

PONTÍFICA UNIVERSIDADE CATÓLICA DO PARANÁ

BRUNO MIGUEL NOGUEIRA DE SOUZA

DETECÇÃO E LOCALIZAÇÃO DE FOGO EM IMAGENS DIGITAIS USANDO
TÉCNICAS DE APRENDIZADO DE MÁQUINA

CURITIBA PR
2019

BRUNO MIGUEL NOGUEIRA DE SOUZA

DETECÇÃO E LOCALIZAÇÃO DE FOGO EM IMAGENS DIGITAIS USANDO
TÉCNICAS DE APRENDIZADO DE MÁQUINA

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

Orientador: Alceu S. Britto Jr.

Co-orientador: David Menotti.

CURITIBA PR
2019

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

S729d
2019 Souza, Bruno Miguel Nogueira de Souza
Detecção e localização de fogo em imagens digitais usando técnicas de
aprendizado de máquina / Bruno Miguel Nogueira de Souza ; orientador:
Alceu S. Britto Jr. ; co-orientador: David Menotti. – 2019.
xvi, 103 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba,
2019
Bibliografia: f. 98-109

1. Informática. 2. Aprendizado do computador. 3. Prevenção de incêndio.
4. Redes neurais. I. Britto Junior, Alceu S. II. Menotti, David. III. Pontifícia
Universidade Católica do Paraná. Programa de Pós-Graduação em
Informática. IV. Título.

CDD 22. ed. – 004

DECLARAÇÃO

Declaro para os devidos fins que o aluno **BRUNO MIGUEL NOGUEIRA DE SOUZA**, defendeu sua tese de doutorado intitulada “**DETECÇÃO E LOCALIZAÇÃO DE FOGO EM IMAGENS DIGITAIS USANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA**”, na área de concentração Ciência da Computação, no dia 23 de abril de 2019, no qual foi aprovado.

Declaro ainda que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade, firmo a presente declaração.

Curitiba, 23 de maio de 2019.



Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná



Para Luke e Roberta

Agradecimentos

Gostaria de agradecer a Deus e a todos meus familiares que me acompanharam nesta caminhada. A minha esposa Roberta Ekuni que tanto me apoiou e teve muita paciência durante todo o processo. Ao meu filho Luke que nasceu recentemente e tem iluminado nossos dias, trazendo muita alegria para nossa casa. Agradeço a "Sueli", minha Sogra, por todo apoio e paciência neste período. A Yuki que ficava me esperando na porta sempre que eu ia pra Curitiba. A Mel, por ser uma das cachorras mais meigas que conheço e sempre tentava "segurar" a Yuki quando esta queria morder alguém. Agradeço aos meus pais (Heidi, Marcus e Lê) por todo apoio e incentivo dado para que eu pudesse concluir mais esta etapa. Agradeço minha irmã Camila e ao cunhado Brayan, por sempre nos receberem tão bem em São Paulo.

Agradeço à CAPES pelo apoio financeiro, pois o presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Agradeço aos professores que me orientaram durante todo o processo. Ao Prof. Jacques Facon, que me aceitou no PPGIa e me orientou, permitindo meu ingresso no programa de doutorado. Ao Prof. Alceu Britto Jr. pela orientação, paciência e ensinamentos. Ao Prof. Zhaozheng Yin da MS&T, por ter permitido minha participação no programa de doutorado sanduíche. Agradeço também ao co-orientador Prof. David Menotti pelas orientações e colaborações com a pesquisa.

Agradeço a equipe de suporte (Jhonatan) e a secretaria do PPGIa (Cheila e Flavia) pelo pronto atendimento sempre quando necessário. Agradeço à Andreia Malucelli, ex-coordenadora do programa, pelo apoio para que fosse possível apresentar um de meus trabalhos em um congresso internacional. Agradeço ao Prof. Emerson e todos os docentes do PPGIa pelo apoio e disponibilidade.

Agradeço os amigos do PPGIa: Cleverton, Eduardo, Vilmar, Alan, Fiuki (Eduardo), Jhonatan e Rafael pela parceria e pelas comilanças nos restaurantes de Curitiba. Aos amigos de laboratório: Flávio, André, Patricia, Fabiano, Marcia, Marcelo e Rodrigo.

Agradeço aos amigos da UENP que possibilitaram o afastamento de minhas atividades durante o doutorado (Biluka (Carlos Eduardo), Ricardo, André, Tiago, Christian, Glauco, Sgarbi, Dani, Della Mura, Merlin, Fabio e Fernando).

Agradeço ao Danilo (Gato Mestre), Kairo, Rafael (Ping) e Lucas por terem me acolhido tão bem na República Kaluai.

Agradeço a Nvidia por disponibilizar uma placa que viabilizou meus experimentos com Redes Neurais Convolucionais.

Abstract

Fire incidents present a constant threat to ecological systems, infrastructure and human lives. Using digital image processing is a viable and low cost alternative as an aid in early fire detection. In order to solve the problem of fire detection in images, the present work is developed based on two main steps: (1) fire and smoke detection and (2) segmentation of fire-like pixels. For that, we created two datasets, one for fire and smoke detection and another dataset with labeled pixels for segmentation. Regarding the detection process, the VGG16, Inception, Resnet50, Xception and MobileNet architectures were modified and evaluated. In addition, by performing architecture variations we propose a Convolution Neural Network (CNN) called Fire and Smoke Detection Network (FSDN). To benchmark this new architecture, we compared it with some architectures in the literature. The results showed that using binary neural networks to detect fire is more efficient than multiclass networks, since the accuracy values for binary networks ranged from 89% to 99% for fire and from 82% to 96% for smoke. On the other hand, in the multiclass networks (fire, smoke, negative) the accuracy range was from 79% to 95%. The proposed *FSDN* architecture obtained 95% and 90% of fire and smoke detection rate, respectively, and was executed 2.9 times faster than the best performing architecture for detection. For fire localization process, two new fire pixel mapping methods were proposed. The first method used probabilistic approaches by mixing color channels of different color spaces using an entropy rule. The second method was created based on probabilities with colorness theories for: Red, Yellow, and Brown used as color channels. As a result, the segmentation proposal using probabilities and entropy obtained better scores of accuracy (86.8%) and F_1 score (84.1%). In addition, a new proposal for feature vectors using different colour channels was used to train machine learning models (SVM, Decision Trees, Random Forests and Adaboost) evidenced significant gains in segmentation quality, which achieves 92% accuracy and 92% F_1 score for segmentation quality. We also propose a mechanism for fire and smoke detection and fire localization, this mechanism is capable of detect fire with 91% to 93% of detection rate and also localize the region affected by fire.

Keywords: Fire detection, Fire Localization, Smoke Detection, Convolutional Neural Network, Machine Learning.

Resumo

Os incêndios representam uma ameaça constante para os sistemas ecológicos, infraestrutura e vidas humanas. A utilização de recursos computacionais por meio do processamento digital de imagens mostra-se uma alternativa viável e de baixo custo como auxílio na detecção precoce de fogo. Para solucionar o problema de detecção de fogo em imagens, o presente trabalho atuou em duas frentes principais: na detecção de fogo e fumaça, utilizando Redes Neurais Convolucionais - RNC (1) e na segmentação de pixels de fogo por meio de técnicas de aprendizagem de máquina (2). Para isso, foram desenvolvidas bases de imagens rotuladas voltadas para detecção de fogo/fumaça e segmentação de fogo. No que diz respeito ao processo de detecção, as arquiteturas VGG16, Inception, Resnet50, Xception e Mobilenet foram modificadas e testadas. Além disso, ao testar diferentes camadas de convolução e variar às estruturas das RNCs, foi possível desenvolver uma arquitetura de RNC menor e mais rápida do que às presentes no estado da arte, nomeada de *Fire Smoke Detection Network - FSDN*. O desempenho desta rede foi comparado ao desempenho das arquiteturas existentes na literatura. Os resultados obtidos apontaram que utilizar redes neurais binárias para detectar fogo é mais eficiente do que redes multiclases, pois os valores de acurácia para redes binárias variaram de 89% a 99% para fogo e 82% a 96% para fumaça enquanto que para redes multiclases (fogo, fumaça ou negativo) a variação da acurácia foi de 79% a 95%. A arquitetura *FSDN* proposta obteve 95% e 90% de taxa de detecção de fogo e fumaça, respectivamente e foi executada 2,9 vezes mais rápida que a arquitetura com melhor desempenho para detecção. Para o processo de localização de fogo, foram propostos dois novos métodos de detecção de pixels de fogo. O primeiro método utilizou abordagens probabilísticas mesclando canais de cor de diferentes espaços de cor em conjunto com cálculos de entropia. O segundo método proposto é baseado nas probabilidades dos índices de cor: Vermelho, Amarelo e Marrom. Como resultados, a proposta de segmentação utilizando probabilidades e entropia obteve melhores pontuações de 86,8% e 84,1%, de acurácia e $F_1 score$ respectivamente. Adicionalmente, uma nova proposta de vetor de características aplicados à algoritmos de aprendizado máquina (SVM, Árvores de Decisão, Florestas Randômicas e Adaboost) evidenciou ganhos significativos na qualidade de segmentação obtendo em média 92% de Acurácia e 92% de $F_1 score$ para a qualidade de segmentação. Como resultado, um mecanismo de detecção de fogo e fumaça e de localização de fogo é proposto, onde o valor da taxa de detecção fogo ficou entre 91% e 93% e após o processo de detecção há a localização de fogo.

Palavras-chave: Detecção de Fogo, Localização de Fogo, Detecção de Fumaça, Redes Neurais Convolucionais, Aprendizado de Máquina.

Sumário

1	Introdução	20
1.1	Definição do Problema	21
1.2	Objetivos	23
1.3	Hipóteses	24
1.4	Propostas	25
1.5	Contribuições	25
1.6	Estrutura do Trabalho	26
2	Fundamentação Teórica	27
2.1	Fogo	27
2.1.1	Queimadas	28
2.1.2	Controle do Fogo	30
2.2	Processamento Digital de Imagens	32
2.2.1	Transformações para índices de cores	33
2.2.2	Entropia aplicada ao processamento digital de imagens	34
2.3	Aprendizagem de Máquina	35
2.3.1	SVM – <i>Support Vector Machine</i>	36
2.3.2	<i>Random Forest</i>	37
2.3.3	Adaboost	38
2.4	Redes Neurais Convolucionais	38
2.4.1	Redes Neurais	38
2.4.2	Convolução em Redes Neurais	40
2.4.3	Função de Ativação e Pooling	40
2.4.4	Técnicas para redução do sobre-ajuste (<i>overfitting</i>)	41
2.4.5	Deteção de Objetos e Redes Neurais Convolucionais	42
2.5	Métricas de avaliação	48
2.5.1	Avaliação de Deteção	50
2.5.2	Avaliação de Segmentação	50
2.6	Considerações Finais	52

3	Estado da Arte	54
3.1	Detecção e localização de incêndio baseados em regras nos espaços de cor . . .	55
3.1.1	Detecção e localização de fogo no espaço de cor RGB	55
3.1.2	Detecção e localização de fogo no espaço de cor HSI	56
3.1.3	Detecção e localização de fogo nos espaços de cor YCbCr e YUV . . .	56
3.1.4	Detecção e localização de fogo no espaço de cor CIELab	58
3.1.5	Detecção e localização de fogo com espaços de cor híbridos	59
3.2	Detecção e localização de incêndio usando aprendizagem de máquina	62
3.3	Detecção de fogo e as Redes Neurais	64
3.3.1	Detecção de fogo e fumaça e as Redes Neurais Convolucionais	65
3.4	Considerações Finais	70
4	Método Proposto	72
4.1	Bases de Imagens	74
4.1.1	Base de imagem para detecção de fogo e fumaça	74
4.1.2	Base de imagem para localização de fogo	80
4.2	Considerações Finais	83
5	Detecção de fogo e fumaça	84
5.1	Método proposto para detecção de fogo e fumaça	86
5.1.1	Configurações dos parâmetros em Redes Neurais Convolucionais . . .	87
5.1.2	Proposta e avaliação de diferentes arquiteturas de RNCs	90
5.2	Resultados experimentais das propostas de modelos arquiteturais de RNCs . . .	91
5.2.1	Comparativo de desempenho das RNCs	97
5.2.2	Avaliação das taxas de detecção de fogo e fumaça nas RNCs estudadas	106
5.3	Comparativo entre métodos de detecção de fogo	109
5.4	Considerações finais	110
6	Localização de fogo	112
6.1	Análise dos canais de cor	113
6.2	Tabelas de consulta baseadas em probabilidades	117
6.3	Localização de fogo usando entropia e rotulação	122
6.4	Localização de fogo usando índices de cor	126
6.5	Localização de píxeis de fogo usando aprendizagem de máquina	132
6.6	Comparativo entre algoritmos de Segmentação de Fogo	136
6.7	Considerações finais	139
7	Monitoramento de fogo e fumaça	140
7.1	Método	140
7.2	Resultados obtidos	142

7.3	Considerações finais	144
8	Conclusões e Trabalhos futuros	146
	Referências Bibliográficas	150
A	Apêndice A - Formação das cores e Espaços de Cor	158
A.1	Formação das cores	158
A.2	Espaços de Cor	159
A.2.1	RGB	159
A.2.2	YCbCr e YUV	160
A.2.3	HSI, HSV e HSL	161
A.2.4	CIEL*a*b*	164
B	Apêndice B - Algoritmos de Aprendizagem de máquina	166
B.1	Algoritmos de clusterização	166
B.1.1	K-Means	166
B.1.2	Nearest Neighbor	166
B.2	Algoritmos de Probabilidades	168
B.2.1	Naïve Bayes	168

Lista de Figuras

1.1	Imagens de não fogo (<i>a, b e c</i>) e fogo (<i>d, e e f</i>).	23
2.1	Estrutura do Fogo.	28
2.2	Tetraedro do Fogo.	29
2.3	Frequência dos focos de incêndio no Brasil por mês em 2019 comparado com a média histórica de 1998 até 2018, Fonte: (INPE, 2019).	30
2.4	Gastos relacionados a incêndios – Fonte: (Brushlinsky et al., 2015).	31
2.5	Exemplo de tipos de regressão para os mesmos dados (a) Regressão Linear e (b) Regressão Polinomial – Fonte: (Murphy, 2012).	36
2.6	Exemplo de um problema separável de duas classes e os vetores de suporte mostrando a distância ótima entre os objetos de classes distintas – Fonte: Cortes and Vapnik (1995).	37
2.7	Neurônio Artificial (perceptron) – Fonte: (Barreto, 2002).	39
2.8	Representação de uma Rede Neural Artificial e suas camadas internas.	39
2.9	Exemplo de maxpooling aplicado em uma porção de uma imagem com máscara de tamanho 2×2	41
2.10	Arquitetura da AlexNet - Fonte: (Krizhevsky et al., 2012).	43
2.11	Configurações das camadas arquiteturais das redes VGG, variando de 11 a 19 camadas - Fonte: (Simonyan and Zisserman, 2014).	44
2.12	Estrutura de um bloco residual de aprendizagem – Fonte: (He et al., 2015a).	44
2.13	Estruturas Arquiteturais das Redes Resnet e medida de operações de números reais por segundo (FLOPs) – Fonte: (He et al., 2015a).	45
2.14	Módulo Inception – Fonte: (Szegedy et al., 2015).	46
2.15	Bloco Inception Extremo – Fonte: (Chollet, 2016).	47
2.16	Arquitetura da Rede Xception – Fonte: (Chollet, 2016).	47
2.17	Arquitetura da MobileNet – Fonte: (Howard et al., 2017).	48
2.18	Matriz de Confusão. Fonte: (Fawcett, 2006)	49
3.1	Arquitetura CNN para detecção de fogo e fumaça proposta por (Frizzi et al., 2016). Fonte: (Frizzi et al., 2016).	65
3.2	Arquitetura da RNC proposta por Zhang et. al. Fonte: (Zhang et al., 2016).	66

3.3	Arquitetura de RNC reduzida proposta por Zhang et. al. Fonte: (Zhang et al., 2016).	66
3.4	Arquitetura RNC proposto por (Maksymiv et al., 2017).	67
3.5	Arquitetura para detecção de fumaça proposta por Yin et al. (2017).	69
4.1	Visão Geral da proposta de Detecção de Fogo e Fumaça e Localização de Fogo em Imagens.	73
4.2	Disposição das Bases de Imagens para Detecção.	75
4.3	Exemplo de sobreposição para extração dos patches das imagens.	77
4.4	Exemplos da base BD1 dos <i>patches</i> de fogo e seus contextos.	78
4.5	Exemplos da base BD1 dos <i>patches</i> Negativos e seus contextos.	78
4.6	Exemplos da base BD1 dos <i>patches</i> de fumaça e seus contextos.	79
4.7	Exemplos de imagens com os <i>patches</i> e seus respectivos rótulos.	80
4.8	Disposição das Bases de Imagens para Localização.	81
4.9	Exemplo de imagens de fogo da base de imagens BL2.	81
4.10	Exemplo de solução ideal para as imagens de fogo da BL2.	81
4.11	Exemplos de imagens de não fogo da BL3.	82
5.1	Etapas metodológicas utilizadas para proposição de arquitetura de Rede Neural Convolucional voltada para detecção de fogo e fumaça.	85
5.2	Evolução da acurácia utilizando CLR (em vermelho) e não utilizando (em azul e verde). Fonte: Smith (2015).	88
5.3	Evolução da taxa de aprendizagem em relação a função de perda (<i>loss</i>).	89
5.4	Rede Neural $FSDN_1$	92
5.5	Rede Neural $FSDN_2$	92
5.6	Rede Neural $FSDN_3$	92
5.7	Rede Neural $FSDN_4$	93
5.8	Rede Neural $FSDN_5$	93
5.9	Redes Neurais $FSDN_6$	94
5.10	Redes Neurais $FSDN_7$ e $FSDN_8$	95
5.11	Redes Neurais $FSDN_9$	96
5.12	Bloco Convolucional.	96
5.13	Bloco Identidade.	97
5.14	Redes Neurais $FSDN_{10}$	97
5.15	<i>Patches</i> de fumaça da base BD1 (a) e <i>patches</i> de fumaça da base BD3 (b).	104
6.1	Etapas metodológicas para criação de método de segmentação de imagens de Fogo.	113
6.2	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor RGB.	114

6.3	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor YCbCr.	114
6.4	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor YUV.	115
6.5	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSV.	115
6.6	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSL.	116
6.7	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSI.	116
6.8	Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor <i>CieLab</i>	116
6.9	Exemplo de cálculo dos canais de cor RGB para consulta na Tabela de consulta.	118
6.10	Fluxograma de criação das tabelas de consulta.	119
6.11	Processo de Consulta de um pixel $P(x,y,c)$ em uma tabela de consulta.	120
6.12	Método Proposto baseado em Probabilidades.	122
6.13	Processo de rotulação em imagens de fogo.	123
6.14	Método de localização de píxeis de fogo utilizando entropia para retirada de elementos de não fogo.	125
6.15	Aplicação do índices Redness: a) Imagem original; b) (Held, 2002); c) (Gasparini and Schettini, 2005); d) (Smolka et al., 2003).	128
6.16	Aplicação do índices de Amarelo: a) imagem original, b) (Doi, 2012), c) (Robinson, 1998) e d) (ASTM, 2015).	129
6.17	Aplicação de índices de Marrom: a) imagem original, b) (Ricauda et al., 2015) e c) (Pathare et al., 2013).	130
6.18	Processo de construção da tabela de consulta para os índices de Vermelho, Amarelo e Marrom.	131
6.19	Exemplo de vetores de uma imagem de Fogo e Não Fogo e os píxeis extraídos em vermelho, utilizando $c=30$	133
7.1	Método utilizado para teste em algoritmos de detecção e localização de fogo, com o objetivo de identificar melhor estratégia para criação de um mecanismo de monitoramento de fogo.	141
7.2	Modelo de Monitoramento Proposto para detecção e localização de fogo.	144
7.3	Exemplos de saída do Método de Monitoramento de Fogo e Fumaça proposto para detecção e localização de fogo.	145
A.1	Espectro eletromagnético visível – Fonte (SOUTO, 2000).	158
A.2	Representações geométricas do espaço de cor RGB.	160
A.3	Representações geométricas do espaço de cor HSI.	162

A.4	Representação geométrica do espaço de cor HSV.	163
A.5	Representação geométrica do espaço de cor HSL.	164

Lista de Tabelas

2.1	Comparativo entre diversos tipos de monitoramento de fogo – Fonte: (Alkhatib, 2014).	31
3.1	Principais características dos métodos estudados	71
4.1	Lista de Palavras utilizadas para busca de imagens de fogo, fumaça e não fogo.	76
4.2	Distribuição dos Patches em categorias e bases de treino, validação e teste.	76
4.3	Distribuição dos patches nos respectivos contextos e classes.	77
4.4	Exemplos da divisão das imagens por <i>patches</i>	79
4.5	Distribuição dos píxeis de fogo nas imagens de fogo coletadas.	82
5.1	Resultado dos testes realizados com 10 diferentes arquiteturas de RNCs.	95
5.2	Referência bibliográfica e sigla utilizada para referenciar o trabalho.	98
5.3	Resultados das arquiteturas binárias para detecção de fogo - Treino em BD1.	100
5.4	Resultados das arquiteturas binárias para detecção de fogo - Treino em BD3.	100
5.5	Resultados com as arquiteturas binárias para detecção de fumaça - Treino em BD1.	102
5.6	Resultados com as arquiteturas binárias para detecção de fumaça - Treino em BD3.	102
5.7	Resultados da detecção de fogo e fumaça em Redes Neurais multiclases - Treino realizado na base BD1.	105
5.8	Resultados da detecção de fogo e fumaça em Redes Neurais multiclases - Treino realizado na base BD3.	105
5.9	Taxa de detecção de fogo e fumaça nas arquiteturas estudadas.	107
5.10	Desempenho em relação ao tempo de processamento, Taxa de detecção (TxD), média de imagens por segundo, espaço em disco dos pesos e número de parâmetros de cada arquitetura testada na base BD2.	108
5.11	Nome da Técnica e Referência bibliográfica correspondente.	109
5.12	Comparativo das taxas de detecção de fogo nas técnicas baseadas em Regras e técnicas que utilizam aprendizagem de máquina.	110
6.1	Valores médios dos desvios padrão.	117
6.2	Valores para o limiar de probabilidades testados e seus resultados.	121

6.3	As 22 melhores combinações dos canais de cor ordenados por valores da métrica F_1score (F1).	121
6.4	Resultados obtidos a partir do método das probabilidades proposto.	122
6.5	Rótulos de Fogo e regras de entropia.	124
6.6	Rótulos de não Fogo e regras de entropia.	124
6.7	Resultados de Segmentação usando Probabilidades.	126
6.8	Resultado da aplicação do método probabilístico aplicado a imagens que contenham fogo.	132
6.9	Resultado da média das métricas dos modelos de aprendizagem de máquina nas bases de Teste e Treino.	135
6.10	Limiars utilizados para cada uma das técnicas estudadas.	137
6.11	Resultado da qualidade de segmentação em BL2.	138
6.12	Resultado da qualidade de Segmentação em BL1.	138
6.13	Resultados obtidos na base BL3.	139
7.1	Taxas de Acerto antes e após processo de segmentação nos patches da base BL3 para as arquiteturas Xception e Arquitetura Proposta.	142

Lista de Acrônimos

ASTM	<i>American Society for Testing and Materials</i> (Associação Americana de Teste e Materiais)
AVHRR	<i>Advanced Very High Resolution Radiometer</i> (Radiômetro Avançado de Alta Resolução)
BD1	Base de Detecção de Imagens 1
BD2	Base de Detecção de Imagens 2
BD3	Base de Detecção de Imagens 3
BL1	Base de Localização de Imagens 1
BL2	Base de Localização de Imagens 2
BL3	Base de Localização de Imagens 3
CMYK	<i>Cyan Magenta Yellow Black</i> (Ciano Magenta Amarelo e Preto)
CNN	<i>Convolutional Neural Network</i>
CTIF	<i>Comité Technique International de prevention et d'extinction de Feu</i> (Comitê Técnico Internacional de prevenção e supressão de Fogo)
DP	Desvio Padrão
ESA	<i>European Space Agency</i> (Agência Espacial Europeia)
FN	<i>False Negative</i> (Falso Negativo)
FP	<i>False Positive</i> (Falso Positivos)
FSDN	<i>Fire and Smoke Detection Network</i>
$FSDN_x$	<i>Fire and Smoke Detection Network variation x</i>
GOES	<i>Geostationary Operational Environmental Satellite</i> (Satélite Geoestacionário de meio-ambiente)
HSI	<i>Hue Saturation Intensity</i> (Matiz Saturação Intensidade)
HSL	<i>Hue Saturation Luminosity</i> (Matiz Saturação Luminosidade)
HSV	<i>Hue Saturation Value</i> (Matiz Saturação Valor)
IA	Inteligência Artificial
IBAMA	Instituto Brasileiro do Meio-Ambiente e dos Recursos Naturais
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
INPE	Instituto Nacional de Pesquisas Espaciais
MAE	<i>Mean Absolute Error</i> (Erro absoluto Médio)

MSG	<i>Meteosat of Second Generation</i> (Satélite Meteorológico de Segunda Geração)
MODIS	<i>Moderate Resolution Imaging Spectroradiometer</i> (Espectro radiômetro de imagem de resolução moderada)
NASA	<i>National Aeronautics and Space Administration</i> (Agência Nacional de Administração Aeronáutica e do Espaço)
NFPA	<i>National Fire Protection Association</i>
PB	Probabilidades
PDI	Processamento Digital de Imagens
RB	<i>Rule Based</i> (Baseado em regras)
RBF	<i>Radial Basis Function</i>
RGB	<i>Red Green Blue</i> (Vermelho Verde Azul)
RNC	Redes Neurais Convolucionais
SRM	<i>Structure Risk Minimization</i>
SVM	<i>Support Vector Machine</i>
TC	Tabela de Consulta
TN	<i>True Negative</i> (Verdadeiro Negativo)
TP	<i>True Positive</i> (Verdadeiros Positivos)
WEKA	<i>Waikato Environment for Knowledge Analysis</i>
YCbCr	<i>Luminance Chrominance blue Chrominance red</i>

Capítulo 1

Introdução

Os incêndios representam uma ameaça constante para os sistemas ecológicos, infraestrutura e vidas humanas. Quando acontece um incêndio numa cidade, dependendo de sua localização, o poder público pode ser obrigado a dar poderes especiais aos serviços de emergência que incluem: ordens de evacuação e eliminação de elementos combustíveis como corte de gás e eletricidade, por exemplo. Quando o incêndio atinge uma floresta, o ecossistema florestal sofre consequências graves e precisa de tempo para se regenerar. Dados da NFPA (2014) - *National Fire Protection Association* - mostram que os incêndios em 2014 causaram a morte de 3.275 pessoas, feriram 15.775 pessoas e deram um prejuízo de mais de U\$ 11,6 bilhões, nos Estados Unidos.

O Centro de Estatísticas dos Incêndios (*Center of Fire Statistics* - CTIF) da Associação Internacional de Serviços de Incêndio e Resgate (*International Association of Fire and Rescue Services*) em Brushlinsky et al. (2018) produziu um relatório sobre incidentes de fogo em 39 países (incluindo alguns países Europeus, Estados Unidos e o Japão). As estatísticas apresentam os incidentes separados por países e por tipos/cenários de incêndio. O referido relatório mostra que a média de distribuição dos incidentes de incêndio durante os anos de 2012 e 2016 foram: Estrutural 35,5%, Rodovia 13,6%, Florestal 3,3%, Áreas Verdes 18%, Lixo 8,9% e outros ambientes 20%.

Tendo estes dados como base, percebe-se que a maioria dos incidentes de fogo ocorrem em locais onde há a presença humana. A prevenção ao incêndio visa ao combate a um dos elementos que combinados causam o fogo: calor, oxigênio (comburente), combustível e a reação em cadeia. Sem dúvida, a prevenção é o melhor aliado à eliminação do incêndio. Porém, é inviável controlar a variável do calor em ambientes abertos, já que estes dependem das características climáticas, como temperatura e umidade. Dessa forma, a detecção e localização de um incêndio em menor tempo são de suma importância para minimizar os danos gerados. Tais informações corroboram com a necessidade do avanço de técnicas de detecção de incêndio que possam ser mais efetivas nos cenários mais afetados como áreas urbanas, rodovias e ambientes agrícolas que juntos correspondem a 67,1% dos incidentes de acordo com o relatório do CTIF (*Comité Technique International de prevention et d'extinction de Feu*) (Brushlinsky et al., 2018).

Existem diversos mecanismos de detecção de fogo, dentre eles, pode-se citar: sistemas de vigilância móvel/fixa, sensoriamento remoto, sensores ópticos, químicos, térmicos, radares, entre outros. Em geral, sensores utilizados para detecção de incêndio, além de possuírem altos custos, devem estar localizados próximos ao início do evento, o que os tornam inviáveis para detecção em ambientes abertos. Nesse sentido, utilizar sensores ópticos (como imagens de câmeras de vigilância, por exemplo) para captura de imagens e então realizar o processamento delas com o auxílio de técnicas de aprendizado de máquina é uma alternativa. Porém, o processo de detecção e localização de fogo em imagens é uma tarefa complexa, pois o fogo possui formas irregulares, assim como alta variabilidade de cores; isso se deve principalmente às reações que ocorrem entre o tipo do material combustível e o oxigênio. Assim, visto que o fogo não possui um padrão para: cor, volume ou formato, o processo de sua localização e detecção torna-se difícil.

Tendo em vista os desafios apresentados para detecção e localização do fogo, o presente trabalho propõe duas etapas para discussão das questões principais: detecção de fogo e fumaça (1) e localização de fogo (2). Para atacar o problema de detecção de fogo e fumaça, estruturas de Redes Neurais Convolucionais foram criadas e comparadas às existentes na literatura. Já para o problema de localização do fogo, abordagens probabilísticas foram aplicadas com o objetivo de mapear píxeis com maior probabilidade de serem de fogo, assim como modelos mais robustos de aprendizado de máquina como: SVM, Árvores de Decisão, Árvores Aleatórias (*Random Forest*) e Adaboost foram testados e comparados com os principais métodos da literatura.

1.1 Definição do Problema

Visando à resolução do problema de detecção e localização de fogo em ambientes abertos, diversos pesquisadores propuseram técnicas de detecção de fogo a partir do processamento digital de imagens. Os trabalhos que tratam da detecção de fogo em vídeos utilizam diferenças entre quadros para remoção do fundo dos elementos estáticos da cena. Após esta filtragem, a classificação dos píxeis é aplicada com o objetivo de localizar o foco do incêndio ou simplesmente detectar se existe algum pixel correspondente a fogo nas imagens. Para esse processo de segmentação, pode-se afirmar que a literatura utiliza duas abordagens principais: criação de regras aplicadas a canais de cor ou a utilização de algoritmos de aprendizagem de máquina.

Regras aplicadas a canais de cor dizem respeito ao mapeamento de intervalo de valores, onde há maior probabilidade de um pixel ser de fogo ou não, como por exemplo define Horng et al. (2005) que para os píxeis de fogo os valores do canal H (Matiz), do espaço de cor HSI, devem estar entre 0° e 60° . Dentre os autores que utilizam regras aplicadas a determinados espaços de cor, pode-se citar: RGB (Phillips et al., 2002; Toreyin et al., 2005; Çelik et al., 2006, 2007); HSI (Horng et al., 2005); YUV (Marbach et al., 2006; Rudz et al., 2009); YUV com RGB (Chen et al., 2004); RGB com HSI (Ko et al., 2009); YCbCr (Çelik and Demirel, 2009); CIELab

em (Çelik, 2010); RGB e YCbCr (Vipin, 2012; Chino et al., 2015) e CIELab e RGB (Toulouse et al., 2015).

Além de abordagens que realizam o mapeamento dos píxeis de fogo por meio de regras, existem abordagens que fazem uso de algoritmos de classificação ou agrupamento (*clustering*) como: K-Means (Rossi et al., 2009), Nearest Neighbor (Wong and Fong, 2014), Naïve Bayes (Chino et al., 2015; Ko et al., 2010), SVM (Collumeau, 2011) e (Ko et al., 2009), Redes Neurais (Angayarkkani and Radhakrishnan, 2010), ADABOST (Dukuzumuremyi et al., 2014a) e Boosting (Ozkan et al., 2008) e, mais recentemente usando Redes Neurais Convolucionais (CNN), (Frizzi et al., 2016; Maksymiv et al., 2017; Zhang et al., 2016; Muhammad et al., 2018).

O levantamento bibliográfico deste tema nos levou ao seguinte problema: as bases de imagens. As bases de imagens encontradas possuem poucas amostras ou estão em resolução insuficiente para o treinamento efetivo de Redes Neurais Convolucionais que requerem imagens com pelo menos 299×299 píxeis. Além disso, alguns trabalhos não disponibilizam suas bases, o que dificulta o processo de replicação dos resultados reportados, assim como, o treino de modelos de aprendizagem de máquina mais robustos. Nesse sentido, o presente trabalho propõe uma nova base com mais de 110 mil *patches* (fragmentos de imagens). Tal base foi dividida em 4 cenários de incêndio rotuladas em: Rodovia, Floresta, Estrutural e imagens que contenham Pessoas próximas a áreas de incêndio. Esta nova base possibilita o treino de modelos de aprendizagem de máquina considerando especificidades dos cenários para a detecção de incêndio e fumaça.

Ainda em relação à Redes Neurais Convolucionais, quando se pretende aplicar estas redes em equipamentos com capacidade computacional limitada, como: celulares, drones e arquiteturas móveis, por exemplo, é preciso se preocupar com o tamanho e desempenho das redes. Redes com muitos parâmetros, ou que ocupem muito espaço em disco podem não ser portáteis a dispositivos móveis, conforme mostra Howard et al. (2017) ao propor as redes MobileNet. Dessa forma, a maioria dos trabalhos que utilizam Redes Neurais Convolucionais para detecção de fogo não apresentam compromisso com eficiência em dispositivos móveis de capacidade computacional limitada.

Já no contexto de localização e segmentação, as bases de imagens encontradas, não estão separadas por contexto/cenários como é o caso da base disponibilizada por Chino et al. (2015), ou são direcionadas somente para fogo florestais, como é o caso de Toulouse et al. (2017). Neste sentido, este trabalho propõe uma base com 1000 imagens de fogo segmentadas em 5 diferentes contextos (Agrícola, Floresta, Rodovia, Indústria e Urbano). Vale destacar que o estudo de segmentação foi realizado somente nos píxeis de fogo, pois a fumaça dispersa rapidamente no ambiente, e a presença do fogo é que vai indicar com maior precisão onde está o foco do incêndio, podendo assim auxiliar as entidades no combate ao incêndio.

Um das maiores dificuldades para a detecção e localização de fogo, reside nas características similares de fogo que são encontradas em imagens que não possuem fogo e na distância em que se encontram os focos de incêndio. A Figura 1.1 ilustra 3 diferentes imagens de não fogo (*a, b e c*) que possuem características de coloração similares ao fogo, como por exemplo:

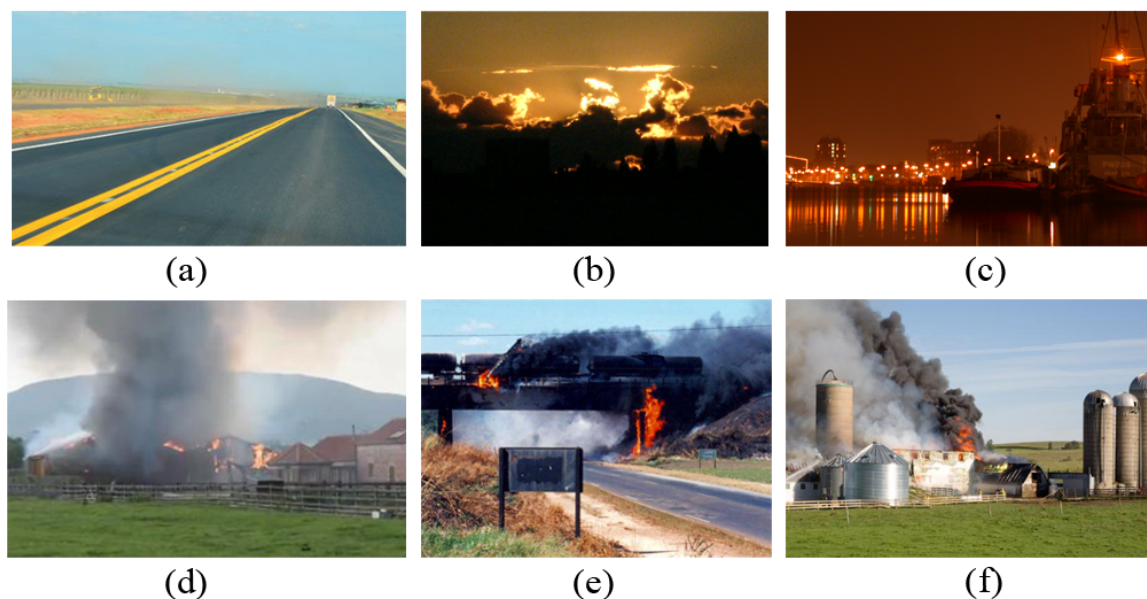


Figura 1.1: Imagens de não fogo (*a, b e c*) e fogo (*d, e e f*).

a faixa amarela de uma estrada (Figura 1.1*a*), o sol e seu reflexo em nuvens (Figura 1.1*b*) ou a iluminação noturna de uma cidade (Figura 1.1*c*). Em relação às imagens que contêm fogo (sub-figuras *d, e e f* da Figura 1.1), uma das dificuldades de localização de fogo encontra-se principalmente em relação à distância e o tamanho dos focos de incêndio que são difíceis de serem detectados, conforme mostra a Figura 1.1*d*.

1.2 Objetivos

O objetivo principal do trabalho consiste em propor um método de detecção de fogo e fumaça e localização de fogo explorando características extraídas de forma automática e manual. A extração de características desta última é elaborada a partir da exploração de métodos de aprendizagem de máquina como: Redes Neurais Convolucionais, SVM e Árvores de Decisão. Já a extração de características de fogo de forma manual consiste em analisar os canais de cor, aplicar abordagens probabilísticas em busca de padrões que possibilitem a classificação de píxeis como fogo ou não fogo. Com base nisso, os seguintes objetivos específicos foram definidos:

- Criar base de imagens que possibilite avaliar a qualidade de detecção de fogo e fumaça. Sendo coletadas imagens de fogo, fumaça e não fogo, assim como rotular as imagens nas três classes;
- Criar base de imagens para avaliação da qualidade de segmentação de píxeis de fogo com a definição das soluções ideais de segmentação para cada uma das imagens, com os píxeis rotulados em fogo e não fogo;

- Testar e avaliar diferentes arquiteturas de redes neurais convolucionais na base de detecção, considerando mudanças nas camadas destas arquiteturas;
- Propor uma nova arquitetura de Rede Neural Convolucional para detecção de fogo e fumaça que possua desempenho compatível com o estado da arte utilizando menos recursos computacionais;
- Testar e avaliar diferentes métodos para segmentação/localização de píxeis de fogo propostos na literatura;
- Treinar e testar modelos de aprendizagem de máquina (SVM, Árvores de Decisão, Árvores Aleatórias e Adaboost) utilizando características pré-selecionadas visando à segmentação de píxeis de fogo;
- Propor novos métodos de segmentação de imagens de fogo realizando estudo nos espaços de cor e selecionando os canais que mais destacam as características de fogo, aplicando técnicas de probabilidades, entropia e aprendizagem de máquina;
- Propor um método de monitoramento de fogo e fumaça, que realize a união de métodos de detecção e localização de fogo capaz de alertar sobre a presença de fumaça e fogo.

1.3 Hipóteses

Neste trabalho, 3 hipóteses principais foram levantadas.

Hipótese 1 – Redes Neurais Convolucionais - RNCs - projetadas para dispositivos móveis e treinadas com uma base representativa de imagens digitais de fogo e fumaça são capazes de apresentar resultados de taxa de detecção compatíveis a métodos existentes na literatura.

Hipótese 2 – A aplicação de algoritmos de aprendizagem de máquina, e abordagens probabilísticas, utilizando diferentes espaços de cor, variando canais de cor, assim como técnicas de índices de cor, aplicadas a píxeis de imagens de fogo e não fogo são capazes de criar mecanismos de localização com melhores taxas de acurácia e $F_1 score$ comparados a métodos existentes.

Hipótese 3 – Um modelo de aprendizagem de máquina utilizando Redes Neurais Convolucionais para detecção de fogo e fumaça em imagens, aliado à abordagem probabilística para localização de píxeis de fogo em imagens resultam em um método de monitoramento de incidentes de fogo robusto e preciso.

À vista disso, as principais questões da pesquisa são:

1) Ao Utilizar Redes Neurais Convolucionais aplicadas à detecção de fogo e fumaça é possível aumentar a taxa de detecção em relação aos resultados da literatura?

2) Utilizar abordagens probabilísticas para identificação de píxeis de fogo a partir de variações de canais de diferentes espaços de cor podem ser utilizados a fim de otimizar o processo de localização de fogo?

3) Aprimorar o processo de segmentação de píxeis de fogo a partir de modelos de aprendizagem de máquina, como: SVM, Árvores de Decisão, Florestas Aleatórias e Adaboost, são mais eficientes que os métodos baseados em regras?

4) Ao utilizar RNCs para detecção de fogo e fumaça, aliado à modelos de aprendizagem de máquina para localização de fogo, seria possível a proposição de um método de monitoramento, com alta taxa de detecção e bom índice de localização ($F_1 score$)?

Portanto, as propostas para responder estas perguntas consistem em elaborar bases de imagens, aplicar técnicas de aprendizagem de máquina, implementar modelos existentes na literatura, propor novos modelos e realizar a comparação entre os modelos propostos e os métodos existentes.

1.4 Propostas

De acordo com as hipóteses e as questões de pesquisa levantadas, a proposta deste trabalho está concentrada em 5 etapas principais: (1) Criação de uma base de imagens, com imagens de situações de fogo, fumaça e não fogo; (2) Elaboração de método de detecção capaz de detectar fogo e fumaça utilizando abordagens de aprendizagem de máquina; (3) Criação de base de imagens para localização de fogo, em que píxeis de fogo serão separados dos píxeis de não fogo; (4) Elaboração de métodos baseados em aprendizagem de máquina e abordagens probabilísticas capazes de localizar píxeis de fogo em imagens e, por fim, (5) a proposição de um método de monitoramento e detecção de fogo e fumaça integrado à localização de fogo em imagens.

1.5 Contribuições

O presente trabalho apresenta contribuições subdivididas em 3 categorias: Sociais, Científicas e Tecnológicas. Em relação às contribuições Sociais, observamos que o auxílio no processo de detecção e localização de fogo em tempo real pode minimizar os danos à sociedade e aos ecossistemas, que poderiam ser causados em um eventual incêndio de grandes proporções.

Já as contribuições Tecnológicas incluem a proposição de duas bases de imagens, sendo uma voltada para detecção de fogo e fumaça com mais de 110 mil imagens rotuladas e outra base voltada para localização de fogo com 1000 imagens segmentadas pixel a pixel apresentando a solução ideal. Cabe ressaltar que as bases criadas possibilitam um alto potencial de inovação de sistemas de monitoramento, uma vez que elas podem ser utilizadas para aprimorar sistemas existentes. Além disso, também é proposto um método de monitoramento de fogo robusto a partir dos métodos criados para detecção de fogo/fumaça e localização de fogo.

No que diz respeito às contribuições Científicas, o presente trabalho traz inovação a partir da elaboração de métodos computacionais de aprendizagem profunda usando conceitos de redes neurais convolucionais, propondo uma nova e compacta arquitetura de rede neural para detecção de fogo e fumaça. Além disso, também é apresentada a análise de desempenho entre os métodos de detecção de fogo e fumaça existentes na literatura, assim como, das principais arquiteturas de redes neurais convolucionais para a resolução deste problema.

Ainda em relação às contribuições Científicas, há a criação de dois novos métodos computacionais voltados para segmentação de imagens de fogo. (1) os quais foram idealizados a partir de abordagens probabilísticas em conjunto com características de entropia e um novo método de segmentação de fogo (2) baseado na probabilidade dos índices das cores vermelha, amarela e marrom. Adicionalmente, abordagens robustas de aprendizagem de máquina foram testadas e um novo vetor de características como elemento de entrada para estes métodos é proposto. Ao final, é apresentada a análise comparativa entre os métodos de segmentação presentes na literatura com os métodos propostos.

1.6 Estrutura do Trabalho

O presente trabalho está subdividido em 7 capítulos: Introdução, Fundamentação Teórica, Estado da Arte, Método proposto, Detecção de fogo e fumaça, Localização de fogo e, por fim, Conclusões e Trabalhos Futuros. No Capítulo 2 serão apresentados tópicos que serviram de base para o desenvolvimento deste trabalho. Os tópicos apresentados são: fundamentos básicos sobre fogo; processamento digital de imagens; Aprendizagem de Máquina, no qual serão abordadas Redes Neurais Convolucionais e métricas de avaliação. No Capítulo 3, temos o Estado da arte que apresentará os principais métodos de detecção e segmentação de fogo presentes na literatura. O Capítulo 4 apresenta a visão geral do método desenvolvido no trabalho, assim como a descrição de como as bases de imagens foram criadas. No Capítulo 5, intitulado "Detecção de fogo e fumaça", há o detalhamento do método utilizado para a elaboração de modelos de Redes Neurais Convolucionais, assim como os protocolos experimentais e resultados obtidos para o processo de detecção de fogo e fumaça. O Capítulo 6, intitulado "Localização de fogo", trata do processo de localização de fogo em imagens e apresenta dois métodos baseados em probabilidades, assim como, propõe um vetor de características para ser utilizado no treinamento de modelos de aprendizagem de máquina. O Capítulo 7 apresenta um método de monitoramento de fogo e fumaça. O Capítulo 8 apresenta às conclusões do trabalho e as proposições de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados os conceitos que nortearam o desenvolvimento deste trabalho. A Seção 2.1 mostra o que é o fogo e quais são as implicações quando o mesmo não é controlado de forma adequada. A Seção 2.3 trata de métodos de aprendizagem de máquina utilizados com foco na localização de píxeis de fogo, os algoritmos abordados foram: SVM, Random Forest, Adaboost e Redes Neurais Artificiais. A Seção 2.4 trata dos conceitos que englobam Redes Neurais Convolucionais, utilizadas como base para a criação de novos modelos de detecção de fogo/fumaça. A Seção 2.5 trata das métricas de avaliação para os processos de detecção e segmentação.

2.1 Fogo

De acordo com Gaylor (apud (Saraiva, 2011), p. 17), o fogo é uma reação química de rápida oxidação de um material combustível, liberando assim, luz, calor, dióxido de carbono (CO_2) e água. Na Grécia antiga, acreditava-se que o fogo era um elemento, porém, hoje sabe-se que o fogo é uma mistura de diversas substâncias em altas temperaturas. As chamas são resultado de reações químicas entre o oxigênio presente no ar (Comburente), um elemento combustível, como madeira ou gasolina por exemplo, e tem seu início a partir de uma energia de ativação (também conhecida como ignição) (Helmenstine, 2016).

As substâncias emitidas durante a reação química de fogo liberam luz e criam uma estrutura de chama que pode ser dividida em 3 partes: base, chamas e fumaça, conforme mostra a Figura 2.1. A base da estrutura de chama é uma porção invisível resultante da mistura do oxigênio com o vapor do elemento combustível. A região das chamas (parte mais visível do fogo) é a região onde ocorrem a maior parte das reações químicas, ou seja, é onde as moléculas se colidem como reação do processo de combustão. E acima da região de combustão, está a região de fumaça que é composta basicamente de gás carbônico, vapor de água ou, dependendo do material de queima, elementos mais nocivos como o monóxido de carbono (CO) ou o dióxido sulfúrico (SO_2) (Helmenstine, 2016).

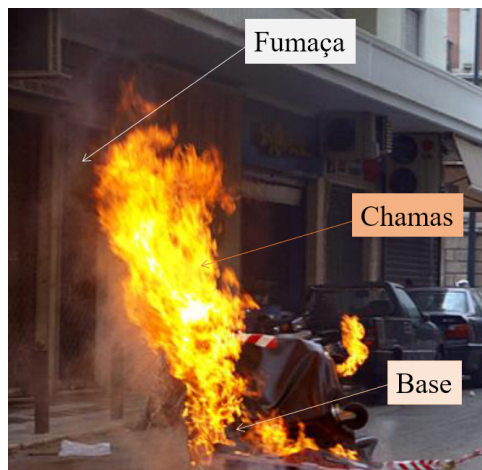


Figura 2.1: Estrutura do Fogo.

Conforme explica (Oliveira, 2005), para a existência de fogo são necessários quatro elementos: calor, comburente (oxigênio), combustível e a reação em cadeia, conforme Figura 2.2. O calor é uma energia na qual o fogo se propaga, pode ser uma faísca, ou o superaquecimento de uma máquina energizada. O Comburente é o elemento que se combina com o vapor do elemento combustível. O Combustível é qualquer material que queime (madeira, gasolina, papel). Já a reação em cadeia é o produto da combustão dos combustíveis que geram mais gases ou vapores combustíveis, criando um ciclo. A falta de qualquer um dos quatro elementos do tetraedro de fogo inibe a existência do mesmo. Durante um incêndio, a combustão continuará existindo até o final do consumo do combustível, ou até que o combustível se esfrie a temperatura abaixo da necessária para combustão (diminuição do calor), ou a reação em cadeia seja interrompida.

2.1.1 Queimadas

No trabalho de (Brushlinsky et al., 2015), houve um levantamento estatístico em 32 países (entre eles Reino Unido, Rússia, Japão e Estados Unidos), realizando um comparativo entre as ocorrências de fogo, tipos de incêndios, e os prejuízos diretos e indiretos causados pelo fogo. Este estudo subdivide os incidentes com fogo em seis tipos de incêndios: estrutural (prediais/urbanos), rodoviários, florestais, áreas agrícolas, áreas de lixões e outros. Nos países estudados, as ocorrências estão distribuídas da seguinte forma: incêndios estruturais (40%), outros (18%), rodoviário (14%), agricultura (14%), lixo (11%) e florestais (3%).

Apesar de ter em média pouca incidência nos países estudados (3%), incêndios florestais, por afetar grandes áreas territoriais, segundo Alkhatib (2014), produzem 30% do CO_2 da atmosfera, além disso, o fogo ainda provoca a destruição de sistemas ecológicos e extinção de espécies raras. O fogo acomete florestas, principalmente por estas serem ambientes isolados e dotados de abundante material combustível, como: madeira e folhas secas. Conforme

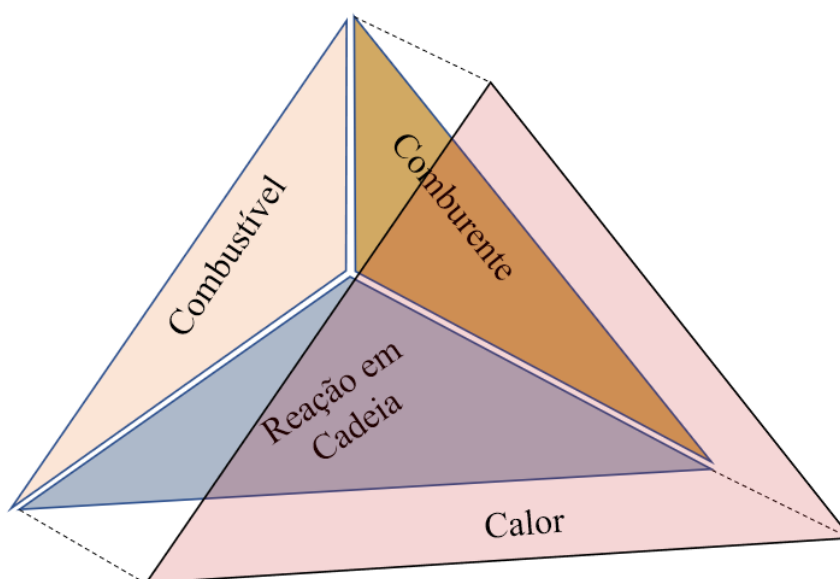


Figura 2.2: Tetraedro do Fogo.

cita Collumeau (2011), as queimadas florestais acontecem principalmente nos períodos de maior seca.

Dados do Instituto Nacional de Pesquisas Espaciais (INPE, 2019) mostram que os focos de queimadas no Brasil, predominantemente ocorrem no segundo semestre do ano, tendo um pico de focos de incêndios no mês de setembro (como mostrado na Figura 2.3). Dados da série histórica mostram que o Brasil possui em média 175.870 focos de incêndio por ano, e as variações deste valor devem-se a condições climáticas e fenômenos naturais como o El-Niño, por exemplo. Em 2015, (ano de passagem do El-Niño) houve o registro de 236 mil focos de incêndio, 53 mil focos de incêndio a mais do que em 2014.

Estes dados mostram a influência das mudanças climáticas sobre os focos de incêndios no país. Este comportamento também se reflete no mundo. Nos EUA, dados da Associação Nacional de Proteção de Fogo dos Estados Unidos, fornecidos em (Evarts, 2018) mostram que em 2017 houve mais de 1,3 milhões de incidentes com fogo, que causaram 3400 mortes de civis e U\$ 23 bilhões em prejuízos.

Já no Brasil, os prejuízos gerados pelas queimadas também são grandes. Um estudo datado de 2002 (Vera et al., 2002), realizou estimativa de prejuízos das queimadas entre 1995 e 1999 em três frentes: perdas agropecuárias, perdas florestais e gastos com saúde humana relacionados a problemas respiratórios. Nesse período, as perdas agropecuárias chegaram a uma média de U\$ 67 milhões de dólares por ano. Em relação a perdas florestais, no ano de 1998 (ano do El Niño), os valores estimados de perda ultrapassam U\$ 13 milhões. No que tange à saúde, em média, 6% dos atendimentos hospitalares das regiões afetadas tinham relação com problemas respiratórios, gerando um custo aproximado anual de U\$ 7,4 milhões.

