

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia Elétrica

Roger Júnio Campos

**DETECÇÃO DE FOGO BASEADA EM SINAIS DE  
VÍDEO POR MEIO DE ANÁLISE DE SÉRIES  
TEMPORAIS**

Belo Horizonte

2023

Roger Júnio Campos

**DETECÇÃO DE FOGO BASEADA EM SINAIS DE  
VÍDEO POR MEIO DE ANÁLISE DE SÉRIES  
TEMPORAIS**

Versão final

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Hani Camille Yehia

Belo Horizonte

2023



C198d	<p>Campos, Roger Júnio.          Detecção de fogo baseada em sinais de vídeo por meio de análise de séries temporais [recurso eletrônico] / Roger Júnio Campos. - 2023.          1 recurso online (132 f. : il., color.) : pdf.</p> <p>Orientador: Hani Camille Yehia.</p> <p>Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Apêndices: f. 102-132.</p> <p>Bibliografia: f. 85-101.          Exigências do sistema: Adobe Acrobat Reader.</p> <p>1. Engenharia elétrica - Teses. 2. Análise de séries temporais - Teses. 3. Detectores - Teses. 4. Fogo - Teses. 5. Aprendizado de máquina - Teses. 6. Redes neurais convolucionais - Teses. I. Yehia, Hani Camille. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.</p>
	CDU: 621.3(043)

Ficha catalográfica elaborada pela Bibliotecária Roseli Alves de Oliveira CRB/6 2121  
 Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

### FOLHA DE APROVAÇÃO

"DETECÇÃO DE FOGO BASEADA EM SINAIS DE VÍDEO POR MEIO DE ANÁLISE DE SÉRIES TEMPORAIS"

**ROGER JUNIO CAMPOS**

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica. Aprovada em 20 de março de 2023. Por:

Prof. Dr. Hani Camille Yehia - DELT (UFMG) - Orientador

Prof. Dr. Adriano Vilela Barbosa - (UFMG)

Prof. Ph.D. Eduardo Mazoni Andrade Marçal Mendes - DELT (UFMG)

Prof. Dr. Adriano Chaves Lisboa - (Gaia Solutions on Demand)

Prof. Dr. Allan Kardec Duailibe Barros Filho - Departamento de Engenharia de Eletricidade (UFMA)



Documento assinado eletronicamente por **Hani Camille Yehia, Professor do Magistério Superior**, em 20/03/2023, às 11:57, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Adriano Vilela Barbosa, Professor do Magistério Superior**, em 22/03/2023, às 14:34, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Mazoni Andrade Marçal Mendes, Professor do Magistério Superior**, em 24/03/2023, às 09:26, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Adriano Chaves Lisboa, Usuário Externo**, em 24/03/2023, às 11:01, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Allan Kardec Duailibe Barros Filho, Usuário Externo**, em 04/04/2023, às 12:39, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2095814** e o código CRC **D7E7448D**.

*Dedico este trabalho à minha família.*

# Agradecimentos

Agradeço a Deus por tudo que ele permitiu na minha vida: por todo crescimento, por todas as lutas, por todas as dádivas. Agradeço à minha família, em especial minha mãe, meu pai e minhas irmãs. À Lílian por todo apoio e pelas palavras de incentivo nos momentos difíceis. Aos meus amados filhos Davi, Matheus e João Pedro (JP), vocês são as engrenagens da motivação. Aos companheiros da Universidade Federal de Itajubá (UNIFEI) por terem me apoiado no período de afastamento. Aos companheiros do laboratório CEFALA obrigado pelos momentos, conselhos e preciosas dicas. Ao Prof. Hani obrigado pela confiança e apoio no desenvolvimento desta pesquisa.

*Ele faz coisas grandes e inescrutáveis, e maravilhas sem número.*

**Jó 5:9**

# Resumo

Nos últimos anos, incêndios florestais de grandes proporções têm sido frequentemente noticiados em âmbito mundial. Quando não controlados, podem provocar impactos irreparáveis, como perda da biodiversidade, prejuízos à saúde humana, a propriedades rurais, indústrias e a todas as comunidades que permeiam as florestas. Outrossim, devido ao desenvolvimento tecnológico e à necessidade de maiores áreas para plantio de alimentos, as florestas, no mundo todo, têm sido cada vez mais suprimidas. Nesse contexto, torna-se indispensável a adição de técnicas eficazes capazes de detectar incêndios automaticamente. Nesse sentido, nos últimos anos, é possível observar um aumento significativo de técnicas que investigam a detecção automática do fogo. Essas técnicas geralmente são baseadas em algoritmos de visão computacional e podem compreender desde o processamento de vídeos ou imagens ao uso de aprendizado profundo, especialmente as redes neurais convolucionais. Neste trabalho é apresentado um método para detecção de fogo em duas etapas baseado em sinais de vídeo por meio de análise de séries temporais. Na primeira etapa, regiões com possibilidade de incidência de fogo são selecionadas por meio de um detector espacial implementado por uma rede neural convolucional. Na segunda etapa, é utilizado um detector temporal baseado na análise das séries temporais (AST) das regiões de interesse (ROIs) identificadas pelo detector espacial. A AST consiste em uma sequência de passos: inicialmente, a ROI do vídeo investigado é dividida em blocos de dimensão  $N \times N$  *pixels* e, para cada um desses blocos, é computada uma série temporal média. Com intuito de realçar o efeito da cintilação do fogo, a primeira diferença temporal é calculada, rotulada e organizada em vetores de entrada para serem classificados por meio de seis algoritmos de aprendizado de máquina. São analisados valores diferentes do tamanho do bloco  $N \times N$ , observando o desempenho computacional. Ao todo são usados 87 vídeos para avaliar a AST, sendo que 35 compõem o conjunto de treinamento dos algoritmos de aprendizado de máquina e 52 compõem o conjunto de testes. A abordagem AST foi aplicada às ROIs com possibilidades de fogo, identificadas pelo detector espacial. O conjunto de teste é constituído por 50% dos vídeos sendo falsos positivos e 50% sendo verdadeiros positivos. Os resultados experimentais obtidos mostram que a adição de informações temporais ao detector espacial produz reduções significativas na taxa de erro. Essa redução depende do tamanho do bloco utilizado. O melhor desempenho foi obtido com regiões de interesse divididas em tamanhos de bloco de  $8 \times 8$  *pixels* e com o classificador SVM. Nesse cenário, a AST é capaz de eliminar todos os falsos positivos sem reduzir a taxa de verdadeiros positivos, adicionando um tempo médio de processamento por vídeo de menos de 1 segundo.

Palavras-chave: Análise de séries temporais. Detecção de fogo. Aprendizado de máquina. Redes neurais convolucional. Detector espacial.

