>Formação: GR. PARANOÁ</p> <p>Geólogo (s): CLAUDIANO MIRANDA/VICENTE PIMENTEL</p>	<p>Coord. UTM Base</p> <p>N: 8.073.602,000 m</p> <p>E: 538.130,000 m</p> <p>Coord UTM Fundo</p> <p>N: 8.073.602,000 m</p> <p>E: 538.130,000 m</p> <p>MR: 543 m</p> <p>Sondador: 1781,00 (- 1238,00) m</p> <p>Perfil: 1781,00 (- 1238,00) m</p>			
Observações:					
CONFIGURAÇÃO DO POÇO					
DADOS DE POÇO E REVESTIMENTOS					
Diam. Broca (pol)	Prof. Sond. (m)	Diam. Rev. (pol)	Prof. Sapata (m)	Observações	
17 1/2"	105.0 -438.0	13 3/8"	95.0 -448.0	REVESTIMENTO DE SUPERFICIE	
12 1/4"	498.0 -45.0	9 5/8"	480.0 -63.0	REVESTIMENTO INTERMEDIARIO	
8 1/2"	1781.0 -1238.0	5 1/2"	1781.0 -1238.0	REVESTIMENTO FINAL	
DADOS DE PERFILAGEM					
Corr.	Diam. Poço (pol)	Perfis Corridos e Intervalos (m)	Data	Max Temp °F	Observações
1	12 1/4"	DSN(93.1-450.4); GR(14.7-446.3); HRI(95.0-474.0);	12/03/2013	130°F	PERFILAGEM INTERMEDIÁRIA CONVENCIONAL
1	12 1/4"	SDL (93.1-466.0)	12/03/2013	130°F	PERFILAGEM INTERMEDIÁRIA CONVENCIONAL
1	12 1/4"	WSTT(95.0-465.0); XRM(11.9-478.0); RSCT(8.8-181.6)	12/03/2013	130°F	PERFILAGEM INTERMEDIÁRIA ESPECIAL
2	8 1/2"	HRLA (479.3-1774.6); LDT(479.3-1770.5); HGNS(479.3-1772.3); GR(385.0-1774.6)	26/03/2013	130°F	PERFILAGEM FINAL CONVENCIONAL
2	8 1/2"	DSI (479.3-1467.0); FMI(1482.0-1780.0)	26/03/2013	135°F	PERFILAGEM FINAL ESPECIAL
2	8 1/2"	CMR(479.3-1475.1); ECS(479.3-1779.2); HNGS(479.3-1772.3)	27/03/2013	135°F	PERFILAGEM FINAL ESPECIAL
2	8 1/2"	VSI(150.0-1478.0); MSCT(1477.0-1776.0);	29/03/2013	135°F	PERFILAGEM FINAL ESPECIAL
2	8 1/2"	MSIP(1430.0-1765.5)	13/04/2013	135°F	PERFILAGEM FINAL ESPECIAL