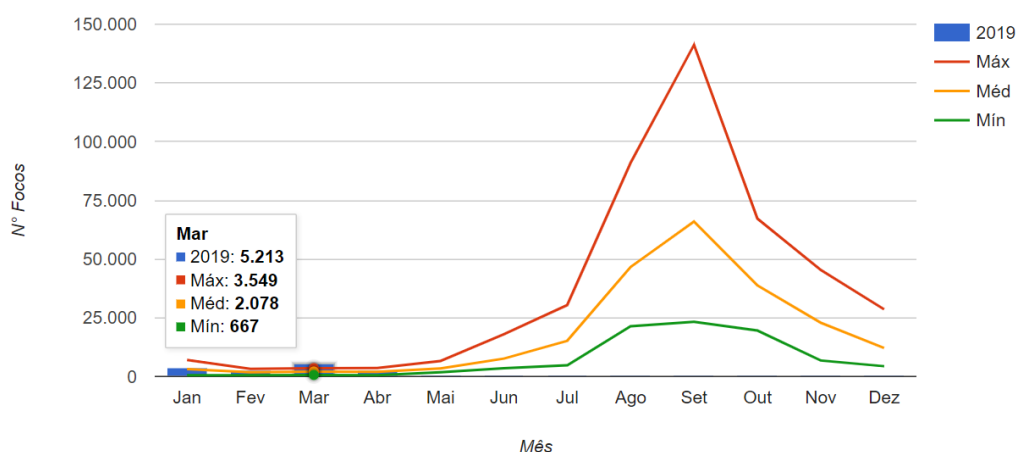


Figura 2.3: Frequência dos focos de incêndio no Brasil por mês em 2019 comparado com a média histórica de 1998 até 2018, Fonte: (INPE, 2019).

2.1.2 Controle do Fogo

Segundo Alkhatib (2014), para realizar o controle de fogo, as autoridades contam com equipes de brigadas de incêndio que recorrem a diversas técnicas. Dentre as técnicas citadas estão: utilização de fogo controlado, estimativa da possibilidade de fogo por meio das condições climáticas em conjunto com as condições dos combustíveis existentes, torres de controle, detecção ótica de fumaça, detectores de iluminação, câmeras infravermelhas, drones, caminhões de água, chamadas móveis (feitas a partir de celulares) e treinamentos sobre prevenção e atuação nos casos de incêndios. Em relação aos sistemas de monitoramento, estes são subdivididos em dois grupos: (1) Denúncias de ameaças de fogo e (2) Sistemas de detecção operacionais. No que tange ao grupo 2, este abrange: torres de controle, monitoramento aéreo (satélites, drones), detectores eletrônicos de iluminação e sistemas de detecção automática.

A Tabela 2.1 mostra um comparativo entre os tipos de sistemas de monitoramento e sua relação com custo, eficiência, falsos alarmes, localização, tempo de detecção, comportamento do fogo. Os autores da Tabela 2.1 classificam o custo do tipo de monitoramento utilizando 4 parâmetros: Baixo, Médio, Elevado e Muito Elevado. Nota-se que o autor considera o monitoramento humano (manual) como sendo considerado de baixo custo, porém com pouca eficácia. Em contrapartida, sensores sem fio demonstram ter o melhor custo benefício quanta à eficácia e ao custo, pois possuem alta eficiência e curto espaço de tempo para detecção porém, tais sensores devem estar próximos a áreas do início do incêndio, o que pode ser inviável quando há a necessidade de monitoramento de grandes áreas territoriais. Em relação a sensores ópticos, É importante observar na Tabela 2.1 a comparação entre os sensores sem fio, pois para os autores a eficiência e acurácia dos sensores ópticos são medianos e o tempo para identificação dos focos de incêndio é longo, ou seja, é mais demorado comparado a sensores sem fio. Isso mostra a necessidade de evolução dos sensores ópticos e, tal evolução pode ser possível utilizando técnicas de processamento digital de imagens aliadas à aprendizagem de máquina. Vale ressaltar que na



Figura 2.4: Gastos relacionados a incêndios – Fonte: (Brushlinsky et al., 2015).

Tabela 2.1 a coluna “Câmera e sensores ópticos” inclui os seguintes equipamentos: câmeras de infravermelho, espectrômetros, câmeras comuns e sensores a laser de fumaça.

Tabela 2.1: Comparativo entre diversos tipos de monitoramento de fogo – Fonte: (Alkhatib, 2014).

Comparação	Observação Humana	Satélite	Câmeras e Sensores ópticos	Sensores sem fio
Custo	Baixo	Muito elevado	Elevado	Médio
Eficiência e Praticidade	Baixo	Baixo	Médio	Alto
Repetição de falsos alarmes	Baixo	Baixo	Médio	Médio
Acurácia de localização	Baixo	Médio	Médio	Alto
Tempo de detecção	Longo	Muito Longo	Longo	Curto
Informações sobre Comportamento do Fogo	-	Sim	-	Sim

No relatório do CTIF (Comitê Técnico Internacional de combate ao Fogo -*Comité Technique International de prevention et d’extinction de Feu*) presente em (Brushlinsky et al., 2015), as chamadas de emergência relacionadas a fogo correspondem a 5,5% das chamadas; porém, em cidades mais secas como Atenas, na Grécia, os chamados chegam a 36%. Os gastos relacionados a incêndios nos países estudados são subdivididos conforme ilustra o gráfico na Figura 2.4. Vale destacar que 22% dos gastos relacionados a incidentes com fogo são de perdas (perdas diretas e indiretas), 55% da verba é destinada para prevenção e seguro e o restante (27%) destinado para os serviços de combate ao incêndio (Corpo de Bombeiros, por exemplo).

No Brasil, o monitoramento de incêndios florestais é realizado pelo IBAMA em conjunto com o INPE. O IBAMA é o responsável por combater os focos de incêndio, enquanto que o INPE disponibiliza os dados dos satélites para este fim. Existem dois tipos principais de satélites utilizados pelo INPE: os satélites polares e os satélites geostacionários. Os satélites

polares circundam a Terra entre os polos e acompanham o movimento do sol. Já os satélites geostacionários orbitam a linha do equador, juntamente com o movimento da Terra. Os satélites utilizados pelo INPE para identificação de queimadas são dotados de sensores que capturam dados da faixa termal de $4 \mu\text{m}$ e as imagens capturadas variam de acordo com os tipos de sensores ópticos utilizados. Satélites polares são dotados de sensores do tipo AVHRR (*Advanced Very High Resolution Radiometer*) e MODIS (*Moderate Resolution Imaging Spectroradiometer*) e estes geram dois conjuntos de imagens por dia. Em relação aos satélites geostacionários, o INPE utiliza os sensores do tipo GOES (*Geostationary Operational Environmental Satellite*) e MSG (*Meteosat of Second Generation*) que realizam a coleta de imagens pelo menos uma vez por hora, sendo o primeiro controlado pela NASA (*National Aeronautics and Space Administration* - Agência Nacional de Administração Aeronáutica e do Espaço) e o Segundo pela ESA (*European Space Agency* - Agência Espacial Europeia). O INPE coleta e processa cerca de 200 imagens por dia e as informações obtidas pelos satélites são diversas, como: temperatura, localização, risco de fogo, indicativo de fumaça (INPE, 2018).

No Brasil, ainda podemos contar com os brigadistas de incêndio que realizam o controle de incêndio em áreas rurais e urbanas. Existem basicamente três categorias de bombeiros atuantes no país: Bombeiros Militares - ligados ao governo, Bombeiros Voluntários e os Bombeiros Civis - não são profissionais de carreira, mas passam por treinamento de brigadas de incêndio para atuar diretamente com sistemas de evacuação e prevenção de acidentes de fogo.

A partir dos dados expostos, conclui-se que a necessidade de prevenção e combate a incêndios é fundamental para preservação de vidas e redução de gastos. Atualmente o uso de câmeras e sensores ópticos apresenta alto custo de implantação e o tempo de resposta ainda não é ideal, porém com o avanço da tecnologia das câmeras de vigilância, aumento na capacidade de processamento dos equipamentos e a progressiva diminuição no custo desses equipamentos, a utilização de recursos computacionais aliado ao processamento digital de imagens pode se tornar uma alternativa de baixo custo e com alta eficiência para minimizar os danos causados pelas queimadas.

2.2 Processamento Digital de Imagens

Processamento Digital de Imagens - PDI - consiste na análise dos dados provenientes de imagens que podem ser capturadas por diversos tipos de sensores. Após o processo de captura, a imagem precisa ser representada de modo a possibilitar o tratamento computacional (Eustáquio et al., 2001). Além de mecanismos de representação da imagem, faz parte do PDI a realização de operações que destaquem características de interesse na imagem, de modo a facilitar o processo de detecção e identificação de objetos. Dentre as operações para este fim, podemos destacar: realce, suavização, extração de características de bordas, texturas e vizinhanças.

Nesse sentido, o presente trabalho utilizou de técnicas de processamento digital com o objetivo de realizar destaque nas regiões de fogo em imagens digitalizadas. Como o fogo é um

elemento instável e que não possui formas regulares, as técnicas de processamento digital de imagens utilizadas foram baseadas principalmente em relação às características de coloração dos píxeis por meio de transformações nos espaços de cor, assim como a utilização de técnicas de análise de vizinhança dos píxeis utilizando cálculos de entropia de regiões previamente selecionadas.

2.2.1 Transformações para índices de cores

Manipular canais de cores e criar índices que destaquem determinadas colorações pode ser uma vantagem quando há a necessidade de destacar objetos com cores características. Dessa forma, para que fosse facilitada a localização de elementos de fogo, estudaram-se algumas destas variações de índices de cores para as seguintes cores: vermelho, amarelo e marrom, cores comumente localizadas nas chamas de imagens de incêndio.

Os índices de vermelho – *Redness* – são utilizados por diversas aplicações, como: detecção de olhos vermelhos, rostos e cabelos, doenças e pestes na agricultura, análise de solo e avaliações da qualidade de uma fruta. Ao realizar uma análise das características gerais dos índices de vermelho foi constatado que as transformações nos canais de cor realizadas nos trabalhos relacionados à remoção de olhos vermelho destacariam as regiões de fogo de coloração avermelhada, razão pela qual selecionamos três trabalhos: Gasparini and Schettini (2005); Held (2002); Smolka et al. (2003). Nas equações de índices de Vermelho, temos que R representa o o valor do canal Vermelho do sistema RGB em um pixel da imagem, assim como G representa o Verde e o B representa o azul. A Equação 2.1 representa o índice Rn_S proposto por (Smolka et al., 2003). A Equação 2.2, representa o índice Rn_G proposto por (Gasparini and Schettini, 2005) e a Equação 2.3 é o índice Rn_H proposto por (Held, 2002).

$$Rn_S = \frac{R - \max\{G, B\}^2}{R} \quad (2.1)$$

$$Rn_G = \frac{4R + [(G + B) - \min\{R, G, B\} - \max\{R, G, B\}]}{R} \quad (2.2)$$

$$Rn_H = R - \min\{B, G\} \quad (2.3)$$

Os índices de amarelo – *Yellowness* – são bastante utilizados por diversos setores industriais como os têxteis, pinturas e plásticos para avaliar o nível de degradação dos materiais ao serem expostos à luz solar, temperatura e umidade. Ao utilizar os índices de amarelo fica possível realizar o destaque de regiões de fogo cuja coloração mais amarelada é predominante. Assim, realizando um levantamento bibliográfico, foram identificados três índices promissores para o destaque de fogo, são eles: ASTM (2015); Doi (2012); Robinson (1998).

Os autores de (Doi, 2012), na Equação 2.4, propuseram um índice de amarelo para medir a intensidade de amarelamento em folhas. Robinson (1998), na Equação 2.5, propôs um índice de amarelo para medir a intensidade de cores dos objetos a partir do espaço de cor CMYK.

Já a Associação Americana de Teste e Materiais (ASTM, 2015), na Equação 2.6, apresenta um índice de amarelamento para plásticos baseados no espaço de cor XYZ, onde $C_x = 1,3013$ e $C_z = 1,1498$.

$$Yn_d = R + G \quad (2.4)$$

$$Yn_r = \frac{Y}{\sqrt{C^2 + M^2 + Y^2}} \quad (2.5)$$

$$Yn_a = 100 \times \frac{C_x \times X - C_z \times Z}{Y} \quad (2.6)$$

Por fim, os índices de marrom – *Brownness* – são principalmente utilizados para verificar a qualidade de alimentos, como frutas e legumes. Para o presente trabalho, visando destacar regiões mais alaranjadas escuras que aparecem nas imagens de fogo, foram selecionados e testados dois índices de marrom, propostos por (Ricauda et al., 2015) e (Pathare et al., 2013). Ricauda et al. criaram um método baseado no espaço de cor RGB, conforme Equação 2.7, para medir a qualidade de frutas congeladas. Nesta equação, o valor de k pode ser definido dentro do intervalo entre 3 e 4, conforme sugerem os autores da pesquisa. Já Pathare, Opara e Al-Said utilizaram o espaço de cor CIELab e propuseram um índice de marrom, conforme a Equação 2.8, para medir e analisar a qualidade de comidas frescas e processadas.

$$Bn_R = k \times R - G - B \quad (2.7)$$

$$Bn_P = \frac{100}{0,17} \times \frac{(a + 1,75L)a}{5,45L + a + 3,012b} - 0,31 \quad (2.8)$$

2.2.2 Entropia aplicada ao processamento digital de imagens

O conceito de entropia é originário da física e a segunda lei da termodinâmica enunciada por Lord Kelvin (William Thomson) e Rudolf Clausius no final do século XIX; descrevem que os sistemas tendem à desordem e que a entropia se incrementa com o tempo até atingir o valor máximo. Sendo assim, a entropia como uma medida do grau de desordem de um sistema indica que quanto maior seu valor, maior é a desordem do mesmo. Em 1870 Boltzmann propôs a seguinte fórmula para a entropia, onde S é a entropia, k é a constante de Boltzmann e W são os microestados possíveis para um sistema:

$$S = k \times \ln(W) \quad (2.9)$$

Shannon em seu trabalho (Shannon, 1948), adaptou a fórmula de Boltzmann e aplicou a distribuição da probabilidade de um determinado evento acontecer. Já no processamento de imagens, a equação pode ser adaptada para o histograma que vai indicar a probabilidade de um pixel ser de uma determinada cor baseada no conhecimento de todas as cores de uma imagem. Essa adaptação para o canal RGB por exemplo, podemos definir da seguinte forma: sendo $P_c = (P_{c,0}, P_{c,1}, \dots, P_{c,i})$ a distribuição de probabilidade da cor c , onde $c \in \{R, G, B, Gr\}$

(onde, Gr equivale a escala de cinza) e $i = [0...255]$, teremos que a entropia de um canal de uma imagem, pode ser definido por:

$$E = \sum_{i=0}^{255} P_{c,i} \times \log P_{c,i} \quad (2.10)$$

Este conceito foi utilizado em diversos trabalhos de processamento de imagens para identificar valores de entropia para determinadas texturas, como pode ser visto nos trabalhos de (Tsai et al., 2008; Kopriva et al., 2017). Tsai e demais autores propõem uma métrica de transmissão de dados em imagens radiográficas por meio da entropia; e Kopriva aplica medições de entropia em imagens para localizar tumores em pulmões a partir de imagens de tomografia.

2.3 Aprendizagem de Máquina

Segundo Murphy (2012), a aprendizagem de máquina é um conjunto de métodos que possibilitam o reconhecimento de padrão dos dados e então utiliza dos dados descobertos para prever os dados futuros ou realizar tomada de decisão sobre as incertezas. O aprendizado da máquina dependerá do processo de aprendizagem e qual algoritmo será adotado. Em relação ao processo de aprendizagem, ele pode ocorrer de duas formas principais: supervisionado e não supervisionado.

No aprendizado supervisionado, o classificador terá um conjunto de dados já rotulados para serem utilizados como base de treinamento para o classificador. Para que isso seja possível, o classificador passa por um processo de treinamento que consiste em identificar qual a melhor forma de separar os dados de acordo com os rótulos identificados de modo a ocorrer o menor número de erros possível. Formalmente, no aprendizado supervisionado, temos $D = \{(x_i, Y_i)\}_{i=1}^N$, onde D representa os dados de treinamento, x_i os dados de entrada e $Y_i \in \{1, \dots, C\}$ os dados de saída esperados. No modelo de aprendizado supervisionado, existem duas categorias de aprendizagem: classificação ou regressão. Na classificação, os dados a serem classificados são categóricos, ou seja, representam uma característica definida. Já os algoritmos de regressão produzem dados contínuos, pois representam valores/medidas (Murphy, 2012).

Algoritmos de classificação realizam o mapeamento das entradas x para uma classe Y , onde $Y \in \{1, \dots, C\}$, onde C representa o número de classes. Caso $C = 2$, então, temos um classificador binário. Nos casos em que C for maior que 2, temos um classificador multiclases. Na classificação, o objetivo é encontrar uma função de aproximação a partir dos dados de treinamento e que esta função possibilite a generalização do problema para a maioria dos casos ainda não conhecidos. Algoritmos de regressão, são muito parecidos com algoritmos de classificação, a diferença principal está na saída destes algoritmos que produzem variáveis contínuas como saída, a Figura 2.5 mostra as regressões Linear e Polinomial para o mesmo conjunto de dados (Murphy, 2012).

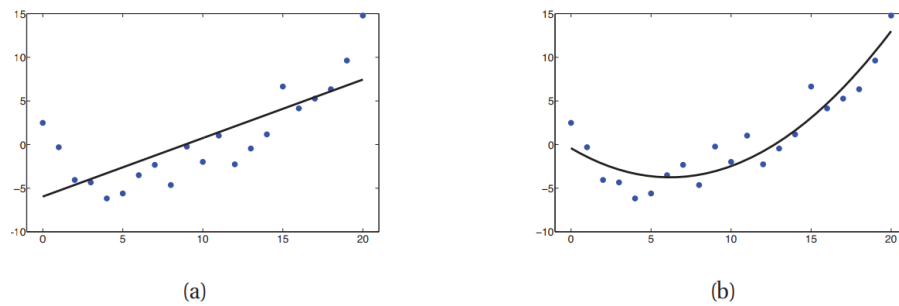


Figura 2.5: Exemplo de tipos de regressão para os mesmos dados (a) Regressão Linear e (b) Regressão Polinomial – Fonte: (Murphy, 2012).

Já o modelo de aprendizagem não supervisionada procura identificar padrões nos dados fornecidos, sem rótulos, e então realizar a separação desses dados de acordo com a proximidade entre os padrões encontrados. Formalmente, temos que o conjunto de treinamento para o aprendizado não supervisionado é dado por $D = \{(x_i)\}_{i=1}^N$. Nesse sentido, temos que os modelos de aprendizagem não supervisionados trabalham com descoberta de conhecimento e podem ser subdivididos em três categorias principais: descoberta de clusterização (consiste em separar os dados similares em grupos), descoberta de fatores latentes e descoberta de estrutura de grafos (realiza a identificação dos padrões a partir da interconexão entre as características do vetor de entrada x) (Batista, 2003; Murphy, 2012).

Nas seções subsequentes serão abordados alguns dos algoritmos de aprendizagem de máquina utilizados com o foco em detecção de incêndios. Os algoritmos de aprendizagem de máquina utilizados foram: SVM, Random Forest, Adaboost e as Redes Neurais Artificiais.

2.3.1 SVM – *Support Vector Machine*

O algoritmo SVM foi criado por Vapnik e sua equipe de pesquisa, publicado no trabalho de Schölkopf et al. (1998) e é derivado do Support Vector Network (Cortes and Vapnik, 1995) é um algoritmo que foi inicialmente criado para classificação binária, porém conforme explica (Murphy, 2012), o SVM pode ser extensível para regressão e classificação multi-classes. O processo de estimativa estatística de classificação do SVM é realizado por meio de um processo de aprendizado baseado na técnica da Minimização do Risco Estrutural (SRM – *Structure Risk Minimization*) que consiste em subdividir os riscos em Risco Empírico e Risco Funcional.

O Risco Empírico diz respeito às classificações incorretas obtidas e o Risco Funcional mostra as probabilidades de erros de generalização do classificador. O princípio da indução da SRM possui dois objetivos: controlar o risco empírico no conjunto de treinamento e controlar a capacidade da função de decisão usada para obter esse valor de risco. A função de decisão do SVM linear é descrita por um vetor de peso p , um limiar l e um padrão de saída x , onde:

$$f(x) = \text{sign}(p \times x + l) \quad (2.11)$$

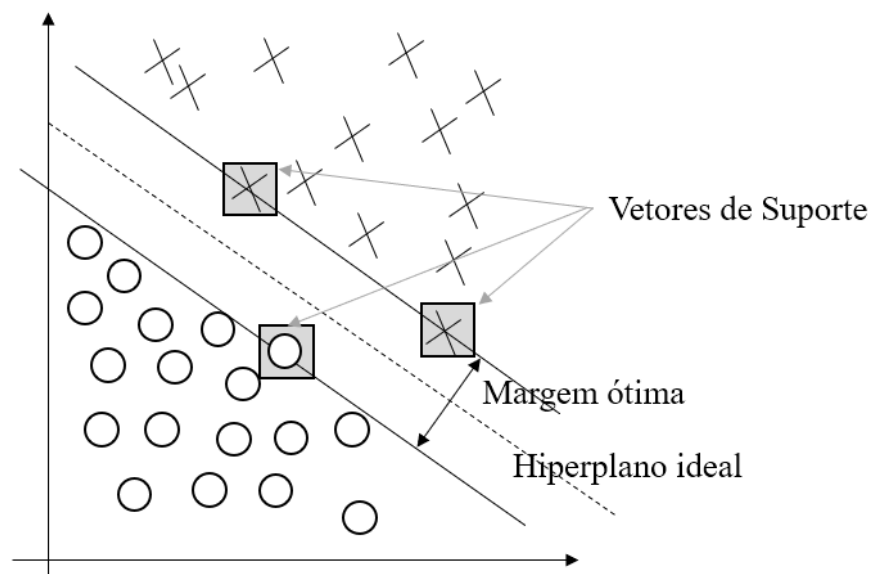


Figura 2.6: Exemplo de um problema separável de duas classes e os vetores de suporte mostrando a distância ótima entre os objetos de classes distintas – Fonte: Cortes and Vapnik (1995).

Uma função Kernel para SVM realiza a transformação não linear entre o espaço original para um de maior dimensão, onde é mais provável que o problema seja linearmente separável (Schölkopf et al., 1998). A Figura 2.6 mostra duas classes representadas em um plano 2D e os vetores de suporte que auxiliam na separação do problema. A Equação 2.12 mostra a função de decisão do SVM, onde α_i e b são parâmetros para o treinamento; x_i é o vetor de características; y_i são os rótulos e $K(x, x_i)$ representa o kernel. O algoritmo SVM pode ser executado com diferentes tipos de Kernel, dentre eles, pode-se citar: Linear, Polinomial, Gaussiano, Sigmoide, Principal Component Analysis (PCA) e Produto Escalar. O custo computacional do SVM depende também do tipo de Kernel selecionado, e a escolha do Kernel é determinante para definir a acurácia do classificador (Cortes and Vapnik, 1995; Schölkopf et al., 1998; Murphy, 2012).

$$f(x) = \sum_i \alpha_i y_i K(x, x_i) + b \quad (2.12)$$

2.3.2 *Random Forest*

Random Forest, ou Floresta Aleatória, é um método de classificação supervisionado baseado na combinação (*ensembles*) de árvores de decisão. Esta combinação depende dos valores do vetor randômico distribuído igualmente por todas as árvores da floresta (Breiman, 2001).

Cada árvore de decisão da floresta é criada por meio de um processo chamado *bagging* que consiste em subdividir o conjunto de treino de forma randômica, mas igualmente, em N conjuntos e para cada um dos conjuntos, criar uma árvore de decisão. Para cada árvore criada

são calculados os erros de generalização baseados na margem de acerto do classificador. A partir dos resultados dos erros de generalização, é possível avaliar como está a qualidade da classificação e assim identificar árvores de decisão com os melhores desempenhos para compor a Floresta (Breiman, 2001).

Segundo (Breiman, 2001), uma definição formal para uma Floresta Aleatória é a seguinte: “Uma Floresta Aleatória é um classificador que consiste em uma coleção de classificadores estruturados em árvores $\{h(x, \theta_k), k = 1, \dots\}$, onde, $\{\theta_k\}$ é independente e distribuído de forma randômica e idêntica em vetores, onde cada árvore de decisão é responsável por indicar o valor classe para a entrada x , que no final é classificada de acordo com a classe que obteve mais votos dentre as árvores da floresta.”

2.3.3 Adaboost

Segundo (Bauer et al., 1999), o Algoritmo de Boosting foi criado por Schapire em 1990 como um método para aperfeiçoar a performance de algoritmos de aprendizado com fraco desempenho. Freund e Schapire, em 1995, a partir do Boosting, criaram o AdaBoost (Adaptative Boosting). Um classificador fraco pode ser definido como um classificador em que a taxa de acertos na classificação é muito baixa; em contrapartida, um classificador forte é aquele em que o número de classificações corretas é alto. A ideia principal do algoritmo de Boosting consiste em iterativamente testar hipóteses diferentes até que a resposta mais próxima da classificação ideal seja alcançada, dessa forma transformando processos de classificação com fraco aprendizado em classificadores mais fortes.

Já o Algoritmo de AdaBoost, utiliza a técnica de gerar um conjunto de classificadores e então realizar o processo de votação entre os que obtiverem melhores resultados, escolhendo assim o melhor classificador.

O objetivo deste algoritmo é minimizar os erros sobre diferentes distribuições de entrada. Dado um inteiro T , que especifique o número de amostras e seus respectivos pesos $S_1, S_2, S_3, \dots, S_T$, então, T classificadores são gerados $C_1, C_2, C_3, \dots, C_T$. Um Classificador final Cl é formado usando o sistema de votos e pesos, onde o peso de cada classificador depende de sua performance na base de treinamento.

2.4 Redes Neurais Convolucionais

2.4.1 Redes Neurais

Segundo (Barreto, 2002), redes neurais artificiais consistem em um conjunto de estruturas com características similares aos neurônios biológicos e são chamados de neurônios artificiais (perceptrons). Um neurônio artificial pode possuir N entradas de dados, processá-las por meio de uma função de ativação e retornar uma saída. Uma Rede Neural é na verdade um conjunto

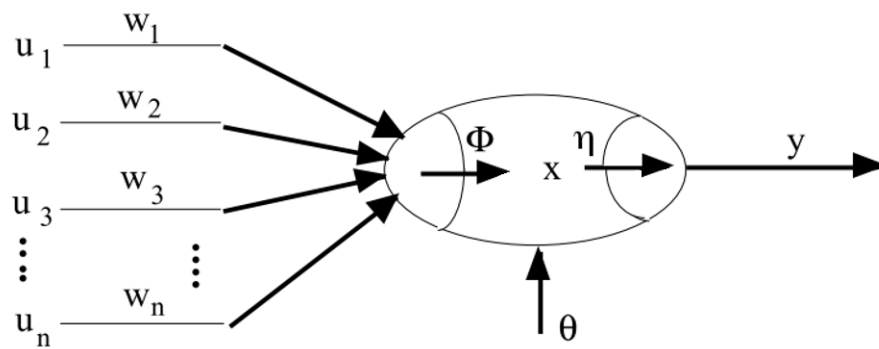


Figura 2.7: Neurônio Artificial (perceptron) – Fonte: (Barreto, 2002).

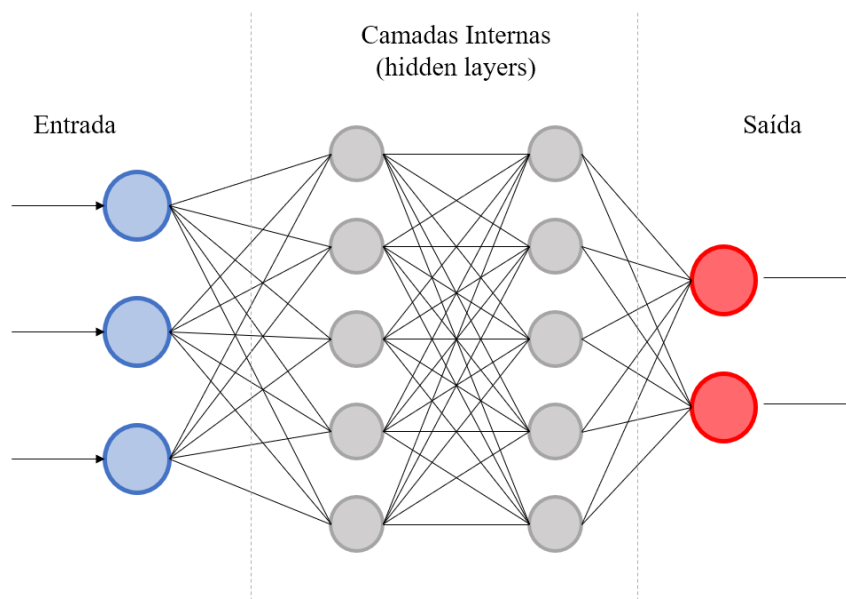


Figura 2.8: Representação de uma Rede Neural Artificial e suas camadas internas.

de neurônios artificiais interligados com o objetivo de resolver um problema. A Figura 2.7 a seguir mostra a estrutura de um neurônio artificial, indicando as entradas u_i e w_i combinadas pela função ϕ (soma ou produto) e ativadas pela função η , produzindo a saída y . Um valor auxiliar θ é uma condicional de polarização, que é aplicada à função η para determinar a saída y .

Combinando diversos neurônios, ligando suas saídas a entradas de outros neurônios, é então criada uma rede neural. O treinamento de uma rede neural ocorre quando um conjunto de treino é apresentado à rede, para que os pesos dos neurônios sejam recalculados a partir das entradas e saídas esperadas do conjunto de treino. Em uma rede neural, conforme Figura 2.8, existem os neurônios da camada de entrada, que recebem os dados para serem processados pela rede, e os neurônios da camada de saída, que apresentam os resultados da rede processados pelos neurônios das camadas internas (hidden layers).

No processo de aprendizagem chamado de retropropagação (backpropagation) a aprendizagem da rede neural ocorre de forma supervisionada, onde são informadas as saídas esperadas

para um determinado conjunto de entrada. Dessa forma, os pesos das camadas internas são atualizados até que o gradiente do erro seja minimizado até um valor de tolerância pré-estabelecido. Após o treinamento da rede, o processo de reconhecimento de padrões de uma rede neural consiste em: para cada instância, executar as funções existentes nas multicamadas, logo a complexidade computacional de uma rede neural fica dependente da função interna dos neurônios artificiais (Barreto, 2002).

2.4.2 Convolução em Redes Neurais

Em 1995, Yan Lecun e Yoshua Bengio apresentaram no trabalho LeCun and Bengio (1995) a utilização de redes neurais convolucionais – RNC – com o foco em reconhecimento de fala, imagens e séries temporais. O processo de convolução em imagens consiste em realizar convoluções a partir de um kernel de tamanho n , onde este kernel desliza sobre os píxeis da imagem e uma nova imagem é criada a partir dos cálculos de convolução dos píxeis e seus vizinhos determinados pelo tamanho do kernel. Formalmente, a convolução discreta de uma imagem I com kernel K resulta na imagem R , conforme mostra a Equação 2.13 (Facon, 2005).

$$R(x, y) = \left(\frac{1}{MN}\right) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n)K(x - m, y - n) \quad (2.13)$$

Os cálculos de convolução a partir dos kernels, destacam características visíveis das imagens como bordas e pontos de fronteira. Dessa forma, ao final do processo de convolução em uma camada, um mapa de características por camada é o resultado deste processo (LeCun and Bengio, 1995). Este mapa de características pode ser utilizado como entrada para a próxima camada da rede, que gera outro mapa de características e este processo pode ser repetido por diversas camadas.

2.4.3 Função de Ativação e Pooling

Em um neurônio artificial, a função de ativação é de suma importância para determinar como será a saída do mesmo. Esta função determinará a saída de cada um dos neurônios presentes nas camadas da rede. Uma função de ativação muito utilizada em Redes Neurais é a função tangente hiperbólica. Além desta função, existem ainda as funções Sigmoid e ReLU (Rectified Linear Unit), representadas pelas equações 2.14 e 2.15 respectivamente.

$$f(x) = \frac{1}{(1 + e^{-xa})} \quad (2.14)$$

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.15)$$

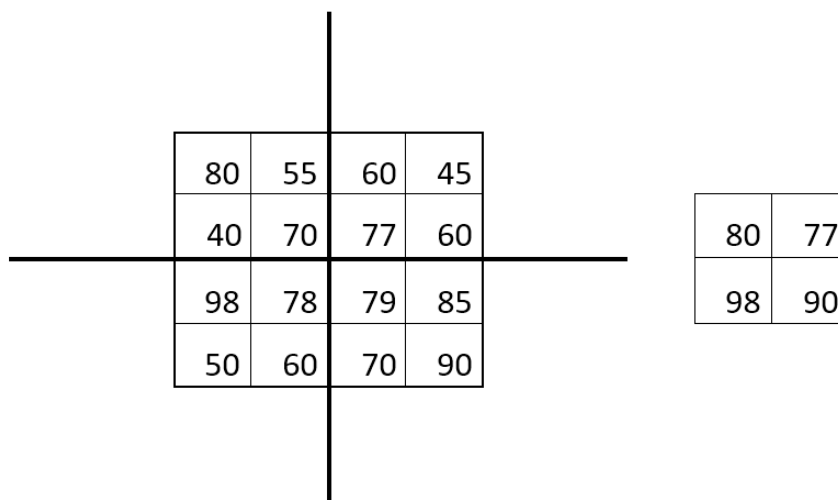


Figura 2.9: Exemplo de maxpooling aplicado em uma porção de uma imagem com máscara de tamanho 2×2 .

No entanto, quando se trata de RNC multicamadas, utilizar as funções de tangente hiperbólica e sigmoid são custosas computacionalmente. No trabalho de (Russakovsky et al., 2015) os autores mostraram que para uma rede com 4 camadas convolucionais a função de ativação ReLU converge à taxa de erro seis vezes mais rápida que a função *tangh*. Além disso, quando combinadas, as funções ReLU aproximam-se das funções sigmodais (Rocha, 2015). Sendo assim, quando se expande o número de camadas convolucionais, a utilização da função de ativação ReLU mostra-se mais eficaz. Além da função de ativação, o processo de pooling é fundamental para o aprendizado da rede. O objetivo desta etapa é diminuir a sensibilidade da rede neural para pequenas alterações nas entradas. Em consequência, utilizar o pooling permite que haja diminuição de redundâncias e eliminação de ruídos na rede neural. A função de pooling mais utilizada nas redes neurais convolucionais é o *maxpooling*, que consiste em selecionar o maior valor para um atributo em uma vizinhança de tamanho determinado pelo tamanho de uma máscara. A Figura 2.9 mostra como é realizado o maxpooling para uma máscara 2×2 em uma porção de uma imagem.

Após sucessivas operações de convolução e pooling nas diversas camadas da rede neural convolucional, um mapa de características é criado. Para classificar estes dados, as camadas totalmente conectadas (*Fully Connected Layers*) realizam combinações específicas destas características que são capazes de classificar os dados de entrada de acordo com as saídas possíveis (Frizzi et al., 2016).

2.4.4 Técnicas para redução do sobre-ajuste (*overfitting*)

Um dos problemas no processo de treinamento de um classificador é a possibilidade de sobre-ajuste, ou *overfitting*. Este problema ocorre quando o classificador aprende características específicas sobre a base de treinamento e acaba não generalizando o aprendizado para a base

de validação, de tal forma que a qualidade da classificação na base de treino destoa de forma significativa em relação à base de validação.

Para reduzir este problema nas redes neurais convolucionais, existem algumas técnicas que podem ser aplicadas tanto na base quanto no processo de aprendizado da rede. O processo de *Dropout* proposto por Hinton em (Hinton et al., 2012) consiste em desconsiderar as partes da rede que geram resultados inconclusivos durante o processo de predição. Ao desconsiderar partes da rede, o autor chegou à conclusão que há uma diminuição na criação de relações complexas entre os elementos do treino o que consequentemente diminuiria o sobre-ajuste.

Para treinar uma rede neural convolucional é necessário que a base de treinamento represente o problema a ser classificado de maneira completa mostrando o máximo de variabilidades. Nesse sentido, treinar uma rede neural para detecção de objetos requer uma base de treino de tamanho significativo. Para auxiliar neste processo é possível aumentar a base de imagens de forma artificial, conforme mostra (Krizhevsky et al., 2012). A este processo é dado o nome de *Data Augmentation* que consiste em ampliar a base de treino a partir da criação de novas imagens resultantes de operações como: rotações, mudanças em escala, translações e transformações nos espaços de cor.

2.4.5 Detecção de Objetos e Redes Neurais Convolucionais

A identificação de objetos e a classificação de imagens é um desafio complexo para máquinas. Por esta razão, o projeto ImageNet lançou em 2010 uma competição para classificar uma base de imagens com mais de 1000 classes diferentes distribuídas em mais de 10 milhões de imagens (the ImageNet Large Scale Visual Recognition Challenge – ILSVRC (Russakovsky et al., 2015)). Em 2012 Krizhevsky et al. (2012) propuseram uma arquitetura de RNC que foi capaz de vencer o desafio de classificação do ILSVRC do ano de 2012 com taxa de erro de 15,3% contra 26,2% do segundo colocado. Este resultado colocou os holofotes sobre as Redes Neurais e encorajou outros cientistas a criar e testar novos modelos arquiteturais de Redes Neurais Convolucionais. No último ILSVRC, em 2017, a equipe WMW venceu a competição de classificação com cerca de 2,25% de taxa de erro realizando modificações nas camadas de convolução do tipo inception (Szegedy et al., 2015) e camadas residuais da resnet (He et al., 2015b).

A arquitetura proposta por Krizhevsky, Sutskever e Hinton em (Krizhevsky et al., 2012), também conhecida como AlexNet, continha 5 camadas de convolução, também nomeadas de camadas escondidas (*hidden layers*) e 3 camadas totalmente ligadas (*full connected layers*) como pode ser visualizada na Figura 2.10. Para otimizar o treinamento e diminuir a possibilidade de sobre-ajuste (*overfitting*) os autores utilizaram as técnicas de *dropout* nas duas primeiras camadas totalmente conectadas (*dense layers* na Figura 2.10) e *Data Augmentation*.

Simonyan and Zisserman em (Simonyan and Zisserman, 2014) propuseram um aumento na profundidade da rede e a redução dos filtros de convolução (3×3) em relação aos apresentados por (Krizhevsky et al., 2012) e a variação da profundidade da rede entre 11 e 19 camadas, que

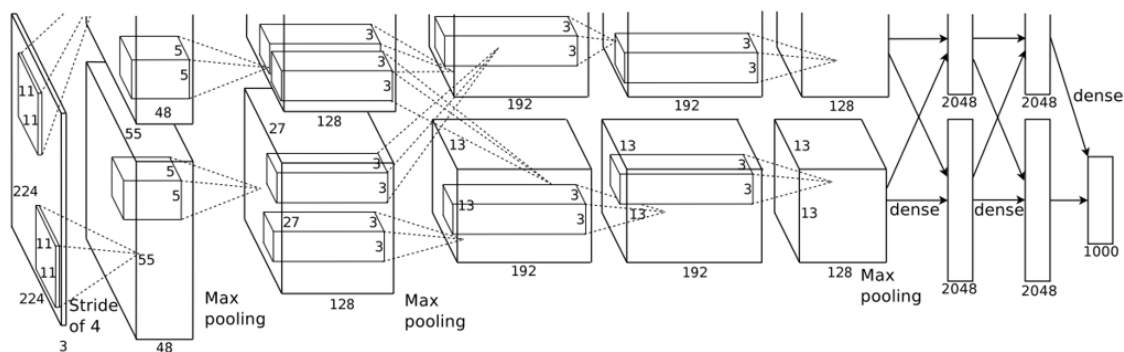


Figura 2.10: Arquitetura da AlexNet - Fonte: (Krizhevsky et al., 2012).

pode ser visualizada na Figura 2.11. Nesta Figura, visualizamos as variações de rede com duas camadas totalmente conectadas com 4096 neurônios cada e uma última camada utilizada para identificar a classe do objeto que está sendo identificado dentre outras 1000 possíveis. A coluna D da Figura 2.11 ilustra a estrutura da rede VGG16 que apresenta desempenho semelhante ao da rede da coluna E (VGG19) apesar de conter menos parâmetros. Esta arquitetura foi a vencedora do desafio de classificação da ILSVRC em 2014.

Em 2015, pesquisadores da Microsoft (He et al., 2015a) introduziram a ideia do aprendizado residual nas camadas de uma RNC. Um dos problemas da inserção de muitas camadas da RNC estava na saturação da acurácia e logo em seguida o início de sua degradação, além do aumento da taxa de erros durante a fase de treinamento. Para resolver estes problemas os desenvolvedores desta arquitetura inseriram a ideia de um bloco residual, como pode ser visualizado na Figura 2.12. O bloco residual permite que o valor identidade de entrada (X) nos blocos, se este já estiver com o peso ideal, não necessite passar por outro processo nas camadas não lineares da rede (representados na Figura 2.8 como sendo “Peso da Camada”), o que possibilita a criação de atalhos para a rede.

Incluindo o conceito de blocos residuais, os autores do trabalho criaram uma rede chamada Resnet de 18 até 152 Camadas, como pode ser visualizada na Figura 2.13. Na Figura ainda é possível verificar nos dados entre colchetes a composição dos blocos residuais, assim como o número de blocos residuais que são empilhados para a construção da arquitetura da rede. Os relatórios comparativos deste trabalho apresentaram para a rede Resnet menores taxa de erro comparadas às redes de mesma dimensão sem blocos residuais. Outra informação relevante na Figura 2.13 é o número de operações realizadas por segundo (FLOPs); esta indica que quanto menor este valor, menor é a complexidade da rede. As redes ResNet50, ResNet101 e Resnet152 superaram as taxas de erro das outras redes que competiram no concurso de identificação de objetos do ILSVR em 2015.

Configuração ConvNet					
A	A-LRN	B	C	D	E
11 camadas peso	11 camadas peso	13 camadas peso	16 camadas peso	16 camadas peso	19 camadas peso
entrada (224×224 RGB)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 2.11: Configurações das camadas arquiteturais das redes VGG, variando de 11 a 19 camadas - Fonte: (Simonyan and Zisserman, 2014).

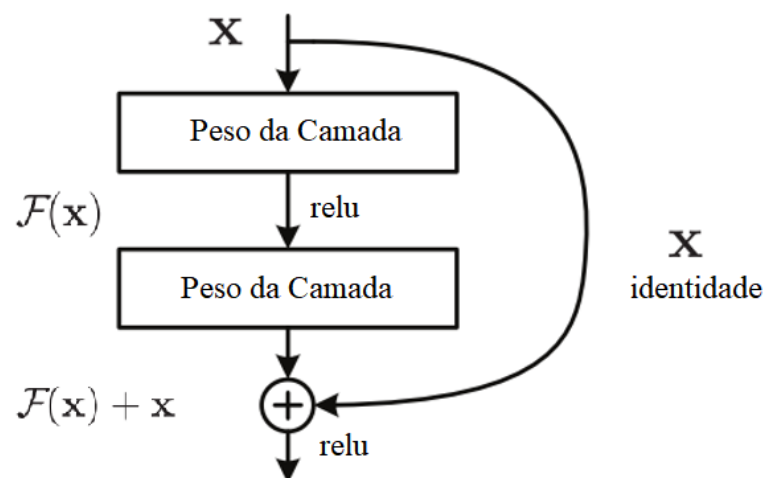


Figura 2.12: Estrutura de um bloco residual de aprendizagem – Fonte: (He et al., 2015a).

Camada	Saída	18-Camadas	34-Camadas	50-Camadas	101-Camadas	152-Camadas
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figura 2.13: Estruturas Arquiteturais das Redes Resnet e medida de operações de números reais por segundo (FLOPs) – Fonte: (He et al., 2015a).

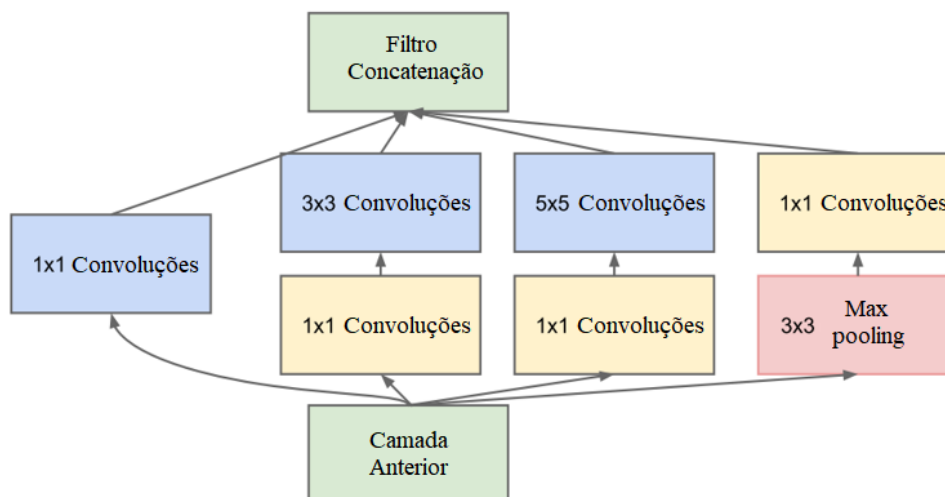


Figura 2.14: Módulo Inception – Fonte: (Szegedy et al., 2015).

Pesquisadores da Google em (Szegedy et al., 2015) introduziram o conceito de módulos Inception com uma RNC de 22 camadas chamado GoogleLeNet. Os autores sugeriram uma arquitetura de Rede Neural que não somente empilha as camadas, mas que também aumente a largura destas camadas utilizando módulos *inception*. Tal ideia possibilitou a melhora na performance da rede mantendo sua eficiência computacional. Na Figura 2.14, é importante destacar a variação do tamanho dos *kernels* entre os diversos filtros de convoluções presentes no módulo *inception*; estas variações permitem a realização de convoluções em escalas menores e com extração de diferentes tipos de características, o que possibilita um maior grau de abstração, principalmente nas camadas superiores (topo da rede) (Szegedy et al., 2015).

Cerca de um ano depois, Chollet (2016) estendeu a ideia do Inception, sugerindo um novo bloco, com camadas ainda mais largas, chamado de Inception Extremo, conforme pode ser visualizado na Figura 2.15. Além disso, a ideia de blocos residuais também foi aplicada na construção da nova rede à qual foi dado o nome de Xception.

A Figura 2.16, mostra as camadas da arquitetura Xception, o sinal de “+” indica os pontos onde são aplicadas as técnicas de informação residual.

Outra arquitetura que merece destaque é a MobileNet, proposta por Howard et al. (2017). A MobileNet é uma rede que foi desenvolvida com o objetivo de reduzir o tempo de detecção e o tamanho da rede em comparação com as redes do estado da arte da época da publicação, a saber: GoogleNet (Inception) Szegedy et al. (2015), VGG de Simonyan and Zisserman (2014) e da Alexnet de Krizhevsky et al. (2012). Para isso, os autores exploraram a utilização de camadas de convoluções de profundidade separáveis, assim como a proposta de um hiperparâmetro que permite a variação do tamanho da rede, bem como o tamanho da entrada da rede. Como resultado, os autores mostram que a MobileNet, cuja arquitetura está ilustrada na Figura 2.17, apresentou melhores resultados em relação à GoogleNet e obteve acurácia 0,9% inferior à rede VGG e que a

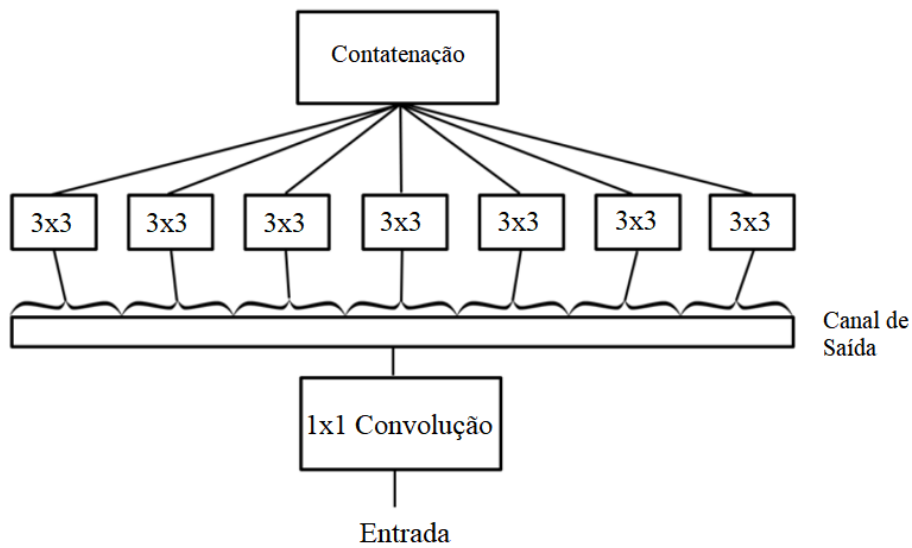


Figura 2.15: Bloco Inception Extremo – Fonte: (Chollet, 2016).

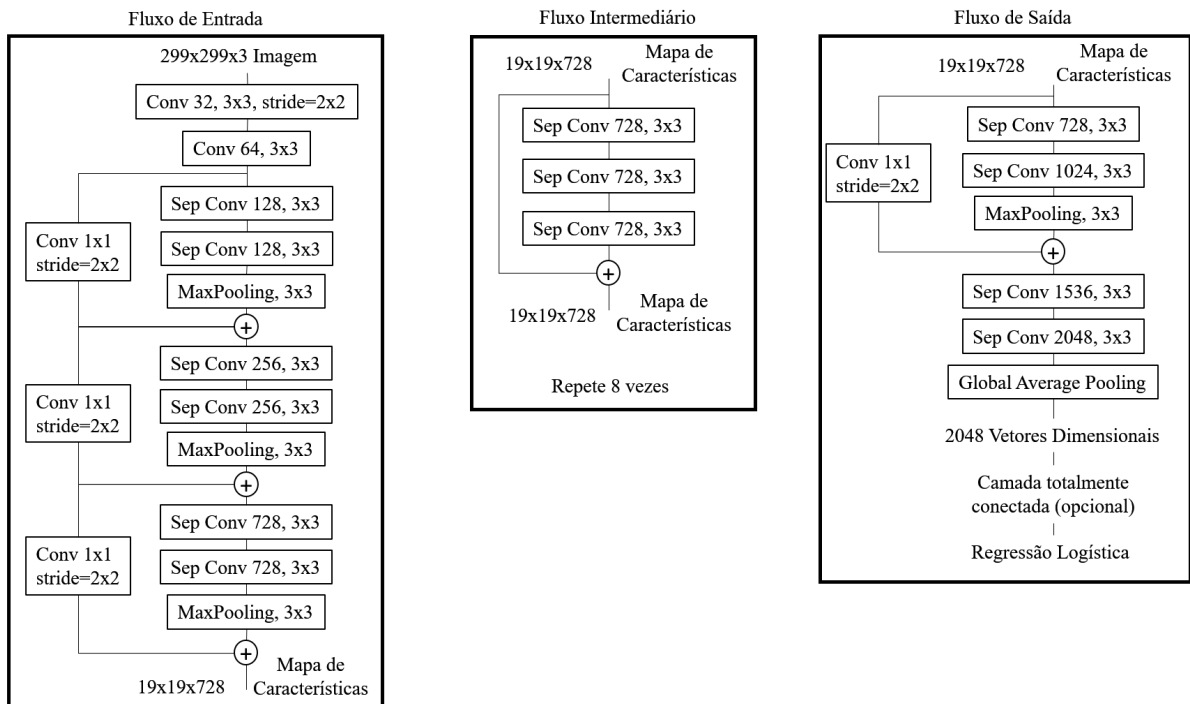


Figura 2.16: Arquitetura da Rede Xception – Fonte: (Chollet, 2016).

Tipo / Passo	Dimensão do Filtro	Dim. de Entrada
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classificador	$1 \times 1 \times 1000$

Figura 2.17: Arquitetura da MobileNet – Fonte: (Howard et al., 2017).

arquitetura aplicada a um fator de redução de 0,5 apresentou acurácia superior em 3% em relação a rede Alexnet.

2.5 Métricas de avaliação

Para o desenvolvimento deste trabalho, dada a necessidade de medição de desempenho de detecção de fogo e fumaça e também da localização do fogo, foram utilizados dois conjuntos de métricas de avaliação: Avaliação de Detecção de Fogo e Fumaça e Avaliação da qualidade de Segmentação. Segundo (Fawcett, 2006), dado um sistema de classificação e uma instância a ser classificada, existem quatro possibilidades de resposta para esta instância: verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo. Estes valores são contabilizados da

		Resultado Esperado	
		Positivo	Negativo
Hipótese	Verdadeiro	Verdadeiro Positivo (TP)	Falso Positivo (FP)
	Falso	Falso Negativo (FN)	Verdadeiro Negativo (TN)
Total Colunas:		P	N

Figura 2.18: Matriz de Confusão. Fonte: (Fawcett, 2006) .

seguinte maneira: caso a instância seja positiva se o classificador a considerar verdadeira, ela será classificada como verdadeira positiva, caso contrário, falso negativo; caso a instancia seja negativa, se classificada corretamente, será verdadeiro negativo, senão falso positivo.

A partir das quatro possibilidades de resposta, pode-se criar uma matriz que confronte os resultados, chamada de matriz de confusão (*Confusion Matrix*). A Figura 2.18 ilustra como é formada uma matriz de confusão.

A partir dessa matriz, diversas métricas foram criadas para medir a qualidade de classificação dos modelos de classificação. Fawcett (2006) descreve as principais métricas para avaliação de classificação, dentre elas, vale destacar as seguintes métricas: taxa de falso positivo (fpr), Equação 2.16, taxa de verdadeiro positivo (tpr), Equação 2.17, taxa de falso negativo (fnr), Equação 2.18, taxa de verdadeiro negativo (tnr), Equação 2.19, precisão, Equação 2.20, revocação, Equação 2.21, $F_1 score$, Equação 2.22, e acurácia, Equação 2.23.

$$fpr = \frac{FP}{N} \quad (2.16)$$

$$tpr = \frac{TP}{P} \quad (2.17)$$

$$fnr = \frac{FN}{P} \quad (2.18)$$

$$tnr = \frac{TN}{N} \quad (2.19)$$

$$Prec = \frac{TP}{TP + FP} \quad (2.20)$$

$$Rev = \frac{TP}{TP + FN} \quad (2.21)$$

$$F_1score = \frac{2 * Prec * Rev}{Prec + Rev} \quad (2.22)$$

$$Acurácia = \frac{TP + TN}{P + N} \quad (2.23)$$

2.5.1 Avaliação de Detecção

Para realizar a avaliação do processo de detecção de fogo e fumaça foram utilizadas as seguintes métricas: Acurácia, F_1score , Perdas (*Log Loss*) e o Erro Absoluto Médio (MAE).