# Abstract

In recent years, large forest fires have often been reported worldwide. When they are not controlled, they can have irreparable impacts such as loss of biodiversity, human health losses, rural properties, industries and all the communities that permeate forests. In addition, due to technological development and the need for higher areas of food planting, forests worldwide have been increasingly suppressed. In this context, it is essential to add effective techniques capable of automatically detecting fires. In this sense, in recent years, it is possible to observe a significant increase in the techniques that investigate automatic fire detection. These techniques are usually based on computational vision algorithms and can understand from video processing or images to the use of deep learning, especially convolutionary neural networks. This work is presented a method for detecting fire in two steps based on video signals by analyzing the temporal series. In the first stage, regions with the possibility of fire are selected through a spatial detector implemented by a convolutionary neural network. In the second stage, a temporal detector is used based on the analysis of the temporal series (AST) of the regions of interest identified by the space detector. AST consists of a sequence of steps: Initially, the investigated video ROI is divided into blocks of dimension  $n \times N$  pixels and, for each of these blocks, an average temporal series is calculated. To improve the glow effect of fire, the first temporal difference is calculated, labeled and organized into input vectors to be classified by six machine learning algorithms. The values different from the size of the Block  $N \times N$  are analyzed, observing computational performance. In all 87 videos are used to evaluate AST, with 35 composing the training set of machine learning algorithms and 52 make up the test set. The AST approach was applied to ROI's with fire possibilities, identified by the space detector. The test set consists of 50 % of the videos being false positive and 50 % being true positive.

Experimental results show that the addition of temporal information to the spatial detector produces significant error reductions. This reduction depends on the size of the block used. The best performance was obtained with regions of interest divided into  $8 \times 8$  pixels and SVM classifier. In this scenario, AST is able to eliminate all false positives without reducing the real positive rate by adding an average video processing time less than 1 second.

*Keywords: Time series analysis. Fire detection. Machine learning. Convolutional neural networks. Spatial detector.*

# Lista de Ilustrações

Figura 1 – Categorias de classificação das técnicas de detecção de fogo e fumaça. . . . .	18
Figura 2 – Sistema híbrido de detecção de fogo proposto baseado em duas etapas: detecção espacial e análise de séries temporais. . . . .	20
Figura 3 – Histórico temporal dos focos ativos de incêndios na América Latina e Brasil no período entre 1998 e 2022. . . . .	21
Figura 4 – Número de trabalhos científicos agrupados por ano de publicação. Período: 1990-2019. . . . .	25
Figura 5 – Imagem de uma fogueira e sua composição pelos três canais: R, G e B. . . . .	36
Figura 6 – Detalhes de <i>pixels</i> fogo, fumaça, mata e céu da imagem e sua composição pelas três intensidades de cores R, G e B. . . . .	37
Figura 7 – Representação de um vídeo com as componentes de cores RGB e comprimento de $k$ quadros. . . . .	38
Figura 8 – Séries temporais de um <i>pixel</i> RGB em um cenário que não contém fogo. . . . .	39
Figura 9 – Séries temporais de um <i>pixel</i> RGB em um cenário que contém fogo. . . . .	40
Figura 10 – Árvore de decisões composta pela raiz, nó e folhas. . . . .	44
Figura 11 – Classes, $C_1$ e $C_2$ , separadas por uma função discriminante linear. . . . .	45
Figura 12 – Classificação usando Naïve para separar duas classes $C_1$ ou $C_2$ . . . . .	46
Figura 13 – Regressão logística separando duas classes $C_1$ ou $C_2$ . . . . .	47
Figura 14 – Algoritmo $K$ nn classificando uma amostra entre duas classes $C_1$ e $C_2$ . . . . .	48
Figura 15 – Hiperplano de uma máquina de suporte vetorial (SVM). . . . .	49
Figura 16 – Esquema do método de validação cruzada $k$ -fold com $k = 5$ . O conjunto de dados é dividido em $k$ partes iguais. . . . .	50
Figura 17 – Processo de detecção do algoritmo YOLO. . . . .	52
Figura 18 – Quadros dos 35 vídeos que compõem o conjunto de dados utilizados na etapa de treinamento e validação. . . . .	57
Figura 19 – Quadros de 30 dos 52 vídeos que compõem o conjunto de dados utilizados na etapa de teste. . . . .	58
Figura 20 – Sistema para detecção de fogo composto por dois estágios sequenciais: (i) detector e (ii) análise de séries temporais. . . . .	59
Figura 21 – Análise de séries temporais (AST): processo multi-passos extrai as séries temporais de sinais de vídeo. . . . .	60



Figura 22 – Região de interesse (falso positivo) detectada pelo algoritmo YOLO em um quadro de um vídeo. . . . .	61
Figura 23 – Quadro de um vídeo dividido em blocos de tamanho $N = 16 \times 16$ pixels. . . . .	62
Figura 24 – Quadro de um vídeo dividido em blocos de tamanho $N = 32 \times 32$ pixels. . . . .	63
Figura 25 – (a) Séries temporais ( $\mathbf{x}$ ) de um bloco pertencente à região de fogo. (b) Séries temporais médias ( $\overline{ST}_{Bloco}$ ). . . . .	64
Figura 26 – (a), (b) e (c) são séries temporais ( $\mathbf{x}$ ) com diferentes taxas: 60 FPS, 30 FPS e 2 FPS. (d), (e) e (f) apresentam as respectivas diferenças temporais ( $\Delta_{ST}$ ). . . . .	65
Figura 27 – Diferenças temporais ( $\Delta_{ST}$ ) das componentes (a) vermelha (R), (b) verde (G) e (c) azul (B). . . . .	67
Figura 28 – Acurácias da etapa de validação dos classificadores para cada um dos tamanhos de blocos $N$ . . . . .	71
Figura 29 – Dimensão das ROIs de cada um dos vídeos usados na etapa de testes. . . . .	73
Figura 30 – Taxas de detecção (%) dos classificadores para os diferentes tamanhos de blocos usando os vídeos do conjunto de teste. . . . .	75
Figura 31 – Quadro de um vídeo de teste que apresenta um evento parecido com fogo (falso positivo). . . . .	79
Figura 32 – Quadro de um vídeo de teste que apresenta incêndio em estágio inicial (verdadeiro positivo). . . . .	80

# Lista de Tabelas

Tabela 1 – Estado da arte da detecção de fogo e da detecção de fumaça: técnicas de detecção que realizam a extração de características destacadas na literatura. . . . .	31
Tabela 2 – Metodologias recentes usando redes neurais convolucionais (CNN) na detecção de fogo e fumaça. . . . .	34
Tabela 3 – Conjuntos de dados usados para o treinamento dos classificadores. . . .	70
Tabela 4 – Tempo de processamento (em segundos) da etapa de treinamento dos modelos para cada um dos tamanhos dos blocos $N$ . . . . .	71
Tabela 5 – Tempo médio ( $\pm$ desvio padrão) em segundos da AST para processar um video. . . . .	72
Tabela 6 – Tempo de processamento em segundos da maior e menor ROI dos vídeos de teste usando o classificador SVM. . . . .	74
Tabela 7 – Desempenho do método AST na detecção de fogo. . . . .	76
Tabela 8 – Métricas do conjunto de testes: impacto da AST na detecção de fogo. .	77