A Função Perda - Log Loss

A função perda irá demonstrar como está a qualidade da classificação a partir das predições e da confiança das predições. Por exemplo, dados dois classificadores $C1$ e $C2$, dada a instância i podendo i pertencer a classe X ou Y . Se $C1$ classifica i como pertencente à classe X com 93% de certeza, e $C2$ classifica i também na classe X , mas com 75% de certeza e na verdade i faz parte da classe Y . Teremos que o resultado da função de perda para $C1$ será maior que a função perda de $C2$, pois esta função penaliza classificadores. A Equação 2.24 mostra como é calculado o valor da perda para problemas binários, sendo que y_i é o valor da instância e p_i corresponde à probabilidade apresentada pelo classificador de y ser da classe 1.

$$\logLoss = \frac{-1}{N} \sum_{i=1}^N y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \quad (2.24)$$

Erro absoluto médio (*Mean Absolute Error – MAE*)

O erro absoluto médio é calculado a partir do valor absoluto da diferença entre a previsão realizada pelo modelo y' e o valor real y e expresso pela Equação 2.25.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i| \quad (2.25)$$

2.5.2 Avaliação de Segmentação

Para realizar a avaliação da qualidade de segmentação foram utilizadas algumas métricas existentes, além da Acurácia, precisão e F_1score ; alguns trabalhos utilizam métricas diferentes

para mensurar a qualidade de segmentação. Entre as métricas, podemos citar: Coeficiente de Correlação de Mathew (MCC), HAF, o erro de classificação (Missclassification Error - ME) e o critério de Martin (MAR).

Coeficiente de Correlação de Mathews (MCC)

No trabalho de (Toulouse et al., 2015), é citada a métrica *Mathews Correlation Coefficient* (MCC), criada por Matthews em 1975, descrita na Equação 2.26.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \quad (2.26)$$

HAF

Descrita por (Hafiane et al., 2007), a métrica HAF constitui-se de dois índices, o primeiro índice é chamado de índice de compatibilidade (M_I) expresso pela Equação 2.27 e o segundo é um índice que leva em consideração a qualidade de segmentação 2.28 e a equação final é dada pela Equação 2.29.

$$M_I = \frac{1}{\#(I_s)} \sum_{j=1}^{NR_s} \frac{\#(R_{i^*}^{GT} \cap R_j^s) \times \#(R_j^s)}{\#(R_{i^*}^{GT} \cup R_j^s)} \quad (2.27)$$

Onde, $\#(x)$ representa o número de píxeis de x , I_s é a imagem resultante do processo de segmentação, NR_s é o número de regiões conectadas na imagem segmentada I_s . Já NR_{gt} é o número de regiões segmentadas das regiões das imagens de referência I_{gt} correspondente. Portanto, o valor de η na Equação 2.28 corresponde ao índice de qualidade de segmentação, sendo ponderado pela alta ou baixa segmentação. A alta segmentação é calculada quando a região segmentada NR_s é maior que a região da solução ideal NR_{gt} . Já a baixa segmentação acontece quando a região da imagem da solução ideal NR_{gt} for maior que a região segmentada NR_s .

$$\eta = \begin{cases} \frac{NR_{gt}}{NR_s}, & NR_s > NR_{gt} \\ \log 1 + \frac{NR_s}{NR_{gt}}, & NR_s \leq NR_{gt} \end{cases} \quad (2.28)$$

Após o cálculo dos índices de compatibilidade e de segmentação, então o índice *HAF* é calculado conforme a Equação 2.29.

$$HAF = \frac{M_I + (m \times \eta)}{1 + m} \quad (2.29)$$

Onde m é o fator de peso da classificação e pode ser variável entre 0 e 1.

Erro de Classificação (ME)

O erro de classificação “Missclassification error” (ME) pode ser expresso a partir da Equação 2.30 (Sezgin and Sankur, 2001). Onde, B_O e F_O denotam os conjuntos de píxeis do fundo e do primeiro plano da imagem original, enquanto B_T e F_T denotam os conjuntos da área de píxeis do fundo e do primeiro plano da imagem de teste respectivamente. O erro de má classificação varia entre 0 para uma segmentação perfeita da imagem e 1 para o pior caso de imagem mal segmentada.

$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|} \quad (2.30)$$

Crítérios de Martin (MAR)

O critério de Martin é utilizado para comparar a qualidade de segmentação e é constituído por duas métricas a MAR_{gce} e a MAR_{lce} (Martin et al., 2001).

Essas métricas são baseadas no cálculo de refinamento de erros locais, definido pela Equação 2.31. S_1 e S_2 como sendo áreas de segmentação de uma imagem cujos valores variam entre 0 e 1 (0 indica não possuir erros) e que S_1 representa a segmentação ideal (groundtruth) e S_2 a segmentação a ser testada. Onde \setminus significa a diferença entre os conjuntos e $|x|$ indica a cardinalidade do conjunto x e p_i sendo o pixel a ser comparado nas áreas segmentadas. A Equação 2.32 mostra o erro de consistência global (*Global Consistence Error*) e a Equação 2.33 mostra como é calculado o erro de consistência local (*Local Consistence Error*).

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|} \quad (2.31)$$

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\} \quad (2.32)$$

$$LCE(S_1, S_2) = \frac{1}{n} \sum_i \min \{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \quad (2.33)$$

2.6 Considerações Finais

Neste capítulo foram apresentados conceitos básicos inerentes ao desenvolvimento deste trabalho. A contextualização dos problemas que envolvem os incêndios acidentais no mundo e no Brasil, o que mostrou o quão importantes são as ações para minimizar os danos causados por esses incidentes. Além disso, foram apresentados os conceitos básicos de processamento digital de imagens, aplicação de entropia em imagens, algoritmos de aprendizagem de máquina e RNCs.

Em relação às RNCs, vale destacar as escolhas das redes a partir de seu desempenho na competição ILSVR. A proposta de redes que apresentam variações nos números de camadas foi escolhida ponderando o tamanho da rede e seu desempenho. Nesse sentido, em relação às redes

VGG, a rede VGG16 foi escolhida por ter desempenho similar à rede VGG19 e menos camadas. Para as redes ResNet, os resultados apresentados pelos autores mostram que a ResNet152 obteve melhor desempenho comparado às outras variações de Resnet, porém a arquitetura Resnet50 apresentou resultados semelhantes realizando menos operações por segundo. Portanto, a rede Resnet50 foi a rede escolhida dentre as redes com blocos residuais.

Também foram apresentados neste capítulo as métricas de avaliação voltadas para detecção e localização de fogo em imagens. No próximo capítulo serão apresentados os principais trabalhos relacionados à detecção e localização de fogo.

Capítulo 3

Estado da Arte

O levantamento dos métodos de detecção de fogo encontrados na literatura apontou abordagens bastante distintas. Dessa forma, pode-se separar as abordagens em dois grupos: 1) baseado em diferentes espaços de cor e 2) as que utilizam de classificadores como auxiliar no processo de detecção de fogo. De uma maneira geral, as técnicas de detecção de fogo são subdivididas em duas etapas: 1) mapeamento de píxeis de fogo e 2) detecção de movimentos para descartar píxeis de fogo suspeitos. Alguns métodos utilizam ordens distintas, realizando primeiro a exclusão de fundo por meio da detecção de movimentos e depois realizando o mapeamento de píxeis de fogo.

Outra característica dos trabalhos estudados, principalmente os que utilizam vídeos, é em relação a forma de cálculo da detecção de fogo. Na maioria dos casos, o sucesso ou não de um método é medido de acordo com a razão entre os *frames* detectados corretamente (tanto fogo e não fogo) e o número total de *frames* do vídeo, conforme acontece nos trabalhos de: (Horng et al., 2005), (Çelik et al., 2007), (Çelik and Demirel, 2009), (Çelik, 2010) e (Santana et al., 2012).

A partir de 2016, diversos pesquisadores adotaram a abordagem de Rede Neural Convolutiva para tentar solucionar o problema de reconhecimento de fogo ou fumaça, a maioria desses trabalhos propuseram novas arquiteturas de redes neurais, os trabalhos encontrados foram: (Frizzi et al., 2016), (Tao and Wang, 2016), (Zhang et al., 2016), (Wang et al., 2017), (Maksymiv et al., 2017), (Sharma et al., 2017) e (Muhammad et al., 2018).

A Seção 3.1 trata das técnicas existentes para detecção e localização de fogo que utilizam regras aplicadas aos espaços de cor. Já a Seção 3.2 mostra os métodos de detecção de fogo que utilizam técnicas de aprendizagem de máquina como auxiliar no processo de mapeamento dos píxeis de fogo.

3.1 Detecção e localização de incêndio baseados em regras nos espaços de cor

Nesta Seção serão apresentados os trabalhos de classificação de píxeis de fogo baseados em regras aplicadas nos espaços de cor ou canais de forma individual. A identificação dos prováveis píxeis de fogo, além de detectarem a presença ou não de fogo também indicam a localização do mesmo. A disposição dos métodos está agrupada de acordo com os espaços de cor utilizados.

3.1.1 Detecção e localização de fogo no espaço de cor RGB

O trabalho de Phillips et al. (2002) traz uma abordagem de localização utilizando o espaço de cor RGB e uma base de treinamento composta de imagens previamente segmentadas. A partir dessas imagens, foi criada uma tabela de consulta (TC) em três dimensões de tamanho $256 \times 256 \times 256$, e com isso foi realizada uma análise dos valores de RGB os quais mais se aproximavam de fogo a partir do histograma das imagens segmentadas. Com isso, um limiar de $k_1 = 0,2$ foi escolhido para classificar os píxeis como sendo candidatos de fogo ou não de acordo com o valor existente na tabela de consultas criadas a partir do treinamento. O Algoritmo criado por estes autores utiliza também as mudanças temporais dos vídeos valendo-se da diferença de intensidade $\Delta I(x, y)$ e consiste em 8 (oito) passos para determinar se uma região da imagem possui ou não fogo. A Equação 3.1 demonstra como é dado o cálculo da probabilidade de uma cor P_{color} e a Equação 3.2 mostra como é realizada a consulta e definição, se a cor possui ou não probabilidade maior que k_1 para ser considerada fogo ou não. Já a Equação 3.3 mostra como é determinado se um pixel de uma imagem é considerado de fogo ou não. Como consequência, os autores apresentam os resultados de detecção de fogo de 93,77% para 11 vídeos; em relação à qualidade de segmentação, os resultados não são apresentados pelos autores.

$$P_{color} = \frac{\sum_{i=1}^n TC(Pi(x, y))}{n} \quad (3.1)$$

$$Color(x, y) = \begin{cases} 1 & \text{se } P_{cor} > k_1 \\ 0 & \text{se } P_{cor} \leq k_1 \end{cases} \quad (3.2)$$

$$Fogo(x, y) = \begin{cases} 1 & \text{se } Color(x, y) = 1 \text{ e } \Delta I(x, y) > k_2 \\ 0 & \text{senão} \end{cases} \quad (3.3)$$

Para localizar os píxeis de fogo, Çelik et al. (2007) utilizaram 0, valores normalizados RGB baseados no modelo de cores da chama, este valores normalizados são propostos para amenizar os efeitos de mudança de iluminação. O modelo genérico é obtido utilizando análise estatística para os planos (r, g) , (g, b) e (r, b) , em conjunto com regras de subtração do fundo, aplicando-se as regras presentes em 3.4, aplicação de morfologia matemática, detecção de

manchas de fumaça e análise se é de fogo ou não. Os autores não apresentam os valores da qualidade de segmentação, porém como resultado, o autor apresenta 98.89% de taxa de detecção para a base que ele utilizou: 10 vídeos com mais de 1.000.000 de *frames* de vídeo.

$$P(x, y) = \text{fogo se } \begin{cases} R(x, y) > R_{media} \\ R(x, y) > G(x, y) > B(x, y) \\ 0,25 \leq \frac{G(x,y)}{R(x,y)+1} \leq 0,65 \\ 0,05 \leq \frac{B(x,y)}{R(x,y)+1} \leq 0,45 \\ 0,20 \leq \frac{B(x,y)}{G(x,y)+1} \leq 0,60 \end{cases} \quad (3.4)$$

3.1.2 Detecção e localização de fogo no espaço de cor HSI

Utilizando a análise de cores, os autores de (Horng et al., 2005) utilizaram o espaço de cor HSI com a justificativa de ser o espaço de cor com maior similaridade à visão humana. Desta forma, ao analisar o comportamento dos canais de cor a partir de uma base de imagens e suas respectivas soluções ideais (*groundtruth*), os autores chegaram às seguintes conclusões: para o Hue (Matiz), os valores dos píxeis de fogo ficaram entre 0° e 60° quando o fogo possui coloração amarelada/vermelha e entre 200° e 280° quando o fogo possui coloração azulada (intervalo de 0° a 360°), os valores de Saturação variam de 40 a 100 (intervalo de 0 a 100) em ambientes claros e de 20 a 100 em ambientes escuros e os valores de Intensidade variam de 127 a 255 (intervalo de 0 a 255) em ambientes claros e de 100 a 255 em ambientes escuros. A partir dessas informações foi criado o seguinte modelo de regras para a realização da segmentação representado em 3.5 e 3.6, sabendo que $f(x, y)$ é a função que representa a imagem de entrada para o pixel nas coordenadas x e y .

$$C([h_1, h_2], [s_1, s_2], [i_1, i_2]) = \{(h, s, i) \mid \begin{cases} h_1 \leq h \leq h_2 \\ s_1 \leq s \leq s_2 \\ i_1 \leq i \leq i_2 \end{cases} \} \quad (3.5)$$

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) \notin C \\ 1 & \text{caso contrário} \end{cases} \quad (3.6)$$

Ainda sobre o trabalho de Horng et al. (2005), após a segmentação, os quadros (*frames*) dos vídeos são processados e os píxeis onde não ocorrem alterações são descartados. Segundo os autores, este método atinge 96,97% de acerto em média na detecção de fogo.

3.1.3 Detecção e localização de fogo nos espaços de cor YCbCr e YUV

Marbach et al. (2006) apresentam uma técnica de processamento de imagens em tempo real para a detecção automática de incêndio em imagens de vídeos com base nas variações temporais

da intensidade do fogo capturadas por uma câmara. As sequências de imagens são analisadas para selecionar regiões candidatas de fogo a partir de regras definidas nas equações 3.7 à 3.11. Características do fogo são extraídas das regiões candidatas através dos canais de luminância, crominância e saturação do espaço YUV e combinadas para determinar a presença de padrões de fogo ou não de fogo. Limiares definidos de forma heurística são empregados para assegurar um bom desempenho em tempo real. A seguir, seguem as equações 3.7 à 3.11 que auxiliam no processo de localização dos píxeis de fogo, dado que $Y(x, y, t)$ representa o pixel $P(x, y)$ para o canal Y dado um tempo t .

$$D(x, y, t) = |Y(x, y, t) - Y(x, y, t - 1)| \quad (3.7)$$

Então, uma matriz cumulativa é criada para armazenar as mudanças temporais do pixel.

$$A(x, y, t) = \alpha A(x, y, t - 1) - (1 - \alpha)D(x, y, t)W(x, y, t)$$

onde, $W(x, y, t) = \begin{cases} Y(x, y, t), & \text{se } Y(x, y, t) \geq \delta \\ 0, & \text{caso contrário} \end{cases}$, (3.8)

onde $Y_{max} > \delta \geq Y_{mean}$

Para identificar os píxeis candidatos a fogo, é extraída uma região Ω_{ROI} com 32×32 píxeis vizinhos a $P(x_{ROI}, y_{ROI})$, sendo que cada pixel é denominado pela equação:

$$P(x_{ROI}, y_{ROI}) = \{P(x, y) | \max\{A(x, y, t)\}\} \quad (3.9)$$

A partir disso, o conjunto de píxeis ativos é dado por:

$$\pi_{ROI}(x, y, t) = \{(x, y) \in \Omega_{ROI} | A(x, y, t) \geq n_1\}, \text{ onde } 0 \leq n_1 \leq 255 \quad (3.10)$$

Então, seis características são retiradas sendo elas: luminância ($I_{ROI}(t)$), frequência ($f_{ROI}(t)$), amplitude ($a_{ROI}(t)$), número de píxeis ativos ($r_{ROI}(t)$), número de píxeis saturados ($s_{ROI}(t)$) e números de píxeis cor de fogo ($c_{ROI}(t)$). Cada característica é então associada a um indicador $I(z, t)$, onde z simboliza a característica. Temos:

$$I(z, t) = \begin{cases} 1, & \text{se } \mu_{low}(z) \geq z_{ROI} \geq \mu_{high}(z) \\ 0, & \text{caso contrário} \end{cases} \quad (3.11)$$

O trabalho ainda cita que os valores de μ_{low} e μ_{high} devem ser encontrados de forma empírica. Os autores não indicam quais foram os valores utilizados. O indicador de fogo é então dado como verdadeiro, caso todos os indicadores da cada característica tenham valor igual a 1. No trabalho, não foram apresentados resultados numéricos referentes à taxa de detecção, tampouco para a qualidade de segmentação.

O Método de Çelik and Demirel (2009) é baseado em um modelo de regras aplicadas ao espaço de cor YCbCr. Algumas regras são identificadas aos píxeis de fogo são aplicados

o Sistema de Inferência Fuzzy que é descrito a partir das regras apresentadas pelas equações 3.12 e 3.13. Como resultado, são relatados 99% de acerto na detecção em imagens de fogo e 31% de falsos alarmes em imagens de não fogo. No trabalho não há menção da qualidade de segmentação dos píxeis de fogo nas imagens utilizadas. Além disso, segundo seus autores, este método não é recomendado para ser utilizado como monitoramento em tempo real.

$$P(x, y) \text{ é fogo se } \begin{cases} I_Y(x, y) > I_{Cb}(x, y) \\ I_{Cr}(x, y) > I_{Cb}(x, y) \\ |I_{Cb}(x, y) - I_{Cr}(x, y)| \geq \tau \\ I_Y(x, y) > \mu_Y \text{ e } I_{Cb}(x, y) < \mu_{Cb} \text{ e } I_{Cr}(x, y) > \mu_{Cr} \\ I_{Cb} \geq fu(I_{Cr}) \cap I_{Cb} \leq fd(I_{Cr}) \cap I_{Cb} \leq fl(I_{Cr}) \end{cases} \quad (3.12)$$

$$\begin{aligned} fu(Cr) &= -2,6 \times 10^{-10}Cr^7 + 3,3 \times 10^{-7}Cr^6 - 1,7 \times 10^{-4}Cr^5 \\ &\quad + 5,16 \times 10^{-2}Cr^4 - 9,1 \times Cr^3 + 9,6 \times 10^2Cr^2 \\ &\quad - 5,6 \times 10^4Cr + 1,4 \times 10^6 \\ fl(Cr) &= -6,77 \times 10^{-8}Cr^5 + 5,5 \times 10^{-5}Cr^4 - 1,76 \times 10^{-2}Cr^3 \\ &\quad + 2,78Cr^2 - 2,15 \times 10^2Cr + 6,62 \times 10^3 \\ fd(Cr) &= 1,81 \times 10^{-4}Cr^4 - 1,02 \times 10^{-1}Cr^3 + 2,17 \times 10Cr^2 \\ &\quad - 2,05 \times 10^3Cr + 7,29 \times 10^4 \end{aligned} \quad (3.13)$$

3.1.4 Detecção e localização de fogo no espaço de cor CIELab

Em Çelik (2010), houve a utilização do espaço de cor *CIELab*, em que o algoritmo proposto consiste em duas etapas: aplicação de um modelo de cores usando o referido espaço de cor (etapa 1) e a detecção de elementos que se movimentem nos vídeos (etapa 2). Para a criação do modelo de cores, o método proposto utiliza uma matriz de probabilidades dividida em 24 níveis para cada composição possível de espaço de cor nos planos (L, a) , (L, b) e (b, a) . Utilizando o modelo de probabilidades, o autor criou 5 regras para definir se um pixel é candidato a fogo ou não. As 5 regras propostas pelo autor estão na Equação 3.14, em que o valor de α pode variar, o autor sugere $\alpha = 0,00016$, que produz 90% de acerto em detecções, mas também produz 36% de falsos alarmes. Para complementar o processo de detecção, os autores também utilizam das mudanças temporais entre os *frames* e ao aplicar esta técnica a taxa de detecção sobe para 99,88%. Em

relação à seleção dos píxeis de fogo e à localização do fogo não são apresentados resultados que indiquem a qualidade de segmentação do método proposto.

$$P(x, y) = \text{fogo se } \begin{cases} L(x, y) \geq L_{media} \\ a(x, y) \geq a_{media}(x, y) \\ b(x, y) \geq b_{media}(x, y) \\ b(x, y) \geq a(x, y) \\ P(L(x, y), \alpha(x, y), b(x, y)) \geq a \end{cases} \quad (3.14)$$

3.1.5 Detecção e localização de fogo com espaços de cor híbridos

O Método de Chen et al. (2004) é baseado em regras aplicadas em uma imagem I ao canal de cor RGB e o canal S do espaço de cor HSI, em que, para ser identificado como pixel de fogo, as três condições mostradas na Equação 3.15 devem ser satisfeitas. Segundo o estudo, os limiares para S_T e R_T devem ficar entre 55 a 65 e 115 a 135 respectivamente.

$$I_c(x, y) \begin{cases} I_R(x, y) > R_T \\ I_R(x, y) \geq I_G(x, y) > I_B(x, y) \\ I_S(x, y) \geq (255 - I_R(x, y)) \frac{S_T}{R_T} \end{cases} \quad (3.15)$$

Ainda sobre Chen et al. (2004), após a verificação das similaridades com os píxeis de fogo, uma análise dinâmica dos *frames* dos vídeos é realizada para que píxeis parecidos com o fogo, mas que não são fogo, sejam descartados. Esta análise consiste em aplicar a verificação de variabilidade entre um *frame* e outro, analisando inclusive se há aumento na região de possível fogo, sendo que os *frames* onde não ocorrem essas mudanças são descartados. O artigo não demonstra que foram as bases utilizadas para teste, tampouco os resultados dos testes, ou seja, resume-se à proposta do método de localização dos píxeis de fogo.

No método de Ko et al. (2009) o autor propôs a utilização de distribuição de probabilidade Gaussiana para cada canal de RGB, definido pela Equação 3.16.

$$p_c(I_c(x, y)) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(I_c(x, y) - \mu_c)^2}{2\sigma_c^2}\right) \quad (3.16)$$

Onde, μ_c e σ_c representam a média e o desvio padrão respectivamente do canal $c \in \{R, G, B\}$ de um pixel de fogo. Neste trabalho, o autor utiliza 100 imagens extraídas de 10 vídeos. Com o cálculo da probabilidade de cada canal do pixel x , calcula-se o produtório das probabilidades de p_c , conforme Equação 3.17. Então, caso $p_f \geq \tau$ então p_f é considerado um pixel de fogo, onde τ é o valor de limiar não informado no trabalho em questão.

$$p_f(I(x, y)) = \prod_{c \in \{R, G, B\}} p_c(I_c(x, y)) \quad (3.17)$$

Após a seleção dos píxeis de fogo, o trabalho ainda utiliza mais três passos: detecção de píxeis de fogo em movimento, remoção de píxeis de não fogo utilizando a variação de luminância em relação ao tempo e a utilização do classificador SVM. Como resultado, os autores apresentam taxa de detecção de 86,3% e taxa de falsos positivos em 0,4%. Em relação à taxa de localização, os autores não apresentam resultados de qualidade de segmentação.

Em Chen et al. (2010) os autores apresentaram um método de detecção de chama em vídeos com base na fusão de características temporais e espaciais de chamas e de cintilação de chamas. Um modelo de mistura gaussiana é primeiramente adotado para extrair objetos em movimento nas cenas de fogo. Em segundo lugar, estes objetos em movimento detectados são então classificados em regiões candidatas e não-candidatas de chama por meio de um algoritmo de filtragem baseado em regras no espaço de cor RGB descritas nas Equações: 3.18, 3.19 e 3.20. Finalmente, um algoritmo de identificação de cintilação de chamas é usado para distinguir as verdadeiras regiões de chama. Segundo os autores, os resultados dos testes mostram que os algoritmos propostos são eficazes, robustos e eficientes. Nenhuma informação sobre a quantidade de vídeos e que tipo de avaliação numérica usada foi fornecida pelos autores.

$$Y1 = \begin{cases} 100 - 0.48 \times B(x, y) & \text{if } B(x, y) \geq 117 \\ 79.27 - 0.311 \times B(x, y) & \text{if } B(x, y) < 117 \end{cases} \quad (3.18)$$

$$Y2 = -2.0147 + 90.59435 \times \exp\left(-\frac{B(x, y)}{77.6027}\right) \quad (3.19)$$

$$P(x, y) \text{ é fogo se } = \begin{cases} R(x, y) \geq R_t \\ R(x, y) \geq G(x, y) \geq B(x, y) \\ Y2 \leq S(x, y) \leq Y1 \end{cases} \quad (3.20)$$

Onde, R_t é um limiar para a cor vermelha e seu valor deve estar entre 115 e 135, segundo informações dos autores.

O trabalho de Vipin (2012), consiste na criação de um conjunto de regras para definir se um pixel é ou não de fogo usando os espaços de cores RGB e YCrCb. Esta proposta é voltada para detecção de píxeis de fogo em imagens estáticas. Segundo o autor, este é um método de baixo custo computacional e de boa performance. Experimentos realizados com 200 imagens mostraram a abordagem eficiente com 99% de detecção de chamas e 14% de falsos positivos. Vale ressaltar que quando o autor menciona detecção, ele indica que foram localizados ao menos 10 píxeis de fogo na imagem testada. O resumo deste método pode ser visualizado na Equação 3.21, nesta equação o valor de Th pode alterar de acordo com os tipos de imagens de fogo, os

autores utilizaram $Th = 70$. Neste trabalho não há um estudo da qualidade de segmentação de imagens, portanto, não é possível estabelecer qual a qualidade de segmentação.

$$P(x, y) \text{ é fogo } \left\{ \begin{array}{l} 1) R(x, y) > G(x, y) > B(x, y) \\ 2) R(x, y) > 190 \wedge G(x, y) > 100 \wedge B(x, y) < 140 \\ 3) Y(x, y) \geq Cb(x, y) \\ 4) Cr(x, y) \geq Cb(x, y) \\ 5) Y(x, y) \geq Y_{mean}(x, y) \wedge Cb(x, y) \leq Cb_{mean}(x, y) \wedge Cr(x, y) \geq Cr_{mean}(x, y) \\ 6) |Cb(x, y) - Cr(x, y)| \geq Th \\ 7) (Cb(x, y) \leq 120) \wedge (Cr(x, y) \geq 150) \end{array} \right. \quad (3.21)$$

Santana et al. (2012) apresentaram um sistema de detecção precoce de incêndio em vídeos. A técnica é baseada em um modelo de análise de frequências Wavelet acoplado a um novo mapeamento de píxeis de fogo baseado nos canais Y do espaço de cor YUV e H do espaço de cor HSI. Esta proposta de mapeamento de píxeis de fogo interage com um sistema de detecção e rastreamento de pessoas, de modo a permitir a redução de falsos alarmes, uma vez que pode evitar que um carro vermelho em movimento, por exemplo, seja identificado como fogo. Para validar o sistema proposto, um conjunto de 12 vídeos obtidos a partir da Internet foi criado e rotulado manualmente, totalizando 21.992 quadros com resolução 300×250 em diferentes ambientes e condições de iluminação. Os experimentos realizados obtiveram uma taxa média de sucesso na detecção de 92,7%, além disso, segundo os autores, em pelo menos 90% dos casos de detecção, o fogo foi corretamente localizado.

O trabalho de Toulouse et al. (2015) traz uma avaliação do desempenho da segmentação de 11 métodos de reconhecimento de fogo. A partir das regras dos métodos existentes, os autores propõem a separação destes métodos em 29 regras que foram analisadas quanto à eficiência de identificação de píxeis de fogo e não fogo. Estas regras, foram utilizadas em conjunto com a técnica de votos majoritários. Além disto, o trabalho propõe um novo método de segmentação baseado em aprendizagem de máquina usando Regressão Logística. Para avaliar a qualidade de segmentação dos métodos estudados, os autores utilizaram a ponderação de valores de verdadeiros positivos para imagens de fogo e verdadeiros negativos para imagens de não fogo. Dos métodos avaliados em relação à qualidade da segmentação foi constatado que a proposta de regressão logística e o trabalho de Phillips et al. (2002) foram os que obtiveram os melhores índices com valores médios de 91% de qualidade de segmentação. Já a técnica do voto majoritário obteve 90% de qualidade de segmentação avaliada. Este trabalho não apresentou dados em relação à qualidade de detecção das imagens, somente em relação à localização das mesmas.

3.2 Detecção e localização de incêndio usando aprendizagem de máquina

Além de abordagens que realizam o mapeamento dos píxeis de fogo por meio de regras, existem abordagens que fazem uso dos algoritmos de classificação como: K-Means, *Nearest Neighbor*, Naïve Bayes, SVM, Redes Neurais, AdaBoost e Boosting.

O Método proposto por Rossi et al. (2009) e em Rossi et al. (2011) consiste em pré-selecionar píxeis de fogo, depois aplicar o classificador K-Means com o canal de cor V do espaço de cor YUV, com a maior área segmentada, então, aplica-se o modelo gaussiano 3D descrito pelas Equações 3.22 e 3.23 aplicadas aos canais RGB.

$$\begin{cases} \mu = (\mu_R, \mu_G, \mu_B) \\ \sigma = (\sigma_R, \sigma_G, \sigma_B) \end{cases} \quad (3.22)$$

$$I(x, y) = \begin{cases} \text{fogo, se } \sqrt{(I_R - \mu_R)^2 + (I_G - \mu_G)^2 + (I_B - \mu_B)^2} \leq c \times \sigma \\ \text{não fogo, caso contrário} \end{cases} \quad (3.23)$$

No trabalho Rudz et al. (2013), os autores estudaram diferentes espaços de cor e propuseram uma técnica utilizando K-Means aplicado ao canal Cb do espaço de cor YCbCr, e após esta pré-segmentação aplicaram regras no canal RGB. Para chegar a este resultado, os autores realizaram testes nos canais de cor U, V, Cb e Cr dos espaços de cor YUV e YCbCr (canais análogos), e variaram o número de classes do k-means de 2 a 9. Os melhores resultados com os critérios de segmentação adotados, foram com o canal Cb e com a utilização de 4 classes. A identificação da classe mais próxima aos valores dos píxeis de fogo foi definida a partir da comparação de proximidade de valores com 2×10^6 píxeis de fogo pré-definidos.

Após a aplicação do k-means, e o agrupamento das classes, dois conjuntos de regras são aplicados, um aplicado a regiões com menor proximidade aos píxeis de fogo pré-definidos (Equação 3.24) e outro para regiões com maior aos píxeis de fogo pré-definidos (Equação 3.25). Onde R, G e B são os canais de RGB, $hist_C^{GT}$, μ_C^{GT} e σ_C são respectivamente o valor de referência do histograma, média e desvio padrão e os valores de τ_R , τ_G , τ_B , c_R , c_G , c_B utilizados foram 40.1, 74, 74, 1.4, 9 e 9, respectivamente.

$$p(x, y) = \begin{cases} \|\mu_R^{GT} - \mu_R\| < c_R \times \sigma_R \\ \|\mu_G^{GT} - \mu_G\| < c_G \times \sigma_G \\ \|\mu_B^{GT} - \mu_B\| < c_B \times \sigma_B \end{cases} \quad (3.24)$$

$$p(x, y) = \begin{cases} \| hist_R^{GT} - hist_R \| < \tau_R \\ \| hist_G^{GT} - hist_G \| < \tau_G \\ \| hist_B^{GT} - hist_B \| < \tau_B \end{cases} \quad (3.25)$$

Os resultados deste método comparado com outras 3 técnicas de reconhecimento de fogo (Çelik and Demirel, 2009; Chen et al., 2004; Rossi et al., 2011) e uma técnica de aplicação de k-means supervisionado apresentaram os seguintes resultados: $MAR_{gcc}=94.2\%$, $MAR_{Ice}=98.6\%$ e $HAF = 65.9\%$, enquanto que Çelik and Demirel (2009) obtiveram: $MAR_{gcc}=87.6\%$, $MAR_{Ice}=96.3\%$ e $HAF = 57.2\%$; Chen et al. (2004) obtiveram: $MAR_{gcc}=91.5\%$, $MAR_{Ice}=98.1\%$ e $HAF = 59.7\%$, Rossi et al. (2011) obtiveram: $MAR_{gcc}=88.6\%$, $MAR_{Ice}=94.8\%$ e $HAF = 52.8\%$ e Chitade e Katiyar (2010) apud (Rudz et al., 2013), obtiveram: $MAR_{gcc}=93.4\%$, $MAR_{Ice}=98.5\%$ e $HAF = 62.0\%$.

O trabalho de (Wong and Fong, 2014) utiliza a abordagem do *Nearest Neighbor* para refinar o método desenvolvido. A técnica criada consiste em aplicar o algoritmo de Otsu (1979) para determinação de 6 limiares, paralelamente, há a aplicação do algoritmo de distribuição de Rayleigh que também determina outros 6 limiares. Estes limiares são então comparados e caso um deles seja igual nos dois métodos, este é aplicado para segmentar a imagem. Após esta segmentação, o algoritmo de *Nearest Neighbor* é aplicado com o intuito de identificar se a região selecionada consiste em uma região de fogo. Segundo o autor, a composição dos algoritmos de Otsu e Rayleigh é promissora. O trabalho não cita o número de imagens usadas no teste, tampouco seus resultados.

No trabalho de Chino et al. (2015) é proposto um método de detecção de píxeis de fogo em imagens estáticas, utilizando os espaços de cor RGB e YCbCr e uma técnica de superpixel baseado em textura. Para classificar os píxeis, os autores adotaram o classificador Naive Bayes e também testaram o classificador KNN. O classificador Naive Bayes utilizou um método de discretização automática valendo-se da metodologia SLIC (Simple Linear Iterative Clustering), descrita por Achanta et al. (2012), porém ao invés de utilizar o espaço de cor CIE, foi utilizado o YCbCr. Já o classificador KNN foi desenvolvido baseado na distância Manhattan.

Além da proposta apresentada, os autores também aplicaram a técnica de textura aos métodos de (Chen et al., 2004; Çelik and Demirel, 2009; Rossi et al., 2011; Rudz et al., 2013). Os autores ainda realizaram o comparativo das qualidades de segmentação do método proposto e dos métodos existentes. Segundo os autores, a técnica de aplicação de textura otimizou as taxas de falso positivo em todos os métodos testados.

No trabalho de Ozkan et al. (2008) foi estudada a eficácia da utilização de algoritmos de Boosting para detecção de queimadas em florestas em imagens de satélites. Os algoritmos testados foram: Adaboost (AB) e Logitboost (LB). Os algoritmos de boosting foram executados em conjunto com outros algoritmos de classificação, sendo: Perceptron Multicamadas (MLP - Redes Neurais), Árvore de Regressão (RT) e Máxima Semelhança (Maximum Likelihood –

ML). Os resultados obtidos sugerem que a acurácia do MLP foi maior que os demais usando dados isolados do espectro e textura. E que a acurácia do método (MLP) foi pouco superior aos classificadores baseados em Adaboost e Logitboost.

Dukuzumuremyi et al. (2014b) utilizaram o Adaboost como auxiliar no processo de detecção de fogo em imagens retiradas a partir de uma webcam. O algoritmo proposto consiste em quatro etapas: classificação usando adaboost, pré-processamento, extração de características e classificação. No primeiro passo, caso o classificador Adaboost não reconheça fogo na imagem, uma classificação adicional é realizada por meio de redes neurais. Foram utilizadas 1000 imagens de fogo e 1000 imagens de não fogo com dimensões 250×250 píxeis. Como critérios de comparação, o método proposto foi comparado a outros dois métodos, com o método BBM de (Ko et al., 2010) e WMF de Kong et al (2004) apud (Dukuzumuremyi et al., 2014b). Como critério de medida de eficácia, foram medidas as quantidades de imagens que continham fogo classificadas de maneira correta ou não. Os resultados mostram que o método proposto alcançou taxas de detecção de 89% e os métodos BBM 84% e o WMF 86%.

Os autores da pesquisa descrita em Collumeau (2011) propuseram um método novo de segmentação por classificador SVM com o espaço de cor RGB e um estudo comparativo no caso da segmentação de incêndios florestais. O novo método foi confrontado com as técnicas de segmentação de fogo de Çelik and Demirel (2009) e de Chen et al. (2004) (com combinação dos espaços de cor RGB e YUV), de Ko et al. (2009) (com o espaço de cor RGB) e Rossi et al. (2009) (agrupador KMeans combinando os espaços de cor RGB e YCrCb). E foi avaliado utilizando as métricas de avaliação F-score, os critérios de Martins MAR_{gce} e MAR_{lce} e o critério de Hafiane HAF . Os experimentos realizados numa base de 76 imagens com a segmentação ground-truth disponível mostraram o método proposto pelos autores ser o com melhores resultados para as métricas: $F_1 score = 80.8\%$, $MAR_{gce} = 95.5\%$, $MAR_{lce} = 98.4\%$ e $HAF = 78.5\%$.

3.3 Detecção de fogo e as Redes Neurais

Em relação ao uso de Redes Neurais - RN, Angayarkkani and Radhakrishnan (2010) utilizam do espaço de cor XYZ aliado à segmentação difusa anisotrópica e classificação baseada em uma Rede Neural. A Rede Neural foi constituída a partir da função de suporte radial (RBF - Radial Basis Function) que possibilitou a criação do modelo a partir de instâncias de treino baseadas no canal XYZ. O processo de reconhecimento de fogo dessa proposta consiste em: transformar imagem RGB em imagem XYZ, aplicar a segmentação difusa anisotrópica e, por fim, executar a segmentação utilizando a Rede Neural criada. O artigo não traz comparativos tampouco mostra os resultados obtidos com a base utilizada.

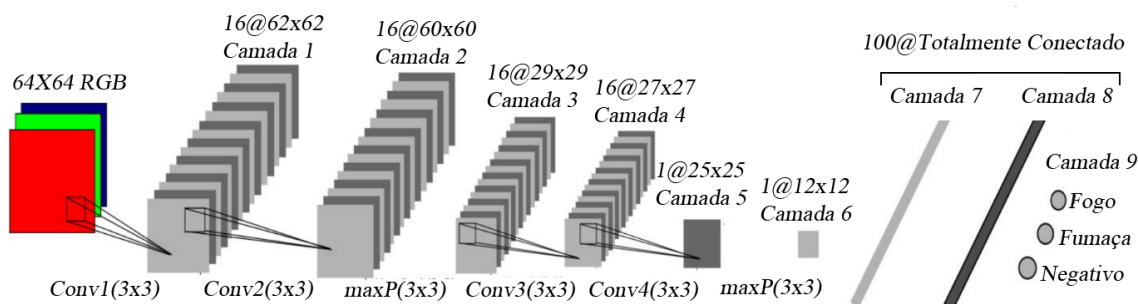


Figura 3.1: Arquitetura CNN para detecção de fogo e fumaça proposta por (Frizzi et al., 2016). Fonte: (Frizzi et al., 2016).

3.3.1 Detecção de fogo e fumaça e as Redes Neurais Convolucionais

Entre os anos de 2016 e 2018 surgiram alguns trabalhos utilizando Redes Neurais Convolucionais – RNC como tentativa de resolver os problemas de detecção de Fogo e/ou Fumaça.

No trabalho de Frizzi et al. (2016), os autores propuseram uma nova arquitetura de RNC para detecção de fogo e fumaça. A proposta arquitetural contém 8 camadas, sendo 6 camadas convolucionais e duas delas completamente conectadas apresentando 3 saídas possíveis (Fogo, Negativo ou Fumaça) conforme ilustra a Figura 3.1. Além da proposta arquitetural, os autores também disponibilizaram sua base de imagens composta de 27.919 imagens na base de treino e 5.584 imagens na base de teste, ambas as bases com imagens de tamanho 64×64 píxeis. Além disso, os autores também propuseram a classificação da imagem como um todo a partir do mapa de características gerado pela camada 6, que possui como saída um *patch* de tamanho 12×12 . Dessa forma, segundo os autores, foi possível reduzir o tempo de processamento sem afetar a acurácia do método. Como resultado, os autores reportam acurácia de 97% na detecção dos *patches* de fogo. Além disso, com os *patches* de tamanho 12×12 os autores fazem um mapeamento na imagem completa para localizar os focos de incêndio e fumaça na imagem toda.

Os autores de Tao and Wang (2016) adaptaram a arquitetura da rede Alexnet (Krizhevsky et al., 2012) para o problema de detecção de fumaça. Dentre as adaptações estão a mudança nas últimas camadas, cuja saída passou a ser binária (fumaça ou negativo) e a inversão nas camadas de *maxpooling* e a normalização de camadas. Para os testes, os autores utilizaram quatro bases de imagens com o respectivo número de imagens: 1383 (S_1), 1505 (S_2), 10712 (S_3) e 10617 (S_4), coletadas e publicadas no trabalho Yuan et al. (2016). As bases S_1 e S_3 foram usadas como treino e as bases S_2 e S_4 como teste. Como resultado, os autores reportam uma taxa de detecção de 94,2% na base S_2 e 99,4% na base S_4 .

O trabalho de Zhang et al. (2016) propõe um sistema de detecção de fogo em florestas utilizando redes neurais convolucionais - RNC. Os autores utilizaram uma base de imagens, extraída de *frames* de vídeo, composta de 178 imagens para treino e 69 imagens para teste, e a partir das imagens extraíram *patches* de 32×32 píxeis, totalizando 12460 *patches* para treino e

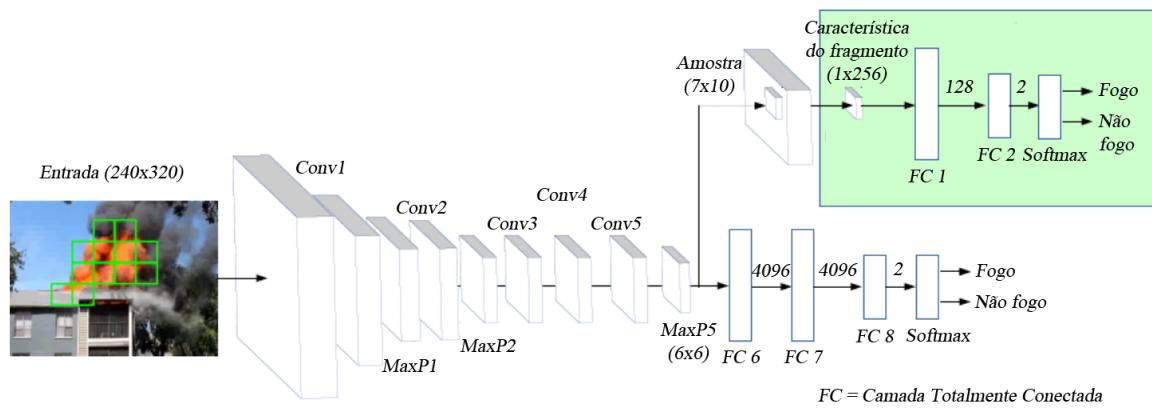


Figura 3.2: Arquitetura da RNC proposta por Zhang et. al. Fonte: (Zhang et al., 2016).

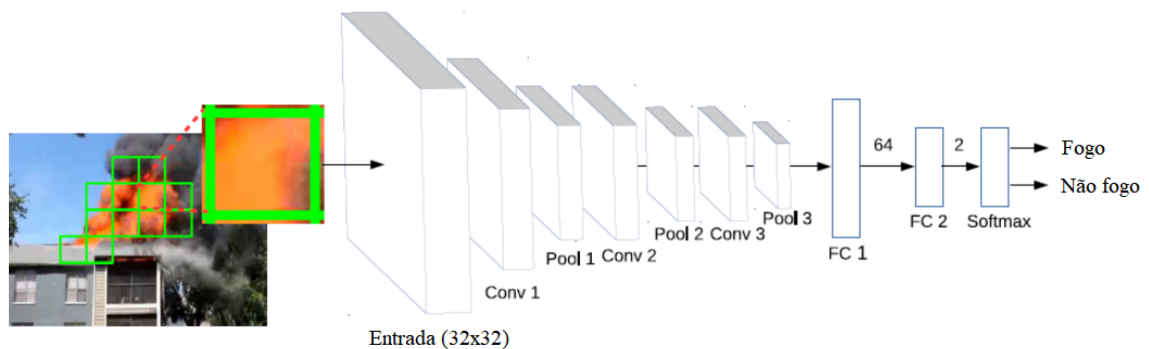


Figura 3.3: Arquitetura de RNC reduzida proposta por Zhang et. al. Fonte: (Zhang et al., 2016).

4130 *patches* para teste. Além disso, a base utilizada pelos autores é uma base desbalanceada, pois havia 1307 *patches* de fogo e 11153 *patches* negativos para o treino do modelo. Além da proposta arquitetural da rede neural convolucional, os autores também realizaram testes comparativos com o classificador SVM. Como resultado, na base teste, considerando as imagens inteiras, a acurácia ficou em 74% para o SVM e 93% para a RNC e a taxa de detecção 24% para o SVM e 84% para a RNC. A arquitetura RNC utilizada para detecção foi criada baseada na rede Alexnet (Krizhevsky et al., 2012) e utilizou a rede pré-treinada. Tal arquitetura pode ser visualizada na Figura 3.2. Além desta arquitetura, Zhang et al. (2016) também propõem uma rede neural convolucional reduzida, que foram aplicadas aos *patches*, ilustrada por meio da Figura 3.3, como resultado, na base teste a acurácia ficou em 89% para o SVM e 74% para a RNC e a taxa de detecção 24% para o SVM e 59% para a RNC.

Utilizando a composição de classificadores e a RNC, Wang et al. (2017) propuseram um método para detecção de fogo em imagens. Em um primeiro momento é realizada a extração da área de interesse utilizando o algoritmo de extração de características Haar com Adaboosting. Após a extração das características, os autores aplicaram uma rede RNC com quatro camadas convolucionais, realizando a classificação final e utilizando o SVM com kernel de função

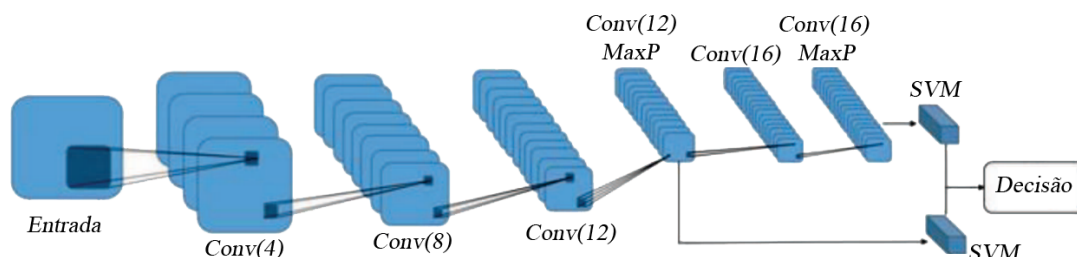


Figura 3.4: Arquitetura RNC proposto por (Maksymiv et al., 2017).

radial. A base de imagens utilizada para a extração de características possui 28.137 *patches* de 32×32 píxeis, sendo que 20.923 deles eram *patches* negativos. Para a rede RNC-SVM foram coletadas 55.197 *patches* de vídeos, utilizando a extração de características Haar, onde 41.300 foram usadas como treino e o restante como teste. Como resultado, para o método proposto RNC-SVM, os autores reportaram 99,14% de precisão. No trabalho em questão não há menção de disponibilidade da base de imagens utilizada.

Uma proposta similar à proposta de Wang et al. (2017) é a de Maksymiv et al. (2017) que também combina um extrator de uma área de interesse e uma RNC classificada ao final com o SVM. Porém, ao invés de utilizar o extrator de características Haar, Maksymiv e sua equipe utilizaram o Adaboost em conjunto com o LBP (*Local Binary Pattern* – Padrão Binário Localizado) para realizar a tarefa. Esta arquitetura está representada pela Figura 3.4. A base de imagens utilizada para treino contém 1876 imagens com fogo e 4634 imagens negativas. Para teste, os autores utilizaram uma base com 150 imagens de fogo, 150 imagens de fumaça e 300 imagens negativas. Como resultado os autores reportam 86,96% de Precisão e 95,2% de Recall.

No trabalho de Sharma et al. (2017) os autores modificaram as camadas finais das arquiteturas Resnet50 (He et al., 2015a) e VGG16 (Simonyan and Zisserman, 2014) uma arquitetura de RNC com 4 camadas totalmente conectadas na arquitetura VGG e uma camada totalmente conectada na arquitetura Resnet50. A base de imagens utilizada possui 651 imagens coletadas manualmente e separadas em 549 imagens para treino (59 de fogo).

Segundo os autores, a base de treino não balanceada é para simular o mundo real, onde a maioria das situações do cotidiano são de não fogo e isso auxiliaria no processo de identificação se o método de detecção está generalizando o problema. A base de teste por sua vez é balanceada e possui 51 imagens de fogo, para corroborar a tese dos autores. Como resultado final, nas redes modificadas, os resultados para acurácia foram 91,18% para a VGG16 e 92,15% para a rede Resnet50.

Para detecção de fumaça, temos a proposição de uma arquitetura de RNC por Yin et al. (2017). Esta arquitetura diferencia-se de outros trabalhos por introduzir o conceito de normalização entre as camadas de convolução também utilizado na rede Inception (Szegedy

et al., 2014) e Xception (Chollet, 2016). Segundo os autores, utilizar este recurso acelerou o processo de treinamento, além de melhorar os resultados de detecção e falsos alarmes comparados com arquiteturas já conhecidas como a AlexNet (Krizhevsky et al., 2012) e a VGG (Simonyan and Zisserman, 2014). A proposta arquitetural pode ser visualizada na Figura 3.5, onde NC_i indica o processo de normalização realizado em cada camada i de convolução. Os autores utilizaram 4 conjuntos de imagens divididos em dois conjuntos de teste (2888 imagens, sendo 1240 de fumaça), um conjunto de treino (10712 imagens, sendo 2201 de fumaça) e um de validação (10617 imagens, sendo 2254 de fumaça). Para reduzir o sobre-ajuste, os autores aplicaram técnicas de *data augmentation* nas imagens de fumaça da base de treino para equiparar a quantidade de imagens negativas e de fumaça. Como resultado, nas bases de teste, os autores alcançaram 97,83% e 98,08% de acurácia, 95,28% e 96,36% de taxa de detecção e 0,48% e 0,48% na taxa de falsos alarmes.

No trabalho de Muhammad et al. (2018), os autores utilizaram 4 bases de imagens, dentre elas a base de vídeo publicada por Foggia et al. (2015), assim como a base de imagens com solução ideal para segmentação de píxeis proposta por Chino et al. (2015). Os autores desta pesquisa propuseram uma rede neural artificial convolucional inspirada na GoogleNet. Como resultado, na base de vídeos proposta por Foggia et al. (2015) os autores relataram acurácia de 93,55% e 11,6% de falsos alarmes. Já na base proposta por Chino et al. (2015) os autores reportaram taxa de F_1score de 88%.

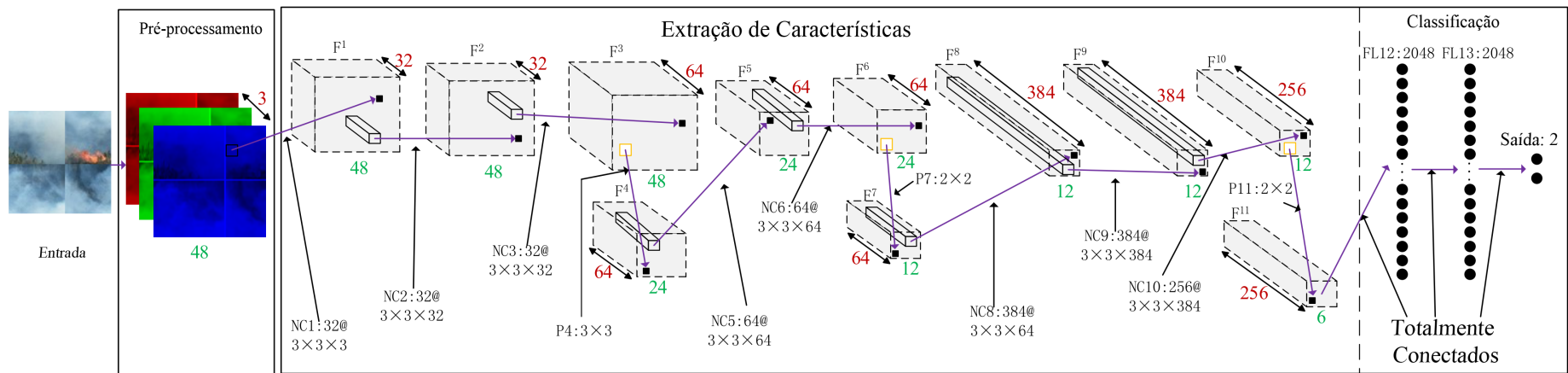


Figura 3.5: Arquitetura para detecção de fumaça proposta por Yin et al. (2017).

3.4 Considerações Finais

Muito embora não seja possível comparar de forma justa os resultados dos métodos encontrados no levantamento bibliográfico, pois cada trabalho utiliza uma base de imagens, assim como há a diferença nos métodos e protocolos experimentais adotados, uma tabela comparativa mostrando as características de cada método foi criada. Na Tabela 3.1, descrevemos de forma sintetizada as principais características dos trabalhos apresentados neste capítulo. Dentre as informações relevantes de cada método, estão: Nome da Técnica; Espaços de Cor utilizados; o tipo de classificação adotado – podendo ser por: Modelo Baseado em Regras (RB), Probabilidades (PB) ou utilizando algum mecanismo de aprendizagem de máquina (IA); a taxa de detecção de fogo se mencionada no artigo; Tamanho e tipo de base de imagens utilizada no trabalho avaliado e por fim se autor disponibilizou as bases de imagens/vídeos.

Vale ressaltar que as taxas de detecção mencionadas dizem respeito a localização ou não de fogo na imagem, ou seja, não tem relação com a qualidade de segmentação destas imagens. Os cálculos para taxas de detecção normalmente são baseados no número de *frames* que contém fogo, número de *frames* que não possuem fogo e número de *frames* que os métodos classificaram corretamente.

Em relação às técnicas de detecção de fogo estudadas, estas mostraram os principais conceitos utilizados na literatura para realizar o processo de detecção. As técnicas de detecção em vídeo aproveitam das mudanças temporais que o vídeo proporciona para auxiliar no processo de detecção. Complementar à detecção de movimento, o mapeamento de píxeis de fogo, também podem ser utilizados após ou antes do processo de detecção de movimentos.

O estudo da literatura nos mostrou que as técnicas baseadas em regras, são utilizadas a partir de um espaços de cor definido, ou a combinação destes espaços, conforme mostra a coluna "Espaços de cor" da Tabela 3.1. O único trabalho que apresenta uma combinação de canais de cor de diferentes espaços de cor é o trabalho de Santana et al. (2012) que combina os canais H do espaço de cor HSI e o Y do canal de cor YCbCr. Outra evidência encontrada é que os métodos baseados em regras, na maioria dos casos é utilizado para detecção de fogo em vídeos. Em relação aos métodos que utilizam técnicas de aprendizagem de máquina, a maioria dos trabalhos utilizam o espaço de cor RGB e atacam o problema da detecção em imagens estáticas, pois não é possível a remoção de fundo por meio de análise temporal como nos vídeos.

No capítulo seguinte será mostrada a visão geral do método para detecção de fogo e fumaça e localização de fogo. Além disso, é apresentado o método utilizado para a proposição de bases de imagens voltadas para detecção de fogo e fumaça e para localização de regiões de fogo.

Tabela 3.1: Principais características dos métodos estudados

Autores	Espaços de Cor	Tipo de Classific.	Pontuação (Métrica) (Se mencionado)	Tamanho e tipo de Base (imagem/vídeo)	Base disponível?
Çelik et al. (2006)	RGB	RB	Não mencionado	Não mencionado	Não
Çelik et al. (2007)	RGB	RB	98,89% (Taxa de detecção)	10vídeos (176x144 40 fps)	Sim
Çelik and Demirel (2009)	YCbCr	RB	99% (Taxa de detecção)	332 imagens de Fogo	Não
Çelik (2010)	CIELab	RB	99,88% (Taxa de detecção)	10 vídeos (176x144 40 fps)	Sim
Chen et al. (2004)	RGB	RB	Não mencionado	Não mencionado	Não
Chen et al. (2010)	RGB+HSI	RB	Não mencionado	13 vídeos	Sim
Horng et al. (2005)	HSI	RB	96,94% (Taxa de detecção)	7 vídeos	Não
Ko et al. (2009)	RGB+HSI	RB + IA	93,5% (Taxa de detecção)	12 vídeos	Não
Phillips et al. (2002)	RGB	PB	93,77% (Taxa de detecção)	11 vídeos	Não
Rossi et al. (2009)	YUV+RGB	IA+RB	Não mencionado	Imagens 3D	Não
Santana et al. (2012)	HY (HSI + YCbCr)	RB	79% (imagens) e 92% (Video) (Taxa de detecção)	217 Fogo imagens e 12 vídeos	Não
Vipin (2012)	RGB+YCbCr	RB	99% (Taxa de detecção)	400 imagens (200 imagens de Fogo)	Não
Foggia et al. (2015)	YUV+RGB	IA	93,5% (Acurácia)	33 vídeos	Sim
Frizzi et al. (2016)	RGB	IA	95,8%	31k imagens	Sim
Chino et al. (2015)	YCbCr	PB	2,6% (Falso Positivo)	226 imagens (119 imagens de fogo)	Sim
Tao and Wang (2016)	RGB	IA	97% (Média de Taxa de detecção)	24217 imagens	Sim
Zhang et al. (2016)	RGB	IA	90,1% (Acurácia)	13767 imagens	Não
Zhang et al. (2016) Reduzido	RGB	IA	88,6% (Acurácia)	13767 imagens	Não
Wang et al. (2017)	RGB	IA	99% (Precisão)	41300 imagens	Não
Sharma et al. (2017)	RGB	IA	92% (Acurácia)	651 imagens (110 fogo)	Sim
Yin et al. (2017) (fumaça)	RGB	IA	96% (Taxa de detecção)	24217 imagens	Sim
Muhammad et al. (2018)	RGB	IA	93,5% (Acurácia)	4 bases	Sim

Capítulo 4

Método Proposto

O presente capítulo tem como objetivo mostrar uma visão geral do método criado para o desenvolvimento deste trabalho, assim como expor a descrição das bases de imagens utilizadas. Com base nas hipóteses levantadas (Capítulo 1, Seção 1.3), 5 diretrizes foram adotadas: 1) criação de bases de imagens para detecção de fogo e fumaça, 2) criação de base de imagens com a solução ideal com foco na localização do fogo, 3) Detecção de Fogo e Fumaça, 4) Localização de Fogo em Imagens e por fim 5) método de monitoramento para detecção e localização de fogo. Fundamentado nestas diretrizes, a estrutura de demonstração dos métodos e resultados do trabalho ficou dividida da seguinte maneira: Neste Capítulo, nas Seções 4.1.1 e 4.1.2 são descritas as criações das bases de imagens para detecção de fogo e fumaça e localização de fogo respectivamente. No Capítulo 5 serão descritos os métodos e os resultados experimentais para detecção de fogo e fumaça. O Capítulo 6 descreve os métodos e os resultados para o problema de localização de fogo. Já o Capítulo 7 descreve o método proposto para o monitoramento de fogo.

Para atingir o objetivo de construir um modelo capaz de detectar fogo e fumaça e identificar a localização de fogo nas imagens, além da elaboração das bases de imagens, foi idealizado um método composto de três etapas principais: 1) Detecção de fogo e fumaça, 2) Localização de fogo e 3) Método de monitoramento de fogo e fumaça. A Figura 4.1 mostra a visão geral deste método.

A etapa de Detecção de fogo e fumaça é detalhada no Capítulo 5. Esta etapa está representada na porção esquerda da Figura 4.1, nela buscou-se validar a hipótese de que a utilização de Redes Neurais Convolucionais, aplicadas à detecção de fogo e fumaça, possibilitam melhorar as taxas de detecção, quando comparados a métodos presentes na literatura. O processo de detecção de fogo e fumaça em imagens consiste em identificar a presença de fogo, fumaça ou a não presença nas imagens avaliadas. Para isso, métodos de detecção de fogo e fumaça propostos na literatura são testados, avaliados em uma mesma base de imagens e um novo modelo de RNC voltado para detecção de fogo e fumaça é proposto.

Em relação à etapa de Localização de fogo, presente no Capítulo 6 e representada na parte direita da Figura 4.1, buscou-se validar a hipótese de que o aprimoramento do processo de segmentação de píxeis de fogo a partir de técnicas de aprendizagem de máquina contribui para a

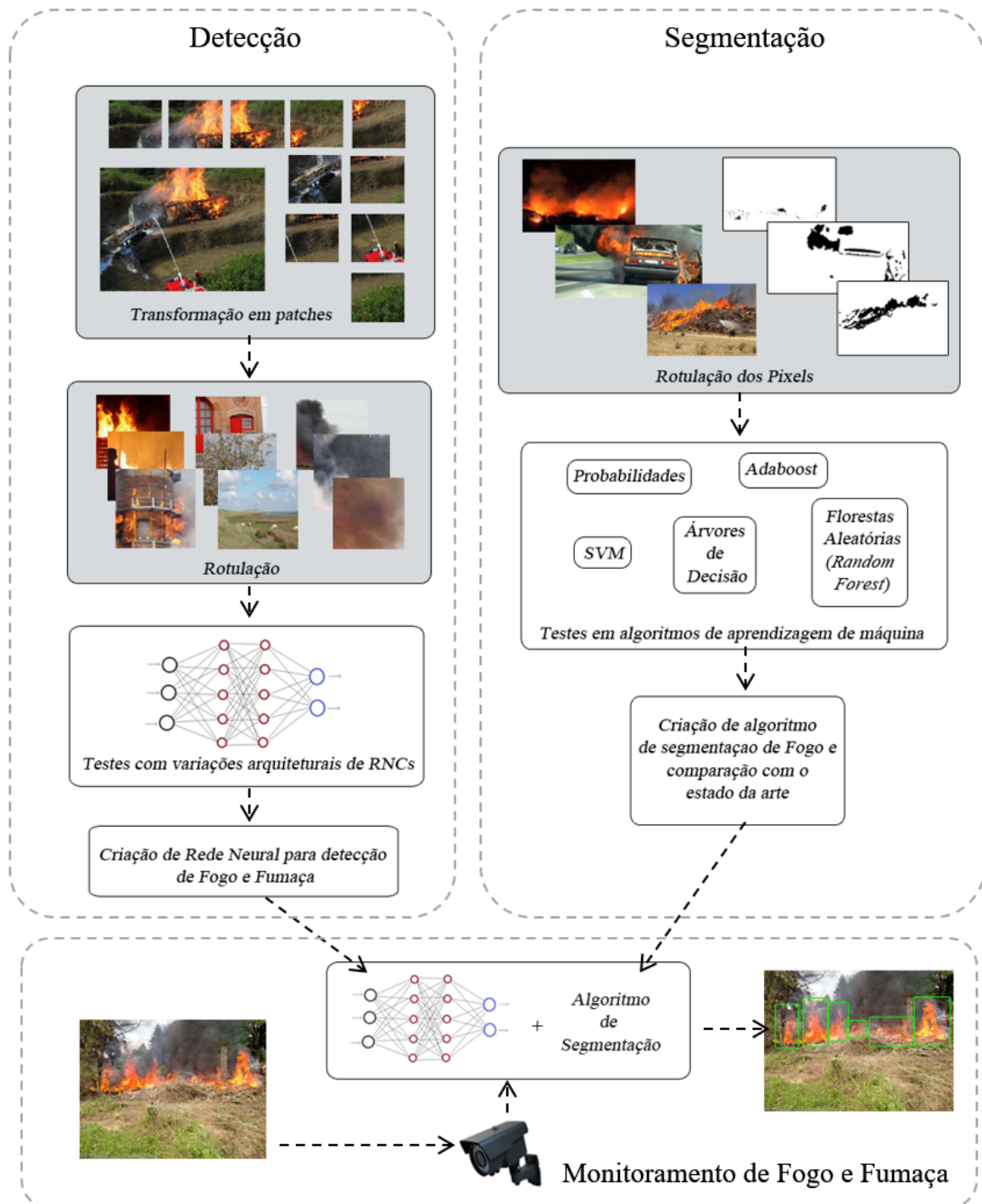


Figura 4.1: Visão Geral da proposta de Detecção de Fogo e Fumaça e Localização de Fogo em Imagens.

localização do fogo. A localização de fogo consiste em identificar por meio de um mapeamento pixel a pixel, se determinado pixel corresponde ou não a um pixel de fogo e ao final, caso um número considerável de píxeis seja localizado, indicar a região na imagem onde está localizado o foco do incêndio. Nessa etapa são testados alguns métodos de segmentação presentes na literatura, assim como experimentos com combinações de canais de cor não testados na literatura a fim de gerar uma nova forma de localização dos píxeis de fogo nas imagens.

A parte inferior da Figura 4.1 mostra a proposta de um método de monitoramento de fogo e fumaça e que realize a localização de fogo a partir da união das técnicas de Detecção e Segmentação. Este método está descrito com mais detalhes no Capítulo 7.

Nas seções seguintes são descritas como foram criadas as bases de imagens utilizadas neste trabalho (Seção 4.1), assim como a ponderação das considerações finais sobre a criação da base e da visão geral dos métodos (Seção 4.2).

4.1 Bases de Imagens

Um dos pré-requisitos fundamentais para criação de um modelo computacional capaz de reconhecer padrões é a base de dados. No caso de classificadores supervisionados para treinar um modelo eficiente, a base de dados deve ser representativa e capaz de generalizar o processo de reconhecimento de padrões para a maior parte dos problemas. Para poder avaliar a capacidade de generalização de um modelo é necessário possuir bases de dados distintas para treino, teste e validação. Assim, as bases de imagens criadas serão separadas proporcionalmente em 60% para treino, 20% para validação e 20% para teste.

A base de treino será utilizada para identificar a função que melhor reconhece as classes dos elementos da base e seus rótulos. A base de validação servirá de parâmetro para verificar quão eficiente está o processo de aprendizagem, assim como identificar possíveis problemas de sobre-ajuste (*overfitting*). Nesse sentido, as bases de treino e validação serão necessárias para identificação dos hiper-parâmetros mais adequados para o modelo de reconhecimento de padrões criado. E por fim, quando os resultados nas bases de treino e validação forem coerentes e o modelo for finalmente criado, a base de teste é utilizada para testar o modelo e poder ter indícios de quão generalista está o modelo de reconhecimento de padrões propostos.

As seções seguintes apresentam as bases de imagens para detecção de fogo e fumaça e localização de fogo. A Seção 4.1.1, apresenta as bases de imagens voltadas para detecção de fogo e fumaça e a Seção 4.1.2 apresenta a base de imagens para segmentação de fogo.

4.1.1 Base de imagem para detecção de fogo e fumaça

Para o problema de detecção de fogo e fumaça foram utilizadas três bases de imagens BD1 (Base de Detecção 1), BD2 (Base de Detecção 2) e BD3 (Base de Detecção 3), conforme Figura 4.2. A base – BD1 – foi criada pelo autor do trabalho e é composta de 101.134 *patches* de

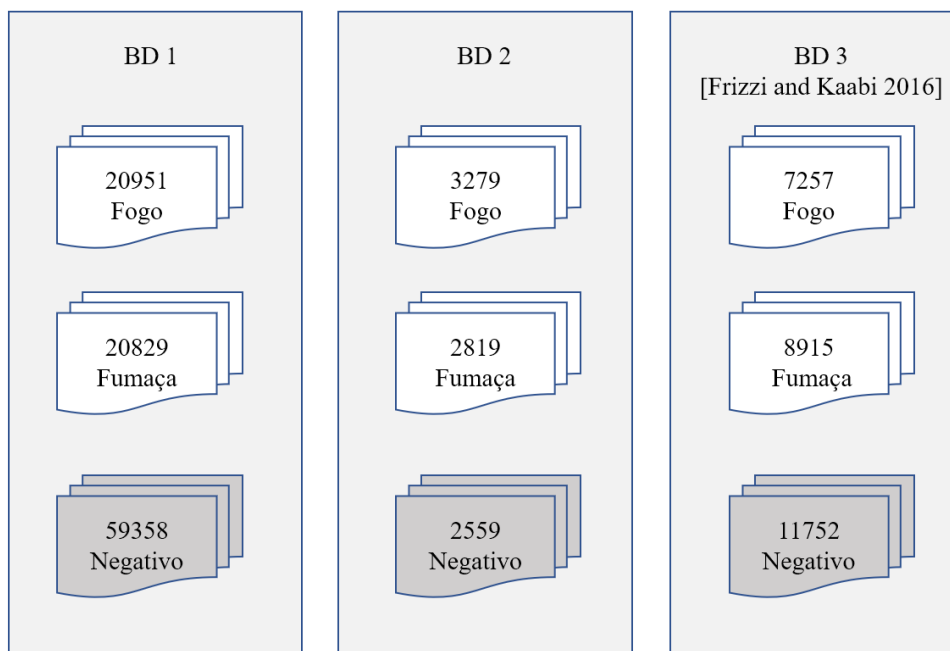


Figura 4.2: Disposição das Bases de Imagens para Detecção.

tamanho 299×299 píxeis divididos em treino, validação e teste, sendo que 62.281 imagens foram utilizadas para treino, 20.599 para validação e 18.254 para teste, adotando o padrão 60/20/20 utilizado para treinamento de algoritmos de aprendizado de máquina. Além desta base, uma segunda base, somente para testes – BD2 – foi criada pelo autor a partir de 2.531 imagens transformadas em 8.657 *patches*, esta base foi utilizada para mensurar a qualidade de detecção na imagem toda a partir das detecções realizadas nos *patches*. As imagens das bases BD1 e BD2 foram obtidas a partir de buscadores de imagens, como: Google Imagens, Baidu e Bing. Já a base – BD3 – foi criada por [Frizzi and Kaabi 2016] e contém 27.924 *patches* de tamanho 64×64 , sendo 8.915 imagens de fumaça, 7.257 imagens de fogo e 11.752 imagens negativas.

A seguir serão descritas as etapas de Aquisição e Rotulação utilizadas para a construção das bases BD1 e BD2. Além disso, será descrito como foi realizado o protocolo de avaliação dos métodos para as bases criadas.

Aquisição

Todas as imagens foram adquiridas por meio de motores de busca de imagens disponíveis: Google, Bing, Baidu e também na base de imagens Imagenet (Russakovsky et al., 2015). Para localizar as imagens, foram utilizadas como critério imagens com tamanho superior a 200×200 píxeis e duas palavras principais: Fogo (fire, fuego) e Fumaça (smoke, humo). Para buscar as imagens, a partir das duas palavras principais, foram realizadas combinações com as palavras dispostas na Tabela 4.1.

Tabela 4.1: Lista de Palavras utilizadas para busca de imagens de fogo, fumaça e não fogo.

Highway	Building	Structure	Grass	Rubbish	Urban
Vehicle	Calle	Industrial	Forest	Estrada	House
Truck	Casa	Bomb	Crop	Agriculture	

A ideia de utilizar diversas palavras chave tem como objetivo abarcar o maior número de contextos em que os incidentes de fogo acontecem. Com este procedimento mais de 30 mil imagens foram coletadas. Então, as imagens foram conferidas e separadas de forma manual em três categorias principais: Fogo, Fumaça e Negativo. Durante este processo de conferência, imagens que não tinham relação com o contexto, desenhos, imagens sintéticas, ou imagens de baixa de qualidade foram eliminadas. Neste processo cerca de 30% das imagens foram descartadas, o que resultou em aproximadamente 20 mil imagens com tamanhos distintos. Destas 20 mil, 2531 imagens foram separadas de forma aleatória para serem imagens de teste para avaliar a qualidade de detecção nas imagens como um todo e outras 18 mil para serem transformados em *patches* para treinamento das redes neurais convolucionais.

Rotulação

Após a seleção das imagens, das 17,5 mil imagens 101.138 *patches* de 299×299 píxeis foram extraídos utilizando a técnica de sobreposição. O tamanho dos *patches* foi escolhido baseado no tamanho aceito pela rede Inception (Szegedy et al., 2014). Vale ressaltar que imagens com menos de 299 píxeis de altura ou largura foram ampliadas para manter o padrão do tamanho da base. A técnica de extração dos *patches* pode ser explicada como uma janela que desliza sobre a imagem a partir de um deslocamento a cada 100 píxeis na horizontal e 100 píxeis na vertical. A Figura 4.3 ilustra este processo, onde cada quadrado representa um *patch* extraído da imagem; no exemplo temos 4 *patches* extraídos da imagem representados pelas cores vermelho, amarelo, azul e verde.

Tabela 4.2: Distribuição dos Patches em categorias e bases de treino, validação e teste.

Categoria	Treino	%	Validação	%	Teste	%
Fogo	12338	19,8	4115	20,6	4498	21,8
Fumaça	12436	20,0	4058	20,4	4333	21,0
Negativo	37508	60,2	11768	59,0	10082	57,1

Com os *patches* criados, então as imagens foram separadas manualmente entre as classes de Fogo, Fumaça e Negativo e distribuídas conforme mostrado na Tabela 4.2. Também foi realizada uma segunda rotulação nos *patches* indicando um dos seguintes contextos: Rodovia, Floresta, Estruturais, Pessoas e não definido. A Tabela 4.3 mostra a distribuição da base a partir destes contextos.



Figura 4.3: Exemplo de sobreposição para extração dos patches das imagens.

O contexto "Rodovia" envolve imagens de: rodovias, caminhões, carros e ruas. O contexto "Floresta" contém imagens de: áreas agrícolas, plantações, florestas e ambientes com vegetação esverdeada. Em relação ao contexto "Estrutural", as imagens englobam: casas, prédios industriais, edifícios e ambientes urbanos. O contexto "Pessoas", são *patches* que contenham pessoas no cenário. O contexto "não definido" contém somente: fogo, fumaça ou uma coloração uniforme. A Figura 4.4 mostra alguns exemplos de *patches* de fogo da base de imagens BD1, assim como a indicação dos contextos ao qual o *patch* faz parte. A Figura 4.5 mostra exemplos de *patches* negativos da base de imagens BD1, além de serem rotulados como negativos, temos os rótulos relacionados aos contextos logo abaixo de cada *patch*. A Figura 4.6 mostra exemplos de *patches* de fumaça da base de imagens BD1 e seus contextos.

Tabela 4.3: Distribuição dos patches nos respectivos contextos e classes.

Contexto	Fogo	%	Negativo	%	Fumaça	%
Rodovia	4517	30,6	15167	31,1	1816	10,2
Floresta	1715	11,6	4370	9,0	5273	29,5
Estrutural	5459	36,9	19011	39,0	4653	26,0
Pessoas	1377	9,3	5838	12,0	863	4,8
Não definido	1715	11,6	4370	9,0	5273	29,5

As 2.531 imagens que foram separadas para testar a qualidade de detecção foram transformadas em 8.657 *patches* com 299×299 píxeis. Esta base foi dividida em 3 classes: fogo,



Figura 4.4: Exemplos da base BD1 dos *patches* de fogo e seus contextos.



Figura 4.5: Exemplos da base BD1 dos *patches* Negativos e seus contextos.



Figura 4.6: Exemplos da base BD1 dos *patches* de fumaça e seus contextos.

fumaça e negativo, totalizando 3.279 *patches* de fogo, 2.559 *patches* negativos e 2.819 *patches* de fumaça.

Para avaliação dos métodos de detecção de fogo foram utilizadas as seguintes métricas: Acurácia, F_1score , MAE, Loss e Taxa de detecção. A taxa de detecção foi aplicada somente em BD2, pois para esta base é possível realizar o mapeamento dos *patches* para a imagem original. Na Tabela 4.4 pode-se notar o arquivo, a classe à qual a imagem pertence (Fogo, Negativo ou Fumaça), a quantidade de *patches* que a imagem possui e a distribuição numérica de cada *patch* para Fogo (F), Negativo (N) e Fumaça (S). Complementar à Tabela, temos a Figura 4.7 que mostra um exemplo para cada tipo de imagem (fogo, negativo e fumaça) e seus respectivos *patches* gerados.

Tabela 4.4: Exemplos da divisão das imagens por *patches*.

Arquivo	Classe	Num. <i>patches</i>	F	N	S
ho_f (104).jpg	Fogo	4	3	1	0
ho_f (10).jpg	Fogo	4	3	1	0
indf_n (14).jpg	Neg.	4	0	4	0
hosm_n (53).jpg	Neg.	4	0	4	0
hw_lo_sm (2).jpg	Fum.	3	0	0	3
indf_sm (99).jpg	Fum.	5	0	2	3

Como critério para considerar acerto da detecção de fogo foi adotado o seguinte: caso a imagem for da classe fogo e pelo menos em um dos *patches* da imagem for detectado fogo de forma correta, então houve acerto na detecção de fogo para esta imagem. Para o cálculo da detecção de fumaça a situação é análoga ao cálculo de detecção de fogo.



Figura 4.7: Exemplos de imagens com os *patches* e seus respectivos rótulos.

4.1.2 Base de imagem para localização de fogo

Para os experimentos de localização de fogo foram utilizadas ao todo 3 bases de imagens, conforme ilustra a Figura 4.8. A base 1 – nomeada de BL1 – foi publicada por Chino et al. (2015) e contém 226 imagens, sendo 117 imagens de fogo em situações diversas e 109 imagens de não fogo. Esta base foi utilizada como base de treino. A segunda base – BL2 – contém 110 imagens, sendo 55 imagens de fogo e 55 imagens de não fogo e foi utilizada como base de teste para dois experimentos que serão descritos no capítulo 6 e foram publicadas em (Souza and Facon, 2016) e (Souza et al., 2017). A terceira base utilizada – BL3 – foi criada com o intuito de se ter uma base mais representativa e separada em 5 cenários: Floresta, Rural, Urbano, Industrial e Rodovias. O total de imagens desta base é de 1038 imagens, sendo 516 imagens de fogo. Esta divisão, tornou possível o teste de métodos de aprendizagem de máquina para diferentes cenários.

A base BL1 possui 119 imagens de fogo; nestas imagens 9.120.396 píxeis são de fogo e 90.526.455 são píxeis de não fogo. A base BL2 possui 55 imagens de fogo, sendo 1.224.967 píxeis de fogo e 16.359.833 píxeis de não fogo. O desenvolvimento da base BL2 foi realizado por meio da coleta de imagens de situações de fogo, e estas situações foram categorizadas da seguinte forma: Veículos, Rodovias, Florestas e áreas urbanas. A Figura 4.9, mostra alguns exemplos de imagens desta base, que foram coletadas e selecionadas a partir de buscas no *Google* Imagens.

Para cada imagem foi então realizada a segmentação manual, com o objetivo de criação das imagens com a solução ideal (*Ground truth*). A Figura 4.10 mostra o resultado da segmentação manual.

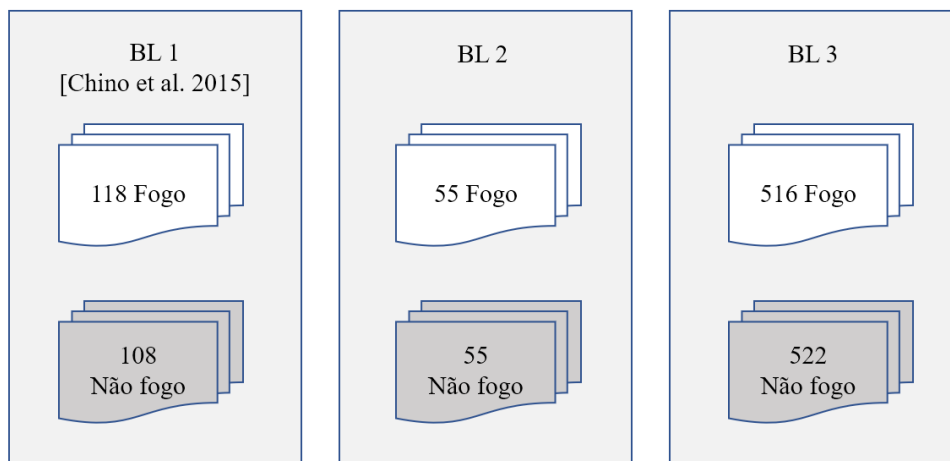


Figura 4.8: Disposição das Bases de Imagens para Localização.

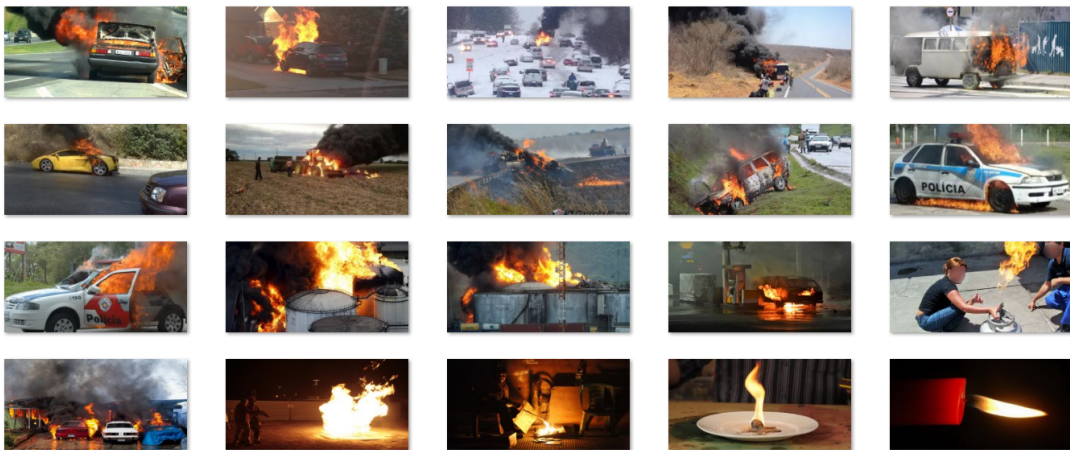


Figura 4.9: Exemplo de imagens de fogo da base de imagens BL2.



Figura 4.10: Exemplo de solução ideal para as imagens de fogo da BL2.

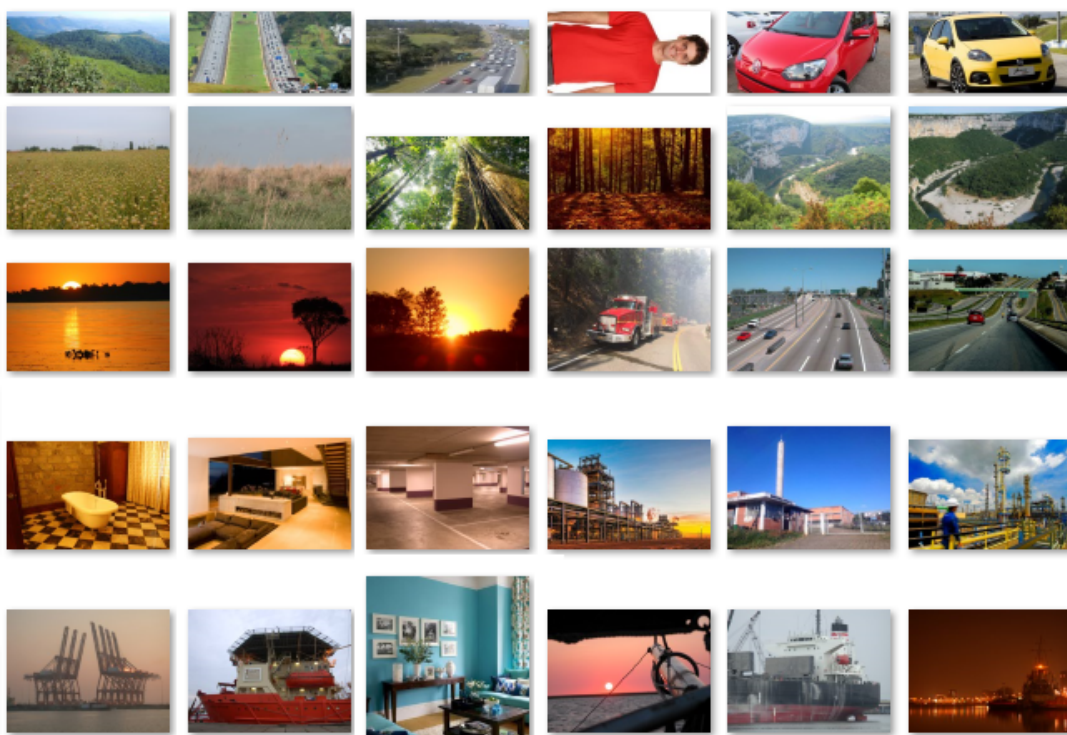


Figura 4.11: Exemplos de imagens de não fogo da BL3.

Com o intuito de criar uma base com mais amostras, a fim de melhorar o processo de aprendizagem, foi criada uma base de imagem, BL3, com a segmentação ideal das imagens de fogo e não fogo. No total, a base é composta por 1038 imagens, sendo 516 de fogo e 522 imagens de não fogo. Além disso, as imagens foram separadas em 5 contextos/cenários: Floresta, Rural, Indústria, Urbano e Rodovia. A Figura 4.11 ilustra uma amostra de imagens de não fogo da base BL3.

Para realizar a coleta de imagens para esta base, foram utilizadas as imagens das bases BL1 e BL2 que totalizam 336 imagens, as outras 702 imagens foram coletadas utilizando bancos de imagens públicos. As imagens de fogo foram então segmentadas de forma manual por 3 especialistas em segmentação de imagens. A Tabela 4.5 traz a distribuição da quantidade de píxeis de fogo presente nas imagens de fogo selecionadas.

Tabela 4.5: Distribuição dos píxeis de fogo nas imagens de fogo coletadas.

Base	Total Imagens	píxeis de Fogo	píxeis de Não Fogo	% fogo
Floresta	102	2.984.036	36.369.054	7,5%
Indústria	113	2.879.626	57.192.719	4,7%
Rural	103	2.215.781	37.052.964	5,6%
Rodovia	100	1.078.479	30.773.556	3,38%
Urbano	100	6.408.037	54.780.278	10,4%

4.2 Considerações Finais

Neste capítulo foram apresentadas a visão geral do método para o desenvolvimento deste trabalho e as bases de imagens utilizadas. A partir das bases de imagens fica possível o treino de modelos de aprendizagem de máquina para detecção de fogo e fumaça e localização de fogo. O detalhamento dos métodos propostos, assim como os resultados obtidos, estão nos capítulos seguintes divididos da seguinte forma: detecção de fogo e fumaça no Capítulo 5, localização de fogo no Capítulo 6 e a proposta de um método para monitoramento de fogo presente no Capítulo 7.

Capítulo 5

Detecção de fogo e fumaça

Para criar um processo de detecção de fogo e fumaça foram utilizadas as seguintes etapas metodológicas: coleta de imagens e criação da base de imagens; estudo das arquiteturas de RNC – Redes Neurais Convolucionais; proposição e testes de diferentes arquiteturas de RNC; teste da base criada com as principais arquiteturas de RNC existentes na literatura e um estudo comparativo com as arquiteturas propostas. O diagrama na Figura 5.1 ilustra o fluxo das etapas desenvolvidas.

Visando à construção de um modelo eficiente de detecção de fogo e fumaça construiu-se uma base de imagens contendo situações de fogo e fumaça em diferentes cenários, descrita na Subseção 4.1.1 do Capítulo 4. A partir desta base foi possível idealizar diversos experimentos utilizando das Redes Neurais Convolucionais como modelo de aprendizagem de máquina a ser adotado para este propósito.

A escolha de modelos baseados em Redes Neurais Convolucionais – RNC – deu-se principalmente pelo fato deste modelo ser o que apresenta os melhores resultados em relação à detecção de objetos, como pode ser verificado nos resultados do concurso ILSVRC (Russakovsky et al., 2015). Tendo em vista estudar os diversos modelos existentes de RNC, foram testados e modificados 5 modelos arquiteturais, de RNC existentes na literatura: VGG-16 (Simonyan and Zisserman, 2014), Resnet (He et al., 2015a), Inception (Szegedy et al., 2014), Xception (Chollet, 2016), MobileNet (Howard et al., 2017). Além disso, arquiteturas específicas propostas para detecção de fogo e fumaça, Frizzi et al. (2016), Yin et al. (2017), Tao and Wang (2016) e a versão reduzida de RNC proposta por Zhang et al. (2016), também foram testadas.

A fim de padronizar as modificações das redes, as camadas totalmente conectadas para os modelos arquiteturais não específicos para detecção de fogo e fumaça foram modificadas com as seguintes configurações: duas camadas totalmente conectadas com 1024 neurônios cada uma e a aplicação da técnica de *Dropout* de 0,5 entre estas camadas. A Subseção 5.1 descreve com mais detalhes os métodos experimentais adotados para que fosse possível realizar um estudo comparativo entre os diferentes modelos de RNCs. Da mesma maneira, a Seção 5.2.1 apresenta os resultados obtidos a partir desses métodos.

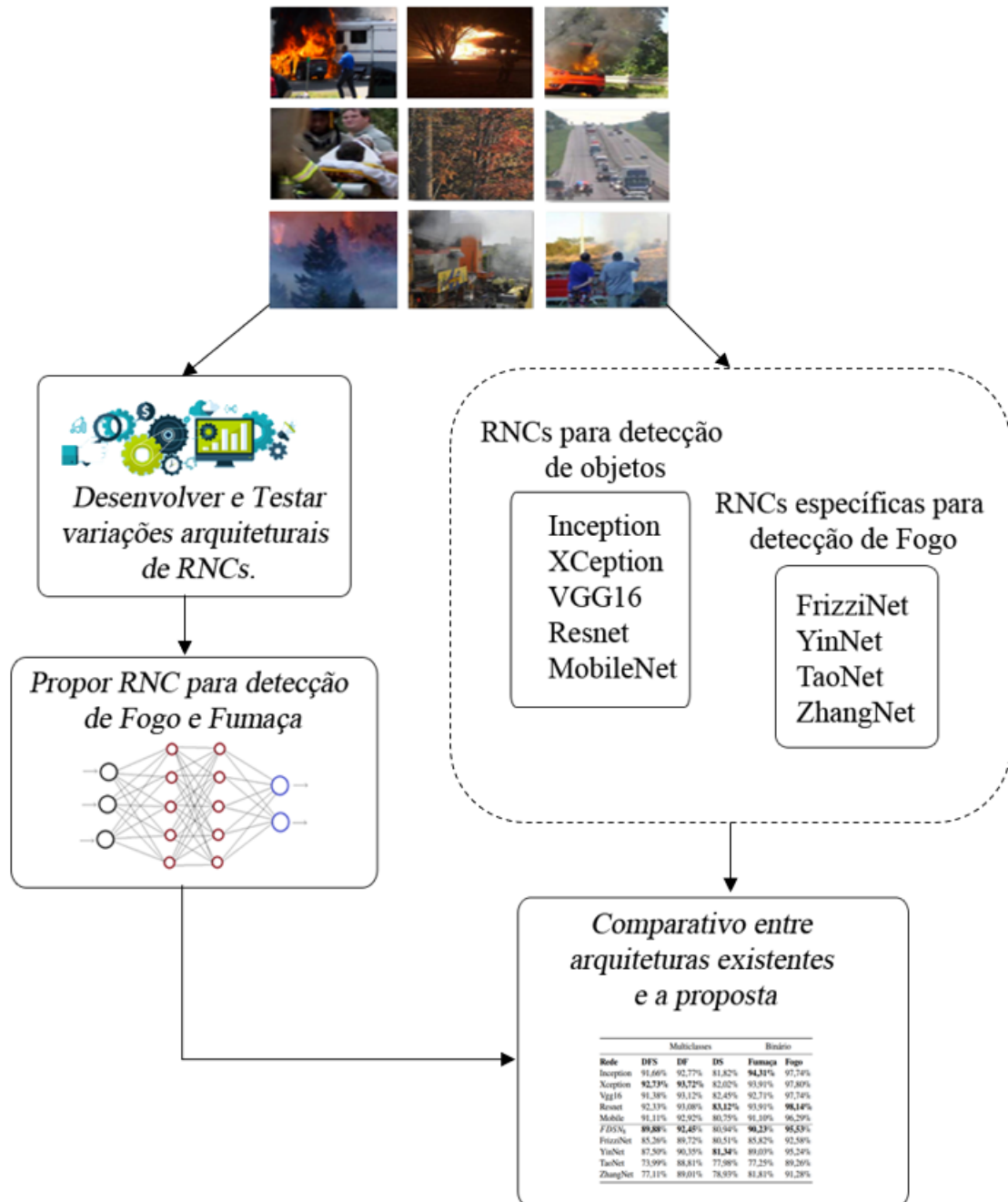


Figura 5.1: Etapas metodológicas utilizadas para proposição de arquitetura de Rede Neural Convolutiva voltada para detecção de fogo e fumaça.

Além dos testes com as arquiteturas presentes na literatura modificadas, foram criadas variações dos modelos arquiteturais de RNCs, conforme método descrito na Subseção 5.1, com o objetivo de criar uma rede neural específica para a detecção de fogo e fumaça. Como as redes VGG-16, Inception, Resnet, MobileNet e Xception foram idealizadas para resolver o problema de detecção de objetos para mais de 1000 classes, uma rede criada especificamente para detecção de fogo e fumaça pode resultar em uma rede menor e com eficiência similar. A Subseção 5.2 descreve cada uma das arquiteturas criadas, assim como apresenta o comparativo entre elas.

5.1 Método proposto para detecção de fogo e fumaça

Conforme descrito previamente, as arquiteturas de RNCs selecionadas foram: Inception (Szegedy et al., 2014), VGG-16 (Simonyan and Zisserman, 2014), Resnet (He et al., 2015a) e Xception (Chollet, 2016). Além destas arquiteturas, a MobileNet (Howard et al., 2017) também foi testada por ser uma proposta de RNC mais compacta, voltada para sistemas embarcados em tempo real, e com taxas de erro similar ao das arquiteturas vencedoras do ILSVR. Em relação às redes neurais específicas para detecção de fogo e ou fumaça foram selecionados as redes propostas pelos seguintes autores: Yin et al. (2017), Tao and Wang (2016), Zhang et al. (2016) e Frizzi et al. (2016). Os testes foram realizados nas bases de imagens BD1, BD2 e também na base de imagens BD3 proposta por Frizzi et al. (2016).

Para a condução dos testes foi realizada a divisão das imagens na base BD1 em: treino, teste e validação, descritas na Subseção 4.1.1. Além dessa divisão foram separadas as bases contendo imagens de fogo e negativo, assim como imagens de fumaça e negativo para realizar testes em redes binárias e avaliar se a RNC é mais eficiente tratando dos problemas de fogo e fumaça ao utilizar abordagem binária ou a abordagem multi-classe; isso se justifica pela hipótese de que pode haver confusão no aprendizado da RNC, uma vez que algumas imagens de fogo possuem fumaça e algumas imagens de fumaça possuem fogo.

As imagens processadas por uma rede neural convolucional – RNC – necessitam ser do tamanho da entrada suportada pela rede, além de serem imagens com resolução quadradas. Para sistemas de detecção em tempo real, as imagens normalmente são retangulares, além de possuírem tamanhos variados. Assim, para medir a eficiência da detecção em uma imagem como um todo é preciso separar esta imagem em quadrados (patches) compatíveis com a entrada da RNC, além de possibilitar mapear a imagem de origem do patch processado.

A base de imagens BD2 descrita na Subseção 4.1.1 foi criada com o propósito de possibilitar o mapeamento dos patches para as suas imagens de origem e assim poder calcular a taxa de detecção alcançada pelas RNCs. Esta base contém 8.943 patches resultantes de 2.530 imagens. Dessa forma, os resultados presentes na Tabela 5.9 mostram as taxas de detecção obtidas para cada uma das arquiteturas de *deep learning* estudadas. Para realizar o teste nesta base, os pesos das redes obtidos por meio do treinamento na base de treino BD1 foi carregado. A escolha desta base de imagens se deve ao fato de que os modelos gerados pela base de treino de

BD1 apresentaram melhores resultados para Acurácia e $F_1 score$ na base de teste BD2 quando comparados aos resultados destas métricas a partir dos modelos treinados pela base de treino da BD3.

Uma vez que as bases de imagens foram preparadas partiu-se para a fase de configuração e otimização das redes neurais. Em um primeiro momento, as camadas totalmente conectadas das redes testadas foram modificadas e configuradas para duas camadas de 1024 neurônios cada e a utilização de *Dropout* entre elas, assim como emitir como saída uma das classes: fogo, fumaça ou negativo. Além dessa modificação nas redes, os valores para os seguintes parâmetros foram testados: taxa de aprendizado, decay e momentum. Estes parâmetros são fundamentais para determinar o aprendizado da RNC. Mais adiante, mais detalhes a respeito do método adotado para identificação dos hiperparâmetros serão apresentados

Para comparar a qualidade de detecção de fogo com os métodos existentes na literatura, a base de localização BL3 é utilizada para ser comparada em relação à taxa de detecção. Para avaliar a taxa de detecção das RNCs desenvolvidas especificamente para detecção de fogo e/ou fumaça existentes na literatura (Frizzi et al., 2016; Yin et al., 2017; Zhang et al., 2016; Tao and Wang, 2016) as imagens da base BL3 foram transformadas em patches para serem utilizadas como entrada nas redes neurais. Além da avaliação da detecção nas RNCs, também são avaliados 12 métodos que utilizam regras para identificação dos píxeis de fogo ((Çelik et al., 2006, 2007; Çelik and Demirel, 2009; Çelik, 2010; Chen et al., 2004, 2010; Ko et al., 2009; Phillips et al., 2002; Horng et al., 2005; Rossi et al., 2009; Santana et al., 2012; Vipin, 2012)).

Os experimentos foram conduzidos utilizando um computador equipado com uma placa GPU GTX GeForce Titan com capacidade de 12 GB de memória RAM e 2.688 núcleos cuda, cedida gentilmente pela NVIDIA. O framework utilizado para a criação e teste das arquiteturas foi o Keras (Cholet, 2015), utilizando como base o Tensor-flow.

5.1.1 Configurações dos parâmetros em Redes Neurais Convolucionais

Em relação à escolha dos hiperparâmetros das redes neurais convolucionais, vale destacar o parâmetro da taxa de aprendizado (*learning rate*), pois o tamanho deste parâmetro influencia diretamente na taxa de aprendizado dos modelos, uma vez que é o parâmetro responsável por delimitar qual intervalo de busca do mínimo local deve ser utilizado. Ou seja, quanto menor a taxa de aprendizado mais demorado será o processo de localização do erro mínimo, conseqüentemente levará mais tempo para o modelo de aprendizagem de máquina convergir.

Para auxiliar no processo de localização da taxa de aprendizado ideal, a técnica proposta por (Smith, 2015) foi utilizada, cujo nome é *Cyclical Learning Rate* (Taxa de aprendizagem cíclica) – CLR. Esta técnica consiste em variar o valor da taxa de aprendizagem durante a execução de uma época (*epoch*) da rede neural convolucional, com isso há a identificação de diferentes mínimos locais, o que leva a RNC convergir para resultados satisfatórios utilizando

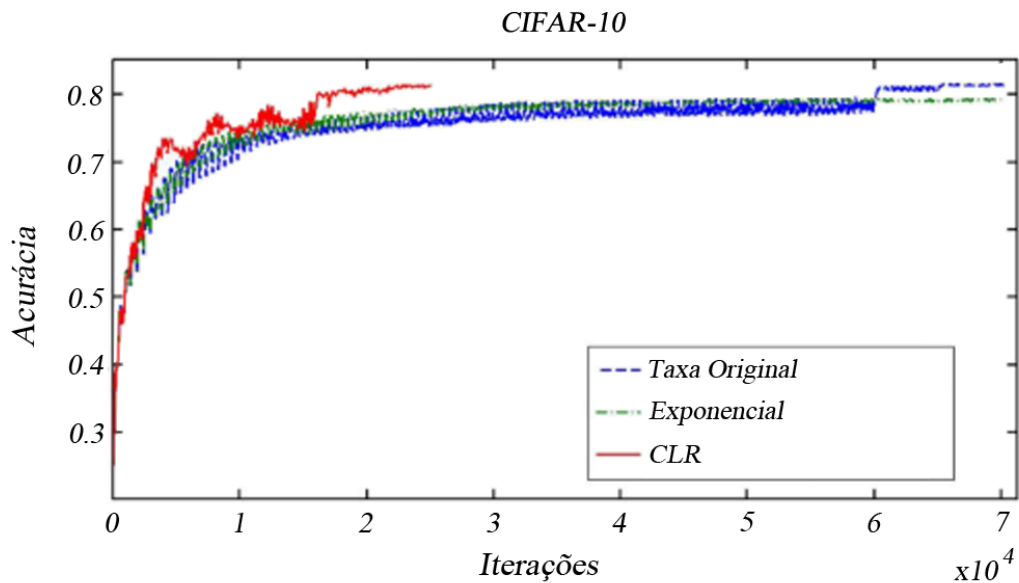


Figura 5.2: Evolução da acurácia utilizando CLR (em vermelho) e não utilizando (em azul e verde). Fonte: Smith (2015).

menos iterações e gerando economia de tempo computacional. A Figura 5.2 mostra a evolução das iterações nos testes realizados utilizando a técnica CLR na base de imagens CIFAR-10.

Em relação à taxa de aprendizagem é esperado identificar os valores ótimos para o aprendizado por meio do teste com diferentes valores para este hiperparâmetro. Nesse sentido, como exemplo a Figura 5.3 mostra a evolução da métrica *loss* com a variação da taxa de aprendizagem. Pelo gráfico nota-se que a queda da métrica *loss* ocorre entre os valores de 10^{-4} e 10^{-2} para a taxa de aprendizagem (*learning rate*). Portanto, este é o intervalo de taxa de aprendizado que é utilizado, neste caso, como base para os experimentos com as arquiteturas de redes neurais convolucionais.

Para localizar outros hiperparâmetros como *decay* e *momentum* foram realizados testes com diferentes valores. Neste sentido, de acordo com os valores para estes hiperparâmetros encontrados na literatura o intervalo de 10^{-4} a 10^{-6} para *decay* e o intervalo de 0,7 a 0,95 para o *momentum* foram testados. A fim de padronizar os testes com estes hiperparâmetros, o tamanho do lote (*batch*) foi fixado em 32 e pelo menos cinco épocas foram executadas para avaliar o desempenho dos parâmetros.

A partir deste protocolo foi possível observar o desempenho da rede em relação às métricas adotadas e identificar os melhores valores para os parâmetros. Para *decay* os melhores resultados foram 10^{-5} e para o *momentum* os melhores resultados foram para o valor 0,9.

Após a localização dos hiperparâmetros, o seguinte protocolo experimental foi adotado para treinamento das RNCs estudadas:

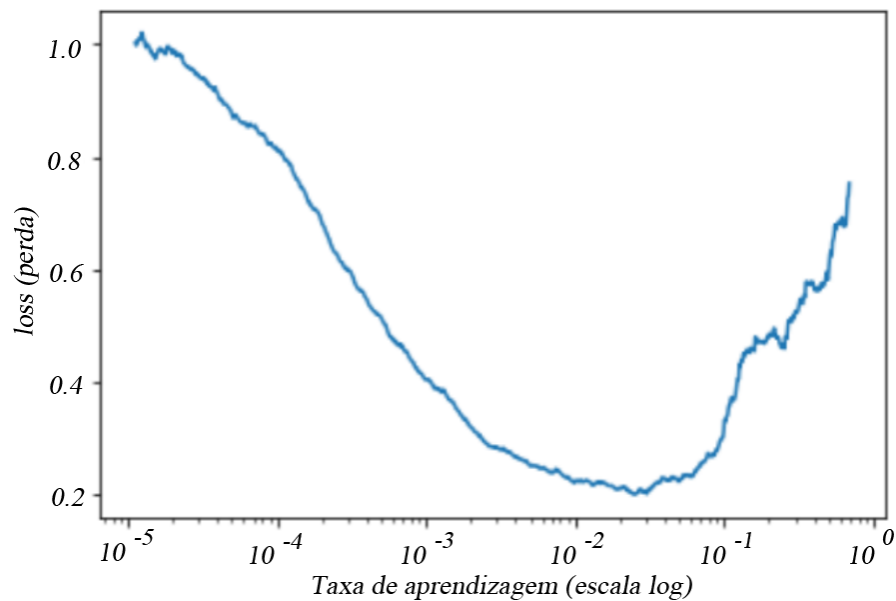


Figura 5.3: Evolução da taxa de aprendizagem em relação a função de perda (*loss*).

- Modificar as camadas totalmente conectadas das arquiteturas selecionadas para a seguinte configuração: 2 camadas com 1024 neurônios cada e com a aplicação do *dropout* de 50% entre estas camadas, para evitar o sobre-ajuste;
- Carregamento dos pesos das redes treinadas pela base de imagens do Imagenet, excluindo os pesos das camadas totalmente conectadas (que foram modificadas);
- Realização de pelo menos 5 épocas para o treinamento exclusivo das camadas totalmente conectadas, congelando as outras camadas com os pesos já carregados;
- Salvar os pesos da rede que obtiverem melhores desempenhos para a métrica *loss*;
- Realizar o treinamento da rede toda, descongelando as camadas para treinamento e carregando os pesos após treinamento das camadas totalmente conectadas por pelo menos 30 épocas;
- Salvar os pesos quando há melhora na taxa de *loss*;
- Executar o modelo salvo na base de testes.

Para as redes Inception, Resnet, VGG16, Xception e MobileNet são utilizados pesos que foram previamente treinados na base ImageNet (Russakovsky et al., 2015). Para as arquiteturas onde não são disponibilizados os pesos pré-treinados, o treinamento da rede é realizado do zero; assim, após as otimizações necessárias, o protocolo adotado para parar o treinamento foi a não melhora das taxas de *loss* por pelo menos 15 épocas.

Outro experimento realizado foi a utilização de transferência de aprendizagem (*transfer learning*) que consiste em utilizar uma rede treinada em uma base e realizar testes em outra base. Para isso, planeja-se treinar as redes nas bases BD1 e BD3 e avaliar o desempenho das métricas nas bases de teste BD1 e na base BD2.

Além dos testes de desempenho da rede em relação às métricas de avaliação, planeja-se aplicar testes de tempo de processamento das imagens, assim como mensurar o tamanho das redes por meio do número de parâmetros e tamanho do arquivo de pesos das redes no disco. A Tabela 5.10 mostra o resultado deste teste.

Com o objetivo de identificar se RNCs binárias são mais eficientes que redes multiclases, idealiza-se realizar um experimento comparativo com estes dois tipos de rede. Desta forma, faz-se necessário o treinamento de três diferentes redes: rede binária para detecção de fogo, rede binária para detecção de fumaça e uma rede multiclasse para detecção de fogo ou fumaça.

5.1.2 Proposta e avaliação de diferentes arquiteturas de RNCs

Para propor uma nova arquitetura de Rede Neural específica para detecção de fogo e fumaça, variações de modelos arquiteturais foram explorados. Tais variações dizem respeito a aspectos que compõem uma Rede Neural Convolutiva, entre eles estão: o número de camadas de convolução, número de camadas de *pooling*, o número de características para cada camada, o tamanho do kernel, dimensões das camadas totalmente conectadas e a escolha do classificador a ser aplicado aos mapas de características gerados pela rede. Para estes modelos de propostas arquiteturais foi dado o nome de Fire and Smoke Detection Network - *FSDN*. Ao todo, dez variações arquiteturais foram testadas. Os testes foram executados nas bases de treino e validação da BD1 e os resultados estão presentes na Seção 5.2.1, na Tabela 5.1 que apresenta os valores obtidos para cada uma das arquiteturas testadas.