# Lista de Abreviaturas e Siglas

AST	Análise de séries temporais
DFV	Detecção de fogo em vídeos
DFV	Detecção de fumaça em vídeos
RGB	Vermelho, verde e azul (do inglês <i>Red, green and blue</i> )
HSI	Tonalidade, saturação e intensidade (do inglês <i>Hue, saturation and intensity</i> )
HSV	Tonalidade, saturação e valor do brilho (do inglês <i>Hue, saturation and value</i> )
YCbCr	Luminância, crominância azul e crominância vermelha (do inglês <i>Luminance component and CB and CR are the blue-difference and red-difference chroma components</i> )
YUV	Luminância, projeção azul e projeção vermelha (do inglês <i>Luminance component, blue projection and red projection</i> )
CNN	Rede neural convolucional (do inglês <i>Convolutional neural network</i> )
CFTV	Circuito fechado de televisão
INPE	Instituto Nacional de Pesquisas Espaciais
ROI	Região de interesse (do inglês <i>Region of interest</i> )
LDA	Análise discriminante linear (do inglês <i>Linear discriminant analysis</i> )
Knn	<i>K</i> -vizinhos mais próximos (do inglês <i>K-Nearest Neighbors</i> )
SVM	Máquinas de Vetores de Suporte (do inglês <i>Support vector machine</i> )
AT	Análise Temporal
YOLO	<i>You Only Look Once</i>

<i>FP</i>	Falso positivo
<i>FN</i>	Falso negativo
<i>VP</i>	Verdadeiro positivo
<i>VN</i>	Verdadeiro negativo
<i>TFP</i>	Taxa de falsos positivos
<i>TFN</i>	Taxa de falsos negativos
<i>TVP</i>	Taxa de verdadeiros positivos
<i>TVN</i>	Taxa de verdadeiros negativos

# Lista de Símbolos

$f(c, l)$	Imagem digital composta por uma matriz bidimensional de números reais
$f(c_0, l_0)$	Valor da função $f(c, l)$ para um determinado pixel com coordenadas $c_0, l_0$
$k$	Número de quadros (frames) de um vídeo
$K$	Número de vizinhos mais próximos
$x_{(c_0, l_0)}(k)$	Série temporal de um pixel nas coordenadas $c_0, l_0$ e comprimento de $k$ quadros
$f_{default}$	Quadro padrão
$D(f(n))$	Imagem resultante da diferença entre um quadro e um quadro padrão
$\Delta_{(c, l)}$	Diferença temporal de uma série temporal $x_{(c_0, l_0)} [k]$
$Acc$	Acurácia
$F_1$	Medida $F_1$
$r$	Revogação ou <i>recall</i>
$p$	Precisão
$n$	Número de séries temporais em um bloco ou região de interesse
$N$	Tamanho do bloco em que a imagem é dividida
$\overline{ST}_{Bloco}$	Série temporal média de um bloco com dimensão $N$

# Sumário

<b>1</b>	<b>Introdução</b>	<b>17</b>
1.1	Motivação	21
1.2	Objetivos	22
1.3	Organização da tese	23
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>25</b>
2.1	Comentários finais	34
<b>3</b>	<b>Fundamentação Teórica</b>	<b>35</b>
3.1	Representação de vídeos em séries temporais	35
3.2	Diferenças Temporais	39
3.3	Aprendizado de máquina	41
3.3.1	Árvore de decisões	43
3.3.2	Análise Discriminante Linear (LDA)	44
3.3.3	Classificador Naïve	45
3.3.4	Regressão logística	46
3.3.5	$K$ -vizinhos mais próximos ( $Knn$ )	47
3.3.6	Máquinas de Vetores Suporte (SVM)	48
3.4	Validação cruzada	50
3.5	Aprendizado de máquina profundo - YOLO	50
3.6	Avaliação de desempenho	52
3.6.1	Acurácia	52
3.6.2	Medida $F_1$ ( $F_1$ -score)	53
3.7	Comentários finais	53
<b>4</b>	<b>Metodologia</b>	<b>55</b>
4.1	Conjunto de dados	55
4.2	Abordagem proposta	58
4.2.1	Região de interesse (ROI)	59
4.2.2	Blocos	60
4.2.3	Série temporal média	62
4.2.4	Diferenças Temporais	63
4.2.5	Classificação das séries temporais usando aprendizado de máquina	66
4.3	Comentários finais	68
<b>5</b>	<b>Resultados</b>	<b>69</b>
5.1	Conjuntos de dados atribuídos aos tamanhos dos blocos $N$	69
5.2	Classificação das séries temporais usando aprendizado de máquina	70
5.3	Teste dos classificadores	72
5.4	Comentários finais	78

<b>6</b>	<b>Conclusão</b>	<b>82</b>
6.1	Trabalhos futuros	83
	<b>Referências</b>	<b>85</b>
	<b>Apêndice A Detecções do conjunto de teste</b>	<b>102</b>
A.1	Árvore de decisões	103
A.2	LDA	108
A.3	Naïve	113
A.4	Regressão logística	118
A.5	$K$ -vizinhos mais próximos ( $Knn$ )	123
A.6	SVM	128

# Capítulo 1

## Introdução

Incêndios não controlados são eventos que podem surgir naturalmente ou podem ser provocados pela ação humana. Independente da forma como se iniciam, são eventos destrutivos, com potencial para causar danos irreparáveis à vida e ao meio ambiente. Além do fogo, que devasta áreas florestais, exterminando diversas espécies da flora e da fauna, a fumaça e a fuligem se espalham por quilômetros, prejudicando a qualidade do ar e interferindo na saúde da população atingida. Outrossim, o fogo ainda pode causar danos a propriedades e a redes de transmissão de energia elétrica com prejuízos imensuráveis (Lisboa et al., 2020; Avazov et al., 2022).

Para minimizar a de incêndios não controlados na natureza, diferentes técnicas de detecção de fogo, com metodologias próprias, têm sido experimentadas e utilizadas (Jadon et al., 2019). Técnicas convencionais adotam sensores que detectam fumaça e fogo. Estes são considerados populares, devido ao baixo custo e simples operação (Yin et al., 2017). Contudo, apresentam limitações, tais como retardo no acionamento do alarme, cobertura restrita a pequenas áreas e necessidade de um sistema para a transmissão dos sinais. Comumente, esses sensores são destinados a espaços confinados e são instalados no teto, o que faz com que o tempo necessário para a fumaça e/ou o calor serem detectados seja longo, resultando em um atraso na detecção. As peculiaridades dos sensores convencionais dificultam sua instalação em espaços abertos, como por exemplo, áreas públicas, estádios, hangares, dentre outros.

Por outro lado, há as técnicas de detecção de fogo em vídeo, que apresentam resposta rápida, detecção em ambientes abertos e fechados, sem necessidade de contato, capacidade de prever o progresso da fumaça e das chamas e abranger extensos limites territoriais, sendo limitados pela abertura da câmera (Ojo e Oladosu, 2014). Além disso, os custos para instalação e manutenção de um sistema dessa natureza podem não ser tão altos (Töreysin et al., 2005). Um sistema composto por câmeras, como por exemplo um Circuito Fechado de Televisão (CFTV), captura as imagens em tempo real possibilitando que as chamas sejam detectadas mais rapidamente em comparação com a detecção por

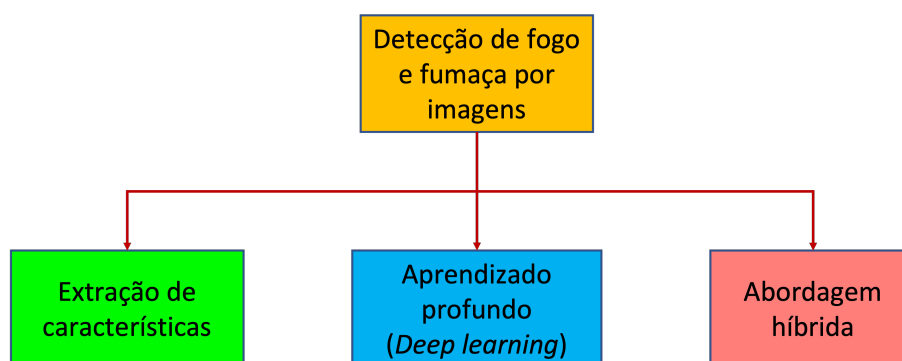


sensores (Çetin et al., 2016).