Em 5 arquiteturas testadas (*FSDN*₁, *FSDN*₂, *FSDN*₃, *FSDN*₄ e *FSDN*₅), o fluxo da rede é realizado de forma serial. Estas arquiteturas possuem camadas de convolução, em que foram variados a quantidade de características dessas camadas, assim como em algumas arquiteturas aplicando a técnica de Zero Padding que consiste em preencher os pixels além da borda da imagem de entrada com zeros, o que faz com que a cada camada o tamanho do mapa de características (imagem de saída da camada) não se altere, mantém-se do mesmo tamanho. Estas 5 arquiteturas testadas tinham como objetivo aferir redes neurais de tamanho bem reduzido e avaliar o desempenho destas redes nas métricas de acurácia e *F1 score* principalmente. Outras 5 arquiteturas (*FSDN*₆, *FSDN*₇, *FSDN*₈, *FSDN*₉ e *FSDN*₁₀) um pouco mais robustas também foram avaliadas, etapa em que se buscou avaliar alguns conceitos presentes na literatura, como os conceitos de blocos inception e inception extremo, presentes nos trabalhos de Szegedy et al. (2014) e Chollet (2016), assim como a utilização de blocos residuais proposto por He et al. (2015a). O que se buscou com os experimentos foi identificar uma arquitetura capaz de aliar eficiência em detecção e velocidade de processamento das imagens.

A partir da definição do método, as Subseções 5.2 e 5.2.1 respectivamente apresentam os resultados experimentais com os modelos arquiteturais testados e um comparativo de desempenho entre o modelo arquitetural proposto e os presentes na literatura.

5.2 Resultados experimentais das propostas de modelos arquiteturais de RNCs

Com o objetivo de construir uma Rede Neural Convolutacional específica para a detecção de fogo e fumaça foram criadas e testadas 10 variações arquiteturais. A estas variações foram dados o prefixo no nome de FSDN, sigla que abrevia os termos em inglês - *Fire Smoke Detection Network*. As variações consistem em testar diferentes estruturas internas na rede que consistem em: variar o número de filtros nas camadas convolucionais, definir o tamanho da camada totalmente conectada para realizar a classificação dos objetos, definir a quantidade de camadas da rede, assim como o formato de cada bloco de convolução. A Tabela 5.1 mostra o resultado da detecção de fogo e fumaça nas arquiteturas testadas nas bases de treino e validação da base de imagens *BD1*.

Para conduzir o experimento, o seguinte procedimento foi adotado: as redes foram criadas para classificação de três classes: fogo, fumaça ou negativo. As taxas de aprendizagem (*learning rate*) foram fixadas no intervalo entre 0,0001 e 0,01 utilizando a técnica de taxa de aprendizagem cíclica (CLR - (Smith, 2015)). O método iterativo de otimização da rede foi o Gradiente Descendente Estocástico (*Stochastic Gradient Descent*) com valores de *momentum* de 0.9 e *decay* de 1×10^{-5} . As redes foram colocadas para serem executadas por pelo menos 30 épocas (aproximadamente 3.150 iterações por época) e a métrica de *loss* foi utilizada como parâmetro de escolha do melhor modelo criado durante a fase de treinamento da rede.

A primeira proposta arquitetural, $FSDN_1$, conforme ilustrado pela Figura 5.4, possui 5 camadas de convolução, utilizando a técnica do Zero Padding, ou seja uma borda com zeros é preenchida ao redor da imagem para que o processo de convolução não diminua o tamanho do mapa de características resultante, utilizando um kernel de tamanho 2×2 . Nas camadas finais da rede foram utilizadas duas camadas totalmente conectadas com 128 neurônios em cada uma delas e entre uma camada e outra foi aplicada a técnica de *dropout* que consiste em desativar parte da rede no momento da classificação a fim de diminuir a possibilidade de sobre-ajuste. Ao final da rede o classificador softmax foi o encarregado de classificar entre três classes o mapa de características gerado pela RNC.

A proposta arquitetural $FSDN_2$, ilustrada na Figura 5.5, possui menos camadas que a $FSDN_1$, porém, as camadas convolucionais são maiores, com 64 e 128 filtros. A técnica de dropout foi utilizada em dois momentos: no meio da rede entre duas camadas de convolução de 128 filtros e ao final da rede entre as camadas totalmente conectadas. Na saída da última camada totalmente conectada então aplica-se o softmax para classificar às saídas da rede.

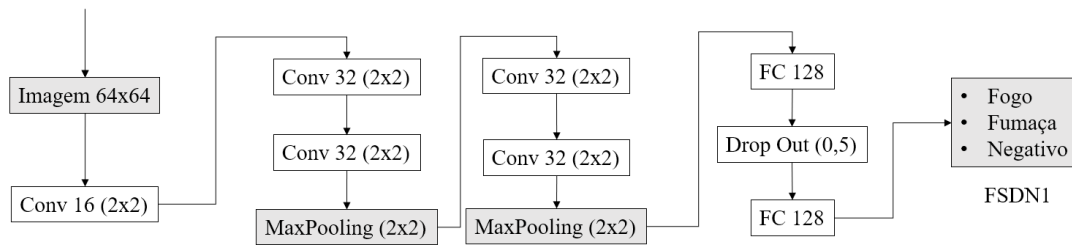


Figura 5.4: Rede Neural $FSDN_1$.

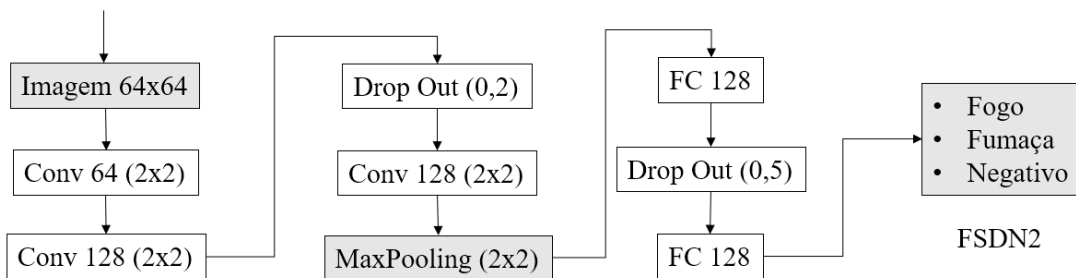


Figura 5.5: Rede Neural $FSDN_2$.

A proposta arquitetural $FSDN_3$, conforme Figura 5.6, é a menor arquitetura que foi testada, possui 3 camadas de convolução com no máximo 32 filtros. A técnica de dropout também foi utilizada ao final da rede entre duas camadas totalmente conectadas e com 64 neurônios. Finalmente, na saída da última camada totalmente conectada então aplica-se o softmax para classificar as saídas da rede. Esta arquitetura mostrou que ao diminuir o tamanho da última camada totalmente conectada, assim como diminuir demais as camadas convolucionais não são benéficas para a rede de detecção de fogo.

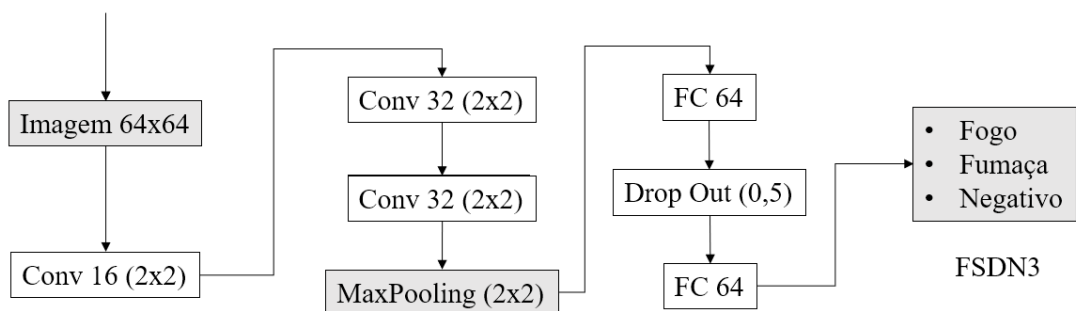


Figura 5.6: Rede Neural $FSDN_3$.

A proposta arquitetural $FSDN_4$, conforme Figura 5.7, possui 6 camadas de convolução, e assim como a arquitetura $FSDN_2$, a técnica de *dropout* também foi utilizada por duas vezes: no meio da rede entre duas camadas de convolução de 64 e 32 filtros e ao final da rede entre

as camadas totalmente conectadas. A inclusão de mais camadas intermediárias em relação à arquitetura proposta $FSDN_2$ trouxe desempenho melhor na base de testes.

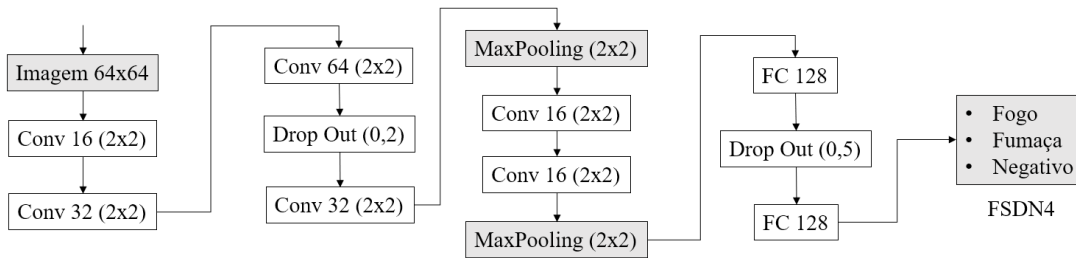


Figura 5.7: Rede Neural $FSDN_4$.

A proposta arquitetural $FSDN_5$, ilustrada pela Figura 5.8, mostrou a importância da camada totalmente conectada; ao aumentar esta camada foi possível otimizar o desempenho da rede, conforme refletido nos resultados sobre as bases de treino e teste. A arquitetura proposta 5 possui a mesma arquitetura da proposta 1, porém com duas mudanças: a não aplicação da técnica de Zero Padding entre as 3 primeiras camadas convolucionais e também que as camadas totalmente conectadas são maiores, contendo 1024 neurônios por camada. Como resultado, esta proposta foi levemente superior à $FSDN_1$.

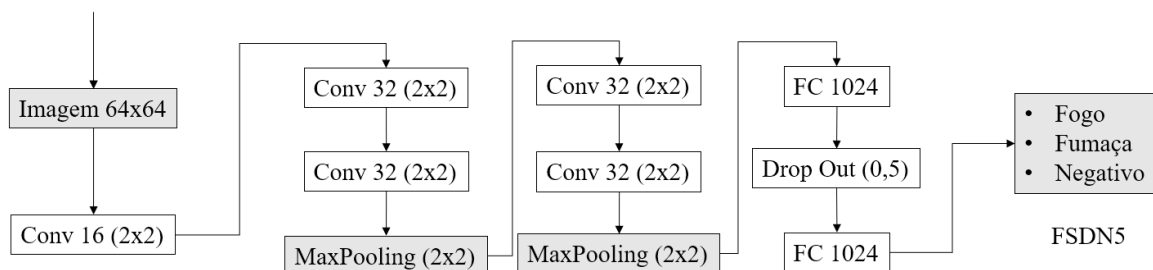


Figura 5.8: Rede Neural $FSDN_5$.

A proposta arquitetural $FSDN_6$ utilizou técnicas presentes na literatura e que otimizam o processo de detecção conforme a evolução dos resultados na competição ILSVRC (Russakovsky et al., 2015). Entre as técnicas, pode-se citar a utilização de camadas residuais, conforme adotado por He et al. (2015a), além das camadas inception apresentadas pela equipe do Google com as redes GoogLeNet (Szegedy et al., 2015) e Xception (Chollet, 2016). Para utilizar a técnica do bloco Inception, o framework adotado (Keras - Cholet (2015)), possui um método chamado de *Separable Convolutions* que consiste em realizar convoluções em profundidade atuando em cada canal de entrada de forma separada e ao final do processo realizar a união das saídas obtidas.

A partir das teorias expostas, a $FSDN_6$ foi proposta, conforme ilustra a Figura 5.9. Esta arquitetura é composta de 8 camadas convolucionais residuais, 2 camadas convolucionais convencionais, 22 blocos inception e duas camadas totalmente conectadas com 1024 neurônios cada.

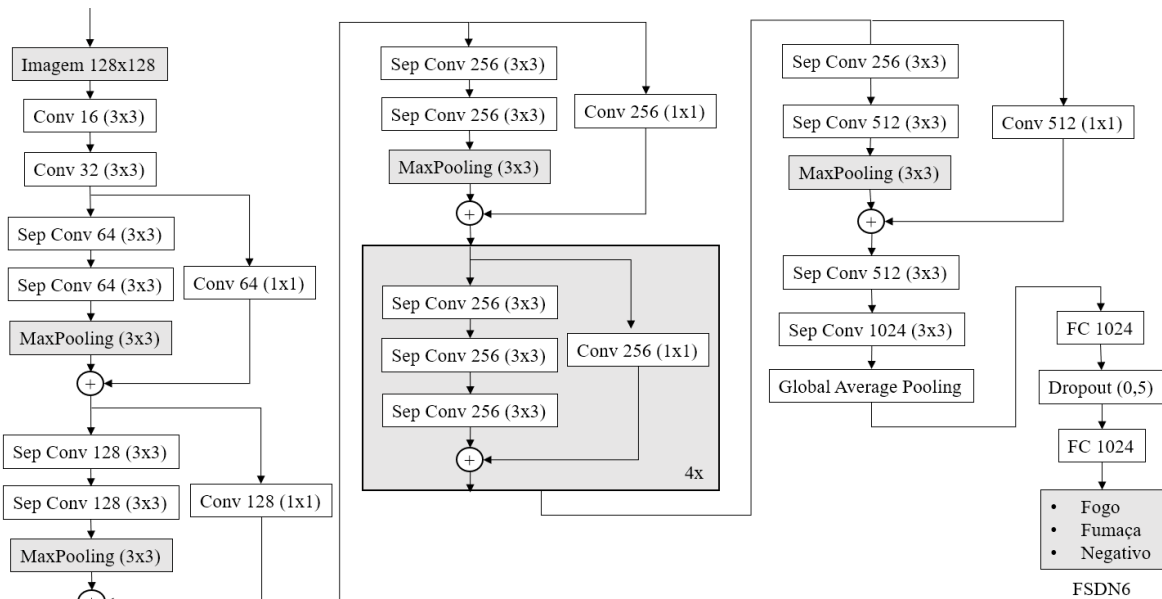


Figura 5.9: Redes Neurais $FSDN_6$.

Partindo das teorias utilizadas para a concepção da $FSDN_6$ e com os resultados obtidos por esta arquitetura, as redes $FSDN_7$ e $FSDN_8$ foram idealizadas com o objetivo de serem mais compactas e tão eficientes quanto a $FSDN_6$. Nesse sentido, houve redução no número de camadas convolucionais e também redução no número de blocos inception que foram substituídos por blocos residuais; além da redução de filtros convolucionais, conforme pode ser visualizado na Figura 5.10, as diferenças entre as arquiteturas $FSDN_7$ e $FSDN_8$ está na camada totalmente conectada onde a primeira possui 1024 neurônios e a segunda 2048 neurônios. Este teste foi idealizado para identificar os impactos na qualidade de detecção e no tamanho da rede quando se diminui o tamanho das camadas de convolução e aumenta o número de neurônios nas camadas totalmente conectadas. Neste caso, o uso da camada totalmente conectada com 2048 neurônios na $FSDN_7$ apresentou resultado na base de validação ligeiramente melhor à rede $FSDN_8$, porém o tamanho do arquivo do peso da rede $FSDN_7$ ficou quase três vezes maior que o arquivo da rede $FSDN_8$.

A arquitetura $FSDN_9$ foi concebida na tentativa de diminuir o tamanho das redes $FSDN_7$ e $FSDN_8$. Para isso, os filtros nas camadas convolucionais foram diminuídos de tamanho, assim como o tamanho da camada totalmente conectada. Porém, com a diminuição do tamanho da rede houve queda no desempenho da rede em mais de 2% em relação às arquiteturas $FSDN_7$ e $FSDN_8$ nas métricas Acurácia e F_1score .

A arquitetura $FSDN_{10}$ foi idealizada utilizando os conceitos da Resnet (He et al., 2015b) de blocos identidade e blocos convolucionais. Os blocos convolucionais concatenam três convoluções e usam um bloco residual com convolução, conforme diagrama da Figura 5.12. Já

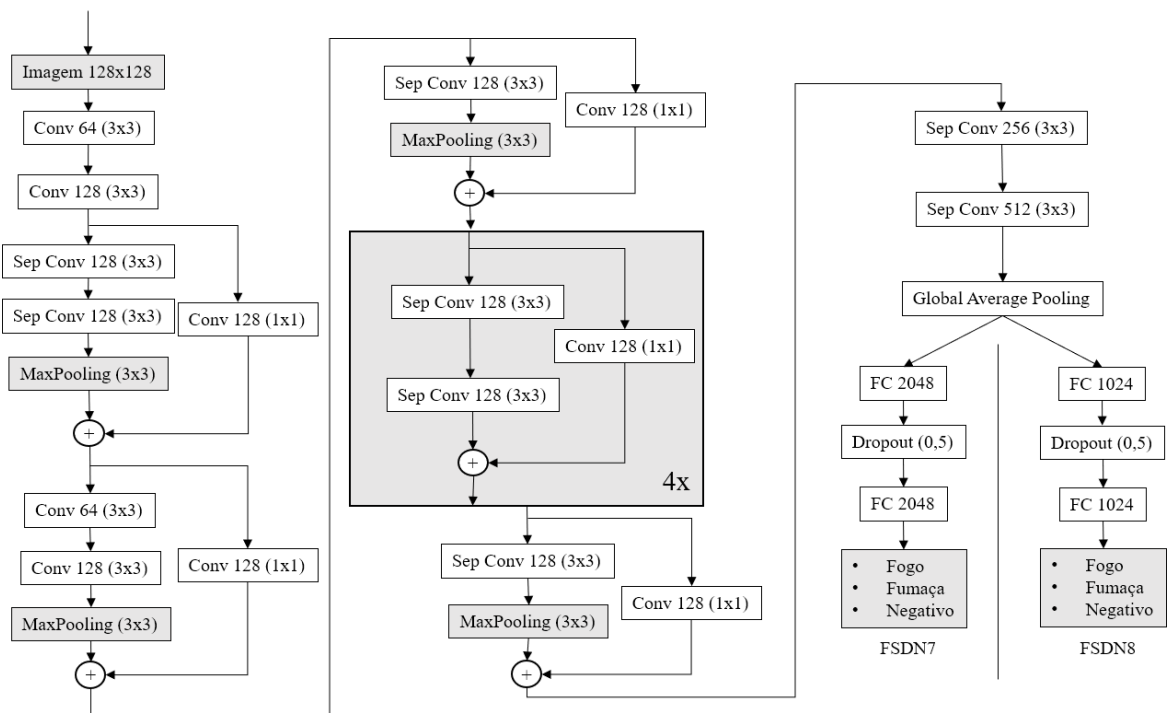


Figura 5.10: Redes Neurais $FSDN_7$ e $FSDN_8$.

os blocos identidade, também realizam 3 convoluções, porém, utilizam o bloco residual sem convolução, conforme diagrama da Figura 5.13.

Tabela 5.1: Resultado dos testes realizados com 10 diferentes arquiteturas de RNCs.

Arquitetura	Tamanho	Treino				Validação			
		Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
$FSDN_1$	9MB	0,855	0,854	0,362	0,141	0,863	0,862	0,352	0,133
$FSDN_2$	65MB	0,870	0,869	0,330	0,128	0,872	0,871	0,342	0,129
$FSDN_3$	16MB	0,835	0,834	0,400	0,159	0,850	0,848	0,392	0,152
$FSDN_4$	5MB	0,839	0,837	0,397	0,155	0,854	0,853	0,374	0,148
$FSDN_5$	111MB	0,779	0,776	0,516	0,204	0,766	0,776	0,523	0,219
$FSDN_6$	33MB	0,959	0,959	0,112	0,042	0,917	0,917	0,239	0,071
$FSDN_7$	46MB	0,959	0,951	0,105	0,040	0,921	0,921	0,249	0,060
$FSDN_8$	17MB	0,971	0,971	0,079	0,029	0,920	0,920	0,297	0,061
$FSDN_9$	4MB	0,918	0,918	0,217	0,081	0,904	0,904	0,273	0,087
$FSDN_{10}$	4 MB	0,915	0,915	0,226	0,084	0,898	0,898	0,279	0,094

A partir dos modelos arquiteturais propostos e da realização de treinos e validações nas redes, a Tabela 5.1 apresenta os resultados obtidos no processo de aprendizagem utilizando a base BD1. Analisando os resultados, temos que as arquiteturas $FSDN_7$ e $FSDN_8$ apresentaram os melhores desempenhos nas métricas testadas. No entanto, a arquitetura $FSDN_8$ possui o melhor custo/benefício na base de validação, pois apesar da rede $FSDN_7$ na base de validação

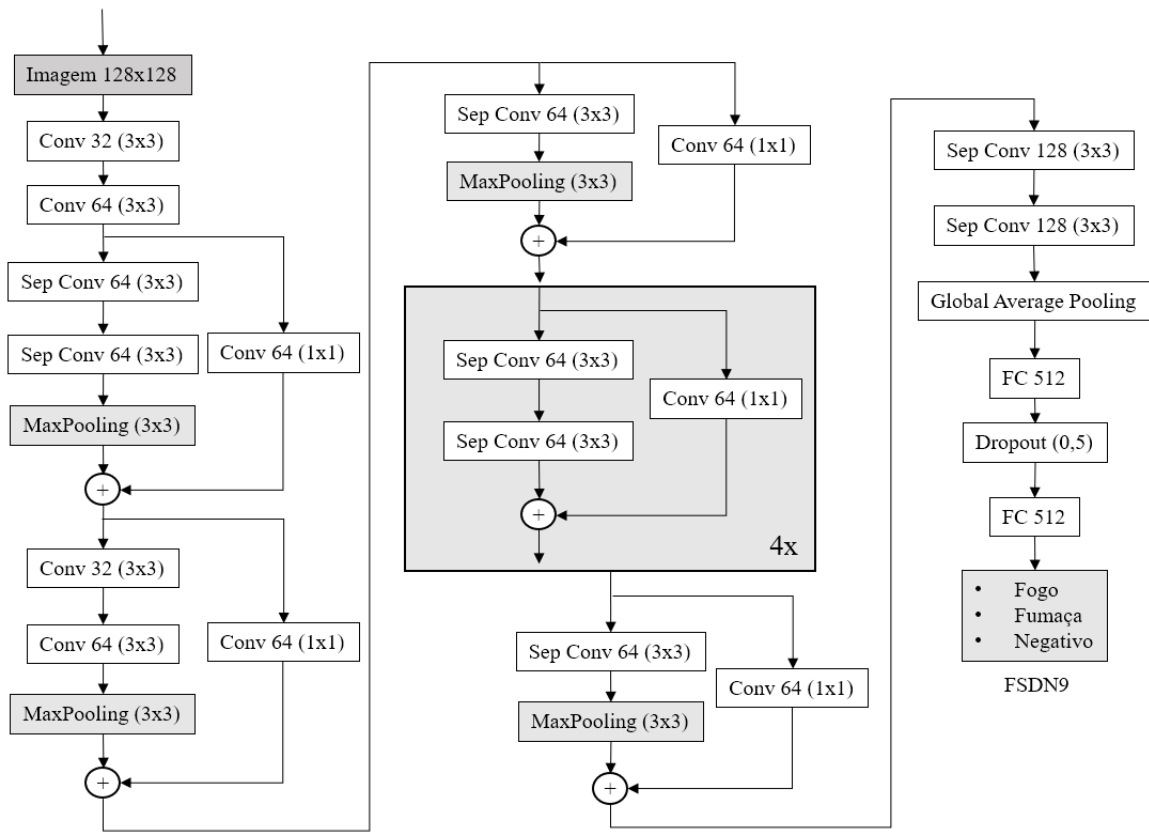


Figura 5.11: Redes Neurais $FSDN_9$.

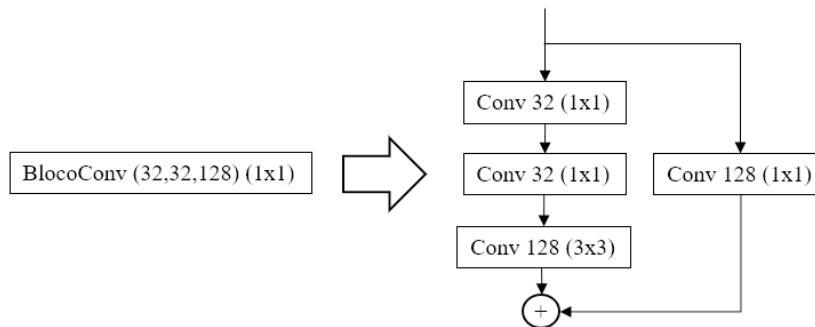


Figura 5.12: Bloco Convolutcional.

ser ligeiramente melhor, apresentando 0,1% de acréscimo no $F_1 score$ e Acurácia para a base de validação, o tamanho da rede $FSDN_8$ foi determinante para a escolha dessa arquitetura, pois o tamanho da rede $FSDN_7$ é quase 3 vezes maior que a rede $FSDN_8$. Portanto, a rede $FSDN_8$ será o modelo arquitetural utilizado para o comparativo com outras arquiteturas presentes na literatura.

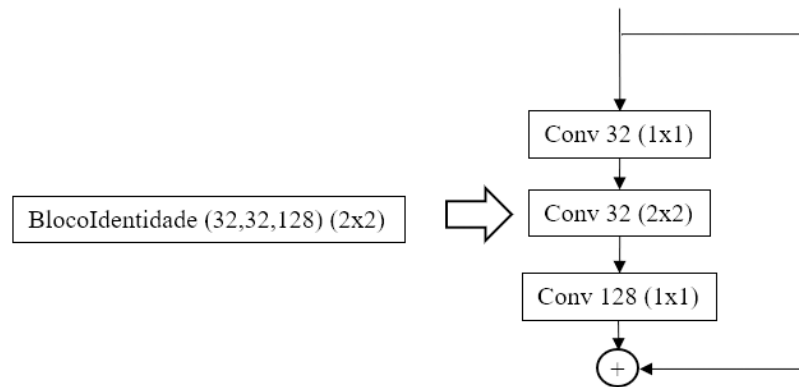


Figura 5.13: Bloco Identidade.

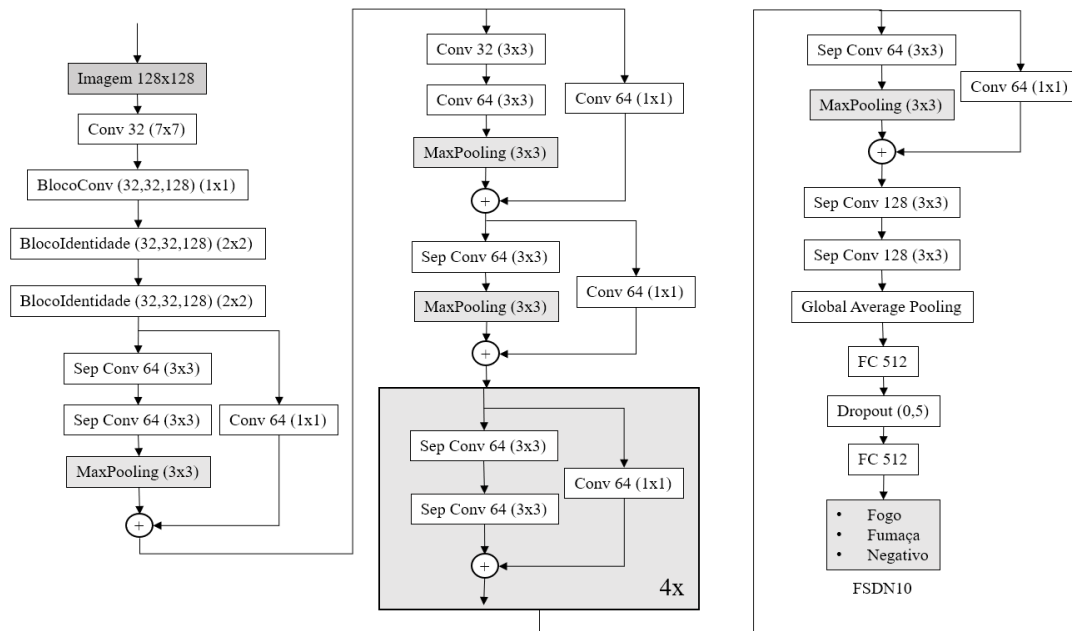


Figura 5.14: Redes Neurais $FSDN_{10}$.

5.2.1 Comparativo de desempenho das RNCs

Os testes nas redes InceptionV3, Resnet50, VGG16, Xception e MobileNet utilizaram a técnica de transferência de aprendizagem, uma vez que os pesos pré-calculados foram carregados e adaptados para as classes de fogo, fumaça e negativo. Neste sentido, a análise dos resultados obtidos deve levar em consideração que as redes pré-treinadas possuem vantagens em relação às redes treinadas do zero, como é o caso da rede proposta neste trabalho e das redes propostas por (Frizzi et al., 2016), (Yin et al., 2017), (Tao and Wang, 2016) e (Zhang et al., 2016). Isto ocorre por causa da transferência de aprendizagem, pois redes treinadas para reconhecer diferentes objetos ao serem submetidas a uma nova classe de problemas, tendem a ter melhor

desempenho conforme mostra Bengio (2012). Como convenção, a Tabela 5.2 mostra o trabalho de origem da rede e a nomenclatura utilizada para referenciar tal rede.

Tabela 5.2: Referência bibliográfica e sigla utilizada para referenciar o trabalho.

Citação	Sigla usada
(Szegedy et al., 2014)	InceptionV3
(He et al., 2015a)	Resnet50
(Simonyan and Zisserman, 2014)	VGG16
(Chollet, 2016)	Xception
(Howard et al., 2017)	MobileNet
(Frizzi et al., 2016)	FrizziNet
(Yin et al., 2017)	YinNet
(Tao and Wang, 2016)	TaoNet
(Zhang et al., 2016)	ZhangNet

Os resultados foram divididos de acordo com o tipo de saída das redes: binária de fogo, binária de fumaça e multiclases (detecção de fogo, fumaça ou negativo). Além disso, foi realizada a verificação cruzada entre as bases *BD1*, *BD2* e *BD3*, onde a rede treinada a partir da base de treino da *BD1* foi testada nas bases de testes das bases *BD1* e *BD2*. De maneira análoga, a rede treinada a partir da base de treino da *BD3* foi testada nas bases de testes da *BD1* e *BD2*.

As Tabelas 5.3 e 5.4 mostram os resultados obtidos com a classificação binária de imagens de fogo e não fogo cujos treinos foram realizados nas bases *BD1* e *BD3*. Quando o treino foi realizado por meio da base *BD1*, Tabela 5.3, a rede que apresentou melhores resultados para as bases *BD1* e *BD2* foi a rede Xception, apresentando F_1score de 99,3% e acurácia de 99,08% na base de validação de *BD1* e 97,5% de F_1score e 96,4% de acurácia na base de teste de *BD1*. Os testes na base *BD2* mostraram que o melhor desempenho de detecção também foi na rede Xception com 98,3% de acurácia e 98,2% de F_1score .

Ainda sobre as redes binárias e o treino realizado na base *BD1*, comparando as redes $FSDN_8$, FrizziNet, YinNet, TaoNet e a ZhangNet, redes mais compactas e treinadas do zero, temos que a rede proposta $FSDN_8$ obteve os melhores resultados de Acurácia e F_1score em todas as bases testadas, atingindo na base de validação de *BD1* 96,9% de acurácia e 97,8% de F_1score e na base de testes da *BD1* 93,7% de acurácia e 95,7% de F_1score . Já na base *BD2* a rede $FSDN_8$ obteve 92,6% de acurácia e 92,4% de F_1score . A rede YinNet obteve os melhores resultados de perda (*Loss*) em relação às outras redes, alcançando 0,118, 0,206 e 0,241 de perda nas bases da *BD1* de validação e teste e na base *BD2*.

Em relação aos resultados com os treinos realizados na base *BD3*, Tabela 5.4, nota-se que o aprendizado intrabase obteve resultados satisfatórios, sendo F_1score de 99,17% e Acurácia de 99,34% na base de validação com a rede Inception. Já em relação ao aprendizado interbases, houve queda significativa nos resultados que foram: 91,08% F_1score e 87,4% de acurácia na base de testes da *BD1* para rede VGG16 e 80,4% de F_1score e 82,7% de acurácia na base *BD2* com a rede $FSDN_8$. Os resultados mostraram que o desempenho das redes, quando treinadas

na base BD3 e testadas nas bases de teste de BD1 e BD2, caíram de forma significativa. Este fato está relacionado diretamente ao tipo de imagens e à heterogeneidade que a base BD3 possui. Dessa forma, verifica-se que a qualidade da base de imagens é fundamental para que o modelo seja o mais generalista possível para o tipo de problema que se deseja tratar.

Outro fator importante é em relação ao treinamento das redes, sendo que as redes pré-treinadas (Inception, VGG16, Xception e Resnet) obtiveram melhores resultados que as bases treinadas do zero (arquiteturas *FSDN₈*, FrizziNet, YinNet, TaoNet e ZhangNet). Ainda sobre a Tabela 5.4 é possível perceber a queda significativa no desempenho das redes quando estas são testadas em outras bases de imagens que não a proposta por Frizzi et al. (2016) (colunas: “Base de Teste - BD1” e “Base de Teste Patches - BD2”). Isso mostra que a transferência de aprendizagem é fraca quando o modelo treinado pela base BD3 é testado em bases diferentes.

Tabela 5.3: Resultados das arquiteturas binárias para detecção de fogo - Treino em BD1.

Detecção de fogo - Rede Neural Binária - Treino realizado na base BD1												
Arquitetura	Validação - BD1				Base de Teste - BD1				Base de teste Patches - BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,9855	0,9897	0,0466	0,0184	0,9467	0,9635	0,3357	0,0559	0,9517	0,9502	0,2519	0,0505
Resnet50	0,9819	0,9874	0,0547	0,0228	0,9580	0,9709	0,1416	0,0484	0,9649	0,9633	0,1011	0,0434
VGG16	0,9824	0,9878	0,0550	0,0229	0,9538	0,9681	0,1785	0,0516	0,9597	0,9579	0,1197	0,0491
Xception	0,9908	0,9934	0,0381	0,0105	0,9640	0,9750	0,1864	0,0370	0,9839	0,9828	0,0758	0,0192
Mobilenet	0,9900	0,9802	0,0909	0,0449	0,9439	0,9616	0,2600	0,0716	0,9332	0,9321	0,2000	0,0924
<i>FSDN₈</i>	0,9690	0,9786	0,1250	0,0388	0,9376	0,9568	0,2900	0,0678	0,9260	0,9246	0,3300	0,0810
FrizziNet	0,9453	0,9615	0,1479	0,0794	0,9255	0,9478	0,2430	0,0976	0,8996	0,8977	0,3142	0,1415
YinNet	0,9643	0,9752	0,1187	0,0465	0,9343	0,9546	0,2067	0,0737	0,9236	0,9216	0,2415	0,0601
TaoNet	0,9296	0,9520	0,1940	0,1112	0,8974	0,9307	0,2921	0,1284	0,8345	0,8456	0,4356	0,1958
ZhangNet	0,9377	0,9573	0,1656	0,0931	0,9102	0,9383	0,2412	0,1167	0,8730	0,8762	0,3588	0,1659

Tabela 5.4: Resultados das arquiteturas binárias para detecção de fogo - Treino em BD3.

Detecção de fogo - Rede Neural Binária - Treino realizado na base BD3												
Arquitetura	Validação - BD3				Base de Teste - BD1				Base de teste Patches - BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,9934	0,9918	0,0227	0,0098	0,7173	0,7571	1,4817	0,2863	0,5850	0,2614	2,3618	0,4111
Resnet	0,9700	0,9612	0,1057	0,0324	0,6899	0,7343	1,7641	0,3144	0,6466	0,5295	1,8158	0,3567
VGG16	0,9539	0,9454	0,1738	0,0467	0,8746	0,9109	0,7460	0,1299	0,7476	0,743	1,2879	0,2580
Xception	0,9558	0,9442	0,1607	0,0478	0,6673	0,7049	1,8458	0,3333	0,6051	0,3454	2,8013	0,3934
<i>FSDN₈</i>	0,9790	0,9749	0,1157	0,0190	0,8741	0,9069	1,4200	0,1347	0,8269	0,8039	1,8062	0,1883
FrizziNet	0,9751	0,9702	0,0943	0,0359	0,8156	0,8644	0,9474	0,1891	0,6943	0,6870	1,4324	0,3123

As Tabelas 5.5 e 5.6 mostram os resultados obtidos com a classificação binária de imagens de fumaça e não fumaça. Para este experimento, com o treino realizado a partir da base BD1, Tabela 5.5, a rede que apresentou melhores resultados para as bases BD1 e BD2 foi a rede Xception, obtendo F_1score de 92,7%, 88,04% e 95,94% para validação e teste de BD1 e teste de BD2 respectivamente. Em relação à Acurácia, a rede Xception obteve 96,8%, 93,4% e 95,6% nas bases de validação e teste da BD1 e na base BD2 respectivamente. Ao comparar as redes treinadas do zero temos que a rede $FSDN_8$ foi superior às demais redes na acurácia, F_1score e MAE, obtendo 93,4%, 88,9% e 87,2% de acurácia, 86,6%, 78,2% e 85,5% de F_1score e 9,7%, 14,4% e 16,9% de erro absoluto médio nas bases de validação e teste da BD1 e na base BD2 respectivamente. Em relação à perda, a rede YinNet obteve os melhores valores, atingindo 0,188, 0,353 e 0,360 para as bases de validação e testes da BD1 e na base BD2.

Em relação aos resultados provindos do treino em BD3, a rede com os melhores resultados na bases de validação foi a rede *Frizzi*. Nota-se que neste caso, fenômeno similar ao que ocorreu com as arquiteturas de detecção de fogo, o aprendizado intrabase obteve resultados satisfatórios, com 97,0% de F_1score e 95,7% de acurácia. Porém, ao utilizar os testes nas bases BD1 e BD2, o desempenho das redes caíram de forma significativa, obtendo F_1score de 78,2% e Acurácia de 88,8% para BD1 com a rede Resnet e F_1score de 85,5% e acurácia de 87,1% para a base de teste BD2. Neste caso, a discrepância dos resultados interbases (treino na BD3 e teste em BD1 e BD2) mostram mais uma vez que a qualidade da base de imagens é fundamental para que o modelo seja o mais generalista possível para o tipo de problema que se deseja tratar.

Tabela 5.5: Resultados com as arquiteturas binárias para detecção de fumaça - Treino em BD1.

Detecção de fumaça - Rede Neural Binária - Treino realizado na base BD1												
Arquitetura	Validação - BD1				Base de Teste - BD1				Base de teste Patches - BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,9654	0,9266	0,1047	0,0433	0,9261	0,8635	0,2504	0,0844	0,9403	0,9373	0,1815	0,0703
Resnet	0,9630	0,9251	0,1315	0,0430	0,9274	0,8665	0,3318	0,0806	0,9339	0,9306	0,2380	0,0744
VGG16	0,9515	0,9006	0,1311	0,0625	0,9110	0,8365	0,2862	0,1023	0,9228	0,9188	0,2368	0,0910
Xception	0,9688	0,9276	0,0911	0,0382	0,9344	0,8804	0,2997	0,0728	0,9562	0,9545	0,1870	0,0576
Mobilenet	0,9500	0,8973	0,1437	0,0824	0,9036	0,8209	0,3357	0,1263	0,9054	0,8983	0,2647	0,1372
<i>FSDN₈</i>	0,9343	0,8668	0,1886	0,0976	0,8889	0,7820	0,4058	0,1444	0,8710	0,8556	0,3686	0,1692
FrizziNet	0,8914	0,7879	0,2693	0,1610	0,8315	0,6685	0,4383	0,2119	0,8226	0,7977	0,4816	0,2395
YinNet	0,9308	0,8637	0,1883	0,1105	0,8775	0,7675	0,3532	0,1612	0,8651	0,8524	0,3606	0,1771
TaoNet	0,8281	0,6703	0,4038	0,2796	0,7415	0,5004	0,5294	0,3312	0,7093	0,6262	0,6018	0,3680
ZhangNet	0,8619	0,7083	0,3306	0,2145	0,7979	0,5737	0,5023	0,2705	0,7517	0,6827	0,5293	0,2960

Tabela 5.6: Resultados com as arquiteturas binárias para detecção de fumaça - Treino em BD3.

Detecção de fumaça - Rede Neural Binária - Treino realizado na base BD3												
Arquitetura	Validação - BD3				Base de Teste - BD1				Base de teste Patches - BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,8078	0,8481	0,4165	0,2685	0,7441	0,2380	0,5912	0,3212	0,5355	0,1121	1,0321	0,4522
Resnet	0,9254	0,9450	0,4249	0,0764	0,7557	0,4718	2,2300	0,2464	0,6468	0,4479	3,3776	0,3531
VGG16	0,9058	0,9299	0,2896	0,0990	0,7513	0,4402	1,6768	0,2509	0,6615	0,4791	2,1985	0,3363
Xception	0,8144	0,8500	0,4026	0,2064	0,7600	0,3448	0,6847	0,2812	0,5920	0,2997	1,0104	0,4229
<i>FSDN₈</i>	0,9013	0,9327	0,3007	0,1873	0,884	0,7680	0,3310	0,2128	0,8710	0,8556	0,3590	0,2495
FrizziNet	0,9576	0,9709	0,1813	0,0909	0,6688	0,3725	2,1332	0,3300	0,5774	0,2654	4,4346	0,4266

Para a classificação multiclases, as Tabelas 5.7 e 5.8 mostram os resultados obtidos com a classificação multiclases de imagens de fogo, fumaça e não fogo/fumaça. Para este experimento a rede que apresentou melhores resultados, usando como treino a base BD1, foi a rede Xception com F_1score de 95,3%, 87,6% e 89,6% e acurácia de 95,3%, 89,9% e 89,6% para as bases de validação e teste de BD1 e teste de BD2 respectivamente.

Ao comparar as redes treinadas do zero, temos que a rede $FSDN_8$ foi superior às demais redes em praticamente todas as métricas, atingindo 92,3%, 86,1% e 86,2% de acurácia e 92,3%, 86,1% e 86,2% de F_1score , assim como 6,1%, 10,1% e 10,4% de erro absoluto. A rede YinNet obteve melhor pontuação somente na perda (*Loss*) na base BD2 com valor de 0,48.

Em relação aos resultados dos treinos realizados a partir da base BD3, a rede que obteve melhores resultados no aspecto intrabase foi a rede Xception com F_1score de 95,3% e acurácia de 95,3%. Já no aspecto interbases, a rede $FSDN_8$ obteve o melhor resultado para F_1score com 57,1% para a base de testes de BD1 e 53,2% para a base de testes BD2. Em relação à acurácia, os melhores resultados para BD1 e BD2 foram apresentados pela rede *Frizzi* com 68,9% e 62,7%.

Os resultados das Tabelas 5.8 e 5.6 mostram que não há transferência de aprendizagem da base BD3 para as demais bases, o que demonstra a distinção entre as bases. Ao analisar as imagens das duas bases, principalmente as imagens de fumaça, nota-se como diferença principal o tipo de imagem de fumaça das imagens selecionadas. Enquanto que as bases BD1 e BD2 possuem imagens de situações de fumaça com outros objetos presentes na cena, conforme pode ser visualizado na Figura 5.15a, a base BD3 de fumaça é composta por imagens em sua maioria da fumaça sem objetos na cena, conforme pode ser visualizado na Figura 5.15b, e isso pode ser um ponto negativo para a base, pois em situações reais a fumaça irá aparecer em conjunto com outros objetos. Portanto, isso explica a discrepância de valores obtidos para Acurácia e F_1score entre a validação da base BD3 e as bases de teste nas bases BD1 e BD2.

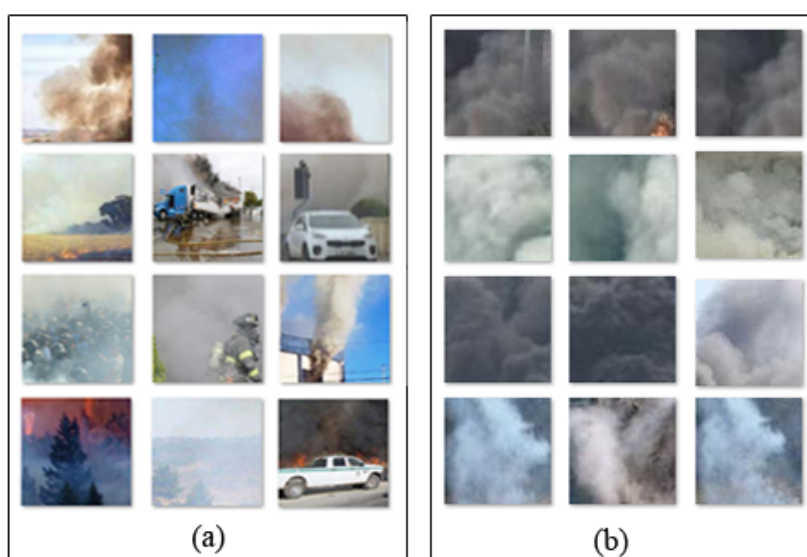


Figura 5.15: *Patches* de fumaça da base BD1 (a) e *patches* de fumaça da base BD3 (b).

Tabela 5.7: Resultados da detecção de fogo e fumaça em Redes Neurais multiclases - Treino realizado na base BD1.

Detecção de fogo e fumaça - Rede Neural Multiclasse - Treino realizado na base BD1												
Arquitetura	Validação - BD1				Base de Teste - BD1				Base de teste Patches – BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,9430	0,9431	0,2111	0,0421	0,8933	0,8711	0,4706	0,0755	0,8856	0,8857	0,4781	0,0836
Resnet	0,9473	0,9472	0,1774	0,0412	0,8901	0,8662	0,4527	0,0789	0,8892	0,8893	0,4441	0,0831
VGG16	0,9339	0,9341	0,1850	0,0570	0,8769	0,8519	0,4156	0,0935	0,8862	0,8856	0,3718	0,0920
Xception	0,9537	0,9536	0,1581	0,0351	0,8996	0,8767	0,4863	0,0709	0,8964	0,8962	0,4307	0,0756
Mobilenet	0,9392	0,9392	0,1900	0,0523	0,8617	0,8513	0,6080	0,1030	0,8724	0,8727	0,4520	0,1008
<i>FSDN₈</i>	0,9235	0,9235	0,2484	0,0612	0,8608	0,8609	0,5362	0,1010	0,8626	0,8625	0,5350	0,1043
FrizziNet	0,8692	0,8600	0,3645	0,1265	0,8000	0,7593	0,5792	0,1653	0,7985	0,7962	0,6081	0,1751
YinNet	0,9067	0,9068	0,2947	0,0845	0,8218	0,8230	0,5500	0,1400	0,8378	0,8354	0,4800	0,1470
TaoNet	0,7725	0,7692	0,6260	0,2437	0,6969	0,7001	0,7290	0,2720	0,6740	0,6620	0,7600	0,2780
ZhangNet	0,8347	0,8290	0,4270	0,1683	0,7479	0,7360	0,7500	0,2160	0,7140	0,7045	0,7050	0,2280

Tabela 5.8: Resultados da detecção de fogo e fumaça em Redes Neurais multiclases - Treino realizado na base BD3.

Detecção de fogo e fumaça - Rede Neural Multiclasse - Treino realizado na base BD3												
Arquitetura	Validação - BD3				Base de Teste - BD1				Base de teste Patches – BD2			
	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE	Acc	F1	Loss	MAE
InceptionV3	0,8859	0,8852	0,4516	0,0858	0,3712	0,3550	5,5607	0,4174	0,4031	0,2600	7,7369	0,3975
Resnet	0,8769	0,8769	0,6044	0,0836	0,6725	0,6119	2,6535	0,2216	0,5506	0,5121	3,7425	0,3017
VGG16	0,9442	0,9450	0,2322	0,0723	0,6063	0,5564	2,4284	0,2685	0,4878	0,4478	3,3995	0,3419
Xception	0,9104	0,9101	0,4386	0,0653	0,6036	0,5620	2,7357	0,2642	0,5589	0,5161	3,4183	0,2942
<i>FSDN₈</i>	0,9444	0,9446	0,4028	0,0405	0,6226	0,5712	4,2400	0,2500	0,5332	0,5329	5,5900	0,3100
FrizziNet	0,9222	0,9227	0,2552	0,0721	0,6891	0,4725	2,0331	0,3900	0,6274	0,3654	4,0346	0,3956

Quando se realiza a comparação dos resultados obtidos com diferentes bases de imagens percebe-se que a utilização de base de imagens com quantidade de amostras significativas (110 mil patches na BD1 vs 28 mil patches na BD3) traz melhores resultados, inclusive quando se utiliza a transferência de aprendizagem com diferentes bases. Portanto, quando há a necessidade de generalizar o problema, como é o caso da detecção de fogo e fumaça, o tamanho da base e sua heterogeneidade fazem a diferença. Outro aspecto a ser considerado é que em redes binárias os resultados obtidos foram superiores aos de redes multiclases, portanto para um sistema de detecção mais eficiente, a utilização de duas RNCs pode ser mais vantajosa, mas deve-se considerar o custo computacional para o processamento de duas redes distintas.

5.2.2 Avaliação das taxas de detecção de fogo e fumaça nas RNCs estudadas

A presente subseção visa à apresentação dos resultados das taxas de detecção de fogo e fumaça realizada na base dos *patches* BD2. Para avaliar a qualidade de detecção, como critério de detecção de fogo (DF), foi utilizado o seguinte: se a imagem possui patches de fogo e pelo menos um patch da imagem de origem for identificado fogo, então a detecção de fogo para esta imagem é classificada como correta. Em relação à fumaça (DS - *Detection Smoke*), o raciocínio é análogo ao da detecção de fogo. Se há na imagem patches de fumaça e pelo menos um dos patches foi identificado como fumaça, então a detecção de fumaça para aquela imagem foi correta. Para classificação multiclases há ainda a taxa de detecção de fogo e fumaça (DFS) que consiste em: se a imagem contiver fogo e for detectado pelo menos um patch de fogo, ou se a imagem contiver fumaça e for localizado pelo menos um patch de fumaça ou se a imagem for negativa e não for localizado nem fogo e nem fumaça, então a classificação é considerada correta.

Os resultados obtidos, presentes na Tabela 5.9, revelam que as detecções das redes binárias são mais eficientes do que a detecção multiclases. O indicador de detecção correta de fogo ou fumaça (DFS) obteve taxa de detecção de 92,7% na rede Xception com o classificador multiclases, frente a 94,31% na taxa de detecção de fumaça na mesma rede e 98,14% na taxa de detecção de fogo para a rede Resnet. Tais resultados levam à conclusão de que utilizar a rede binária para detecção de fogo e fumaça é mais eficiente para evitar falsos alarmes. Este resultado reflete a ambiguidade presente nas bases de fogo e fumaça, sendo que em muitas imagens de fogo também pode haver fumaça, existe a possibilidade de o classificador a confundir as duas classes no momento da classificação. Havendo duas redes para detecção, uma para fogo e outra para fumaça, trabalhando de maneira independente, ocorrem melhores resultados para o problema de detecção.

O tamanho dos modelos de aprendizagem de máquina é outro fator que pode ser considerado quando da necessidade de inclusão deste modelo em um sistema de detecção em tempo real. Normalmente, estes sistemas utilizam software embarcado em arquiteturas específicas, como o Raspberry Pi, por exemplo e possuem limitações nas suas respectivas memórias internas,

Tabela 5.9: Taxa de detecção de fogo e fumaça nas arquiteturas estudadas.

Rede	Multiclasses			Binário	
	DFS	DF	DS	Fumaça	Fogo
Inception	91,66%	92,77%	81,82%	94,31%	97,74%
Xception	92,73%	93,72%	82,02%	93,91%	97,80%
Vgg16	91,38%	93,12%	82,45%	92,71%	97,74%
Resnet	92,33%	93,08%	83,12%	93,91%	98,14%
Mobile	91,11%	92,92%	80,75%	91,10%	96,29%
<i>FSDN₈</i>	89,88%	92,45%	80,94%	90,23%	95,53%
FrizziNet	85,26%	89,72%	80,51%	85,82%	92,58%
YinNet	87,50%	90,35%	81,34%	89,03%	95,24%
TaoNet	73,99%	88,81%	77,98%	77,25%	89,26%
ZhangNet	77,11%	89,01%	78,93%	81,81%	91,28%

o que retarda o processamento em modelos de redes neurais maiores como a Inception, Resnet e Xception. O apelo da rede MobileNet é justamente apresentar uma alternativa de Rede Neural Convolutacional, eficiente e com resultados compatíveis com redes maiores (Howard et al., 2017). dessa forma, a utilização de redes menores, como a proposta por este trabalho (*FSDN₈*) e as arquiteturas menores propostas por outros autores como a FrizziNet (Frizzi et al., 2016) e a ZhangNet (Zhang et al., 2016) podem ser uma alternativa para o processo de detecção.

A Tabela 5.10 mostra o resultado de desempenho das redes ao serem submetidas ao processo de detecção em redes multiclasses para detecção de fogo e fumaça, assim como para as redes binárias de fogo e fumaça. Os testes foram realizados em uma rede por vez, o computador que executou os testes ficou exclusivamente para este propósito para não interferir na aferição do tempo de processamento. A base utilizada para este teste foi a base *BD2*. As redes multiclasse processaram 8.944 imagens, enquanto que as redes binárias de fogo processaram 6.154 imagens e as redes binárias de fumaça processaram 5.660 imagens.

Como resultado, temos que a rede com melhor desempenho em relação ao tempo de processamento e o menor espaço em disco ocupado foi a rede FrizziNet, processando até 486 imagens por segundo. Porém, o desempenho dessa rede em relação à detecção de fogo e fumaça fica aquém de outras arquiteturas. Nesse sentido, a rede proposta *FSDN₈* processou em média 410 imagens por segundo com número de parâmetros menores que a metade da rede MobileNet, ocupando 10% do espaço em disco que a rede Xception e com as melhores taxas de detecção de fogo e fumaça comparado às outras arquiteturas específicas para a detecção de fogo e fumaça, como pode ser visualizado na Tabela 5.10.

Tabela 5.10: Desempenho em relação ao tempo de processamento, Taxa de detecção (TxD), média de imagens por segundo, espaço em disco dos pesos e número de parâmetros de cada arquitetura testada na base BD2.

Rede	Multiclasse	TxD (%)	Bin Fogo	TxD (%)	Bin Fumaça	TxD (%)	Média	Tam Rede	Total Parâmetros
Inception	42,2 s	91,6	30,4 s	97,7	28,2 s	94,3	205 img/s	195MB	21.808.931
Resnet	30,9 s	92,3	22,1 s	98,1	20,5 s	93,9	281 img/s	209MB	23.593.859
VGG16	29,3 s	91,4	21,7 s	97,7	19,8 s	92,7	292 img/s	127MB	134.272.835
Xception	58,1 s	92,7	40,6 s	97,8	37,5 s	93,3	152 img/s	187MB	20.867.627
Mobilenet	49,1 s	91,1	25,5 s	96,2	23,7 s	91,1	221 img/s	62MB	5.329.089
<i>FSDN₈</i>	20,5 s	89,8	15,4 s	95,5	14,4 s	90,2	410 img/s	17MB	2.255.299
FrizziNet	17,8 s	85,3	12,8 s	92,6	11,9 s	85,8	486 img/s	1,98MB	248.311
YinNet	22,1 s	87,5	15,9 s	95,2	14,7 s	89,0	392 img/s	300MB	25.616.481
TaoNet	35,1 s	73,9	76,0 s	89,3	20,9 s	77,2	202 img/s	280MB	23.954.183
ZhangNet	21,1 s	77,1	15,3 s	91,3	14,2 s	81,1	409 img/s	3,5MB	213.287

5.3 Comparativo entre métodos de detecção de fogo

Os resultados obtidos em relação à detecção de fogo dizem respeito a 16 diferentes técnicas descritas na literatura e outras 4 técnicas propostas neste trabalho. A proposta “BrunoCombina” tem os resultados descritos na Seção 6.3. A proposta “BrunoColorness” está descrita na Seção 6.4. A elaboração de um vetor de característica específico para o “SVM”, assim como as configurações estão presentes na Seção 6.5. O processo de construção da rede neural “FSDN₈” está descrito na Seção 5.2. Para aprimorar o processo de leitura foi atribuído o nome do primeiro autor para identificar as técnicas implementadas; dessa forma, nas tabelas comparativas apresentadas neste capítulo, as técnicas comparadas e seus respectivos trabalhos estão mostrados na tabela 5.11.

Tabela 5.11: Nome da Técnica e Referência bibliográfica correspondente.

Nome da Técnica	Referência
Celik RGB	(Çelik et al., 2006)
Celik RGB2	(Çelik et al., 2007)
Celik YCbCr	(Çelik and Demirel, 2009)
Celik Lab	(Çelik, 2010)
Chen RGB	(Chen et al., 2004)
Chen RGB+HSI	(Chen et al., 2010)
Ko	(Ko et al., 2009)
Philips	(Phillips et al., 2002)
Horng	(Horng et al., 2005)
Rossi	Rossi et al. (2009)
Santana	(Santana et al., 2012)
Vipin	(Vipin, 2012)

Os resultados das taxas de detecção de fogo podem ser divididos em dois grupos principais : métodos baseados em regras ou probabilidades e métodos que utilizam técnicas de aprendizagem de máquina mais robustas. A Tabela 5.12 mostra os valores das taxas de detecção das técnicas implementadas da literatura e das propostas apresentadas neste trabalho. Na coluna da Tabela 5.12 “Abordagem”, as siglas Re, PB e AM são referentes a: método baseado em Regras (Re), métodos baseados em Probabilidades (PB) e métodos que utilizam aprendizagem de máquina (AM).

Os resultados mostram a superioridade das Redes Neurais Convolucionais para a detecção de fogo em imagens. A nossa proposta de rede neural foi a técnica que obteve a melhor taxa de detecção para a base BL3, alcançando 91,8% de taxa de detecção de fogo para 1036 imagens de fogo (516 imagens) e não fogo (520 imagens). Ao comparar os métodos baseados em Regras ou Probabilidades temos que as propostas de segmentação deste trabalho também apresentaram melhores resultados que o estado da arte obtendo 74,4% de taxa de detecção na base BL3.

Tabela 5.12: Comparativo das taxas de detecção de fogo nas técnicas baseadas em Regras e técnicas que utilizam aprendizagem de máquina.

Técnica	Detecção	Abordagem
CelikRGB	30,9%	Re
CelikRGB2	62,7%	Re
CelikLab	11,3%	Re
CelikYCbCr	39,0%	Re
Chen	30,6%	Re
Chen10	33,4%	Re
HornG	36,5%	Re
Ko	29,1%	Re
Philips	37,6%	PB
Rossi	29,1%	Re
Santana	31,0%	Re
Vipin	65,7%	Re
BrunoCombina	74,4%	PB
BrunoColorness	68,5%	PB
SVM	64,0%	AM
FDSN8	91,8%	AM
FrizziNet	85,2%	AM
YinNet	90,1%	AM
TaoNet	85,9%	AM
ZhangNet	85,2%	AM

Após os testes realizados em relação ao problema de detecção de fogo e fumaça foram desenvolvidos mais experimentos, mas com o intuito de localizar o foco de incêndio nas imagens. Este processo de localização é de suma importância para o problema de vigilância de ambientes por meio de sensores ópticos. Dessa forma, a subseção a seguir trata de um método de segmentação de imagens que utiliza como base o uso de probabilidades e entropia para píxeis e regiões identificadas como fogo.

5.4 Considerações finais

Neste Capítulo foram expostas as etapas metodológicas visando aos testes em modelos de arquiteturas de RNCs na Seção 5.1. Após a exposição dos métodos, os protocolos experimentais e seus respectivos resultados foram apresentados. A Seção 5.2 apresentou 10 propostas de arquitetura de RNC para detecção de fogo e fumaça, entre elas foi selecionado o modelo arquitetural *FSDN₈*, sendo um modelo que utiliza recursos de blocos inception extremo aliado a aprendizado residual. Este modelo apresentou bom compromisso entre precisão (taxa de acerto) e eficiência (recursos). Além disso, ficou evidente que RNCs binárias desempenham melhor a tarefa de detecção de fogo e fumaça que RNCs multiclases. Os próximos capítulos tratam dos

métodos e protocolos experimentais para a localização de píxeis de fogo no Capítulo 6, assim como propõe um método para monitoramento de fogo e fumaça no Capítulo 7.

Capítulo 6

Localização de fogo

Para criar um processo de localização de fogo, o protocolo experimental foi estruturado a partir das seguintes etapas metodológicas: coleta de imagens e realização da segmentação ideal destas imagens; análise do comportamento dos canais de cor para píxeis de fogo e não fogo; implementação dos métodos de reconhecimento e segmentação de fogo existentes na literatura; estudo de técnicas e modelos de aprendizagem de máquina voltados para localização de fogo; proposição de técnicas de localização de fogo em imagens e, por fim, a análise e comparação dos resultados obtidos. O diagrama na Figura 6.1 ilustra o fluxo temporal do protocolo experimental definido para o processo de localização de fogo.

As etapas para construção das bases utilizadas para localização de fogo estão descritas na Seção 4.1.2 do Capítulo 4. Em relação ao estudo e desenvolvimento de modelos de aprendizagem de máquina para localização de fogo, tal etapa está subdividida em 5 etapas: análise de 7 espaços de cor e seus canais em imagens de fogo (1); combinação dos canais de cor promissores com aplicação de abordagens probabilísticas (2); aplicação de características de entropia em regiões identificadas com fogo (3); aplicação de testes probabilísticos combinando índices de cor (4) e a aplicação de métodos de aprendizagem de máquina mais robustos para localização de píxeis de fogo (5).

Baseado nestas etapas, a estrutura deste capítulo está disposta da seguinte maneira: na Seção 6.1 será apresentado um estudo dos histogramas de sete diferentes espaços de cor. A Seção 6.2 apresenta o método de criação das tabelas de consulta. Já a Seção 6.3 apresenta um método de segmentação baseado em probabilidades e o uso do cálculo da entropia. A Seção 6.4 trata de outra proposta de segmentação de imagens de fogo por meio das teorias de índices de cor, tendo como base a utilização dos índices Vermelho, Amarelo e Marrom. A Seção 6.5 apresenta o experimento realizado com os seguintes modelos de aprendizagem de máquina: SVM, Árvores de Decisão, Random Forest e Adaboost com árvores de decisão. Tais experimentos foram importantes para identificar a qualidade de segmentação de imagens de fogo usando abordagens de aprendizagem de máquina diferentes, assim como, demonstrar os métodos que obtiveram o melhor desempenho e custo operacional para a tarefa de localização de fogo em imagens. A Seção 6.6 mostra o comparativo entre 12 técnicas presentes na literatura e às técnicas propostas,

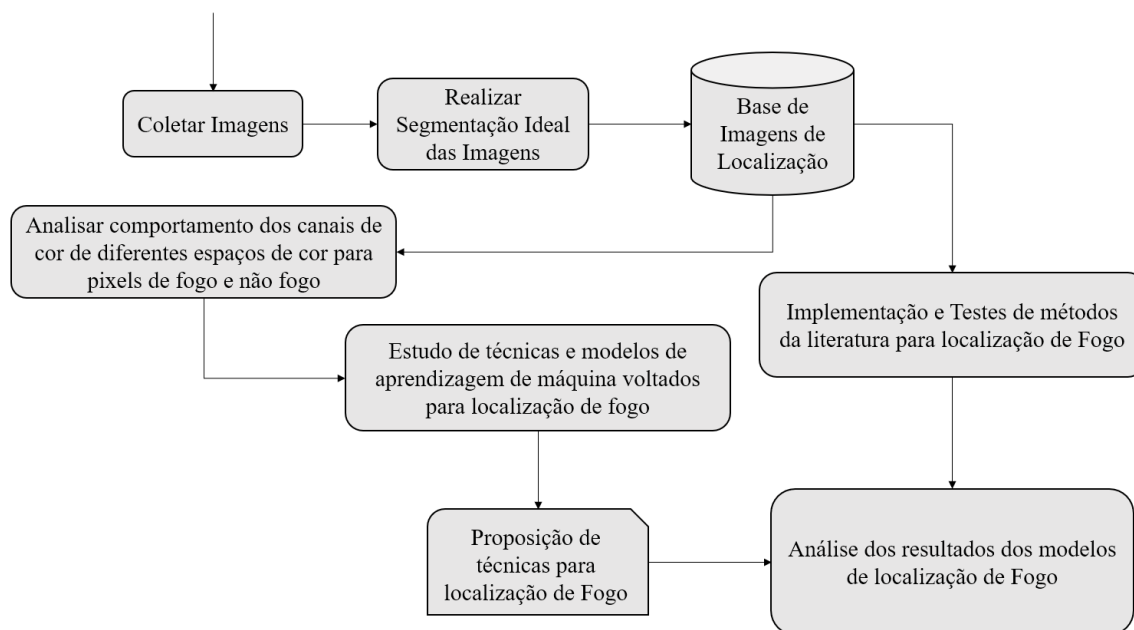


Figura 6.1: Etapas metodológicas para criação de método de segmentação de imagens de Fogo.

visando demonstrar a qualidade da segmentação em imagens de fogo. Como produto final, o Capítulo 7 demonstra uma proposta de monitoramento de Fogo e Fumaça unindo os métodos de detecção usando RNCs, presentes no Capítulo 5 em conjunto com o processo de localização de píxeis de fogo, presentes neste capítulo.

6.1 Análise dos canais de cor

Com o intuito de detectar características promissoras para a detecção de píxeis de fogo os seguintes espaços de cor foram estudados: *RGB*, *YCbCr*, *YUV*, *HSI*, *HSV*, *HSL* e *CIE Lab*.

Para realizar a análise dos histogramas de cada canal, então da base BL2 retiraram-se 334.310 píxeis de maneira aleatória, sendo a metade correspondente a fogo e a outra metade a não fogo. Os dados dos canais foram então normalizados entre 0 e 1 para cada canal de cor. Então, para todos estes píxeis foram gerados histogramas para cada canal de cor com o intuito de observar o comportamento dos píxeis de fogo comparado com os píxeis de não fogo. Os histogramas foram gerados a partir de amostras de píxeis de fogo e não fogo presentes na base BL1.

A análise dos canais de cor foi conduzida como segue. Nos gráficos dos histogramas, a cor azul representa os píxeis de não fogo e a cor vermelha os píxeis de fogo; como o gráfico é quantitativo, o tamanho da barra representa a quantidade de amostras em uma faixa de cor, por exemplo: se o tamanho da barra vermelha for maior que a barra azul, significa que naquela faixa de valor os píxeis de fogo são maiores em quantidade do que os píxeis de não fogo. Esta análise

foi de suma importância para a tomada de decisão quanto aos canais de cores a serem adotados nos métodos baseados em probabilidades propostas.

RGB

Em relação ao espaço de cor RGB, a partir do gráfico presente na Figura 6.2 temos: canal R – Percebe-se um crescente na quantidade de píxeis de fogo a partir do valor 0,55, tendo um pico nos valores máximos; canal G – A quantidade de píxeis de fogo fica concentrada na primeira porção do gráfico com um pequeno pico em seu valor máximo. Mas no geral, os valores de G devem estar acima de 0,05; canal B – Assim como o canal G, a concentração de píxeis de fogo é maior na porção inicial do gráfico. No geral valores de B devem ser menores que 0,55.

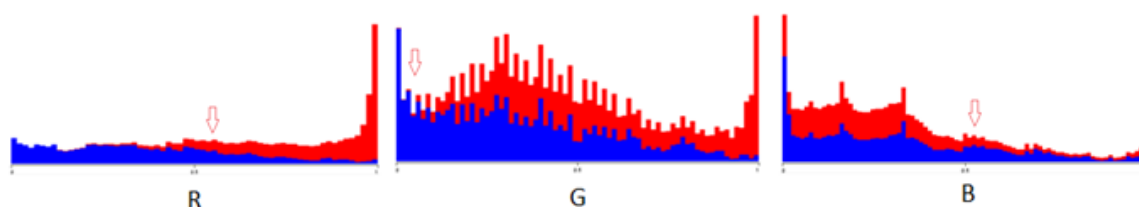


Figura 6.2: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor RGB.

YCbCr e YUV

Para o espaço de cor YCbCr, a partir do gráfico presente na Figura 6.3, temos: no canal Y, uma crescente no número de píxeis a partir dos valores de 0,23, sendo que a medida em que aumenta o valor de Y, o número de píxeis de não fogo diminui consideravelmente; Canal Cb(YCbCr) – No Canal Cb, percebe-se maior concentração de píxeis de fogo para os valores abaixo de 0,44 e uma concentração maior de píxeis de não fogo em 0,55; Canal Cr – Para o canal Cr, percebe-se um aumento no número de píxeis de fogo e uma diminuição no número de píxeis de não fogo a partir dos valores de 0,57.

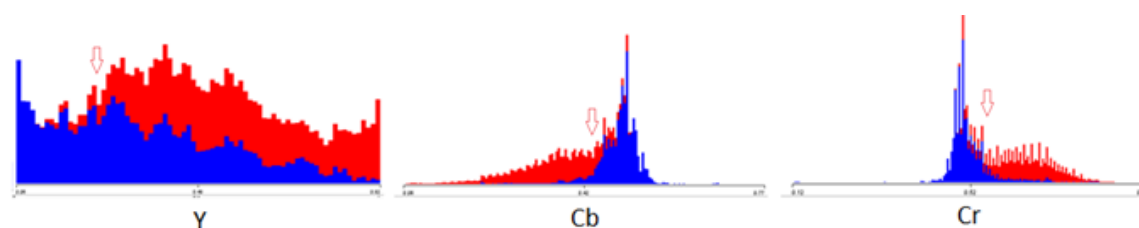


Figura 6.3: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor YCbCr.

Como os canais YCbCr e YUV são canais análogos, conforme descreve Rudz et al. (2009), temos que a partir do gráfico presente na Figura 6.4, no canal Y, percebe-se uma crescente no número de píxeis a partir dos valores de 0,23, sendo que a medida em que aumenta o valor de Y, o número de píxeis de não fogo diminui consideravelmente. Canal U – No Canal U, percebe-se maior concentração de píxeis de fogo para os valores abaixo de 0,46. Canal V – Para o canal V, há um aumento no número de píxeis de fogo e uma diminuição no número de píxeis de não fogo a partir dos valores de 0,57.

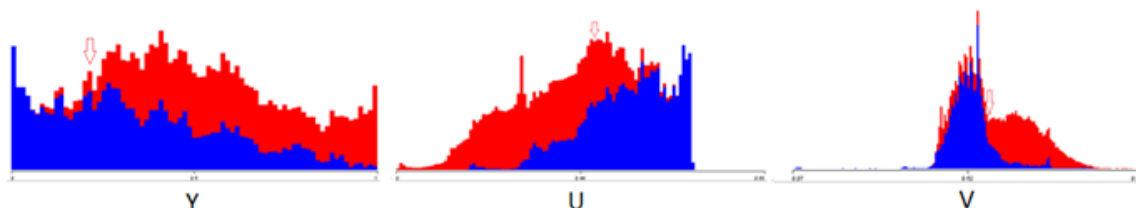


Figura 6.4: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor YUV.

HSV, HSL e HSI

Nos espaços de cor HSV (Figura 6.5) e HSL (Figura 6.6), temos: canal H – O número de píxeis de fogo fica mais próximo dos valores máximos, sendo considerados valores maiores que 0,82 indicados na seta, onde há a maior concentração de píxeis de fogo. Pode-se dizer que nos valores entre 0,071 e 0,79 para o canal H, os píxeis neste intervalo podem ser considerados não fogo, pois em um universo de 334mil píxeis, 317 (0,1%) apresentaram características de fogo; canal S – Para valores maiores que 0,328 até 0,985, há um aumento no número de píxeis de fogo e diminuição no número de píxeis de não fogo; para o Canal V, foi observado que os valores maiores que 0,55 correspondem aos píxeis de fogo; em relação ao Canal L, os valores maiores que 0,15 correspondem em maioria aos píxeis de fogo.

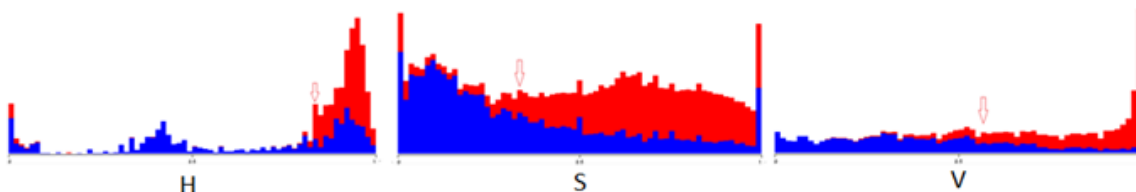


Figura 6.5: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSV.

Para o canal HSI (Figura 6.7), o autor Horng et al. (2005), havia sugerido que para o canal H, os valores deveriam ser menores que 0,16, os valores de S ser maiores que 0,2 e os valores de I ser maiores que 0,4. No nosso estudo, constatou-se que os valores de H devem ser menores que 0,16, os valores de S maiores que 0,2 e os valores de I devem ser maiores que 0,32.

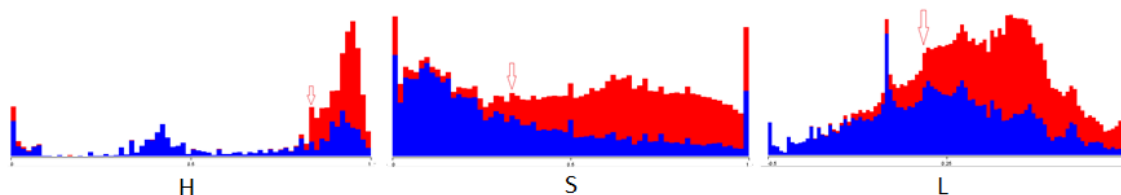


Figura 6.6: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSL.

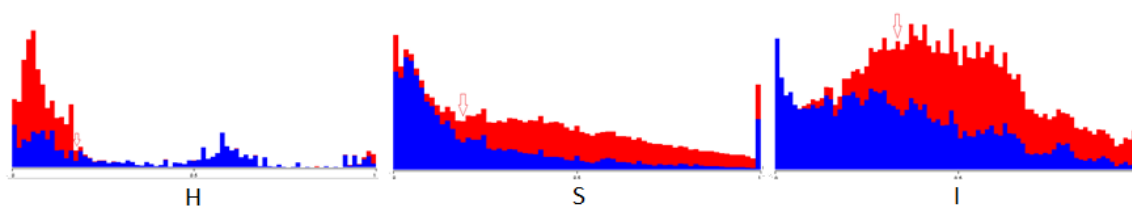


Figura 6.7: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor HSI.

CIELab

Para os canais *CieLab*, conforme ilustra o gráfico na Figura 6.8, os valores de cada canal devem ser: canal L^* maior que 0,338, canal a^* maiores que 0,537 e canal b^* maiores que 0,63.

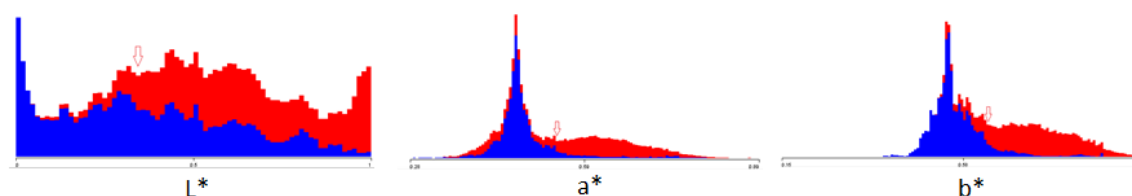


Figura 6.8: Visualização gráfica do quantitativo de píxeis de fogo e não fogo nos canais do espaço de cor *CieLab*.

Conclusão da análise dos canais de cor

A fim de resumir a análise dos histogramas e espaços de cor, a equação 6.1 mostra os pontos principais de mudança que indicam as maiores possibilidades de píxeis serem considerados de fogo ou não. Porém, esta análise precisa de um refinamento, uma vez que é necessário identificar os canais de cor que possuem melhor potencial para destacar os píxeis de fogo. Analisando os

canais de cor, notamos que as diferenças mais acentuadas encontram-se nos canais: $R(RGB)$, Cb e $Cr(YCbCr)$, $V(YUV)$, $H(HSI$ e $HSV)$, $V(HSV)$ e finalmente a e $b(CIE\text{Lab})$.

$$p(x, y) = \begin{cases} (RGB) \rightarrow R > 0,55 \wedge G > 0,05 \wedge B < 0,55 \\ (YCbCr) \rightarrow Y > 0,23 \wedge Cb < 0,44 \wedge Cr > 0,57 \\ (YUV) \rightarrow Y > 0,23 \wedge U < 0,46 \wedge V > 0,57 \\ (HSV) \rightarrow H > 0,82 \wedge S > 0,328 \wedge V > 0,55 \\ (HSL) \rightarrow H > 0,82 \wedge S > 0,328 \wedge L > 0,15 \\ (HSI) \rightarrow H < 0,16 \wedge S > 0,2 \wedge I > 0,32 \\ (CIE\text{Lab}) \rightarrow L > 0,33 \wedge a > 0,53 \wedge b > 0,63 \end{cases} \quad (6.1)$$

Ainda analisando os valores de cada um dos canais, vale destacar os valores de desvio padrão para os píxeis de fogo. Quando o desvio padrão é baixo, significa que há uma menor variação dos valores dos píxeis de fogo, o que indica maior proximidade dos valores à média. Dessa maneira, a Tabela 6.1 ilustra os valores de desvio padrão para cada um dos canais de cor para os píxeis de fogo.

Tabela 6.1: Valores médios dos desvios padrão.

H (hsi)	Cr	V (yuv)	Cb	U (yuv)	b	a	R	V (hsv)	L (hsl)
0.08	0.10	0.10	0.11	0.11	0.11	0.13	0.15	0.15	0.16
H (hsv)	L (Lab)	I (hsi)	Y (ycber)	Y (yuv)	B	S (hsv)	S (hsl)	G	S (hsi)
0.17	0.21	0.21	0.23	0.23	0.28	0.28	0.28	0.29	0.29

Usando um limiar heurístico baseado na mediana, o valor de 0,16 para o desvio padrão foi escolhido para selecionar canais mais promissores. Nesse sentido, os canais: R , Cb , Cr , $V(HSV)$, $H(HSI)$, $L(HSL)$, a e $b(CIE\text{Lab})$ foram os escolhidos.

É necessário explicar que o canal YUV , apesar de utilizado por alguns trabalhos presentes na literatura, não foi incluído neste estudo das probabilidades por ser análogo ao canal $YCbCr$, conforme explicou (Rudz et al., 2009).

6.2 Tabelas de consulta baseadas em probabilidades

Realizando então a combinação dos canais de cor selecionados na seção anterior (Seção 6.1) e utilizando o conceito de tabelas de consulta (TC) de 3 dimensões introduzida por Phillips et al. (2002), 56 tabelas de consulta foram criadas para definir as melhores combinações ($C_{8,3} = \frac{8!}{((8-3)!3!)} = 56$). Além disso, a tabela de consulta para cada espaço de cor também foi criada, totalizando 62 tabelas de consulta.

No entanto, ao invés de utilizar 256 níveis como no trabalho de (Phillips et al., 2002), intervalos menores foram definidos para distribuir os valores de cada canal de cor; desse modo,

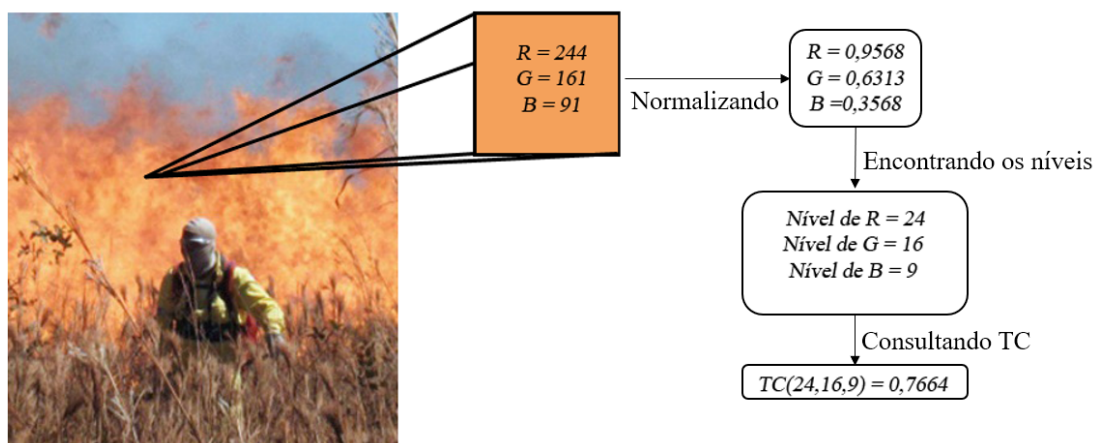


Figura 6.9: Exemplo de cálculo dos canais de cor RGB para consulta na Tabela de consulta.

as TCs criadas possuirão dimensão de $26 \times 26 \times 26$. Com esta medida, a tabela de consulta fica reduzida de 16.777.216 possibilidades para 17.576 possibilidades.

Para a criação da tabela de consulta de probabilidades, a partir da base de imagens coloridas BL1 e da segmentação ideal (*groundtruth*) disponível, foi realizado a extração dos píxeis da imagem, denominados $P(x, y, c)$ – onde x e y correspondem à posição do píxel na imagem e c corresponde ao canal de cor, sendo $c \in \{ch_1, ch_2, ch_3\}$. O cálculo das probabilidades é realizado a partir do número de píxeis de fogo ($\#Fogo(i, j, k)$) dividido pelo número de píxeis ($\#pixeis(i, j, k)$) como mostrado em 6.3, onde $level_{ch1} = i$, $level_{ch2} = j$ e $level_{ch3} = k$.

$$levelc = \frac{P(x, y, c)}{0,04} \quad (6.2)$$

$$TC_{i,j,k} = \frac{\#Fogo_{i,j,k}}{\#pixeis_{i,j,k}} \quad (6.3)$$

Na Equação 6.2 o valor 0,04 foi calculado a partir da divisão do valor máximo de cada canal, que é 1, pelo número de níveis ($\frac{1}{26} \cong 0,04$). A Figura 6.9 ilustra o processo de conversão de um píxel no espaço de cor RGB para resgatar o valor da probabilidade na Tabela de Consulta proposta. Neste exemplo, a probabilidade do píxel selecionado ser de fogo é de 76,64%. O fluxograma, ilustrado a partir da Figura 6.10, mostra como foram criadas as tabelas de consulta.

Outra tarefa importante foi definir um limiar (T) para o valor de probabilidade que melhor represente se um píxel é ou não fogo. Este limiar servirá como um parâmetro de decisão para o algoritmo de localização se o píxel analisado será identificado como fogo ou não. Por exemplo, se o limiar T definido for de 40%, e a probabilidade do píxel analisado de uma imagem for 38%, este píxel **não** será marcado como píxel de fogo. O autor Phillips et al. (2002) sugere um valor acima de 20% em sua técnica, para que um píxel seja considerado um candidato de

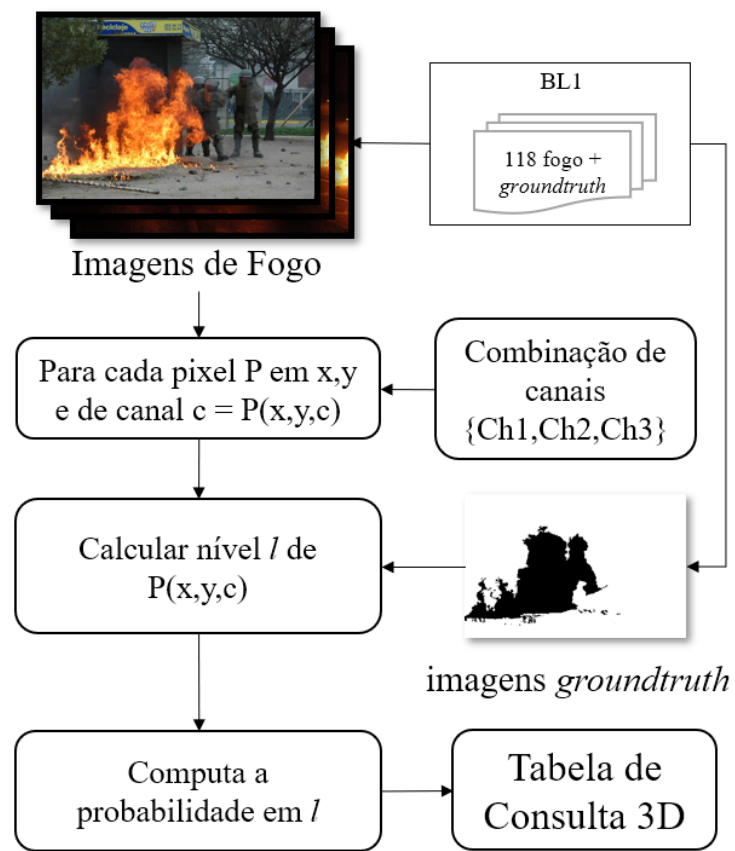


Figura 6.10: Fluxograma de criação das tabelas de consulta.

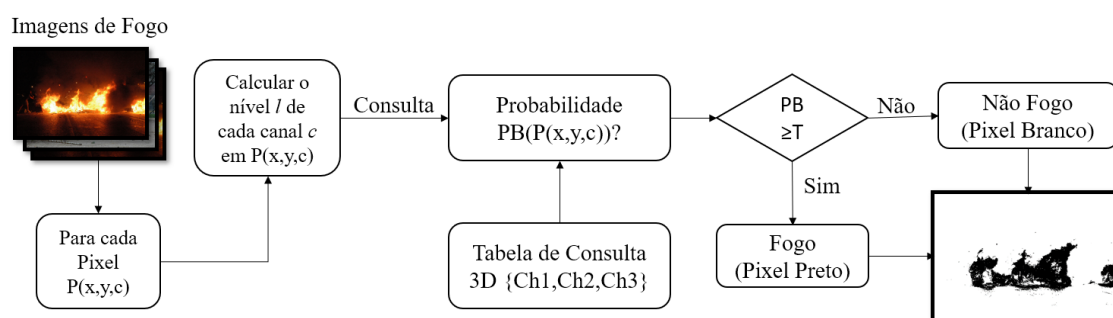


Figura 6.11: Processo de Consulta de um pixel $P(x,y,c)$ em uma tabela de consulta.

fogo. No entanto, para tentar encontrar um valor de limiar ideal (T), foram testados os seguintes valores de probabilidades : 20%, 30%, 40%, 50%, 60%, 70%, 80% e 90%. A Figura 6.11 ilustra como funciona o processo de consulta em uma Tabela de Consulta, dada a probabilidade (PB) de pixel ser ou não um pixel de fogo.

A combinação de canais de cor de espaços de cor diferentes é uma proposta inédita em relação à literatura estudada. Os estudos analisados utilizam canais de cor de um mesmo espaço de cor. Espera-se que as combinações de canais de espaços de cor diferentes ao serem utilizados em conjunto, possam corroborar com o processo de identificação dos píxeis de fogo nas imagens, comparado ao processo de cálculo probabilístico quando utilizados os canais do mesmo espaço de cor.

A Tabela 6.3 destaca os valores encontrados para as 22 combinações que obtiveram os melhores resultados. Destas, 7 combinações obtiveram scores de $F_1 score$ maior que 81%: $(Cb, V(hsv), a)$; $(V(hsv), a, b)$; (RGB) ; (R, a, b) ; $(Cb, L(hsl), a)$; $(Cb, Cr, V(hsv))$ e (R, Cb, Cr) . A partir daí foi criado um método de detecção de fogo que utiliza as 7 melhores combinações dos canais de cor.

A partir dos conceitos expostos até aqui em relação às Tabelas de Consulta, foram conduzidos testes dos limiares de probabilidade de um pixel ser de fogo ou não. Este teste foi conduzido nos 6 espaços de cor na base BL2 e os melhores valores médios para verdadeiros positivos, precisão, acurácia e $F_1 score$ foi para um limiar superior a 70%, como pode ser visualizado a partir da tabela 6.2.

Então, cada tabela de consulta foi testada para a mistura dos canais na base BL1 de Chino et al. (2015) com 118 imagens de fogo. Para definir os melhores valores, foram utilizadas 5 métricas de avaliação: $F_1 score$ (F1), Acurácia (Acc), Precisão, Verdadeiro Positivo (TP) e Falso Positivo (FP). Para melhor comparar as melhores combinações, os valores obtidos por meio dos seis espaços de cor também estão incluídos. Das 62 TCs criadas, foram selecionados os 22 melhores resultados que estão descritos na Tabela 6.3. Os valores de DP, dizem respeito ao valor do desvio padrão da métrica.

Tabela 6.2: Valores para o limiar de probabilidades testados e seus resultados.

Limiar	TP	FP	F_1score
20%	65.7%	2.70%	73.14%
30%	70.22%	3.05%	76.27%
40%	75.08%	3.47%	79.71%
50%	75.91%	4.15%	80.07%
60%	77.17%	4.72%	80.41%
70%	80.09%	4.96%	81.38%
80%	79.83%	5.36%	79.50%
90%	64.41%	5.90%	67.32%

Tabela 6.3: As 22 melhores combinações dos canais de cor ordenados por valores da métrica F_1score (F1).

Combinação	TP %	DP %	FP %	DP %	F1 %	DP %	Acc %	DP %
Cb, V(hsv), a	80,91	25,28	5,09	7,07	81,64	20,63	85,76	12,32
V(hsv), a, b	80,45	25,38	5,09	7,04	81,33	20,67	85,53	12,32
RGB	80,19	25,11	5,02	6,89	81,29	20,20	85,44	12,18
R, a, b	80,36	25,39	5,09	7,04	81,28	20,67	85,48	12,31
Cb, L(hsl), a	80,25	25,46	5,07	7,03	81,22	20,61	85,44	12,37
Cb, Cr, V(hsv)	80,32	25,70	5,09	6,97	81,16	21,07	85,47	12,50
R, Cb, Cr	80,31	25,72	5,09	6,97	81,16	21,09	85,46	12,51
R, L(hsl), a	79,98	25,61	5,12	7,12	80,99	20,72	85,28	12,41
YCbCr	79,95	25,62	5,11	7,08	80,97	20,82	85,26	12,45
CIELab	79,96	25,89	5,03	6,96	80,95	21,12	85,32	12,57
R, Cr, b	79,78	25,76	5,14	7,07	80,81	20,92	85,17	12,45
R, Cr, b	79,78	25,76	5,14	7,07	80,81	20,92	85,17	12,45
HSI	79,93	26,43	5,38	7,48	80,68	21,22	85,13	12,77
HSV	79,72	26,61	5,17	7,27	80,58	21,46	85,13	12,86
Cb, L(hsl), H(hsi)	80,04	26,83	5,43	7,58	80,57	21,94	85,16	12,98
Cr, L(hsl), b	79,08	26,08	5,12	7,03	80,34	20,98	84,83	12,59
Cr, V(hsv), a	78,85	26,27	5,12	7,01	80,16	21,03	84,72	12,66
V(hsv), H(hsi), b	79,51	27,33	5,34	7,52	80,14	22,34	84,94	13,14
R, Cr, a	78,74	26,32	5,12	7,01	80,08	21,05	84,66	12,67
Cr, V(hsv), b	78,94	26,68	5,08	7,05	80,05	21,86	84,78	12,81
HSL	79,00	27,23	5,10	7,21	79,98	22,08	84,80	13,13
R, H(hsi), b	79,31	27,83	5,26	7,49	79,87	23,09	84,88	13,35

Utilizando as Melhores Combinações

Os resultados presentes na Tabela 6.3 destacam que 7 combinações obtiveram scores de F_1score maior que 81%: (Cb, V(hsv), a); (V(hsv), a, b); (RGB); (R, a, b); (Cb, L(hsl), a); (Cb, Cr, V(hsv)) e (R, Cb, Cr). A partir disto, foi criado um método de detecção de fogo que utiliza as 7 melhores combinações dos canais de cor. O fluxograma ilustrado na Figura 6.12 mostra como funciona o método proposto baseado em probabilidades.

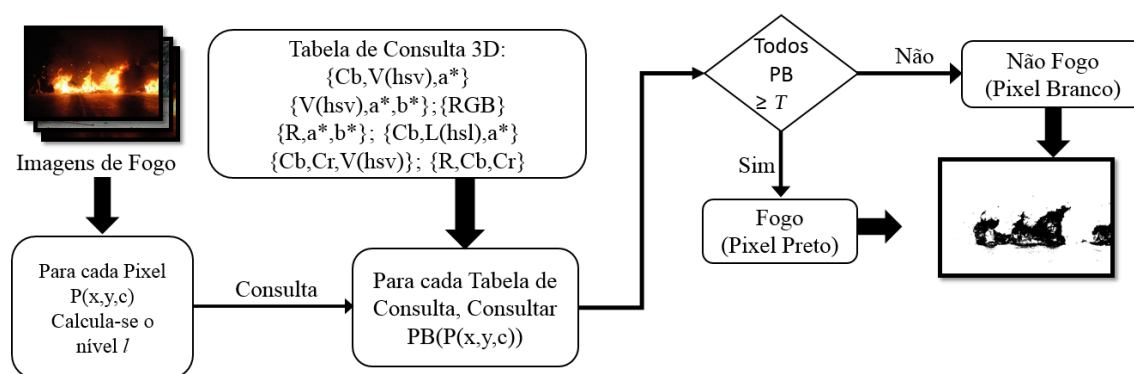


Figura 6.12: Método Proposto baseado em Probabilidades.

As bases BL1 e BL2 foram testadas para verificar a qualidade de segmentação do método proposto. Vale salientar que para as imagens de não fogo, as métricas computadas são as taxas de Verdadeiros Negativos (TN), pois em imagens de não fogo, a solução ideal só possui Verdadeiros Negativos, o que implica em zerar métricas que dependam da taxa de Verdadeiros Positivos (TP). A Tabela 6.4 resume os resultados obtidos com o método baseado nas probabilidades proposto, onde TP corresponde a Verdadeiros Positivo, FP aos Falsos Positivos e TN são os Verdadeiros Negativos.

Tabela 6.4: Resultados obtidos a partir do método das probabilidades proposto.

Base de Imagens	TP	FP	Precisão	Acurácia	$F_1 score$	TN
BL1	80.20%	5.72%	93.90%	87.61%	81.32%	91.30%
BL2	81.08%	4.98%	88.12%	88.36%	82.5%	97.10%

Como pode ser observado, a utilização de Tabelas de Consulta das probabilidades parece promissor para a detecção de fogo. A Taxa de falsos positivos é baixa, sendo 5,72% e 4,98% para as bases BL1 e BL2 respectivamente.

6.3 Localização de fogo usando entropia e rotulação

Depois de analisar diversas imagens foi possível observar que as regiões de fogo parecem ser mais irregulares e instáveis do que as regiões de não fogo. Tal ideia motivou um experimento adicional, utilizando o conceito de Entropia, que nada mais é do que uma medida estatística que representa a desordem de um sistema, conforme descrito na Subseção 2.2.2, do Capítulo 2.

Para aplicar a fórmula da entropia à base BL1, a seguinte estratégia foi utilizada: (1) pre-segmentação das imagens de não fogo, (2) Rotulação das áreas detectadas como fogo nas imagens de não fogo e Rotulação das imagens de fogo a partir da solução Groundtruth e (3) Análise dos valores de entropia calculados para cada um dos rótulos destacados das imagens.



Figura 6.13: Processo de rotulação em imagens de fogo.

O Processo de rotulação consiste em separar em rótulos as áreas segmentadas conectadas por píxeis vizinhos. Nas imagens de fogo da BL1 utilizou-se o processo de rotulação para poder analisar o comportamento dos valores de entropia em cada rótulo para os canais R, G e B e para os valores em escala de cinza, assim como mostrado na Figura 6.13.

O mesmo processo foi repetido para as imagens de não fogo, porém, uma pré-segmentação foi realizada utilizando a técnica de probabilidades aplicada ao canal *RGB*, com limiar de probabilidade $PB \geq 0,5\%$ (para capturar mais elementos de não fogo). Depois deste processo, as imagens binárias geradas pela pré-segmentação, foram processadas para extração dos rótulos e respectivos cálculos de entropia.

Resultados Obtidos

Baseado no método descrito, foi então realizada a separação dos rótulos nas imagens de fogo e não fogo. Deste processo foram encontrados 674 rótulos das imagens de fogo e 1.230 rótulos das imagens de não fogo. Para os rótulos de Fogo, é importante incluir a maior parte dos rótulos (linha SIM na Tabela 6.5) e para os rótulos de não fogo é importante excluir a maior parte dos rótulos (linha NÃO na Tabela 6.6). Com isto em mente, de maneira empírica, foi possível encontrar uma regra de entropia 6.4 e 6.5 que contemplasse 94,2% dos rótulos de fogo e que excluiria pelo menos 39,51% dos píxeis de não fogo. Nas tabelas 6.5 e 6.6 os valores E_R correspondem à Entropia do canal Vermelho, E_G à entropia do canal Verde, E_B à entropia do canal Azul e E_{Gr} à entropia do canal em escala de cinza.

Tabela 6.5: Rótulos de Fogo e regras de entropia.

Rótulos de Fogo e regras de entropia						
	$E_R < E_G$	$E_R < E_B$	$E_R < E_B \wedge E_R < E_G$	$E_R < E_{EC}$	$E_G > E_B$	Regra
SIM	66.1%	44.9%	78.2%	65.4%	76.2%	94.2%
NÃO	33.9%	55.1%	21.8%	34.6%	23.8%	5.8%

Tabela 6.6: Rótulos de não Fogo e regras de entropia.

Rótulos de não fogo e regras de entropia						
	$E_R < E_G$	$E_R < E_B$	$E_R < E_B \wedge E_R < E_G$	$E_R < E_{EC}$	$E_G > E_B$	Regra de Entropia
SIM	30.51%	34.13%	77.98%	28.30%	50.58%	60.49%
NÃO	69.49%	65.87%	22.02%	71.70%	49.42%	39.51%

$$\text{Regras} \left\{ \begin{array}{l} R1 = E_G > E_B \\ R2 = E_R < E_G \wedge E_R < E_B \\ R3 = E_R > E_G \wedge E_R > E_B \\ R4 = E_R < E_{Gr} \end{array} \right. \quad (6.4)$$

$$\text{Regra de Entropia } (R_E) = \{R1 \vee [(R2 \wedge R4) \vee (R3 \wedge R4)]\} \quad (6.5)$$

A partir de cada um dos rótulos, e os valores de entropia para os canais R, G, B e escala de cinza, conforme mostram as Tabelas 6.5 e 6.6 um padrão de valores de entropia para rótulos de fogo e não fogo foi identificado. A identificação do padrão possibilita a remoção de elementos de não fogo que possuem alta probabilidade de terem coloração próxima aos píxeis de fogo e dessa forma reduzir o número de falsos alarmes. Nesse sentido, a Figura 6.14 mostra o fluxograma do método proposto que combina canais de cores em conjunto com a rotulação de áreas com alta probabilidade de ter fogo com o cálculo de entropia.

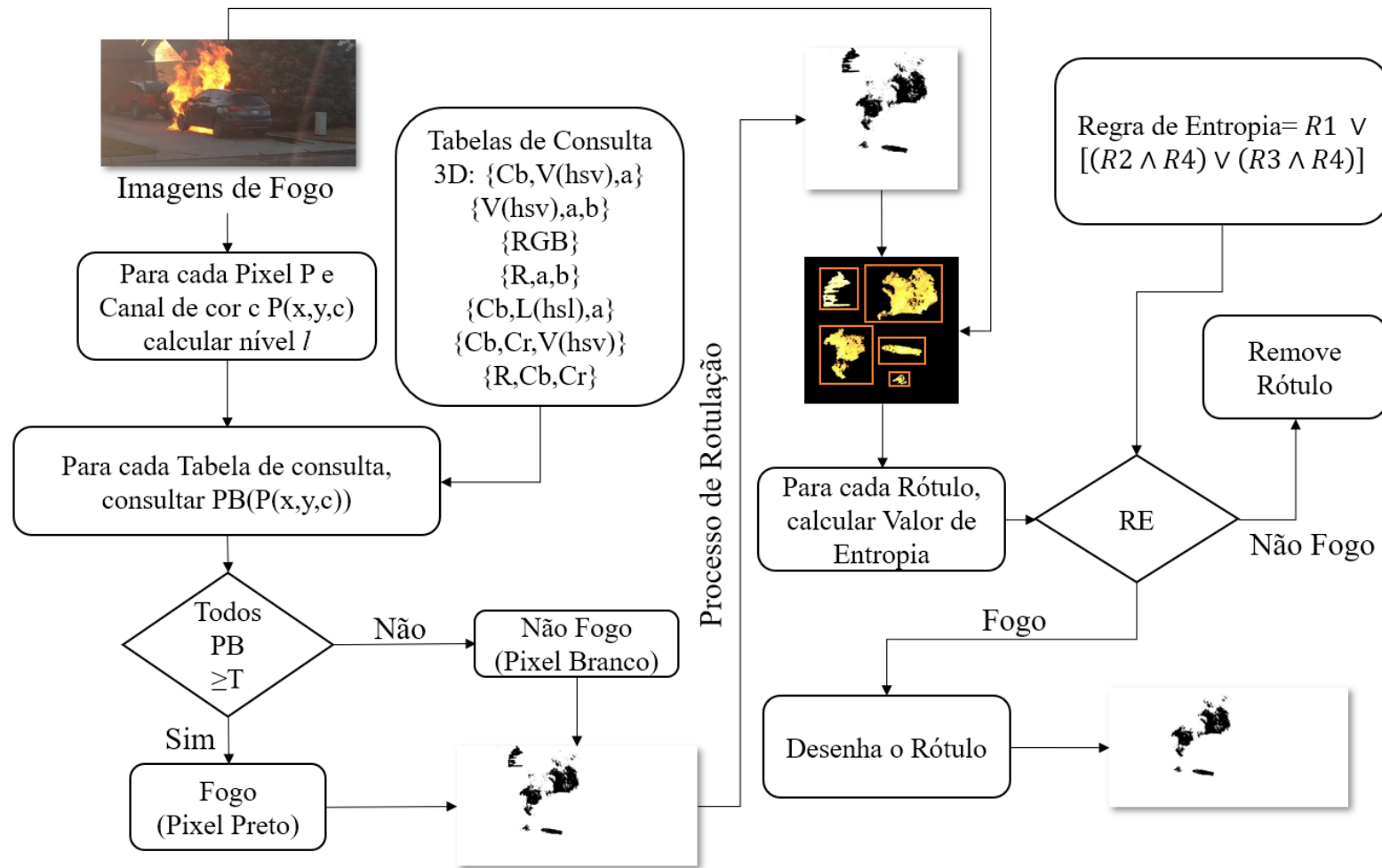


Figura 6.14: Método de localização de píxeis de fogo utilizando entropia para retirada de elementos de não fogo.

Aplicando a técnica de probabilidades em conjunto com a regra de entropia foi criado um método híbrido e este foi testado nas três bases de imagens de localização BL1, BL2 e BL3. A Figura 6.14 mostra o fluxograma do método proposto e a Tabela 6.7 mostra os resultados nas bases, utilizando o recurso de entropia (R_E -Sim) e sem o uso de entropia (R_E -Não).

Tabela 6.7: Resultados de Segmentação usando Probabilidades.

Base de Imagens	TP	FP	Acurácia	F_1score	TN
BL1 – (R_E - Sim)	82.2%	5.8%	88.4%	85.1%	92.6%
BL 1 – (R_E - Não)	80.2%	5.7%	87.6%	81.3%	91.3%
BL 2 – (R_E - Sim)	82.1%	5.1%	89.4%	83.3%	98.1%
BL 2 – (R_E - Não)	81.1%	5.0%	88.36%	82.5%	97.10%

Os resultados da Tabela 6.7 obtidos mostram que com a utilização de Probabilidades em conjunto com as regras de Entropia as taxas de qualidade de segmentação melhoram para a maioria das métricas estudadas tanto na base BL1 e na base BL2. Como resultado, a qualidade de segmentação dos píxeis de fogo aplicados na base de testes BL2 foram: 82,2% de Verdadeiros Positivos (TP - *True Positive*), 88,6% de Acurácia, 85,0% de F_1score e 92,6% de taxa de verdadeiro negativo (TN). Sendo assim, esta é uma técnica que pode beneficiar os métodos de detecção de fogo, com vistas a uma melhora na qualidade de detecção e diminuição do número de falsos alarmes.

Visando estudar outras possibilidades para o problema de segmentação e consequente localização de fogo em imagens, um estudo com índices de coloração foi desenvolvido e é apresentado na subseção seguinte.

6.4 Localização de fogo usando índices de cor

A aplicação de índices de cor é amplamente utilizada na indústria, conforme já exposto na seção 2.2.1 do Capítulo 2, porém não existem registros na literatura da utilização de índices de cor para localização de píxeis de fogo. Nesse sentido, planeja-se testar diferentes índices de cor que destaquem a presença de fogo em imagens. Para isso, os índices de cor para Vermelho, Amarelo e Marrom foram escolhidos para serem estudados, testados e comparados. Estes índices foram selecionados, pois a maioria dos incidentes com fogo emitem chamas de cores alaranjadas, em tons mais claros, ou seja, mais próximos ao amarelo ou em tons mais escuros, os quais são mais próximos das cores vermelho e marrom.

Em relação ao índice de Vermelho (*Redness*) foram selecionadas três técnicas para serem avaliadas, (Gasparini and Schettini, 2005; Held, 2002; Smolka et al., 2003), tais técnicas foram utilizadas na literatura como complemento no processo de redução de olhos vermelhos das imagens. Ambas as três técnicas utilizam a transformação em índice de cor vermelha a partir dos canais do espaço de cor RGB.

Ao avaliar as fórmulas de conversão do espaço de cor RGB para os índices de vermelho, percebe-se que no trabalho de Smolka et al. (2003) há a maximização do canal azul ou verde (ver Equação 2.1 no Capítulo 2) por meio do cálculo do quadrado máximo entre azul e verde, de tal forma que as imagens resultantes da aplicação dos índices de vermelho para esta técnica tendem a dar destaque também para as cores azul ou verde. A Figura 6.15 (d) deixa isto evidente, uma vez que o letreiro de fundo azul (logo acima das chamas) fica com mais destaque que o fogo.

Avaliando a fórmula de (Gasparini and Schettini, 2005) temos que há o fator 4 de multiplicação sobre o canal vermelho, somado aos canais verde e azul e subtraídos dos canais com maior e menor valor. Este cálculo maximiza a possibilidade de partes avermelhadas de fogo terem destaque nas imagens. A Figura 6.15 (c) evidencia o destaque às regiões de fogo e mostra poucos ruídos (regiões de não fogo que foram destacadas pelo método).

Já em relação à fórmula proposta por (Held, 2002) temos na transformação para o índice de cor vermelha a subtração do canal vermelho do valor mínimo entre os canais verde e azul. Dessa forma, há o destaque do canal vermelho na imagem formando o índice de vermelho do autor. Como a coloração de fogo alaranjado tem alta concentração de vermelho, espera-se que haja destaque nas regiões de fogo nas imagens. A Figura 6.15 (b) mostra um exemplo de transformação do espaço de cor RGB para o índice de vermelho proposto por Held e sua equipe de pesquisa. Como exemplo dos resultados das transformações realizadas a partir dos valores RGB das imagens, as imagens ilustradas pela Figura 6.15 mostram os resultados dessas transformações. A partir de um estudo com diversas imagens a proposta de índice de vermelho defendida por (Held, 2002) dá um maior destaque às regiões de fogo, razão pela qual planeja-se aplicar este índice de vermelho a imagens de fogo.

Em relação aos índices de amarelo, temos a utilização de diferentes espaços de cor para cada uma das três técnicas selecionadas para este estudo. Os espaços de cor utilizados além do RGB (Doi, 2012), são o CMYK (Robinson, 1998) e o XYZ (ASTM, 2015). Na proposta de índice de amarelo de (Doi, 2012), o valor do índice é calculado a partir da soma dos canais vermelho e verde do espaço RGB, isso destaca porções mais amareladas da imagem. A Figura 6.16(b) mostra um exemplo da transformação do espaço RGB para o Yellowness proposto por Doi.