A partir de 1980 a detecção de fogo em vídeo ganhou notoriedade, uma vez que foi nesta época que os sensores para captação de imagens se tornaram mais acessíveis. Um dos primeiros trabalhos visando a Detecção de Fogo em Vídeo (DFV) e a Detecção de Fumaça em Vídeo (DFV) foi publicado por Cappellini et al. (1989). Desde então, tem aumentado as publicações que demonstram a viabilidade da DFV e DFV como alternativas aos métodos tradicionais, sendo úteis em resolver vários problemas associados com os sensores convencionais de fogo e fumaça (Verstockt, 2011; Xiong et al., 2007; Çetin et al., 2013).

As técnicas de detecção de fogo e fumaça por imagens podem ser classificadas em três categorias: extração de características, aprendizado profundo e abordagem híbrida, conforme é demonstrado na Figura 1 (Gaur et al., 2020; Subbiah et al., 2020). Em relação à primeira categoria, é necessário conhecimento prévio das características do fogo e da fumaça para extraí-las. A partir daí, podem ser utilizados modelos simples de classificação binária. O uso de classificadores demanda organizar as características de tal forma que seja possível utilizá-las como vetores de entrada para serem classificadas. Por sua vez, a ausência de classificadores exige que sejam propostas regras para definir regiões de interesse que envolvam espaços de cores, análise de movimentos, dentre outras. Já a detecção por meio de técnicas de aprendizado profundo exige obter uma visão geral das diversas arquiteturas para garantir generalização das características espaciais, de tal forma que seja possível detectar eventos reais de fogo, mesmo que para isso ocorram detecções de objetos parecidos com o fogo. A obtenção de um detector que utilize a abordagem híbrida exige conhecimento tanto das características do fogo e fumaças, quanto das técnicas de aprendizado profundo (Gaur et al., 2020).

Figura 1 – Categorias de classificação das técnicas de detecção de fogo e fumaça.



Fonte: Adaptado de Gaur et al. (2020).

Se na abordagem que utiliza a extração de características, com uso de aprendizado de máquina, é difícil processar os dados no formato bruto, pois é requerido conhecimento prévio para se projetar um extrator de características a fim de transformar esses dados em

um vetor de características que possam ser classificadas (Lecun et al., 2015), a abordagem com técnicas de aprendizado profundo remove estas dificuldades. O aprendizado profundo tem ganhado força na última década, e tem sido bastante utilizado para a detecção de fogo e fumaça (Muhammad et al., 2018a; Li et al., 2020; Muhammad et al., 2018b; Majid et al., 2022). No entanto, ao aprendizado profundo estão associadas algumas limitações importantes, como longos tempos de treinamento, necessidade de conjuntos de dados robustos e diversificados para o treinamento, falta de controle sobre os processos internos e a incapacidade de detectar padrões temporais (Schmidhuber, 2015). Como consequência, podem produzir falsos alarmes (falsos positivos) ou perder eventos reais (falsos negativos) (Venâncio et al., 2021).

A abordagem híbrida surge como tentativa de melhorar o desempenho da detecção de fogo automática (Gaur et al., 2020). A tendência desta técnica é combinar informações espaciais e temporais para detectar fogo de modo a propor arquiteturas mais robustas (Xie et al., 2020; Nguyen et al., 2021; Shahid et al., 2022). Comumente, as abordagens híbridas se desenvolvem em duas etapas. Na primeira, são extraídas as características em função de regras baseadas principalmente em cores e movimentos, definindo, assim, regiões de interesse. Na segunda etapa, é aplicado o aprendizado profundo de máquina para extrair as características espaciais do fogo.

Caminhando nessa direção, este trabalho realiza um estudo bibliográfico de técnicas de detecção baseadas em visão computacional aplicadas à detecção de fogo e fumaça publicadas nas últimas décadas. Outrossim, é proposto um método para detecção de fogo baseado em duas etapas, conforme é mostrado na Figura 2. Na primeira etapa é realizada a detecção espacial baseada em aprendizado profundo utilizando o algoritmo YOLO (*You Only Look Once*) (Redmon et al., 2016). Neste projeto, vídeos são transmitidos a uma taxa de 2 quadros por segundo. Essa taxa é muito baixa para que seja possível uma análise espectral convencional, pois a cintilação do fogo atinge pelo menos 12 Hertz (Hz). O objetivo desta etapa é definir uma região de interesse com grande potencial de haver a presença de fogo. O algoritmo YOLO possui elevado nível de implementação e possui altas taxas de detecção de incêndios. Gaur et al. (2020) e Subbiah et al. (2020) apresentam diversas arquiteturas que utilizam aprendizado profundo aplicadas na detecção automática de fogo.

Na primeira etapa, a detecção do fogo a partir das características espaciais foi desenvolvida por Venâncio et al. (2021). O algoritmo YOLO usado para detecção de incêndios está em uso no projeto *Apaga o FOGO*<sup>1</sup> (CEMIG, UFMG, Gaia, RaroLabs e UFVJM, 2021).

---

<sup>1</sup> O *Apaga o Fogo* utiliza câmeras para monitorar, em tempo real, as matas e florestas, detectando automaticamente os focos de incêndio através de um software de inteligência artificial. [www.apagaofogo.eco.br](http://www.apagaofogo.eco.br).

Figura 2 – Sistema híbrido de detecção de fogo proposto baseado em duas etapas: detecção espacial e análise de séries temporais.



Fonte: Autoria própria.

O segundo passo surge com o intuito de potencializar a detecção espacial, realizando a análise das características temporais. Dessa forma, este passo visa incorporar as características temporais de forma a solucionar as detecções que o YOLO não é capaz de identificar corretamente. Conforme será descrito, a AST é uma abordagem multipassos que permite classificar quais séries da região delimitada pelo detector espacial pertencem às chamadas de fogo ou não. O desempenho de seis classificadores diferentes das séries temporais é comparado. Junto a isto, as regiões de interesse são divididas em blocos. Cada bloco é composto por vários *pixels* e uma única série temporal média o representa. Quanto maior a dimensão do bloco, menor o custo computacional e, assim, é possível avaliar o desempenho dos classificadores à medida que a dimensão dos blocos é alterada.

A diferença temporal de cada série temporal média de bloco é proposta com intuito de realçar as amplitudes das séries que possuem baixa taxa de quadros. Os vídeos usados para treinamento, validação e teste apresentam taxa de 2 quadros por segundo e duração de 30 segundos, resultando em um total de 60 quadros por vídeo. Uma outra maneira de analisar as séries temporais seria observar o domínio da frequência, aplicando por exemplo a transformada de Fourier. No entanto, a cintilação das chamadas de fogo apresentam oscilações maiores que 12 Hz. Sendo assim, a análise no domínio da frequência é inviável, pois seria necessário considerar vídeos com taxas acima de 24 quadros por segundo, de acordo com o Teorema da Amostragem (Aguirre, 2007).