Já a proposta de índice de amarelo proposta por Robinson (1998) utiliza o espaço de cor CMYK, que é baseado nas cores Ciano, Magenta, Amarelo e Preto. Na proposta de Robinson, o índice de amarelo é calculado baseado na divisão do canal amarelo pela raiz da soma dos quadrados dos canais Ciano, Magenta e Amarelo. Com esta formulação matemática, temos o destaque da cor amarela suavizada pelos valores dos canais Magenta, Ciano e amarelo. A Figura 6.16(c) mostra um exemplo da transformação do espaço RGB para o índice de amarelo proposto por Robinson.

A ASTM (*American Society for Testing and Materials*) no trabalho ASTM (2015) propôs o índice de amarelo para identificar o grau de amarelamento de plásticos. No diagrama cromático da Comissão Internacional de Iluminação, quanto maior o valor de X, mais próximo ao amarelo a cor está, e quanto maior o valor de Y, mais próximo ao verde a cor; já o canal Z tem

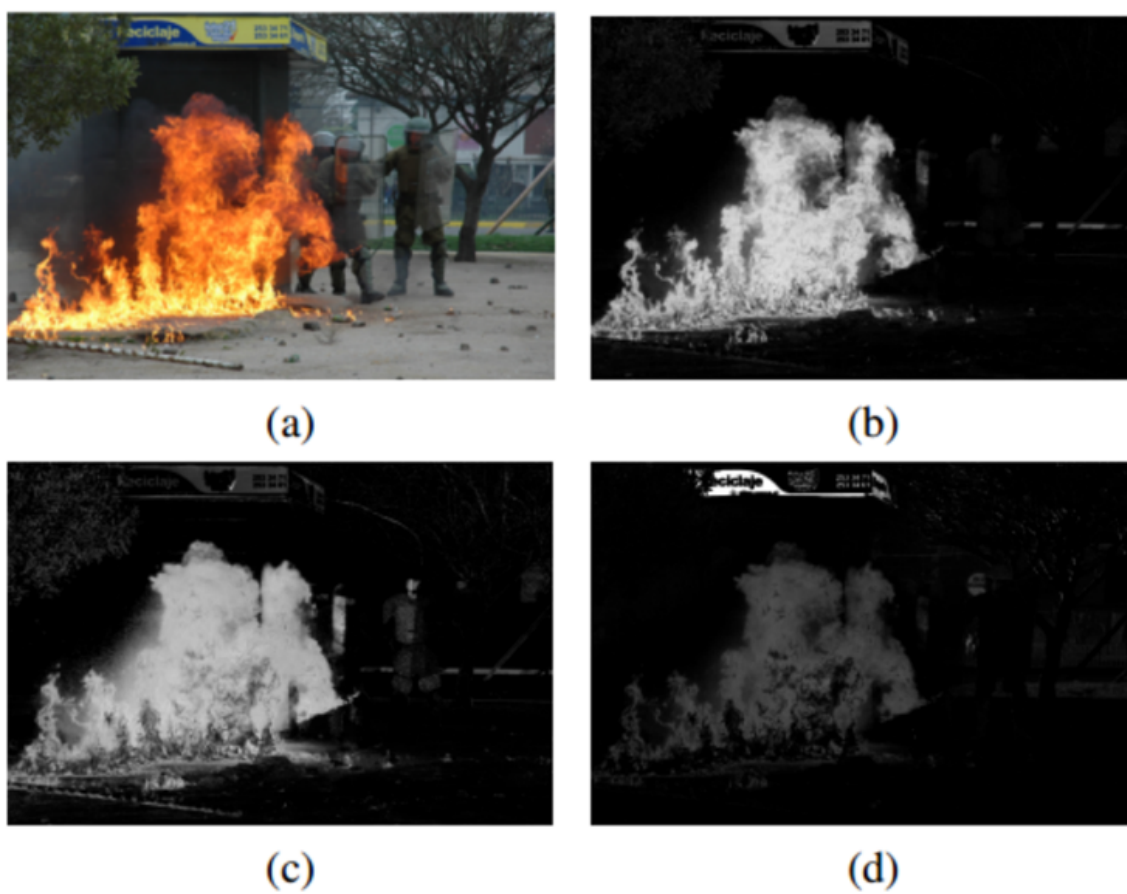


Figura 6.15: Aplicação do índices Redness: a) Imagem original; b) (Held, 2002); c) (Gasparini and Schettini, 2005); d) (Smolka et al., 2003).

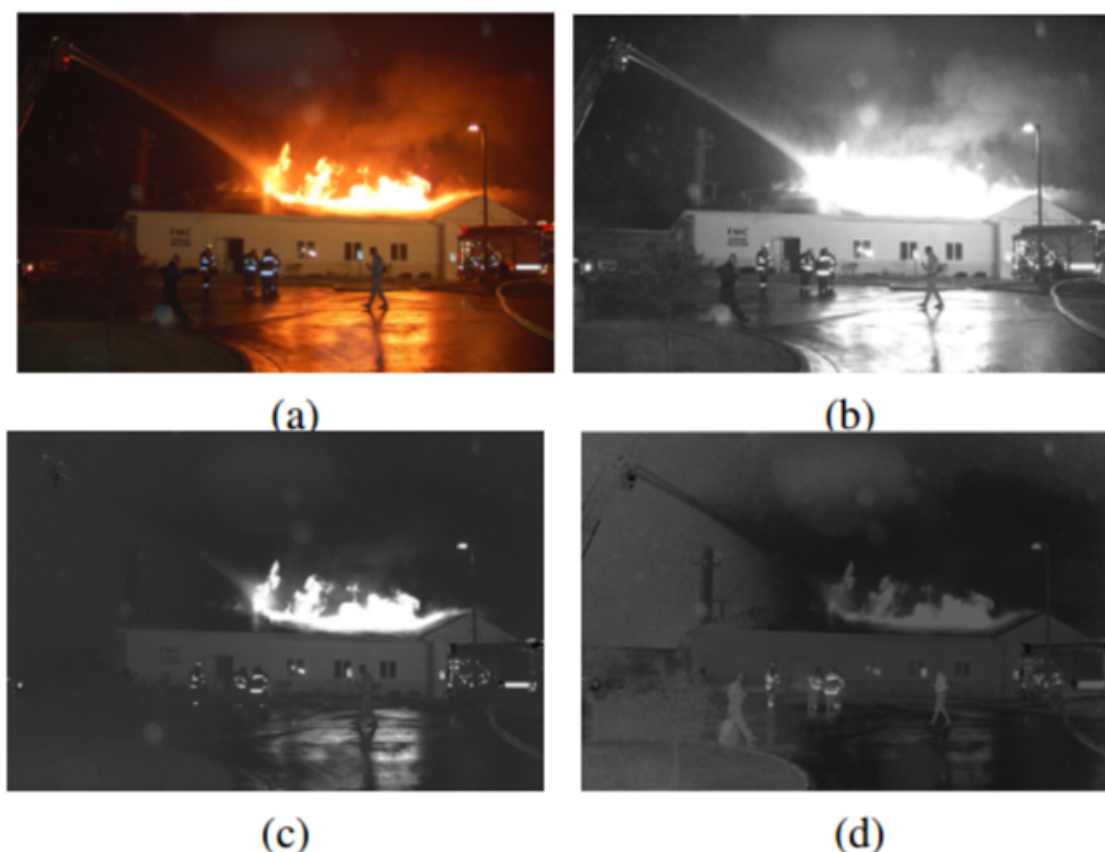


Figura 6.16: Aplicação do índices de Amarelo: a) imagem original, b) (Doi, 2012), c) (Robinson, 1998) e d) (ASTM, 2015).

maior proximidade à cor azul. Dessa forma, ao realizar a multiplicação dos canais X e Z por coeficientes distintos para C_x e C_z , respectivamente, e posteriormente subtrair estes dois valores, dividindo o resultado por Y, conforme Equação 2.6 os valores do índice de amarelamento tendem a equalizar os tons amarelados quando os valores de X e Y forem muito grandes e o valor de Z for menor. A Figura 6.16 (d) mostra um exemplo de imagem de fogo convertida para o índice de amarelo proposto pela ASTM.

Ao aplicar e avaliar os três métodos de índices de Amarelo, presentes na Subseção 2.2.1 no Capítulo 2 nas imagens de fogo, foi possível visualizar, como exemplificado pela Figura 6.16, que o método proposto por (Robinson, 1998) foi o que melhor destacou as regiões amareladas do fogo alaranjado; por este motivo, planeja-se utilizar este índice para auxiliar no processo de localização de píxeis de fogo em imagens.

Após a seleção dos índices de vermelho e amarelo, outra cor que pode auxiliar no processo de detecção de fogo é a cor marrom, pois é uma cor próxima ao laranja, cor bastante presente nas imagens de fogo amarelado. A análise dos índices de marrom nas imagens de fogo foi realizada a partir dos trabalhos de Ricauda et al. (2015) e Pathare et al. (2013).



Figura 6.17: Aplicação de índices de Marron: a) imagem original, b) (Ricauda et al., 2015) e c) (Pathare et al., 2013).

Ricauda et al. (2015) utilizou o índice de marrom para medir a qualidade de frutas congeladas; para isso, sua formulação é baseada no sistema RGB e leva em consideração um coeficiente k que deve estar entre 3 e 4 aplicado ao canal Vermelho; o autor ainda subtrai os canais verde e azul após a aplicação do coeficiente. Dessa maneira, conforme pode ser visualizado na Figura 6.17 (b) há o destaque de áreas alaranjadas do fogo.

Já no trabalho de Pathare et al. (2013), os autores optaram por utilizar o espaço de cor CIELab para analisar a qualidade de frutas e comidas frescas. A Figura 6.17 (c) mostra um exemplo da aplicação do algoritmo de Pathare em imagens de Fogo. Na imagem é possível observar destaque aos contornos do fogo.

A análise dos dois métodos de índices de marrom nos mostrou que o método proposto por (Ricauda et al., 2015) destaca as áreas de fogo de maneira mais eficiente que o método proposto por (Pathare et al., 2013). Assim, a utilização do índice de marrom proposto por Ricauda foi o selecionado para auxiliar no processo de detecção de fogo nas imagens.

Determinados os três índices de cor que serão utilizados, o modelo de detecção de pixel de fogo baseado nos índices de cor, será montado conforme Figura 6.18. O diagrama mostra o método proposto, onde para cada pixel, primeiro se calcula o nível numérico correspondente conforme Equação 6.2, e então é realizada a consulta da probabilidade na tabela de consulta 3D. Caso o pixel possua probabilidade maior ou igual a T de ser um pixel de fogo, então este é marcado com a cor preta, caso contrário, com a cor branca.

A Figura 6.18 mostra o fluxograma que representa o método proposto, onde para cada pixel, primeiro se calcula o nível numérico correspondente conforme Equação 6.2, e então é realizada a consulta da probabilidade na tabela de consulta 3D. Caso o pixel possua probabilidade maior ou igual a T de ser um pixel de fogo, então, este é marcado com a cor preta, caso contrário, com a cor branca.

Resultados Obtidos

O método para segmentação de pixel de fogo usando índices de coloração é dividido em duas etapas: 1) Criação de uma tabela de consulta 3D que armazena valores de probabilidade de

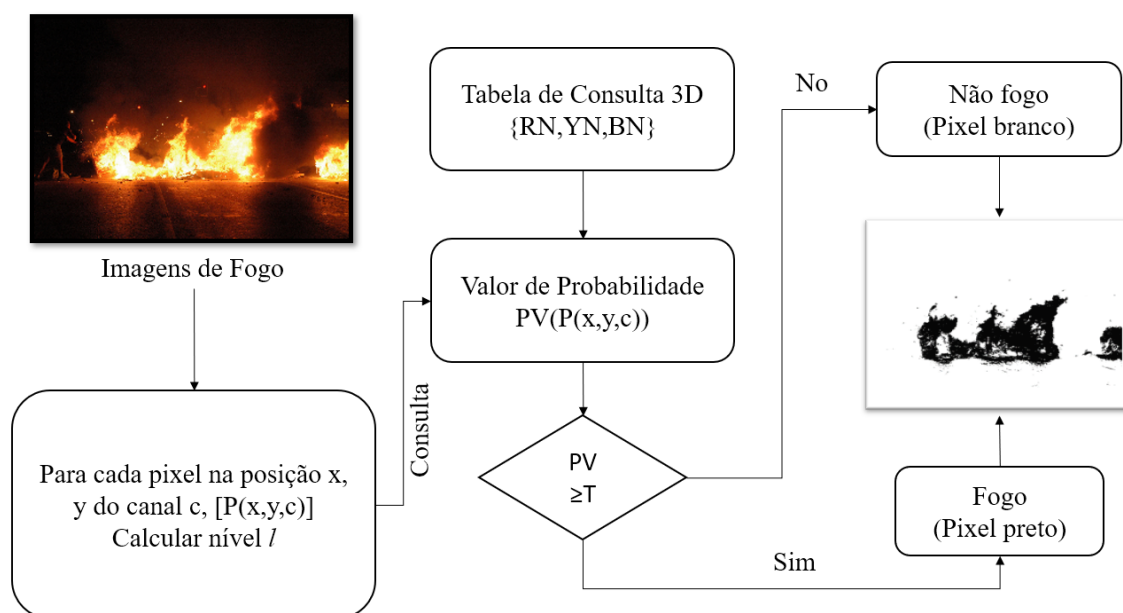


Figura 6.18: Processo de construção da tabela de consulta para os índices de Vermelho, Amarelo e Marrom.

coloração de um pixel de cor sendo ou não um pixel de fogo; 2) Realização de testes para selecionar os melhores valores limite probabilísticos.

Combinando índices de vermelho R_{nh} , índices de amarelo Y_{nr} e índices de marrom B_{nr} foi aplicada a ideia de usar as tabelas de consulta propostas por Phillips, Shah, e Lobo (Phillips et al., 2002), conforme já explanado na Seção 6.2. Adicionalmente, foram criadas tabelas de consulta para os seguintes espaços de cor: RGB , $YCbCr$, HSI , HSL , HSV e $CieLab$, totalizando 7 tabelas de consulta de probabilidade que serão utilizadas como critério comparativo à proposta de detecção utilizando índices de coloração.

A seleção dos valores para definição dos limiares consiste em testar vários valores e avaliar quais expressam melhor que um pixel pode ou não ser considerado como candidato a ser um pixel de fogo.

Utilizando como base o modelo de probabilidades definido na Seção 6.2, cujo cálculo do nível correspondente de uma cor, segue a equação definida em 6.2 e a tabela de consulta é analisada a partir da equação 6.3. Optou-se por utilizar um limiar de 70% para identificar se um pixel de um canal c é considerado fogo ou não. A avaliação do método de coloração foi realizada usando dois conjuntos de imagens de imagens de fogo: BL1 proposto por (Chino et al., 2015) com 226 imagens (119 imagens de fogo) e BL2 com 110 imagens (55 imagens de fogo) criadas para esta proposta.

Os resultados da Tabela 6.8 mostram que a combinação de coloração dos índices de vermelho, amarelo e marrom tem eficiência semelhante à da combinação dos seis espaços de cores estudados. As métricas utilizadas para avaliar tal situação foram: Positivo Verdadeiro (TP),

Falso Positivo (*FP*), Acurácia e *F₁score*. Estes resultados mostram que esta combinação pode ser uma alternativa para detectar píxeis de fogo.

Tabela 6.8: Resultado da aplicação do método probabilístico aplicado a imagens que contenham fogo.

Espaço de Cor	TP	FP	Acurácia	<i>F₁score</i>
RGB	82,49%	5,02%	88,74%	85,29%
YCbCr	82,25%	5,11%	88,56%	84,97%
CIELab	82,26%	5,03%	88,62%	84,95%
Rn,Yn,Bn	82,05%	5,23%	88,40%	84,69%
HSI	82,23%	5,38%	88,43%	84,68%
HSV	82,02%	5,17%	88,43%	84,58%
HSL	81,30%	5,10%	88,10%	83,98%

Nesta seção foi proposto um método de segmentação de píxeis de fogo baseado em índices de coloração baseado em: vermelho, amarelo e marrom. Estes índices foram combinados e computados em uma tabela de consulta que armazena as probabilidades de um pixel ser de fogo. Então, para identificar se o pixel é de fogo ou não, é realizada uma consulta na Tabela de Consulta e a probabilidade do pixel ser de fogo for maior que o limiar de 70%, então, este pixel é considerado um pixel de fogo. Esse método foi ainda comparado a outras 12 técnicas de segmentação de fogo oriundas do estudo bibliográfico. Os resultados comparativos serão apresentados na Seção 6.6.

Este estudo, assim como o presente na Seção 6.4, utilizou das tabelas de probabilidade para combinar canais de cor modificados por meio das teorias de índices de cor para as cores Vermelho, Amarelo e Marrom, como resultado da qualidade de segmentação, a combinação das colorações Vermelha, Amarela e Marrom foram capazes de obter 82,05% de píxeis verdadeiros positivos, 5,23% de falsos positivos, 88,4% de acurácia e 84,69% de *F₁score*.

6.5 Localização de píxeis de fogo usando aprendizagem de máquina

Para o experimento que envolve o uso de algoritmos de aprendizagem de máquina voltados para a segmentação, foi utilizada a base BL3, com um total de 1038 imagens segmentadas (sendo 516 imagens de fogo). Os algoritmos de aprendizagem de máquina testados foram: SVM (*Support Vector Machine*), Árvores de Decisão (*Decision Trees – DT*), *Random Forest* (RF) – e Adaboost com árvore de decisão (*Decision Stumb*). A execução dos testes foi realizada utilizando as bibliotecas de aprendizagem de máquina do software WEKA (Hall et al., 2009) e a biblioteca scikit-learn (Pedregosa et al., 2011) para o python.

Para preparar a base de imagens com o intuito de treinar e testar algoritmos de aprendizagem de máquina, primeiro foi necessário identificar o vetor de características que iria

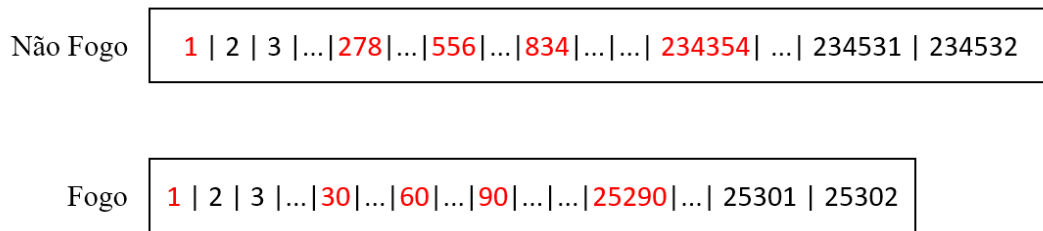


Figura 6.19: Exemplo de vetores de uma imagem de Fogo e Não Fogo e os píxeis extraídos em vermelho, utilizando $c=30$.

ser carregado nos algoritmos classificadores. Isto foi feito convertendo os valores dos píxeis RGB das imagens para os canais dos seguintes espaços de cor: RGB, YCbCr, HSI e CIELab, o que resultou em um arquivo com 12 características por pixel. Após este processo de conversão foram encontradas dificuldades para rodar o treinamento dos modelos para todos os arquivos da base, pois os arquivos de treino da base ficaram muito extensos, o que tornou o processo de aprendizagem muito lento.

Visando contornar tal problema, foram coletadas amostras da base. Para que isso fosse possível, foi estabelecida a criação de 10 arquivos de treino e 10 arquivos de testes diferentes (10 *folds*), intercalando os píxeis selecionados e as imagens para cada um dos 5 contextos, a saber: Agricultura, Floresta, Urbano, Rodovia e Indústria. Para a criação dos arquivos, as imagens da base eram carregadas, e então os píxeis de fogo e não fogo eram separados. Após a separação foi definido um coeficiente de espaçamento (Ce), conforme a Equação 6.6, que foi aplicado a fim de coletar proporcionalmente o mesmo número de píxeis de fogo e não fogo espalhados por toda a imagem. Em relação à equação do coeficiente de espaçamento, a variável c determina a distância mínima entre os píxeis da amostra com menor número de píxeis nas imagens, sendo que $\#PNF$ indica a quantidade de píxeis de não fogo e $\#PF$ a quantidade de píxeis de fogo.

A partir da Equação 6.6, garantiu-se uma amostra representativa e balanceada dos píxeis das imagens de fogo e não fogo representativa, pois as amostras eram retiradas de pontos dispersos (a uma distância mínima de C entre os píxeis) nas imagens e balanceadas, pois dessa forma o número de píxeis de fogo e de não fogo selecionados são iguais.

$$Ce(c) = \begin{cases} \text{Se } \#PNF > \#PF \text{ então, } \frac{\#PNF}{\#PF} * c \\ \text{Senão, } \frac{\#PF}{\#PNF} \times c \end{cases} \quad (6.6)$$

Para exemplificar, considerando uma imagem com 259.825 píxeis. Se $c = 30$, e temos $\#PF = 25.302$ e $\#PNF = 234.523$ então temos que extrair 843 ($\#PF/c$) píxeis de fogo mais 843 píxeis de não fogo de amostra da imagem a cada 30 píxeis (c) do conjunto de píxeis de fogo e a cada 278 ($\frac{\#PNF}{\#PF} \times c$) píxeis do conjunto dos píxeis de não fogo. A Figura 6.19 exemplifica este processo por meio dos valores marcados em vermelho nos vetores de fogo e não fogo.

Um algoritmo para separar as imagens que iriam para a base de treino e teste foi então utilizado na proporção de 60% para treino e de 40% para teste. As bases de localização foram divididas em seus respectivos contextos em dez imagens, cada bloco. Dessa maneira, como cada contexto possui pelo menos 100 imagens de fogo e outras 100 imagens de não fogo, a cada iteração na base de imagens era realizada a divisão com 120 imagens para treino (60 de fogo) e 80 imagens para teste (40 de fogo). A partir desta divisão, os píxeis de fogo eram extraídos das imagens com fogo e os píxeis de não fogo, eram selecionados das imagens de não fogo. Tal divisão foi realizada por 10 iterações para cada uma das categorias, criando assim um total de 50 arquivos com píxeis de treino e 50 arquivos com píxeis de teste.

Em relação à escolha dos parâmetros principais dos modelos de aprendizagem de máquina selecionados foi utilizado o método de *grid-search*, que consiste em definir uma matriz de possibilidades de parâmetros e então verificar os resultados para cada tupla de parâmetros, selecionando os parâmetros com melhores resultados. Para o modelo SVM, o kernel RBF apresentou melhores resultados que o kernel linear e os seguintes parâmetros $C = 0,5$ e $gamma = 0,0001$ foram utilizados.

Resultados obtidos

Para apresentar os resultados, os experimentos foram desenvolvidos a partir de quatro diferentes abordagens de aprendizagem de máquina: SVM, Florestas Aleatórias, Árvores de Decisão e Adaboost. A Tabela 6.9 apresenta os resultados obtidos nas bases de teste e treino. Os resultados em questão representam as médias e os respectivos desvios padrão (DP) para as seguintes métricas: taxa de Verdadeiro Positivo (TP), taxa de Falso positivo (FP), Acurácia (Acc) e F_1score (F1).

Os testes foram conduzidos a partir da criação de arquivos que representassem uma amostra significativa das bases estudadas. Para isso, foram criados 10 arquivos de treino e 10 arquivos de teste para cada um dos 5 contextos selecionados: Agrícola, Floresta, Rodovia, Indústria e Urbano. Sendo que para criação dos arquivos, não havia a sobreposição de imagens para os arquivos de treino e teste, respeitando a proporção 60% para treino e 40% para teste.

Tabela 6.9: Resultado da média das métricas dos modelos de aprendizagem de máquina nas bases de Teste e Treino.

Contexto	Modelo	Teste								Treino							
		TP	DP	FP	DP	Acc	DP	F1	DP	TP	DP	FP	DP	Acc	DP	F1	DP
Agrícola	SVM	0,838	0,022	0,162	0,022	0,866	0,023	0,864	0,022	0,826	0,014	0,154	0,016	0,878	0,009	0,878	0,010
	DT	0,836	0,043	0,164	0,043	0,851	0,045	0,843	0,054	0,904	0,012	0,096	0,012	0,922	0,005	0,919	0,005
	RF	0,851	0,047	0,149	0,047	0,858	0,048	0,849	0,056	0,905	0,009	0,095	0,009	0,920	0,005	0,916	0,006
	Adaboost	0,851	0,055	0,149	0,055	0,861	0,046	0,854	0,050	0,894	0,009	0,106	0,009	0,908	0,007	0,905	0,008
Floresta	SVM	0,829	0,017	0,171	0,017	0,863	0,009	0,862	0,011	0,831	0,006	0,169	0,006	0,869	0,006	0,869	0,003
	DT	0,847	0,016	0,153	0,016	0,869	0,008	0,867	0,009	0,867	0,010	0,133	0,010	0,898	0,005	0,897	0,006
	RF	0,847	0,021	0,153	0,021	0,875	0,006	0,874	0,006	0,861	0,004	0,139	0,004	0,898	0,004	0,898	0,006
	Adaboost	0,850	0,019	0,150	0,019	0,874	0,004	0,871	0,007	0,857	0,006	0,143	0,006	0,887	0,004	0,886	0,006
Rodovia	SVM	0,847	0,017	0,153	0,017	0,875	0,023	0,877	0,024	0,852	0,025	0,148	0,025	0,881	0,018	0,884	0,016
	DT	0,897	0,024	0,103	0,024	0,913	0,015	0,913	0,014	0,928	0,006	0,072	0,006	0,943	0,006	0,943	0,006
	RF	0,898	0,021	0,102	0,021	0,916	0,014	0,917	0,014	0,922	0,009	0,078	0,009	0,942	0,005	0,942	0,005
	Adaboost	0,894	0,019	0,106	0,019	0,913	0,015	0,913	0,015	0,909	0,011	0,091	0,011	0,932	0,006	0,932	0,005
Industria	SVM	0,814	0,047	0,186	0,047	0,853	0,023	0,854	0,021	0,827	0,016	0,173	0,016	0,867	0,011	0,866	0,010
	DT	0,855	0,049	0,145	0,049	0,878	0,018	0,876	0,015	0,913	0,011	0,087	0,011	0,928	0,007	0,925	0,007
	RF	0,864	0,053	0,136	0,053	0,888	0,021	0,886	0,017	0,906	0,012	0,094	0,012	0,926	0,007	0,923	0,007
	Adaboost	0,847	0,049	0,153	0,049	0,876	0,021	0,874	0,017	0,886	0,017	0,114	0,017	0,907	0,009	0,905	0,009
Urbano	SVM	0,879	0,029	0,121	0,029	0,896	0,016	0,885	0,020	0,887	0,012	0,113	0,012	0,901	0,008	0,890	0,009
	DT	0,884	0,024	0,116	0,024	0,911	0,009	0,903	0,010	0,901	0,010	0,099	0,010	0,926	0,005	0,919	0,006
	RF	0,888	0,024	0,112	0,024	0,913	0,008	0,906	0,009	0,901	0,010	0,099	0,010	0,925	0,005	0,918	0,005
	Adaboost	0,895	0,026	0,105	0,026	0,912	0,010	0,903	0,011	0,904	0,010	0,096	0,010	0,920	0,005	0,912	0,006

Para o contexto "Agrícola" o método que apresentou melhores resultados para Acurácia e F1-Score na base de testes foi o SVM, com $0,866 \pm 0,023$ de Acurácia contra $0,861 \pm 0,046$ do Adaboost e obtendo $0,864 \pm 0,022$ de F1-Score contra $0,854 \pm 0,05$ do Adaboost.

Em relação ao contexto "Floresta", o melhor desempenho para Acurácia e F1-Score foram localizados com o método de Florestas Aleatórias (*Random Forest*) obtendo $0,875 \pm 0,006$ para acurácia e $0,874 \pm 0,006$ de F1-Score contra $0,874 \pm 0,004$ de Acurácia e $0,871 \pm 0,007$ de F1-Score com o método Adaboost.

No contexto de "Rodovia", o método de Florestas Aleatórias (*Random Forest*) foi o que obteve os melhores desempenhos, sendo $0,916 \pm 0,014$ para acurácia e $0,917 \pm 0,014$ de F1-Score contra $0,913 \pm 0,015$ de Acurácia e $0,913 \pm 0,014$ de F1-Score com o método de Árvores de Decisão (DT - *Decision Tree*).

Para o contexto "Indústria", o método de Florestas Aleatórias (*Random Forest*) foi o que obteve os melhores desempenhos, sendo $0,888 \pm 0,021$ para acurácia e $0,886 \pm 0,017$ de F1-Score contra $0,878 \pm 0,018$ de Acurácia e $0,876 \pm 0,015$ de F1-Score com o método de Árvores de Decisão (DT - *Decision Tree*).

Para áreas "Urbanas", o o método de Florestas Aleatórias (*Random Forest*) foi o que obteve os melhores desempenhos, sendo $0,913 \pm 0,008$ para acurácia e $0,906 \pm 0,009$ de F1-Score contra $0,912 \pm 0,010$ de Acurácia e $0,903 \pm 0,011$ de F1-Score com o método Adaboost.

Os resultados obtidos mostram quão similar é o desempenho dos métodos de aprendizagem de máquina aqui estudados. O método que obteve os melhores índices para a maioria dos contextos foi aquele referente às Florestas Aleatórias (*Random Forest*), exceto no contexto "Agrícola", onde o SVM se mostrou ligeiramente superior aos outros métodos.

A Seção 6.6 mostra o comparativo de qualidade de segmentação entre os métodos de segmentação de píxeis de fogo presentes na literatura com os métodos propostos neste trabalho.

6.6 Comparativo entre algoritmos de Segmentação de Fogo

Os resultados obtidos em relação à segmentação de fogo dizem respeito a 12 diferentes técnicas descritas na literatura, conforme a Seção 6.6. O comparativo entre os métodos foi realizado a partir de taxas de segmentação separadas em dois grupos: avaliação de imagens de fogo e imagens de não fogo. Para imagens de fogo, as seguintes métricas foram utilizadas: Verdadeiro Positivo (*TP*), Acurácia e *F₁ score*. Para imagens de não fogo, como não há Verdadeiros Positivos a métrica de *F₁ score* fica sempre igual a zero, portanto, para avaliar a qualidade de segmentação as taxas de Verdadeiro Negativo (*TN*) foram utilizadas.

Outro fator importante foi a definição dos limiares dos métodos. Os métodos que indicavam os valores em seus trabalhos foram implementados como tal, porém métodos que não sugeriam valores para os limiares, os mesmos foram localizados a partir de testes e os valores utilizados estão representados na Tabela 6.10.

Tabela 6.10: Limiares utilizados para cada uma das técnicas estudadas.

Técnica	Limiares utilizados
Celik RGB1	-
Celik RGB2	-
Celik YCbCr	$Cr - Cb \leq 40$
Celik Lab	$\alpha = 0.05$
Chen RGB	$St = 60$ and $Rt = 125$
Chen RGB+HSI	$R > 115$
Hornng	$0 < H < 60$, $S > 0.4$ and $I > 0.5$
Ko	$\tau \geq 0.000246$ and $\tau \leq 0.000289$
Philips	$k1 = 0.2$
Rossi	$K = 2$
Santanna	$H < 60$
Vipin	$Th = 70$
Prop. Entropia	$T = 0,6$
Prop. Índice de Cor	$T = 0,7$

Além das técnicas estudadas, os resultados dos experimentos realizados também são mostrados, onde: “Prop Entropia” refere-se à proposta contida na Seção 6.3 e “Prop Coloração” refere-se aos resultados obtidos na proposta contida na Seção 6.4. Os resultados comparativos estão descritos nas tabelas: Tabela 6.11, Tabela 6.12 e Tabela 6.13.

A partir dos dados da Tabela 6.11 (utilizando o BL2) é possível observar que o método com os melhores resultados das métricas para imagens de fogo foi a proposta de ChenRGB, porém o número de Verdadeiros Negativos nas imagens de não fogo está bem abaixo da proposta de entropia apresentada na Seção 6.3, e isso influencia o processo final, uma vez que mostra a ocorrência de muitos falsos alarmes.

A partir dos dados da Tabela 6.12, cujos resultados remetem aos métodos executados na base BL1, é possível observar que o método com os melhores resultados das métricas para imagens de fogo foi a proposta de Phillips, porém assim como ocorreu com a base BL2, o número de Verdadeiros Negativos nas imagens de não fogo, está bem abaixo da proposta de entropia apresentada na seção 6.3.

Além das duas bases BL1 e BL2, também foram testados os referidos métodos na base BL3 compostos por 516 imagens de fogo e 522 imagens de não fogo. Nos resultados na Tabela 6.13 vale ressaltar os resultados apresentados pelos métodos de aprendizagem SVM, DT, RF e Adaboost foram aplicados a amostras da base BL3, conforme descrito na seção 6.5. Nos resultados aplicados na base toda é possível identificar que dentre os métodos de segmentação propostos na literatura, o que obteve melhor score para a F_1score foi o método proposto por Phillips et al. (2002) com 82,9% e desvio padrão de 16%. A proposta que envolve a aplicação de probabilidade e entropia, apresentada por este trabalho, alcançou pontuações superiores a esta medida, alcançando F_1score de 84,1% e desvio padrão de 15,9%. Em relação aos métodos de aprendizagem de máquina SVM, Árvores de Decisão (DT), Florestas Randomicas (RF) e

Tabela 6.11: Resultado da qualidade de segmentação em BL2.

	TP		Acurácia		F_1score		TN	
	μ	σ	μ	σ	μ	σ	μ	σ
Celik RGB1	81,1%	19,8%	80,0%	11,9%	79,1%	15,3%	76,7%	16,9%
Celik RGB2	22,5%	19,8%	59,6%	11,2%	32,5%	24,2%	97,0%	7,5%
Celik Lab	50,5%	28,7%	72,5%	17,0%	60,2%	30,6%	91,5%	12,4%
Celik YCbCr	58,2%	29,8%	76,9%	16,5%	66,9%	28,8%	88,6%	13,1%
Chen RGB	85,0%	15,9%	83,6%	11,1%	83,4%	12,3%	72,6%	21,8%
Chen RGB+HSI	9,7%	14,9%	50,9%	7,6%	14,2%	18,4%	91,0%	12,2%
Horng	62,3%	24,1%	78,5%	13,3%	71,6%	22,0%	86,4%	15,8%
Ko	43,3%	33,0%	62,5%	20,3%	48,4%	32,0%	68,6%	17,6%
Philips	74,5%	21,0%	84,1%	11,0%	74,5%	21,0%	87,8%	20,6%
Rossi	47,6%	36,2%	46,4%	32,7%	45,7%	35,1%	44,2%	28,3%
Santanna	67,2%	24,9%	77,4%	13,6%	72,2%	21,9%	81,0%	16,4%
Vipin	29,3%	20,7%	63,9%	11,3%	41,4%	24,0%	95,6%	9,3%
Prop. Entropia	82,1%	30,1%	89,4%	14,3%	83,3%	27,8%	98,1%	4,0%
Prop. Índice de Cor	78,8%	29,2%	88,0%	15,3%	81,2%	26,7%	96,8%	4,3%

Tabela 6.12: Resultado da qualidade de Segmentação em BL1.

	TP		Acurácia		F_1score		TN	
	μ	σ	μ	σ	μ	σ	μ	σ
Celik RGB	75,7%	20,3%	78,6%	11,7%	76,8%	15,0%	68,1%	15,1%
Celik RGB2	23,6%	18,5%	60,2%	10,2%	34,2%	23,0%	93,7%	8,2%
Celik Lab	56,1%	24,4%	74,4%	15,3%	66,0%	56,1%	84,6%	14,5%
Celik YCbCr	64,8%	23,1%	80,0%	13,4%	74,1%	20,7%	83,6%	13,2%
Chen RGB	84,0%	15,0%	83,9%	10,4%	89,5%	11,5%	65,8%	20,9%
Chen RGB+H	4,0%	0,1%	49,1%	0,1%	6,3%	10,9%	91,0%	11,4%
Horng	63,0%	19,3%	79,1%	10,6%	73,5%	15,9%	82,9%	14,6%
Ko	45,9%	27,9%	64,7%	16,3%	52,7%	26,6%	79,5%	17,1%
Philips	87,5%	11,0%	90,8%	6,1%	90,2%	0,1%	79,0%	21,9%
Rossi	51,9%	26,6%	69,5%	18,7%	60,3%	27,7%	68,6%	15,8%
Santanna	70,2%	19,8%	79,1%	12,1%	75,8%	16,4%	74,6%	15,5%
Vipin	41,1%	18,1%	69,9%	9,6%	55,5%	18,6%	92,5%	9,8%
Prop. Entropia	82,2%	25,6%	88,6%	12,4%	85,0%	20,1%	92,6%	10,1%
Prop. Índice de cor	82,1%	24,3%	88,4%	13,6	84,7%	21,1%	91,8%	10,3%

o Adaboost os melhores scores para F_1score foram apresentados pelas Árvores de Decisão, alcançando 92,1% com desvio padrão de 0,6%.

Conforme se observa nas Tabelas 6.12, 6.11 e 6.13 os métodos de segmentação propostos apresentaram melhores resultados nas bases de teste BL1 e BL3, principalmente o método que une a utilização das regras de entropia e probabilidades. Já em relação aos modelos de aprendizagem de máquina, as árvores de decisão apresentaram melhores resultados apresentando $92,1 \pm 0,6\%$ de F_1score e $92,3 \pm 0,6\%$ de acurácia relativo à qualidade de segmentação. Vale

Tabela 6.13: Resultados obtidos na base BL3.

	TP		Acurácia		F_1score		TN	
	μ	σ	μ	σ	μ	σ	μ	σ
Celik RGB	80,0%	19,4%	79,9%	11,1%	78,4%	16,0%	80,0%	19,4%
Celik RGB2	23,0%	18,7%	60,5%	11,2%	33,3%	23,2%	23,0%	18,7%
Celik Lab	50,8%	27,5%	71,4%	16,8%	60,3%	27,6%	89,0%	16,4%
Celik YCbCr	61,1%	26,5%	77,8%	14,8%	69,5%	25,1%	61,1%	26,5%
Chen RGB	85,5%	17,4%	82,8%	10,3%	82,0%	14,5%	85,5%	17,4%
Chen RGB+H	6,4%	10,8%	49,3%	6,3%	3,8%	13,9%	6,4%	10,8%
Horng	64,2%	23,2%	79,0%	11,9%	72,2%	20,7%	64,2%	23,2%
Ko	44,7%	31,5%	62,4%	18,3%	49,1%	29,4%	44,7%	31,5%
Philips	79,3%	20,9%	85,4%	10,7%	82,9%	16,0%	88,8%	16,1%
Rossi	49,3%	32,8%	48,2%	29,4%	47,3%	30,6%	49,3%	32,8%
Santanna	68,7%	23,3%	77,3%	12,6%	72,5%	20,3%	68,7%	23,3%
Vipin	35,4%	20,3%	67,4%	11,1%	48,4%	23,0%	35,4%	20,3%
Prop. Entropia	81,1%	19,9%	86,8%	9,5%	84,1%	15,9%	81,1%	19,9%
Prop. Índice de cor	80,9%	16,2%	85,4%	8,5%	83,3%	13,7%	80,9%	16,2%
SVM	84,5%	1,5%	87,9%	1,0%	87,7%	1,0%	90,4%	3,3%
DT	90,3%	1,0%	92,3%	0,6%	92,1%	0,6%	92,2%	2,0%
RF	89,9%	0,9%	92,2%	0,5%	91,9%	0,6%	92,5%	2,0%
Adaboost	89,0%	1,1%	91,1%	0,6%	90,8%	0,7%	91,8%	1,5%

destacar que os métodos de aprendizagem de máquina mais robustos configurados com o vetor de características propostos apresentaram melhor desempenho que os métodos baseados em regras ou probabilidades.

6.7 Considerações finais

Neste capítulo foram apresentados os protocolos experimentais e os resultados dos experimentos realizados em relação à localização de fogo. Além da análise dos canais de cor e das criações das tabelas de consulta, três experimentos principais foram executados. O Primeiro experimento, na Seção 6.3, utilizou abordagens probabilísticas combinando canais de cor de diferentes espaços de cor em conjunto com cálculos de entropia para eliminar possíveis falsos alarmes. O segundo experimento, na Seção 6.4, utilizou técnicas de coloração a partir de índices de cor para identificar píxeis de fogo por meio de abordagens probabilísticas. O terceiro experimento, na Seção 6.5, testou 4 modelos de aprendizagem de máquina (SVM, Árvores de Decisão, Random Forest e Adaboost) aplicado a píxeis de imagens de fogo e não fogo. O Próximo Capítulo trata de um método de monitoramento de fogo baseado nas redes neurais convolucionais para detecção e nos métodos probabilísticos para a localização do foco de incêndio.

Capítulo 7

Monitoramento de fogo e fumaça

A criação de métodos eficientes para detecção de fogo e fumaça, assim como o processo de segmentação de píxeis de fogo possuem papel fundamental para a identificação precoce de incêndio e minimização dos danos que poderiam ser causados com este tipo de incidente. Com o objetivo de aliar o processo de detecção e localização, este trabalho propõe um método de detecção de fogo e fumaça, assim como a localização de fogo. Para construir este mecanismo, faz-se necessária a preparação de uma base de imagens que possibilite a avaliação dos métodos de detecção e de segmentação propostos.

O presente capítulo traz a abordagem utilizada para a concepção de um método que utiliza as Redes Neurais Convolucionais como auxiliar no processo de detecção e métodos de segmentação de imagens de fogo para realizar a localização dos focos de incêndios nas imagens. Para construir este modelo é preciso avaliar os diversos modelos testados anteriormente para detecção de fogo e fumaça, assim como os métodos que realizam a segmentação dos píxeis de fogo. Para realizar esta análise, a Subseção 7.1 seguinte aborda o método utilizado para selecionar os modelos adequados para detecção de fogo e fumaça e localização de fogo. A Subseção 7.2 traz os resultados obtidos, assim como apresenta uma possível configuração para um sistema de monitoramento.

7.1 Método

As imagens da base de localização BL3 foram transformadas em *patches* para serem utilizadas como entrada nas Redes Neurais Convolucionais (RNC). Deste método, das 1038 imagens foram extraídos 8.991 *patches*, sendo 5.469 *patches* negativos e 3.522 *patches* contendo fogo. A partir desta transformação da base BL3 fica possível utilizá-la para ser testada em Redes Neurais Convolucionais. Neste sentido, esta base será utilizada para medir a performance em termos de taxas de detecção das redes neurais estudadas. Além desta análise, também é levado em consideração o tempo de processamento destas redes que foram apresentadas na Tabela 5.10 do Capítulo 5. Para os algoritmos de segmentação, a avaliação destes métodos levará

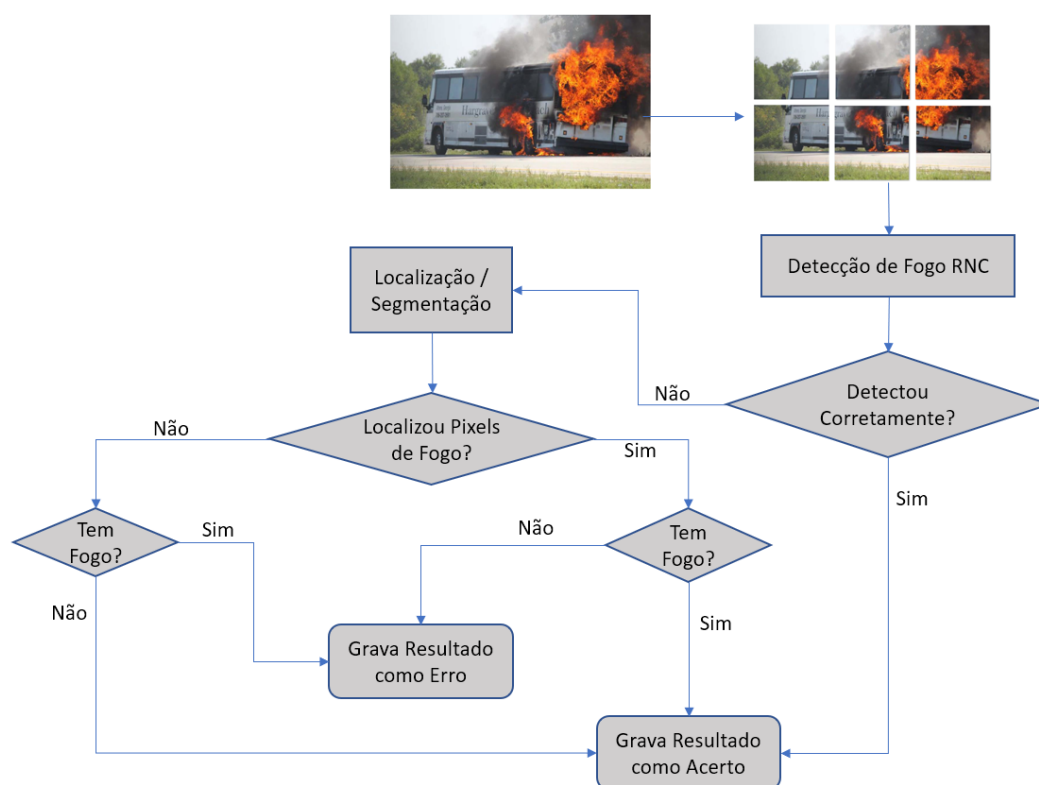


Figura 7.1: Método utilizado para teste em algoritmos de detecção e localização de fogo, com o objetivo de identificar melhor estratégia para criação de um mecanismo de monitoramento de fogo.

em consideração as taxas de $F_1 score$ e Acurácia, pois estas métricas indicam a qualidade de segmentação dos métodos estudados. Os resultados considerados para avaliação da segmentação estão presentes na Tabela 6.13.

Outro experimento, consiste em verificar se a segmentação poderia auxiliar no processo de detecção. Neste sentido, a Figura 7.1 mostra o fluxo metodológico adotado. O método utilizado para a avaliação da otimização da qualidade de detecção utilizando segmentação, foi dividido em 4 etapas principais: 1) Separar as imagens da base BL3 em *patches* rotulados em fogo e não fogo; 2) Classificar os *patches* utilizando RNC e caso houvesse erro de classificação pela rede, armazenar as imagens que apresentaram erro; 3) Executar o processo de segmentação a fim de localizar possíveis foco de incêndio ou não que foram ignorados pela rede neural. 4) Contabilizar os acertos e erros de detecção a partir da utilização de métodos de segmentação complementares à RNCs.

Além do processo avaliativo exposto, outro experimento consistiu em executar o algoritmo de localização de píxeis de fogo em todos os *patches* (antes da RNC), e avaliar como se deu a qualidade de detecção de fogo para o algoritmo de segmentação com os melhores resultados e comparar o resultado obtido nas métricas com as obtidas pelas RNC.

Para o algoritmo de localização, como critério de classificação da detecção de fogo que a imagem necessita ter pelo menos 30 píxeis identificados corretamente como fogo. Após a avaliação dos resultados obtidos a partir do método proposto e ilustrado pela Figura 7.1 pretende-se compor os métodos de detecção de fogo e fumaça e localização de fogo.

7.2 Resultados obtidos

Aplicando o método experimental proposto e exposto anteriormente na Seção 7.1, a base de imagens *BL3* foi transformada em 3.521 patches de fogo e 5.469 píxeis de não fogo, totalizando 8.990 patches. Para as Redes Neurais Convolucionais (RNCs), foram selecionadas dois modelos arquiteturas a Arquitetura *FSDN₈* proposta na seção 5.2.1 e a rede *Xception* por terem obtido os melhores desempenho entre as redes treinadas do zero e as redes pré-treinadas, respectivamente, na maior parte das métricas, com destaque na métrica *F₁score* na base de teste da BD1 que alcançou *F₁score* de 97,5% na *Xception* e 95,6% na rede *FSDN₈*. Em relação ao processo de segmentação, ao realizar a análise de desempenho dos métodos estudados e presentes na Tabela 6.13 constatou-se que o método dentre os baseados em regras, que possui o melhor índice de qualidade de segmentação foi a proposta de entropia, presente na Seção 6.3 do Capítulo 6 que engloba abordagens probabilísticas e Entropia. Outra possibilidade seria a escolha árvores de decisão, quando se opta por utilizar abordagens de aprendizagem de máquina, porém, para isso, deve-se levar em consideração que os testes foram realizados com amostras de píxeis das imagens.

Aplicando-se as redes arquiteturas *FSDN₈* e a *Xception*, e posteriormente o método de entropia aos *patches* da base *BL3*, foi possível criar a Tabela 7.1. Esta tabela mostra o resultado aplicado aos patches *BL3*, indicando o número de patches classificados de forma incorreta por cada arquitetura (Erro) e a taxa de acerto nos patches para as arquiteturas RNC (% Acerto RNC), nas colunas “Corrigiu (Seg)” estão os patches detectados como píxeis de fogo pelo processo de segmentação para cada classe e como ficou a taxa de acerto após a utilização do algoritmo de segmentação (% Acerto RNC+Seg).

Tabela 7.1: Taxas de Acerto antes e após processo de segmentação nos patches da base *BL3* para as arquiteturas *Xception* e Arquitetura Proposta.

Classe	<i>FSDN₈</i>				<i>Xception</i>			
	Erro	% Acerto RNC	Corrigiu (Seg)	% Acerto RNC+Seg	Erro	% Acerto RNC	Corrigiu (Seg)	% Acerto RNC+Seg
Fogo	458	87,0%	185	92,2%	212	94,0%	114	97,2%
Não Fogo	237	95,7%	43	96,5%	160	97,1%	48	98,0%

Analisando a Tabela 7.1 nota-se uma melhora na taxa de detecção final das arquiteturas, principalmente para os patches de fogo quando da aplicação do algoritmo de segmentação partindo de 87,6% para 92,2% de acerto na classificação dos patches para a Arquitetura proposta

e de 94% para 97,2% de acerto para a arquitetura Xception. Em relação às imagens de não fogo, a mudança não foi tão significativa, variando +0,8% para a arquitetura proposta e +0,9% para a arquitetura Xception. Porém, em uma aplicação no mundo real, não se tem conhecimento da classe à qual a imagem de entrada pertence. Neste sentido, aplicou-se o algoritmo de segmentação na base BL3, calculando a taxa de detecção das imagens de fogo, assim como nas imagens de não fogo.

Sendo assim, ao aplicar o algoritmo de localização de fogo proposto pela seção 6.3 diretamente às imagens da base BL3, composta de 520 imagens de não fogo, e ajustando a taxa de probabilidade para 35% para que houvesse maior identificação das regiões de fogo. Dessa forma, apenas 44,7% das imagens de não fogo foram corretamente classificadas como não fogo. Os resultados obtidos mostram que somente o processo de segmentação indicaria um alto índice de falsos alarmes; porém, quando se aplica a mesma técnica a imagens de fogo, a taxa de detecção sobe substancialmente, atingindo 97,4% de acerto. Cabe ressaltar que o critério para definir se houve acerto ou não na detecção, o algoritmo deveria identificar ao menos 30 píxeis com probabilidade de serem píxeis de fogo. Assim, tendo como base os resultados obtidos, principalmente nas imagens de não fogo, optou-se por utilizar o processo de segmentação como um processo complementar para localização de fogo após a execução do processo de detecção pela Rede Neural Convolutiva.

Os resultados obtidos com as diferentes arquiteturas de Redes Neurais Convolutivas - RNC - binárias, obtiveram melhores taxas de acerto que as redes multi-classes. Dessa maneira, a proposta de um mecanismo de monitoramento de incêndios utiliza RNCs binárias para detecção de Fogo e Fumaça, conforme pode ser visualizado no fluxograma na Figura 7.2. Este mecanismo pode ser implementado em um sistema de monitoramento e é composto por duas redes neurais binárias, sendo uma para detecção de fogo e outra para detecção de fumaça, e um algoritmo de localização de fogo, que indica o local do foco do incêndio.

O processo de funcionamento se dá a partir da entrada de imagens (podem ser frames de vídeos) que alimentam as duas redes binárias e indicam se encontraram Fogo ou Fumaça. Caso seja detectada fumaça, apenas um alerta é emitido. Quando a rede binária de fogo detecta a presença de incêndio, então o algoritmo de segmentação é executado e indica o local do incidente na imagem.

Em um sistema de tempo real, a ocorrência de fumaça pode indicar o início de um foco de incêndio, e, ao ser detectada fumaça no sistema, haveria a sinalização de status de alerta. Os resultados obtidos nas RNCs binárias de Fumaça apontaram para a base de teste taxas de detecção na base BD2 de 93,9% para rede Xception e 90,2% para a arquitetura $FSDN_8$ proposta na Seção 5.2.1, em caso de limitações de hardware. Além do processamento de ocorrência de fumaça, paralelamente o processo de detecção de fogo pode ser realizado por meio das redes: Resnet, que obteve na base BD2 98,14% de taxa de detecção, ou em caso de limitações de hardware, a $FSDN_8$, com 95,5% de acerto no processo de detecção. Após os processos de detecção, caso haja a identificação de fogo pela RNC binária, então o algoritmo de segmentação baseado nas

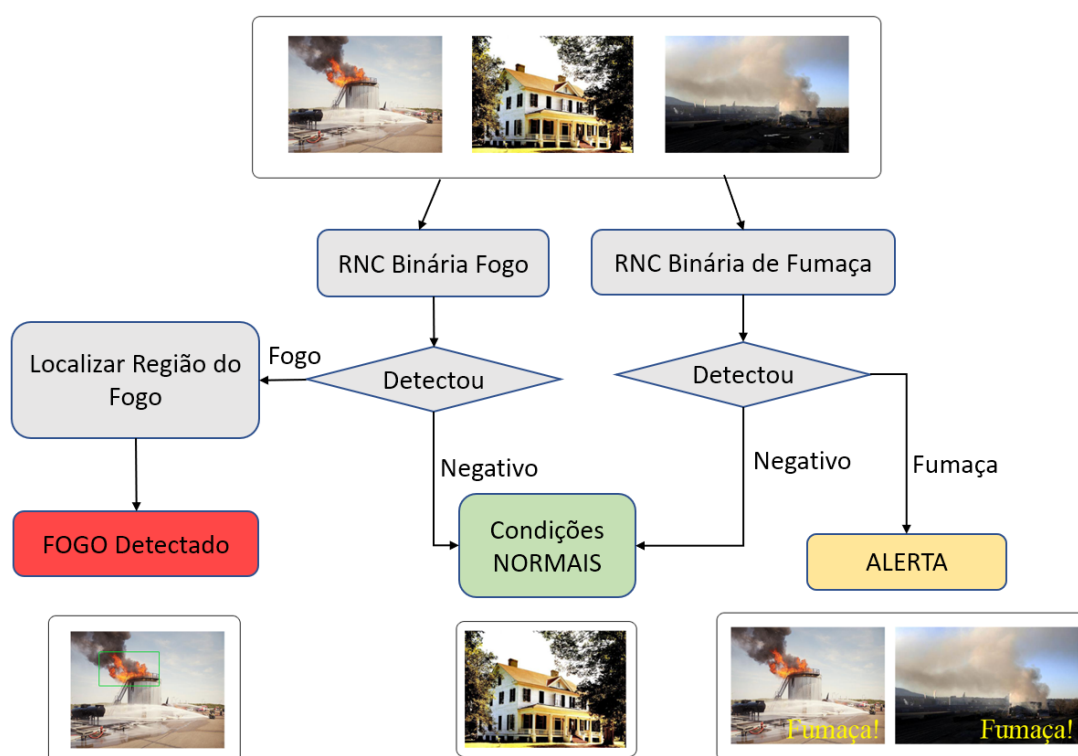


Figura 7.2: Modelo de Monitoramento Proposto para detecção e localização de fogo.

probabilidades e regras de entropia é executado a fim de localizar o foco do incêndio e informar qual a localização do mesmo na imagem.