Para obtenção dos resultados são utilizados ao todo 87 (oitenta e sete) vídeos, sendo que destes, 35 (trinta e cinco) são utilizados para as etapas de treinamento e validação e 52 (cinquenta e dois) vídeos utilizados para avaliar o desempenho da AST na etapa de teste. Os resultados da análise do comportamento temporal da região de interesse, usando a AST, permite, a um baixo custo computacional, diminuir significativamente a taxa de falsos positivos sem depreciar a taxa de verdadeiros positivos. Considerando o conjunto de vídeos de teste, a AST todos os falsos positivos, adicionando um tempo médio de processamento por vídeo de 0,54 segundos utilizando o classificador SVM (máquina de vetores de suporte) e a dimensão dos blocos igual a  $8 \times 8$  *pixels*. A abordagem atinge 100% de acurácia. A AST apresenta algumas limitações, como por exemplo, não é capaz de ser aplicada a

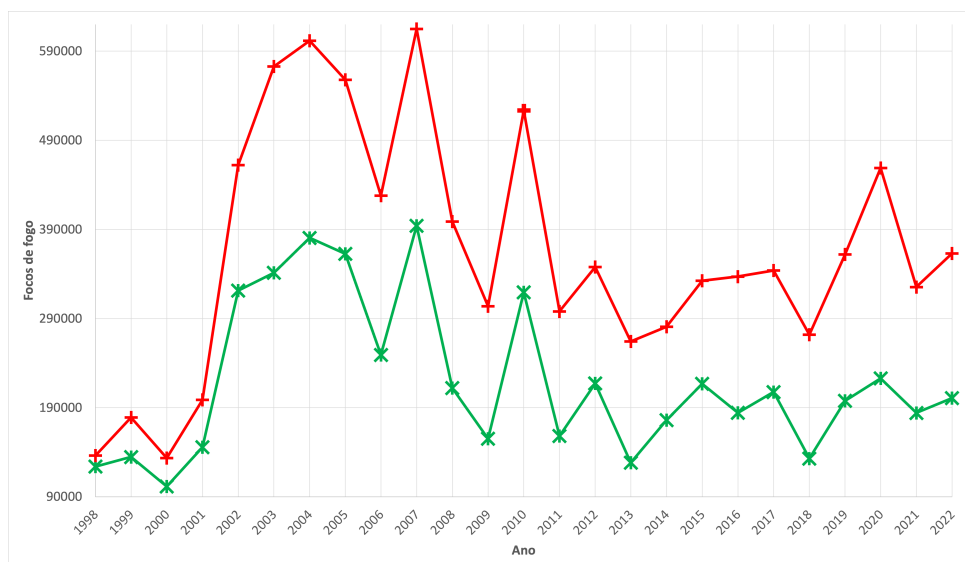
vídeos de câmeras dinâmicas, ou seja, câmeras com movimentos vertical, horizontal e zoom. E também, a AST suprime apenas falsos positivos, sendo incapaz de suprimir falsos negativos, uma vez que, neste trabalho, ela é utilizado após a detecção realizada pelo detector espacial. Os resultados obtidos mostram que a AST é uma importante ferramenta para auxiliar na detecção de incêndio a partir de vídeos.

## 1.1 Motivação

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), nos anos de 2021 e 2022, foram registrados 688.252 focos ativos de incêndio na América do Sul. Apenas o Brasil atingiu 384.844, ou seja, cerca de 56% desse total (INPE, 2022). Para contextualização, na Figura 3 são apresentadas as séries temporais dos focos ativos de incêndio identificados na América Latina e Brasil no período entre 1998 e 2022.

Por si só, essas considerações motivam o estudo da detecção de incêndios. Porém, detectar fogo a partir de um vídeo cuja câmera é criteriosamente configurada é um grande desafio. Esse desafio torna-se maior em um contexto em que as câmeras de vídeo possuem configurações realizadas automaticamente. Sensibilidade, abertura e velocidade do obturador são algumas das configurações que comumente são automaticamente ajustadas na maioria das câmeras de monitoramento.

Figura 3 – Histórico temporal dos focos ativos de incêndios na América Latina (-+) e Brasil (-\*) no período entre 1998 e 2022.



Fonte: Adaptado de INPE (2022).

Mesmo diante dessas dificuldades, o número de trabalhos sobre DFV e DFV usando visão computacional cresceu consideravelmente nas últimas décadas (Çetin et al., 2016). Não obstante esse crescimento, a detecção de fogo não é um problema completamente

resolvido. No Capítulo 2 são descritos diversos trabalhos abordando as mais diferentes técnicas aplicadas com um único objetivo: detectar fogo e/ou fumaça. Cada método visa representar as características do fogo e da fumaça para situações diferentes, sejam elas cenários abertos ou fechados, cidades ou florestas, durante o dia ou noite, e assim por diante. Modelos ou métodos globais, que abrangem todas as situações, são representações que, devido à complexidade computacional e à quantidade de informações de cada uma das situações podem se tornar inviáveis a curto e médio prazo.

Essas ponderações motivam o desenvolvimento deste trabalho que propõe um novo método para detecção de fogo em vídeos. Para isso, as informações temporais dos vídeos são consideradas e representadas como amostras igualmente espaçadas no tempo, nomeadas como *séries temporais*.

O uso das séries temporais para representar vídeos é limitado na literatura. Em [Töreyn et al. \(2006\)](#) os autores obtêm as variações temporais de determinados *pixels*, porém para delimitarem possíveis regiões de fogo, analisam as variações dos subsinais obtidos pela decomposição por um banco de filtros utilizando a transformada Wavelet. Os autores mostram que os *pixels* que pertencem às regiões de fogo, em especial às bordas, apresentam nos seus subsinais variações que estão diretamente relacionadas à cintilação do fogo. Ou seja, as variações se devem à alternância das cores de fogo e do plano de fundo desse determinado pixel sob análise. No entanto, a análise do histórico temporal dos *pixels* só é realizada após a aplicação de outras técnicas que possibilitam detectar regiões de interesse, limitando a quantidade de *pixels* analisados. O uso do histórico temporal de *pixels* de fogo analisados sob a ótica dos subsinais da transformada Wavelet pode ser encontrado também em [Ko et al. \(2009\)](#) e [Zhang et al. \(2008\)](#).

[Hamad e Tsumura \(2012\)](#) utilizam séries temporais de *pixels* para detectar e extrair silhuetas de objetos do primeiro plano de vídeos. O método proposto utiliza, além da análise estatística das séries temporais dos *pixels*, outras técnicas para detecção de movimento e identificação das regiões do plano de fundo.

Adotando o conceito de séries temporais dos *pixels* em vídeos, desbravado por [Töreyn et al. \(2006\)](#) e [Hamad e Tsumura \(2012\)](#), visa-se, neste trabalho, analisar o comportamento temporal a partir das séries temporais das regiões de interesse, definidas por um detector espacial, com intuito de auxiliar na detecção de fogo, diminuindo a taxa de falsos alarmes, aumentando assim a eficiência do procedimento.