Quando aplicado o algoritmo proposto na base de patches da BL3, e realizando o mapeamento dos patches para as imagens originais, os resultados obtidos para taxa de detecção e localização foram de 93,9% (973 acertos de 1036) para a rede Xception e 91,8% (951 acertos de 1036) para a rede $FSDN_8$, conforme mostra a Tabela 5.12. Após a detecção, o algoritmo de segmentação realiza a localização dos píxeis de fogo indicando o local do incêndio na imagem processada. Um exemplo de saída do mecanismo de monitoramento de fogo e fumaça é exemplificado na Figura 7.3.

7.3 Considerações finais

Neste capítulo foi apresentado um método de monitoramento de fogo baseado nas redes neurais convolucionais para detecção e nos métodos probabilísticos para a localização do foco de incêndio. Neste método, para maior eficácia, é proposta a utilização das redes neurais convolucionais binárias para detecção de fogo e outra RNC binária para detecção de fumaça. Já como mecanismo de localização dos píxeis de fogo, é proposto utilizar o método probabilístico com as regras de entropia, conforme demonstrado. O próximo capítulo traz as conclusões deste trabalho, assim como apresenta as possibilidades de trabalhos futuros.

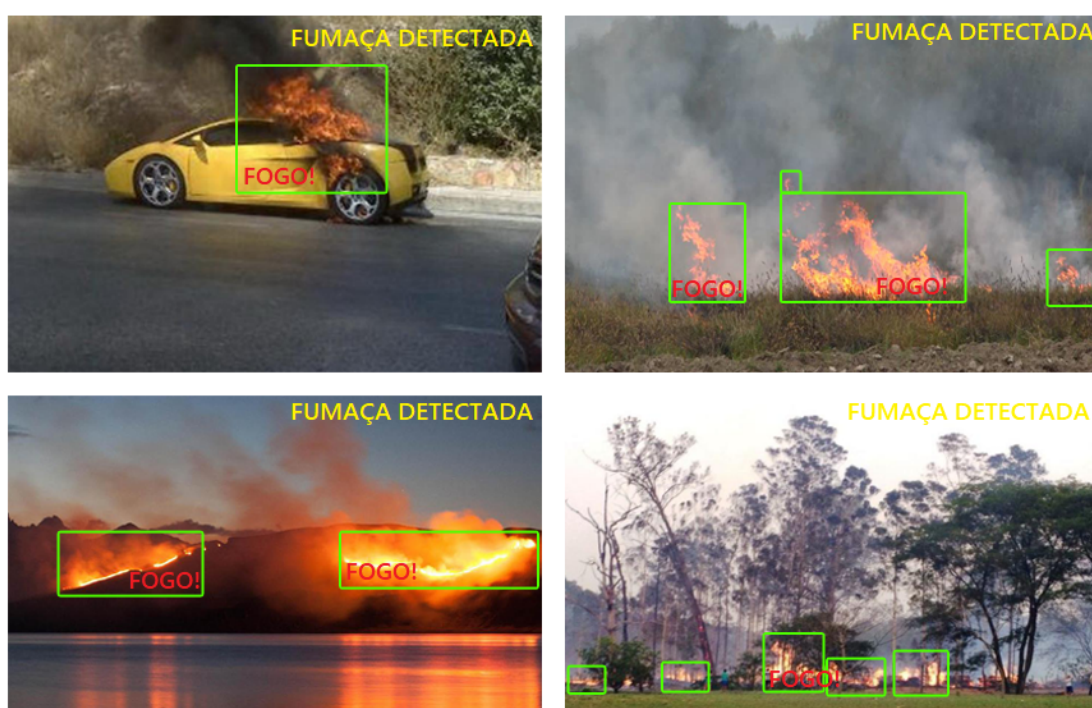


Figura 7.3: Exemplos de saída do Método de Monitoramento de Fogo e Fumaça proposto para detecção e localização de fogo.

Capítulo 8

Conclusões e Trabalhos futuros

O presente trabalho apresentou os conceitos e o estado da arte em relação à detecção automática de fogo e fumaça por meio de técnicas pertinentes ao Processamento Digital de Imagens e abordagens que englobam conceitos de Aprendizagem de Máquina. Nesse sentido, três bases de imagens robustas foram criadas, duas voltadas para testes com detecção de fogo e fumaça, *BD1* e *BD2*, e outra voltada para a localização/segmentação de fogo em imagens a *BL3*.

A construção da base de imagens para detecção de fogo e fumaça consistiu na coleta de mais de 110 mil *patches* de imagens (*BD1* e *BD2*). Este processo foi relevante pois permitiu o aprimoramento do treino dos modelos de aprendizagem de máquina baseados em Redes Neurais Convolucionais - RNCs. Além disso, esta base possibilitou testes de transferência de aprendizagem com a base de imagens proposta e disponibilizada por Frizzi et al. (2016) (*BD3*). Os testes de transferência de aprendizagem evidenciaram as diferenças entre as bases *BD1* e *BD3*. Utilizando como base a arquitetura para detecção de fogo proposta por este trabalho a *FSDN₈*, quando esta rede foi treinada pela base de treino da *BD3*, a métrica de acurácia na base de validação de *BD3* foi de 97,9%, porém quando a rede treinada é submetida a testes em bases diferentes, bases *BD1* e *BD2*, o desempenho da rede caiu de forma significativa, obtendo acurácia de 87% e 82%, respectivamente. Em contrapartida, ao utilizar a base de treino de *BD1*, os resultados para acurácia nas bases de testes de *BD1* e *BD2* atingem 93,7% e 92,6% respectivamente. Os resultados evidenciaram que a diversidade da base é fundamental para o treinamento de um modelo de detecção de fogo que seja mais eficiente.

Ao realizar testes com variações arquiteturais nas RNCs constatou-se que para o problema de detecção de fogo o tamanho das camadas totalmente conectadas influencia diretamente no desempenho da rede, porém deve-se balancear o custo benefício em relação ao tamanho do modelo e tempo de processamento de acordo com as limitações de hardware. Além disso, os resultados recentes no concurso do ILSVRV (Russakovsky et al., 2015) mostraram que utilizar camadas compostas (Inception, Xception) e blocos residuais (Resnet) auxiliam na tarefa de detecção. Optou-se então por criar um modelo que abarcasse a utilização de camadas do tipo Inception, e blocos residuais para uma rede mais compacta do que as redes presentes na literatura. De fato, a utilização desses recursos na arquitetura RNC mostrou melhora no

desempenho na base de validação, apresentando ganhos de até +10,1% para acurácia e +10,2% para $F_1 score$ ($FSDN_2 \times FSDN_8$), assim como no erro absoluto com diminuição do índice e -9,9% ($FSDN_2 \times FSDN_8$) em comparação com a arquitetura que apresentou os melhores resultados sem a utilização de blocos residuais ou camadas Inception. Os testes de desempenho também mostraram que a rede proposta $FSDN_8$ foi a mais eficiente em relação ao custo benefício, pois apesar de ter desempenho inferior à rede Xception, a taxa de processamento da rede ficou em média com 410 imagens por segundo (cerca de 3x mais rápida que a rede Xception) e seus pesos ocupando 17MB de espaço em disco (10x menor que a Xception), com perda de qualidade de detecção de cerca de 3% em média em relação às métricas F1-Score e Acurácia comparada ainda à rede Xception.

Além dos testes de transferência de aprendizagem e da criação de uma nova arquitetura de Rede Neural Convolutacional foram avaliadas redes com pesos pré-treinados na base Imagenet, a saber: Inception, Xception, Resnet, VGG-16 e MobileNet. Adicionalmente, RNCs criadas especificamente para detecção de fogo ou fumaça também foram avaliadas (Frizzi et al., 2016; Yin et al., 2017; Zhang et al., 2016; Tao and Wang, 2016), porém tais redes foram treinadas do zero. A rede proposta $FSDN_8$ obteve melhor desempenho comparada às arquiteturas treinadas do zero, e também apresentou melhor taxa de detecção de fogo, obtendo 91,8% na BL3, comparada a outras 17 técnicas de detecção de fogo.

Outra constatação possível foi a de que o desempenho das Redes Binárias apresentou melhores resultados comparados às redes multiclases para o problema de detecção de fogo e fumaça. Utilizando como exemplo os melhores resultados de Acurácia, $F_1 score$ e erro absoluto médio (MAE), aplicados na base de testes BD2, nas redes multiclases os resultados foram de 89,6%, 89,6% e 7,5% respectivamente para a rede Xception. Já realizando as detecção de fogo e fumaça separadamente, por meio de redes binárias, para a detecção de fogo, os resultados para Acurácia, $F_1 score$ e MAE foram de 98,4%, 98,3% e 1,9% (rede Xception) e para detecção de fumaça os resultados foram de 95,6% para acurácia, 95,4% para $F_1 score$ e 5,7% para MAE também na rede Xception. Em relação às taxas de detecção, temos que a melhor rede multiclasse desempenhou 93,7% para detecção de fogo na rede Xception, e 83,1% para detecção de fumaça, enquanto que nas redes binárias, as taxas de detecção ficaram em 98,1% para detecção de fogo e 94,3% para detecção de fumaça. Os resultados obtidos mostram que utilizar RNCs binárias trazem uma melhor taxa de detecção, o que indica menor possibilidade de falsos alarmes, principalmente no que diz respeito à detecção de fogo. Além disso, as redes binárias para detecção de fumaça podem ser utilizadas como elemento de alerta para uma possível ocorrência de fogo.

Após o processo de criação e testes de modelos voltados para a detecção de fogo e fumaça foram realizados estudos em relação à localização de fogo em imagens, presentes nas Seções 6.3, 6.4 e 6.5. Para conduzir estes experimentos a base BL3 (1036 imagens) foi idealizada com a solução de segmentação ideal para os píxeis de fogo. Além disso, a base BL3 foi separada em diferentes contextos. Os estudos conduzidos utilizaram abordagens de

aprendizagem de máquina, sendo que os experimentos contidos nas Seções 6.3 e 6.4 utilizaram abordagens probabilísticas e o terceiro focou em modelos de aprendizagem de máquina mais robustos, como: SVM, Árvores de Decisão (DT), Florestas Randômicas (RT) e Adaboost. Os resultados comparativos na base *BL3*, apresentados na Seção 6.6 mostram que em relação à qualidade de segmentação a proposta de segmentação utilizando probabilidades e entropia obteve melhores pontuações para Acurácia (86,8%) e F_1score (84,1%). Em relação à aplicação dos modelos de aprendizagem de máquina SVM, Árvores de decisão (DT), Florestas Aleatórias (RF) e Adaboost foi proposto um novo modelo de entrada de dados composto de 4 espaços de cor (RGB, YCbCr, HSI e CIELab); os resultados nas amostras da base de teste apresentaram taxas superiores para a maioria das métricas, com F_1score para o modelo Árvores de decisão (DT) em $92,8\% \pm 0,6\%$ e acurácia em $92,3\% \pm 0,6\%$.

A partir dos resultados obtidos para detecção e segmentação de fogo, uma proposta de um método de monitoramento de fogo foi proposta e este consiste em utilizar duas RNCs binárias, uma para detecção de fogo e outra para detecção de fumaça. Além disso, a proposta consiste em, caso haja identificação de fogo, então o método de segmentação deve ser acionado com o intuito de localizar o foco de incêndio. Os testes na base de localização *BL3* mostraram que a taxa de detecção de fogo ficou em 93,9% para a rede *xception* e 91,8% para a arquitetura *FSDN₈* proposta. Em relação à qualidade de segmentação, os resultados foram de 86,8% de Acurácia e 84,1% de F_1score ; isso significa que em 86,8% dos píxeis classificados foram classificados de forma correta, o que indica uma boa relação para a localização do fogo nas imagens onde houve detecção de fogo. Estes resultados evidenciam a eficiência do método proposto para monitoramento de fogo e fumaça.

Como trabalhos futuros, os métodos de detecção e localização de fogo e fumaça apresentados ainda necessitam ser melhorados. Apesar dos avanços não há a apresentação de uma solução definitiva para o reconhecimento e detecção de fogo e fumaça. Neste sentido, faz-se necessário aprimorar e adicionar outras técnicas que poderiam beneficiar o processo. Sugere-se a investigação de teorias recentes de aprendizagem de máquina, como as Redes GANs- Generative Adversarial Networks Goodfellow et al. (2014), que consistem em duas RNCs competindo entre si, sendo que uma rede gera imagens para serem usadas como entrada na outra rede, e por *backpropagation* gerar imagens realistas que serão interpretadas pela outra rede como uma imagem real ou não. Dessa forma, a utilização de GANs poderia ser utilizada para aumentar o número de amostras de imagens de fogo e fumaça. Consequentemente, há a possibilidade de aprimorar os métodos existentes.

Em relação à base de imagens proposta, outras estratégias de detecção podem ser exploradas. Dentre elas está a identificação de cenários que podem aprimorar o sistema de monitoramento de fogo, o que implica na criação de modelos de detecção para contextos específicos como detecção de fogo em Rodovias ou Cidades. Outros experimentos que podem ser realizados dizem respeito à utilização de pré-segmentação dos frames de vídeos, extraindo o

fundo que represente elementos estáticos na imagem e após este processo, realizar o treino na RNC utilizando os pesos pré-treinados com a base *BD1* proposta.

Referências Bibliográficas

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281.
- Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-Based Learning Algorithms. *Machine Learning*, 6(1):37–66.
- Aleodin, D. (2011). Luz por Isaac Newton.
- Alkhatib, A. A. A. (2014). A Review on Forest Fire Detection Techniques. *International Journal of Distributed Sensor Networks*, 2014(4):1–12.
- Angayarkkani, K. and Radhakrishnan, N. (2010). An Intelligent System For Effective Forest Fire Detection Using Spatial Data. *arXiv:1002.2199*, 7(1):202–208.
- ASTM (2015). Standard Practice for Calculating Yellowness and Whiteness Indices from Instrumentally Measured Color Coordinates. Technical Report C, ASTM International, West Conshohocken.
- Barreto, J. M. (2002). *Indrodução às Redes Neurais Artificiais*. Departamento de Informática e estatística - UFSC, Florianópolis - SC.
- Batista, G. E. A. P. A. (2003). *Pré-processamento de Dados em Aprendizado de Máquina Supervisionado*. PhD thesis, Universidade de São Paulo.
- Bauer, E., Kohavi, R., Chan, P., Stolfo, S., and Wolpert, D. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36(August):105–139.
- Bengio, Y. (2012). Deep Learning of Representations for Unsupervised and Transfer Learning. In JMLR, editor, *JMLR: Workshop and Conference Proceedings*, volume 27, pages 17–37. Proceedings of Machine Learning Research.
- Breiman, L. (2001). Random Forests. Technical report, University of California, Berkeley, CA.

- Brushlinsky, N. N., Ahrens, M., Sokolov, S. V., and Wagner, P. (2015). Center of Fire Statistic. Technical report, International Association of Fire and Rescue Services.
- Brushlinsky, N. N., Ahrens, M., Sokolov, S. V., and Wagner, P. (2018). World Fire Statistics. Technical report, International Association of Fire And Rescue Services.
- Çelik, T. (2010). Fast and efficient method for fire detection using image processing. *ETRI Journal*, 32(6):881–890.
- Çelik, T. and Demirel, H. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44(2):147–158.
- Çelik, T., Demirel, H., and Ozkaramanli, H. (2006). Automatic fire detection in video sequences. In *Proc. of the 14th European Signal Processing Conference (EUSIPCO)*.
- Çelik, T., Özkaramanl, H., and Demirel, H. (2007). Fire and smoke detection without sensors. In *15th European Signal Processing Conference*, pages 1794–1798.
- Chen, J., He, Y., and Wang, J. (2010). Multi-feature fusion based fast video flame detection. *Building and Environment*, 45(5):1113–1122.
- Chen, T.-H., Wu, P.-H., and Chiou, Y.-C. (2004). An early fire-detection method based on image processing. *International Conference on Image Processing*, 3:1707–1710.
- Chino, D. Y., Avalhais, L. P. S., Rodrigues Jr, J. F., and Traina, A. J. M. (2015). BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. In *Sibgrapi - Conference on Graphics, Patterns and Images*, pages 95–102.
- Cholet, F. (2015). Keras: The Python Deep Learning library.
- Chollet, F. (2016). Xception: Deep Learning with Separable Convolutions. *arXiv preprint arXiv:1610.02357*, pages 1–14.
- Collumeau, J.-F. (2011). FIRE SCENE SEGMENTATIONS FOR FOREST FIRE CHARACTERIZATION : A COMPARATIVE STUDY. *Image (Rochester, N.Y.)*, pages 3034–3037.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.
- Doi, R. (2012). Simple luminosity normalization of greenness, yellowness and redness/greenness for comparison of leaf spectral profiles in multi-temporally acquired remote sensing images. *Journal of Biosciences*, 37(4):723–730.

- Dukuzumuremyi, J. P., Zou, B., and Hanyurwimfura, D. (2014a). A Novel Algorithm for Fire / Smoke Detection based on Computer- Vision. *International Journal of Hybrid Information Technology*, 7(3):143–154.
- Dukuzumuremyi, J. P., Zou, B., Mukamakuza, C. P., Hanyurwimfura, D., and Masabo, E. (2014b). Discrete Cosine Coefficients as Images features for Fire Detection based on Computer Vision. *Journal of Computers*, 9(2):295–300.
- Eustáquio, J., Queiroz, R. D., and Gomes, H. M. (2001). Introdução ao Processamento Digital de Imagens Considerações Iniciais. *RITA*, VIII(8):1–31.
- Evarts, B. (2018). Fire Loss In The United State During 2017. Technical Report October, National Fire Protection Association.
- Facon, J. (2005). *Processamento e Análise de Imagens*. Pontifícia Universidade Católica do Paraná, Curitiba.
- Fawcett, T. (2006). An introduction to ROC analysis. *2Pattern Recognition Letters*, 27:861–874.
- Foggia, P., Saggese, A., and Vento, M. (2015). Real-time Fire Detection for Video Surveillance Applications using a Combination of Experts based on Color, Shape and Motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 8215(c):1545–1556.
- Ford, A. and Roberts, A. (1998). Colour space conversions. Technical report, Westminster University, London.
- Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., and Fnaiech, F. (2016). Convolutional neural network for video fire and smoke detection. *IECON Proceedings (Industrial Electronics Conference)*, pages 877–882.
- Gasparini, F. and Schettini, R. (2005). Automatic redevye removal for smart enhancement of photos of unknown origin. In *Proc.of the 8th International Conference on Visual Information Systems*, volume 3736, pages 226–233.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., and Warde-farley, D. (2014). Generative Adversarial Nets. *arXiv*, 1:1–9.
- Hafiane, A., Chabrier, S., Rosenberger, C., and Laurent, H. (2007). A New Supervised Evaluation Criterion for Region Based Segmentation Methods. *Advanced Concepts for Intelligent Vision Systems*, 4678:439–448.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18.

- He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385v1*, 7(3):171–180.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015b). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:1026–1034.
- Held, A. (2002). Model-Based Correction of Red-Eye Defects. In *Color Imaging Conference*, pages 223–228, Scottsdale, Arizona, USA.
- Helmenstine, A. M. (2016). What is the State of Matter of Fire or Flame? Is it a Liquid, Solid, or Gas?
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, pages 1–18.
- Horng, W.-b., Peng, J.-w., and Chen, C.-y. (2005). A New Image-Based Real-Time Flame Detection. In *IEEE Networking, Sensing and Control*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*.
- INPE (2018). INPE - Instituto Nacional de Pesquisas Espaciais - Portal do Monitoramento de Queimadas e Incêndios.
- INPE (2019). INPE - Instituto Nacional de Pesquisas Espaciais - Portal do Monitoramento de Queimadas e Incêndios.
- John, G. and Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, page 8, San Mateo. Morgan Kaufmann.
- Ko, B., Cheong, K.-H., and Nam, J.-Y. (2010). Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks. *Fire Safety Journal*, 45(4):262–270.
- Ko, B. C., Cheong, K.-H., and Nam, J.-Y. (2009). Fire detection based on vision sensor and support vector machines. *Fire Safety Journal*, 44(3):322–329.
- Kopriva, I., Ju, W., Zhang, B., Shi, F., Xiang, D., Yu, K., Wang, X., Bagci, U., and Chen, X. (2017). Single-Channel Sparse Non-Negative Blind Source Separation Method for Automatic 3-D Delineation of Lung Tumor in PET Images. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1656–1666.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9.
- LeCun, Y. and Bengio, Y. (1995). Convolution Networks for Images, Speech, and Time-Series. Technical Report 1, MIT.
- Maksymiv, O., Rak, T., and Peleshko, D. (2017). Real-time Fire Detection Method Combining AdaBoost , LBP and Convolutional Neural Network in Video Sequence. In *CADSM*, pages 351–353, Polyana-Svalyava, Ukraine. IEEE.
- Marbach, G., Loepfe, M., and Brupbacher, T. (2006). An image processing technique for fire detection in video images. *Fire Safety Journal*, 41(4):285–289.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to\nevaluating segmentation algorithms and measuring ecological statistics. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2(July):416–423.
- Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., and Baik, S. W. (2018). Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access*, 6:18174–18183.
- Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective*. MIT Press, Cambridge, Massachusetts, 1 edition.
- Newton, I. (1714). *Opticks: treatise of the reflexions, refraxions, inflexions end colours of light*. Royal Soci.
- NFPA (2014). National Fire Protection Association.
- Oliveira, M. D. E. (2005). *Estudo sobre incêndios de progresso rápido*. Especialização em planejamento e gestão em defesa civil, Universidade Federal de Santa Catarina.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.
- Ozkan, C., Berberoglu, S., and Donmez, C. (2008). Effectiveness of Boosting Algorithms in Forest Fire Classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII(Part B7):625–630.
- Pathare, P. B., Opara, U. L., and Al-Said, F. A. J. (2013). Colour Measurement and Analysis in Fresh and Processed Foods: A Review. *Food and Bioprocess Technology*, 6(1):36–60.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher,

- M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Phillips, W., Shah, M., and Lobo, N. V. (2002). Flame recognition in video. *Pattern Recognition Letters*, 23:319–327.
- Ricauda, D., Barge, P., Comba, L., Gay, P., Occelli, A., and Tortia, C. (2015). Computer Vision for Laboratory Quality Control on Frozen Fruit. *Chemical Engineering Transactions*, 44:175–180.
- Robinson, D. A. (1998). Method and System for Automatically Monitoring the Colors of an Object at a Vision Station.
- Rocha, R. H. S. (2015). *Reconhecimento de Objetos por Redes Neurais Convolutivas*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Rossi, L., Akhloufi, M., and Tison, Y. (2009). Dynamic fire 3D modeling using a real-time stereovision system. *Journal of Communication and Computer*, 6(10):54–61.
- Rossi, L., Akhloufi, M., and Tison, Y. (2011). On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. *Fire Safety Journal*, 46(1-2):9–20.
- Rudz, S., Chetehouna, K., Hafiane, A., Laurent, H., and Séro-Guillaume, O. (2013). Investigation of a novel image segmentation method dedicated to forest fire applications. *Measurement Science and Technology*, 24(7):075403.
- Rudz, S., Chetehouna, K., Hafiane, A., and Sero-guillaume, O. (2009). On the Evaluation of Segmentation Methods for Wildland Fire. In *ACIVS*, pages 12–23.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Santana, P., Gomes, P., and Barata, J. (2012). A vision-based system for early fire detection. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pages 739–744.
- SANTOS, M. A. d. S. (2018). *A Dispersão da Luz Branca*.
- Saraiva, E. A. (2011). *Detecção de incêndios florestais e queimadas com radar meteorológico*. Tese de doutorado, Universidade Federal do Paraná.
- Schölkopf, B., Smola, A., Müller, K., Burges, C., and Vapnik, V. (1998). Support vector methods in learning and feature extraction. In *Ninth Australian Conference on Neural Networks*, page 7.

- Sezgin, M. and Sankur, B. (2001). Selection of thresholding methods for non-destructive testing applications. *Image Processing*, 2001:764–767.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(July 1928):379–423.
- Sharma, J., Ole-christoffer, G., Goodwin, M., and Fidje, J. T. (2017). Deep Convolutional Neural Networks for Fire Detection in Images. *Engineering Applications of Neural Networks*, 744:183–193.
- Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint*, pages 1–10.
- Smith, L. N. (2015). Cyclical Learning Rates for Training Neural Networks. *arXiv*, 1506.01186(April):10.
- Smolka, B., Czubin, K., Hardeberg, J. Y., Plataniotis, K. N., Szczepanski, M., and Wojciechowski, K. (2003). Towards automatic redeye effect removal. *Pattern Recognition Letters*, 24(11):1767–1785.
- SOUTO, R. P. (2000). *Segmentação de imagem multispectral utilizando-se o atributo matiz*. PhD thesis, Instituto Nacional de Pesquisas Espaciais.
- Souza, B. M. N. and Facon, J. (2016). A Fire Color Mapping – based Segmentation Fire pixel segmentation approach. In *IEEE/ACS International Conference on Computer Systemas and Applications - AICCSA*, page 8, Agadir, Morocco. IEEE Computer Society.
- Souza, B. M. N., Facon, J., and Menotti, D. (2017). Colorness Index Strategy for Pixel Fire Segmentation. In *International Joint Conference on Neural Networks (IJCNN)*, page 7, Anchorage, AK, USA. IEEE.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:1–9.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., Hill, C., and Arbor, A. (2014). Going Deeper with Convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.
- Tao, C. and Wang, P. (2016). Smoke Detection Based on Deep Convolutional Neural Networks. In *Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, 2016 International Conference on, page 4. IEEE.

- Toreyin, B. U., Dedeoglu, Y., and Cetin, a. E. (2005). Flame detection in video using hidden Markov models. *Image Processing 2005 ICIP 2005 IEEE International Conference on*, 2:1230–1233.
- Toulouse, T., Rossi, L., Akhloufi, M., Celik, T., and Maldague, X. (2015). Benchmarking of wildland fire colour segmentation algorithms. *Iet Image Processing*, 9(12):1064–1072.
- Toulouse, T., Rossi, L., Campana, A., Celik, T., and Akhloufi, M. A. (2017). Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Safety Journal*, 92(June):188–194.
- Tsai, D.-Y., Lee, Y., and Matsuyama, E. (2008). Information Entropy Measure for Evaluation of Image Quality. *Journal of Digital Imaging*, 21(3):338–347.
- Vera, C., Nepstad, D., Mendonça, M., Motta, R. S., Alencar, A., Gomes, J. C., and Ortiz, R. A. (2002). O Prejuízo Oculto do Fogo : Custos Econômicos das Queimadas e Incêndios Florestais na Amazônia. Technical report, IPAM, IPEA e WHRC.
- Vipin, V. (2012). Image Processing Based Forest Fire Detection. *International Journal of Emerging Technology and Advanced Engineering*, 2(2):87–95.
- Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained K-means Clustering with Background Knowledge. *International Conference on Machine Learning*, pages 577–584.
- Wang, Z., Wang, Z., and Zhang, H. (2017). A Novel Fire Detection Approach Based on CNN-SVM Using Tensorflow. In *International Conference on Intelligent Computing*, volume 10363, pages 682–693.
- Wong, A. K. K. and Fong, N. K. (2014). Experimental study of video fire detection and its applications. *Procedia Engineering*, 71:316–327.
- Wu, Y., Ianakiev, K., and Govindaraju, V. (2002). Improved k-nearest neighbor classification. *Pattern Recognition*, 35(10):2311–2318.
- Yin, Z., Wan, B., Yuan, F., Xia, X. U. E., and Shi, J. (2017). A Deep Normalization and Convolutional Neural Network for Image Smoke Detection. *IEEE Access*, 5:18429–18438.
- Yuan, F., Shi, J., Xia, X., Fang, Y., Fang, Z., and Mei, T. (2016). High-order local ternary patterns with locality preserving projection for smoke detection and image classification. *Information Sciences*, 372:225–240.
- Zhang, Q., Xu, J., Xu, L., and Guo, H. (2016). Deep Convolutional Neural Networks for Forest Fire Detection. In *Proceedings of the 2016 International Forum on Management, Education and Information Technology Application*, pages 568–575.

Apêndice A

Apêndice A - Formação das cores e Espaços de Cor

A.1 Formação das cores

Um dos experimentos de Isaac Newton presentes em (Newton, 1714) utilizou de uma sala escura e um feixe de luz solar incidindo em um prisma polido que resultou em uma decomposição de cores como as contidas no arco-íris. A princípio acreditava-se que a luz oriunda do sol era pura e que as impurezas do vidro faziam com que houvessem variações nas cores (SANTOS, 2018). Para comprovar que a luz branca era uma composição de diversas cores e não fruto de impurezas da refração, Newton utilizou um segundo prisma para refratar as cores decompostas pelo primeiro prisma, desta forma, ele obteve como resultado a luz branca (Aleodin, 2011). Com isso, demonstrou-se que a luz branca na verdade é uma composição com todos os espectros de cores visíveis e que cada cor possui um comprimento de onda diferente dentro do espectro eletromagnético.

O olho humano é sensível a apenas uma pequena faixa de onda do espectro eletromagnético e as ondas visíveis possuem de 400nm a 700nm de comprimento, conforme ilustrado pela Figura A.1 (SOUTO, 2000). A percepção da cor de um objeto é equivalente ao comprimento de onda que este objeto reflete quando ele é submetido ao processo de iluminação.

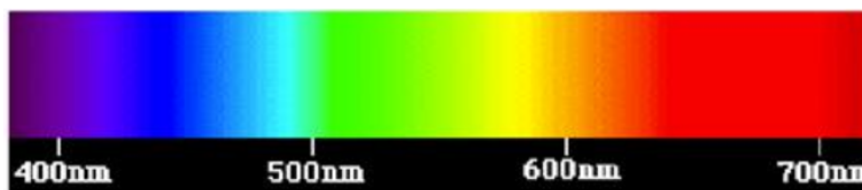


Figura A.1: Espectro eletromagnético visível – Fonte (SOUTO, 2000).

Para que o ser humano possa perceber a cor, ele é dotado de estruturas biológicas específicas. Dentre elas, podemos destacar: pupila, retina, lente, cones e bastonetes. Em linhas gerais, a luz passa pela pupila e é focalizada na retina pela lente. A retina possui fotorreceptores sensíveis a luz – os cones e os bastonetes. Os fotorreceptores por sua vez transformam as ondas eletromagnéticas em estímulos neurais responsáveis por formar a imagem no cérebro completando assim o ciclo de percepção da cor. Estas percepções, segundo (Ford and Roberts, 1998), podem ser definidas como:

- Brilho: Percepção de mais ou menos luz;
- Matiz: Percepção de que uma área ser similar a outra;
- Coloração: Percepção humana da aparência da cor em exibir mais ou menos Matiz;
- Luminância: Percepção de uma área ser mais brilhante comparado a uma referência branca na imagem;
- Croma: A coloração de uma área comparada ao brilho do branco;
- Saturação: A coloração de uma área comparada a seu brilho.

A computação das cores é difícil por meio do espectro de cores. Desta forma, para representar as cores foram propostos diversos espaços de cor que destacam as percepções enumeradas anteriormente. Os espaços de cor são então definidos como valores de coordenadas que em conjunto representam um valor finito para uma cor.

Os espaços de cor, são combinações de cores pelas quais podemos especificar, criar e visualizar cores (Ford and Roberts, 1998). Na percepção humana, pode-se definir cor por seus atributos de brilho, matiz e coloração. Os diferentes espaços de cor existentes são transformações que alteram a percepção de uma cor em três parâmetros distintos. Neste estudo, os espaços de cor estudados são: RGB, YCbCr, YUV, HSI, HSL, HSV e $CIEL^*a^*b^*$. No Apêndice A, encontram-se os cálculos de conversão para os espaços de cor utilizados provenientes do espaço RGB.

A.2 Espaços de Cor

Esta seção traz as transformações nos espaços de cor para imagens dos espaços RGB para os espaços: YCbCr, YUV, HSI, HSV, HSL e $CIEL^*a^*b^*$.

A.2.1 RGB

Uma imagem digital no espaço de cor RGB possui três canais: vermelho (R), verde (G) e azul (B). A combinação destas três cores possibilita a representação de uma cor em um ambiente digital (Ford and Roberts, 1998).

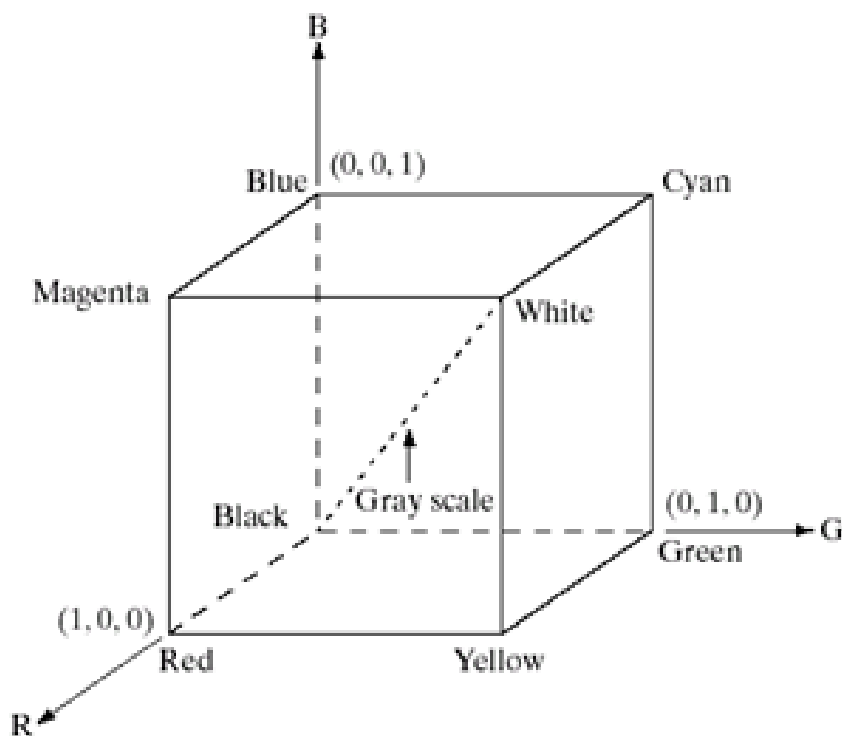


Figura A.2: Representações geométricas do espaço de cor RGB.

Nas imagens digitais, os valores de R, G e B são distribuídos em 256 níveis (8 bits por plano de cor) para cada canal. As cores resultantes são combinações dos valores de cada canal, por exemplo, branco é representado pelos valores máximos de R, G e B e preto é representado pelos valores mínimos de R, G e B.

Uma imagem colorida é composta de pixels, onde, cada pixel é representado por uma localização espacial baseada na grade retangular (x, y) e um vetor de cor $(R(x, y), G(x, y), B(x, y))$ correspondendo a localização espacial nos pontos (x, y) . Na Figura A.2, os canais estão quantificados com valores entre 0 e 1 em 3 planos: R, B; R, G e B, G.

A.2.2 YCbCr e YUV

Segundo Ford and Roberts (1998), este grupo de espaços de cor representam os espaços de cor utilizados pelo sistema de transmissão de televisão, sendo que o espaço YUV é o utilizado nos sistemas PAL e o sistema YCbCr é o sistema para o padrão digital.

No espaço de cores YCbCr, Y indica a luminância, enquanto Cb indica a Crominância Azul e Cr a Crominância Vermelha. A equação A.1, mostra a fórmula de conversão de RGB para YCbCr.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,2568 & 0,5041 & 0,0979 \\ -0,1482 & -0,2910 & 0,4392 \\ 0,4392 & -0,3678 & -0,0714 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (\text{A.1})$$

Já no espaço de cor YUV, o canal Y também representa a Luminância, e os canais UV representam a crominância. A fórmula de transformação deste espaço de cor a partir do espaço RGB é definido pela Equação A.2

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.2})$$

A.2.3 HSI, HSV e HSL

Os espaços de cor HSI, HSV e HSL possuem como característica principal a similaridade com a separação existente na interpretação das cores da visão humana, de tal forma, que este espaço reflete a maneira como os seres humanos veem a cor. Os canais de cor comuns aos três espaços de cor são Matiz (Hue) e a Saturação. Tendo como base um duplo cone como apresentado na Figura 3 a seguir, a Matiz corresponde ao ângulo em que a tonalidade da cor se encontra na circunferência do plano em relação a um ponto que corresponde à cor (0° corresponde a cor vermelha) e a saturação indica a distância do centro do plano até a extremidade (Ford and Roberts, 1998).

Para o espaço de cor HSI, o canal I indica a intensidade da cor, e na Figura A.3 é mostrado que este é um eixo perpendicular a Matiz. Já o espaço de cor HSV, o canal V representa o Valor (Value) da cor, este é responsável por regular o brilho da cor, onde a cor preta possui brilho zero. No espaço de cor HSL, o canal L representa a Luminosidade da cor.

Segundo Ford and Roberts (1998), na transformação do canal RGB (normalizados em valores entre 0 e 1) para o canal HSI, temos as equações A.3, A.4, A.5 e A.6:

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\} \quad (\text{A.3})$$

$$H = \begin{cases} \theta & \text{se } B \leq G \\ 360^\circ - \theta & \text{se } B > G \end{cases} \quad (\text{A.4})$$

$$I = \frac{R + G + B}{3} \quad (\text{A.5})$$

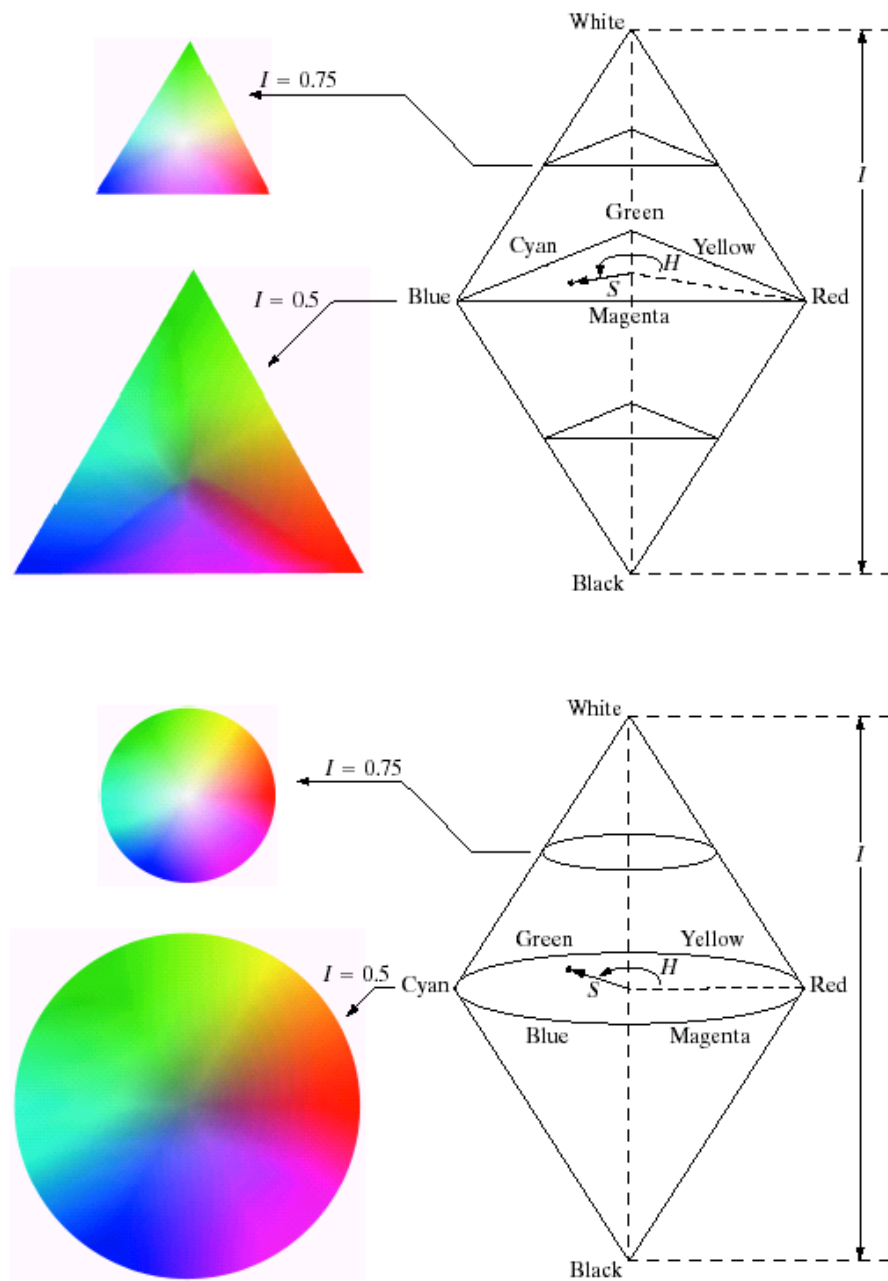


Figura A.3: Representações geométricas do espaço de cor HSI.

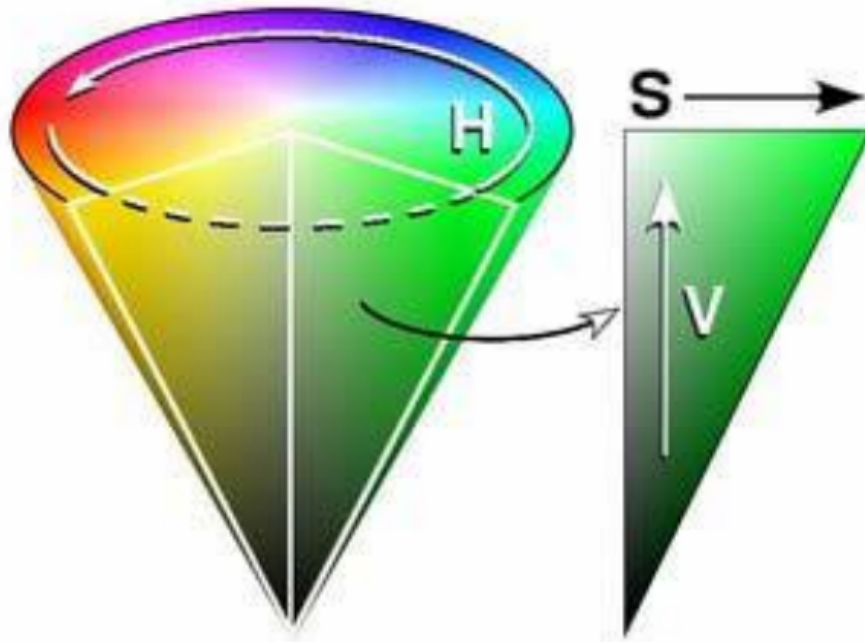


Figura A.4: Representação geométrica do espaço de cor HSV.

$$S = 1 - \frac{3}{R + G + B} * [\min(R, G, B)] \quad (\text{A.6})$$

Seguem as transformações do canal RGB (normalizados em valores entre 0 e 1) para o canal HSV, presentes nas equações: A.7, A.8 e A.9, tendo em vista que $MAX = \max R, G, B$ e $MIN = \min R, G, B$.

$$H = \begin{cases} 60 * \left(\frac{G-B}{MAX-MIN} \right) + 0, & \text{se } MAX = R \text{ e } G \geq B \\ 60 * \left(\frac{G-B}{MAX-MIN} \right) + 360, & \text{se } MAX = R \text{ e } G < B \\ 60 * \left(\frac{B-R}{MAX-MIN} \right) + 120, & \text{se } MAX = G \\ 60 * \left(\frac{R-G}{MAX-MIN} \right) + 240, & \text{se } MAX = B \end{cases} \quad (\text{A.7})$$

$$S = \frac{MAX - MIN}{MAX} \quad (\text{A.8})$$

$$V = MAX \quad (\text{A.9})$$

Na transformação do canal RGB (normalizados em valores entre 0 e 1) para o canal HSL, segundo Ford and Roberts (1998), ocorre somente a mudança do valor do canal L, dado que os valores da matiz H, da saturação S são calculados pelas fórmulas A.7 e A.8 respectivamente, o canal L é calculado a partir da equação A.10.

$$L = V - \frac{S}{2} \quad (\text{A.10})$$

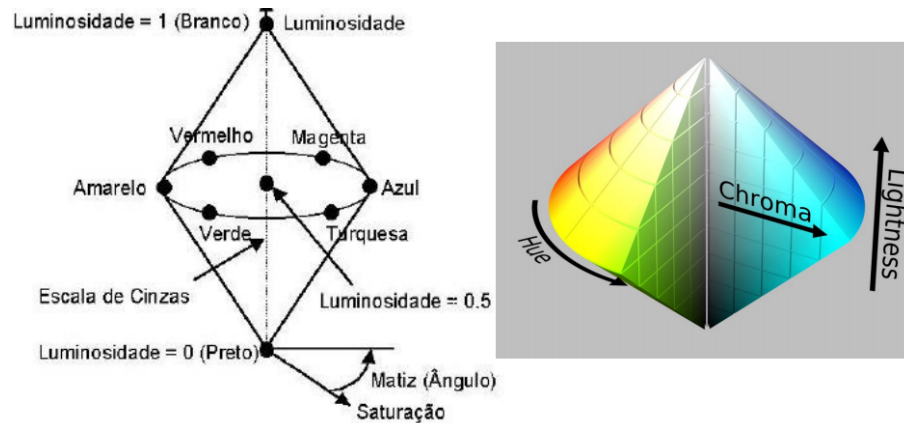


Figura A.5: Representação geométrica do espaço de cor HSL.

A.2.4 CIEL*a*b*

Para realizar a transformação do espaço de cor RGB para o CIEL*a*b*, primeiro é necessária a conversão para o espaço XYZ. Este é um espaço de cor que busca representar a imagem de uma maneira independente do dispositivo utilizado. É composto de três canais (X, Y e Z).

A formulação matemática empregada para a conversão do espaço RGB para o XYZ está estabelecido pela Equação A.11.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,4125 & 0,3576 & 0,1804 \\ 0,2125 & 0,7154 & 0,0721 \\ 0,0193 & 0,1192 & 0,9502 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.11})$$

A formulação matemática utilizada na conversão do espaço de cor XYZ para o espaço de cor CIELab é ilustrado pelas Equações A.12, A.13 e A.14 (Ford and Roberts, 1998).

$$L = f(n) = \begin{cases} \text{Se } n > 0,008856 \text{ então } f(n) = 116 * n^{\frac{1}{3}} - 16 & | n = \frac{Y}{Y_N} \\ \text{Senão } f(n) = 903,3 * n \end{cases} \quad (\text{A.12})$$

$$a = 500 * \left[g\left(\frac{X}{X_N}\right) - g\left(\frac{Y}{Y_N}\right) \right] \quad (\text{A.13})$$

$$b = 200 * \left[g\left(\frac{Y}{Y_N}\right) - g\left(\frac{Z}{Z_N}\right) \right] \quad (\text{A.14})$$

$$\text{Onde, função } g(t) = \begin{cases} \text{caso } t > 0,008856 \text{ então } g(t) = t^{\frac{1}{3}} \\ \text{senão, } g(t) = 7,787 * t + \frac{16}{116} \end{cases} \quad (\text{A.15})$$

X, Y e Z são os valores já calculados pela conversão de RGB para XYZ. Os valores X_n , Y_n , Z_n são conhecidos como tristimulus e fazem referência ao valor branco usado de referência, cujo valores são mostrados na matriz A.16.

$$\begin{aligned} X_n &= 0,95047 \\ Y_n &= 1,0 \\ Z_n &= 1,08883 \end{aligned} \tag{A.16}$$

Apêndice B

Apêndice B - Algoritmos de Aprendizagem de máquina

B.1 Algoritmos de clusterização

B.1.1 K-Means

Segundo (Wagstaff et al., 2001), K-means é um algoritmo de agrupamento (clustering) concebido em 1967 por MacQueen usado para particionar dados em k grupos. Seu funcionamento, no contexto do processamento digital de imagens, baseia-se no processo de selecionar k pixels iniciais (centroides) C_j , onde $0 \leq j < k$, de uma imagem F composta de i pixels. Então, iterativamente estes centroides de acordo com os passos descritos a seguir:

1. Para cada instância F_i é atribuído o centroide mais próximo;
2. Cada centroide C_j é então atualizado como a média das instâncias inseridas na partição j .

Este processo se repete até que não haja mais mudanças nos valores dos centroides de cada partição (cluster).

O processo de classificação usando K-means é bastante eficiente para realizar segmentações, porém o custo computacional para sua execução é considerado alto. Isto se deve ao fato de que para um número de instâncias n , estas são consultadas pelo menos $k \times n$ vezes, o que traz um custo computacional de ordem $O(k \times n)$. Este valor pode aumentar dependendo do número de iterações necessárias para finalizar o processo de atribuição de classes às instâncias.

B.1.2 Nearest Neighbor

Os autores Cover & Hart (Cover and Hart, 1967) citam que o padrão de classificação Nearest Neighbor – NN (Vizinho mais próximo) consiste em armazenar instâncias selecionadas (instâncias de treino) e a partir delas realizar a predição a qual classe outras instâncias farão parte. Esta predição é realizada a partir do cálculo das distâncias entre as instâncias de treino e as instâncias

a serem classificadas. Para cada instância testada verifica-se qual se mostra mais próxima da instância de teste e então atribui-se um valor de classe à instância testada.

A partir do algoritmo NN surgiram outros algoritmos de classificação que refinam o processo de classificação, dentre eles pode-se citar o K-NN, que utiliza a técnica do voto majoritário das instâncias de uma classe C para que uma instância seja atribuída a esta classe. O funcionamento do algoritmo K-NN utiliza a seguinte sequência para as predições:

1. Definir o valor de k como parâmetro do algoritmo.
2. Calcular a distância (Euclidiana) entre a amostra desconhecida e as de treinamento da classe C .
3. Ordenar as distâncias e buscar pelos k vizinhos mais próximos ou k menores distâncias.
4. Utilizar o voto majoritário entre os k vizinhos mais próximos para prever o valor de amostra desconhecida, atribuindo uma classe C a esta amostra.

Outro algoritmo derivado do NN é o IBL, definido por (Aha et al., 1991), o qual se diferencia de K-NN por utilizar a normalização dos intervalos dos atributos e com uma política de tolerância de valores diferenciada e utilizando os cálculos de similaridades para determinar cada atribuição de classe a uma amostra desconhecida. Vide Algoritmo 1 e na Equação B.1.

Algoritmo 1 Algoritmo de Classificação IBL – Fonte: (Aha et al., 1991)

```

procedure IBL( $x, y$ ) // IBL algorithm
   $CD \leftarrow \emptyset$ 
  for  $x \in TrainingSet$  do
    for  $y \in CD$  do
       $Sim[y] \leftarrow Similarity(x, y)$ 
    end for
     $y_{max} \leftarrow$  some  $y \in CD$  with maximal  $Sim[y]$ 
    if  $class(x) = class(y_{max})$  then
       $classification \leftarrow correct$ 
    else
       $classification \leftarrow incorrect$ 
    end if
     $CD \leftarrow CD \cup \{x\}$ 
  end for
end procedure

```

$$Similarity(x, y) = -\sqrt{\sum_{i=1}^n f(x_i, y_i)} \quad (B.1)$$

O trabalho de (Wu et al., 2002) mostra as deficiências deste método e propõe duas melhoras no método para amenizar os custos computacionais sem diminuir a eficiência do método. As deficiências apontadas são: complexidade computacional alta e processo de busca

das amostras conhecidas se mostrando ineficientes e custosas. As estratégias abordadas para otimizar o algoritmo K-NN são de compactação de clusters homogêneos e o pré-processamento que consiste em filtrar uma larga porção de clusters com grandes diferenças comparando-os aos elementos de padrões desconhecidos. Segundo o autor, este processo aumentou a velocidade do K-NN sem a perda de acurácia.

B.2 Algoritmos de Probabilidades

B.2.1 Naïve Bayes

Os classificadores Bayesianos conforme citados em (John and Langley, 1995) são criados a partir do conhecimento de probabilidades, onde há o uso do conjunto de treinamento/aprendizado, ou seja a aplicação do Naïve Bayes é indicada para processos de indução supervisionados. Para a realização do processo de classificação, primeiro são definidas as instâncias de treinamento em vetores X de atributos para as classes C , onde para cada C existem k valores em X . Então, selecionando c como uma classe qualquer do conjunto C e x um valor randômico a ser testado, podemos definir a probabilidade de x pertencer à classe c como na Equação B.2 e então, a partir destes valores, incluir x na classe com maior probabilidade. O cálculo da probabilidade pode seguir a distribuição Gaussiana, descrita na Equação B.3.

$$p(C = c|X = x) = \frac{p(C = c)p(X = x|C = c)}{p(X = x)} \quad (\text{B.2})$$

$$p(X = x|C = c) = g(x, \mu_c, \sigma_c) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp \frac{(x-\mu_c)^2}{2\sigma_c^2} \quad (\text{B.3})$$

Em geral, Naïve Bayes provê um eficiente sistema de indução baseado em probabilidades. O custo computacional para k classes e n instâncias de teste é de $O(k * n)$. Para treino, dado um conjunto de t elementos de treino e k classes o custo computacional é de $O(k \times t)$. A vantagem do Naïve Bayes em relação ao K-Means é que no segundo há o processo iterativo para a definição dos centroides a cada nova definição, enquanto que em Naïve Bayes isto é realizado durante o processo de treinamento, com custo fixo de $O(k \times t)$.