## 1.2 Objetivos

O presente trabalho tem como objetivo geral diminuir a taxa de falsos positivos em sistemas reais de detecção automática de incêndios. Para isso, é proposta uma abordagem que auxilie na detecção automática de incêndios florestais a partir das características

temporais. A análise das séries temporais é realizada em regiões de interesse detectadas pelo algoritmo YOLO proposto por [Venâncio et al. \(2021\)](#) para detectar incêndios. Os objetivos específicos podem ser enumerados conforme listado:

1. coleta de vídeos contendo ocorrências de incêndio em diversas fontes de informações gratuitas, visando qualidade e variedade;
2. representação de sinais de vídeos em séries temporais;
3. construção de um conjunto de dados de séries temporais rotuladas pertencentes às regiões de fogo e regiões comuns;
4. analisar a divisão das regiões de interesse em blocos de tamanhos diferentes, obtendo, assim, a melhor relação entre custo computacional e desempenho dos classificadores;
5. utilização de modelos capazes de classificar as séries temporais. Para este passo seis diferentes classificadores são comparados: árvore de decisões, análise de discriminante linear, naïve, regressão logística,  $K$  vizinhos mais próximos e máquina de vetores suporte;
6. avaliar os modelos levando-se em consideração a diminuição da taxa de falsos positivos, sendo que para isso, a taxa de verdadeiros positivos seja mantida.

### 1.3 Organização da tese

Este trabalho apresenta mais cinco capítulos. No Capítulo 2 é realizada uma revisão bibliográfica contendo os conceitos e técnicas envolvidos no âmbito da detecção automática de fogo e fumaça.

O Capítulo 3 apresenta a fundamentação teórica necessária para a compreensão deste trabalho. A representação de sinais de vídeos em séries temporais é tratada, como também o cálculo da primeira diferença de uma série temporal, chamada de diferença temporal. Os classificadores que usam aprendizado de máquina são retratados e também a sua metodologia de validação. O algoritmo de aprendizado profundo, YOLO, também é destacado. As principais métricas usadas para avaliar o desempenho dos classificadores são descritas.

No Capítulo 4 são apresentados dois bancos de dados distintos. O primeiro foi usado exclusivamente para treinar e validar os classificadores. O segundo foi empregado na etapa de testes dos modelos. A abordagem Análise de Séries Temporais (AST) é detalhada.

Os resultados da classificação usando aprendizado de máquina são apresentados no Capítulo 5. O tamanho do bloco é avaliado com o objetivo de obter a melhor relação

---

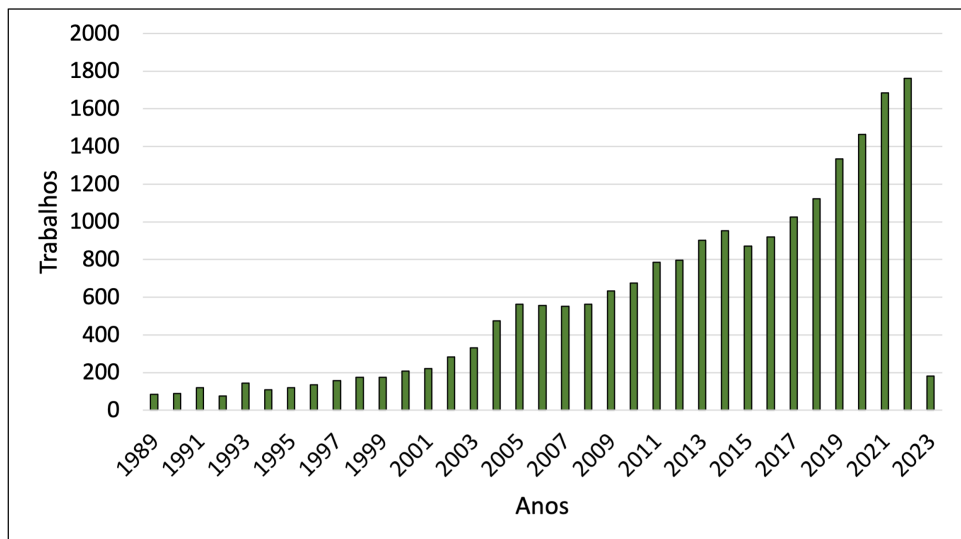
entre custo computacional e desempenho dos classificadores. As conclusões e os próximos passos a serem seguidos são descritos no [Capítulo 6](#).

## Capítulo 2

# Trabalhos Relacionados

Nas últimas três décadas, diversos trabalhos que permeiam a detecção de fogo têm sido publicados. Se for considerada apenas a base de dados bibliográficos Compendex<sup>1</sup> da Elsevier, esse número tem crescido consideravelmente, conforme pode ser visto na Figura 4. De acordo com a base Compendex, em 1989 o número de trabalhos que continham o termo “detecção de fogo” foi igual a 84 e, em 2022, atingiu 1761 trabalhos.

Figura 4 – Número de trabalhos científicos agrupados por ano de publicação. Período: 1990-2019.



Fonte: Adaptado de [Elsevier \(2022\)](#).

Os números apresentados graficamente na Figura 4 mostram que o tema “detecção de fogo” é um problema complexo que ainda precisa ser explorado. Investigar esse assunto

<sup>1</sup> Compendex é um banco de dados bibliográficos de engenharia publicado pela Elsevier. O Compendex indexa a literatura científica referente a materiais de engenharia, ciência da computação e temas correlacionados. O conteúdo oferece mais de 20 milhões de registros, com referências de milhares de revistas científicas especializadas e anais de conferências. A partir de 1884, foi compilado manualmente sob o título original de Engineering Index. O nome “Compendex” significa *COMP*uterized *EN*gineering *in*DEX (Elsevier, 2022).



é primordial, face a diversos prejuízos que as queimadas não controladas podem causar à fauna, à flora, às indústrias e às áreas urbanas.

Os trabalhos encontrados na literatura costumam compartilhar técnicas para detectar fogo e fumaça. Obviamente, essas técnicas consideram parâmetros diferentes para cada tipo de detecção, pois as características da fumaça e do fogo são diferentes. Conforme descrito em [Beutling \(2009\)](#), o fogo é definido como uma rápida oxidação de um material em um processo químico exotérmico liberando calor, luz e vários produtos resultantes da reação química, ou seja, uma reação de combustão que produz luz. Em incêndios florestais, geralmente irá ocorrer a combustão incompleta do material combustível (por exemplo, a madeira), produzindo uma grande quantidade de partículas ou fuligem ([Quintiere, 2016](#)). Nesse caso, a cor da chama é determinada principalmente pela radiação térmica emitida por essas partículas ou pela fuligem, em que a cor varia do amarelo claro ou branco até o vermelho ([Company, 2010](#)).

A fumaça, por sua vez, é um forte indicativo de que há fogo e, em um incêndio, é resultante de gases, vapor de água, materiais particulados finos e resíduos da combustão ([Ye et al., 2015](#)). É possível enxergar duas cores de fumaça bem distintas: branca e cinza. A fumaça branca é resultante de material combustível fino, com pouco material lenhoso e com alto teor de umidade. Essa coloração se deve, principalmente, ao volume de vapor d'água nos combustíveis. A fumaça cinza é resultante de material combustível mais seco e com maior quantidade de material lenhoso ([Rasbash e Drysdale, 1982](#); [Seito et al., 2008](#)). Geralmente, o contorno, a forma, a velocidade, a direção do movimento e a textura interna do bloco de fumaça são fortemente influenciados pelo fluxo de ar. As bordas da fumaça apresentam cintilação com frequências que variam entre 1 e 3 Hz, enquanto as bordas de uma chama cintilam com frequências de 2 a 15 Hz ([Yamagishi e Yamaguchi, 2000](#); [Phillips et al., 2002](#); [Töreyn et al., 2005](#); [Memarzadeh e Mohammadi, 2015](#)).

O comportamento da fumaça e das chamas de um incêndio não controlado são diferentes quando consideradas a distância e a iluminação. Além disso, as câmeras possuem características distintas entre si, com diferentes sensores e algoritmos de cores e balanceamento de iluminação. Esses fatores produzem diferentes imagens e vídeos para uma mesma cena, influenciados também pelas configurações internas das câmeras e dos seus algoritmos ([Ruas, 2009](#)).

Em relação à abordagem que utiliza técnicas de extração, com o uso ou não de classificadores, são comumente utilizadas na detecção de fogo em vídeo (DFV) e detecção de fumaça em vídeo (DFV) a detecção por cores, a detecção de objetos em movimento, a análise do movimento e cintilação, a análise dinâmica de textura e padrão, a análise de desordem, a análise de localização, a divisão dos quadros em blocos menores (*sub-blocking*), dentre outras ([Çetin et al., 2013](#); [Matlani e Shrivastava, 2018](#); [Bu e Gharajeh, 2019](#)).

A detecção por cores foi uma das primeiras técnicas de detecção em vídeo e ainda é

usada em quase todos os métodos de detecção. Existem vários espaços de cores aplicados na DFV e na DFV, destacando-se o HSI/HSV (Yamagishi e Yamaguchi, 1999), YCbCr (Celik et al., 2007b), YUV (Dedeoğlu et al., 2005; Ebert e Shipley, 2009),  $L^*a^*b^*$  (Celik, 2010) e sistemas de detecção por cores mais complexos que usam técnicas baseadas em modelos de mistura de Gaussianas (Töreyn et al., 2006; Zhao et al., 2011) e modelos estatísticos do espaço de cores (Celik et al., 2007b; Borges e Izquierdo, 2010). A maioria das abordagens baseadas em cores usa o espaço de cores RGB (*Red-Green-Blue*), principalmente em razão da baixa complexidade computacional e porque quase todas as câmeras de alcance visível possuem sensores que detectam vídeo no formato RGB (Frizzi et al., 2016; Yu et al., 2013; Bortolini et al., 2014).

A análise de objetos em movimento é também amplamente usada na detecção em vídeo porque elimina os objetos estacionários, enquanto chamas e fumaça, que apresentam movimentos, são evidenciadas (Verstockt et al., 2009; Zaman et al., 2018). Essa etapa exige alta complexidade computacional e também é sensível ao ruído (Matlani e Shrivastava, 2018). Uma análise detalhada da região investigada é necessária para diferenciar objetos comuns (e.g. carros, nuvens, pessoas, etc) de fumaça e chamas (Çetin et al., 2016). Para isso, alguns métodos são utilizados:

- subtração do fundo da imagem (Celik et al., 2007a; Chen et al., 2010; Ko et al., 2010);
- diferenças temporais (Lee e Han, 2007; Yu et al., 2013);
- fluxo óptico (Gomez-Rodriguez et al., 2002; Rodríguez et al., 2003; Rinsurongkawong et al., 2012);
- modelo de mistura de Gaussianas (GMM) (Vijayalakshmi e Muruganand, 2017; Han et al., 2017);
- métodos que extraem o fundo visual (ViBE) (Barnich e Van Droogenbroeck, 2011) e SubSENSE (St-Charles et al., 2015).

A análise de cintilação pode ser aplicada usando as transformadas de Fourier e Wavelet (Yamagishi e Yamaguchi, 2000; Phillips et al., 2002; Memarzadeh e Mohammadi, 2015). As chamas têm comportamento turbulento e cintilam em chamas de fogo, logo, podem ser utilizadas para diferenciar o fogo de objetos comuns. Essa análise foca no comportamento temporal das chamas, pois *pixels* coloridos de chama aparecem e desaparecem nas bordas turbulentas das chamas com uma frequência de até 15 Hz.

A análise de diferenças espaciais de cores foca na característica das chamas de incêndio não controlado terem variações de cores mesmo dentro de uma pequena área, diferentemente de objetos comuns que apresentam cores iguais para determinadas áreas (Verstockt, 2011). No caso da fumaça, a sua densidade é desigual em toda a sua extensão

(Rasbash e Drysdale, 1982). As variações espaciais das cores nos *pixels* das regiões de fogo são observadas e diferenciadas de objetos comuns usando a análise de variância e histogramas (Borges e Izquierdo, 2010), filtros de alcance (*range filters*) (Ebert e Shipley, 2009), e a análise espacial Wavelet (Töreyn et al., 2006).

A análise dinâmica de padrões e texturas em vídeo pode ser definida como a movimentação da textura no vídeo. Embora as texturas dinâmicas sejam facilmente observadas pelos olhos humanos, são difíceis de discernir usando métodos de visão computacional, pois a localização espacial e a extensão das texturas dinâmicas podem variar com o tempo e podem ser parcialmente transparentes (Celik, 2010). Objetos comuns em vídeo, como pessoas andando, têm bordas bem definidas, apresentando um limite bastante estável ou quase estacionário ao longo do tempo. Por outro lado, regiões descontroladas de chamas e fumaça exibem contornos sem formas definidas. Alguns métodos utilizados na análise de padrões e texturas dinâmicas em vídeo estão intimamente relacionados à análise de diferenças espaciais (Hamme et al., 2010; Wang et al., 2017).

O grau de desordem do fogo e da fumaça pode ser analisado como a quantidade de *pixels* de uma chama ou fumaça entre duas imagens consecutivas (Horng e Peng, 2001). Alguns exemplos de métricas usadas frequentemente para a análise de desordem são aleatoriedade do tamanho da área (Borges et al., 2008), rugosidade dos limites (Töreyn et al., 2006) e variação da turbulência (Xiong et al., 2007; Labati et al., 2013). Verstockt et al. (2009) citam que, embora essas métricas sejam diferentes em definição, o resultado de cada uma delas é praticamente idêntico.

A técnica de *subblocking* consiste em dividir uma imagem, ou sequência de imagens, em blocos menores. Essa técnica reduz os distúrbios da medição, ou seja, filtra erros e imprecisões na medição, uma vez que se analisa o bloco e não mais o pixel (Verstockt et al., 2009; Piccinini et al., 2008; Yasmin, 2009). As imagens de entrada são subdivididas em blocos de dimensão  $n \times n$ , e um valor de bloco é calculado como a média de todos os valores de pixel no bloco. Posteriormente, análises adicionais são realizadas no nível do bloco e não mais no nível do pixel. A partir de cada bloco da imagem, técnicas como, por exemplo, reconhecimento de padrões, são utilizadas para identificar blocos positivos (com presença de fogo ou fumaça) ou blocos negativos (Gubbi et al., 2009; Lee et al., 2012; Ojo e Oladosu, 2018).

Todas as técnicas referenciadas são utilizadas na DFV e na DFV com o objetivo de filtrarem as características das imagens e vídeos. Após extrair as principais características da fumaça e do fogo, uma opção é utilizar aprendizado de máquina para processar essas características e classificá-las. Dessa maneira, o treinamento de um determinado classificador é usado para criar modelos de cores de fundo e cores de fogo que potencializam a detecção de objetos em movimento e a detecção de fogo, diminuindo ainda mais a ocorrência de falsos-positivos (Phillips et al., 2002; Celik et al., 2007a). Em Mao et al. (2018); Bhogal

e Rawat (2020) são destacadas algumas técnicas que utilizam aprendizado de máquina na DFV e na DFV.

Por outro lado, os métodos baseados em aprendizado profundo, especialmente as redes neurais convolucionais, do inglês *Convolutional Neural Network* (CNN), têm sido alternativa para detectar fogo e/ou fumaça (Yin et al., 2017; Lin et al., 2019a; NAMOZOV e CHO, 2018). Ao contrário da abordagem convencional, as CNNs são um tipo de redes neurais multicamadas que reconhecem padrões visuais diretamente dos *pixels* de imagens com o mínimo pré-processamento. Em todo o processo do sistema de detecção de incêndio, os quadros obtidos do vídeo são usados como entrada para o método baseado em CNN e o resultado da previsão é transmitido ao sistema de alarme. A maioria dos métodos existentes usa o modelo CNN sequencial, que consiste em camadas convolucionais, de amostragem descendente e totalmente conectadas (Mao et al., 2018; Jadon et al., 2019; Saeed et al., 2020; Frizzi et al., 2016; Dunning e Breckon, 2018; Zhang et al., 2016; Sharma et al., 2017; Maksymiv et al., 2017; Son et al., 2018; Yang et al., 2019; Li et al., 2020). As camadas profundamente empilhadas extraem efetivamente várias características das imagens de entrada, que alcançam alto desempenho na detecção de incêndio e reduzem os alarmes falsos.

Algumas arquiteturas, ou modelos, existentes de CNNs, como por exemplo o GoogleNet (Szegedy et al., 2015), AlexNet (Krizhevsky et al., 2017) e VGGNet (Simonyan e Zisserman, 2014) têm sido aplicados em tarefas de detecção de incêndio e fumaça (Muhammad et al., 2018a,b). Para detecção de incêndio em tempo real, Muhammad et al. (2019b) aplicaram o MobileNet (Howard et al., 2017), que tem desempenho semelhante aos outros modelos de CNN, mas reduz consideravelmente o custo computacional. Em Gaur et al. (2020) e Subbiah et al. (2020) é apresentado um levantamento detalhado das principais arquiteturas envolvendo as CNNs aplicadas à detecção de fumaça e fogo.

Após a citação dos trabalhos de grande relevância na literatura sobre a detecção de fogo e fumaça publicados nas últimas três décadas, são apresentadas duas tabelas que sintetizam as principais obras. A Tabela 1 mostra o estado da arte da detecção de fogo e da fumaça. Essa tabela foi adaptada de Çetin et al. (2013). Existem outros trabalhos que relatam estudos sobre a detecção de fogo e fumaça utilizando a extração de características: Mengxin et al. (2013); Alkhatib (2014); Mahdipour e Dadkhah (2014); Ojo e Oladosu (2014); Çetin et al. (2016); Buhus e Apatean (2017); Petra et al. (2017); Matlani e Shrivastava (2018); Bhogal e Rawat (2020) e Gaur et al. (2020). As referências destacam os métodos com foco na extração das características espaciais, temporais e também em características espectrais. Já a segunda tabela, Tabela 2, sintetiza algumas das metodologias usando as CNNs aplicadas na detecção de fogo e fumaça e apresenta também a taxa de detecção obtida por cada uma delas.



Tabela 1 – Estado da arte da detecção de fogo e da detecção de fumaça: técnicas de detecção que realizam a extração de características destacadas na literatura. Fonte: Autoria própria.

Artigo	Detecção por cor	Detec. movim.	An. cintil.	An. dif. espacial	An. padrão /textura	An. desordem	Sub- blocos	Trein.	Pós- process.	An. de local.	Detec. fogo	Detec. fumaça
Cappellini et al. (1989)		X					X	X			X	X
Healey et al. (1993)					X		X				X	
So e Chan (1994)		X				X					X	X
Noda e Ueda (1994)	RGB										X	X
Foo (1996)		X									X	
d. Breejen et al. (1998)												
Yamagishi e Yamaguchi (1999)	RGB/HSV		X			X	X	X	X	X	X	
Arrue et al. (2000)												
Yamagishi e Yamaguchi (2000)	RGB/HSV		X			X	X	X	X	X	X	
Vicente e Guillemant (2002)	RGB	X		X	X	X	X					X
Phillips et al. (2002)	RGB		X	X				X	X		X	
Gomez-Rodriguez et al. (2002)		X	X			X						X
Rodríguez et al. (2003)		X	X									X
Gómez-Rodríguez et al. (2003)												
Thou-Ho Chen et al. (2003)	RGB/HSI	X				X					X	
Thou-Ho Chen et al. (2004)	RGB/HSI	X				X					X	X
Che-Bin Liu e Ahuja (2004)	RGB/HSV	X	X			X		X			X	
Töreyn et al. (2005)	YUV	X	X			X						X
Dedeoğlu et al. (2005)	RGB/HSV/YUV	X	X								X	
Töreyn et al. (2005)	RGB/HSI	X	X	X		X		X			X	
Wen-Bing Horng et al. (2005)	RGB/HSI	X				X			X		X	
Töreyn et al. (2006)	YUV	X	X			X						X
Chen et al. (2006)	RGB/HSI					X						X

Continua na próxima página

Tabela 1 – Continua na página anterior

Töreyn et al. (2006)	RGB	X	X	X					X	
Marbach et al. (2006)	YUV	X							X	
Celik et al. (2007a)	RGB/YCbCr/HSV								X	X
Xiong et al. (2007)		X	X			X				X
Xu e Xu (2007)		X	X			X	X			X
Celik et al. (2007a)	RGB	X				X	X	X	X	
Celik et al. (2007b)	YCbCr/RGB								X	X
Töreyn e Çetin (2007)	RGB/HSI	X	X				X		X	
Lee e Han (2007)	RGB	X					X	X	X	X
Piccinini et al. (2008)	RGB	X	X				X			X
Yang et al. (2008)	RGB	X				X				X
Cui et al. (2008)			X	X			X		X	X
Horng e Peng (2008)	RGB/HHSI	X					X		X	
Yasmin (2009)	RGB/HSI	X				X	X		X	X
Kim e Wang (2009)	YUV	X				X	X			X
Ebert e Shipley (2009)	YUV/RGB/HSV		X	X				X	X	
Gubbi et al. (2009)			X				X	X		X
Chunyu et al. (2010)	RGB/HSI	X					X			X
Celik (2010)	CIE L*a*b	X				X	X	X	X	
Borges e Izquierdo (2010)	RGB			X	X	X	X		X	
Chen et al. (2010)	RGB/HSI	X	X				X		X	
Gunay et al. (2010)	RGB/HSI	X	X	X		X	X		X	
Ko et al. (2010)	RGB	X	X				X		X	
Kolesov et al. (2010)		X			X		X		X	X
Rafiee et al. (2011)										
Genovese et al. (2011)	RGB/YUV	X		X		X		X		X
Hamme et al. (2010)	HSV				X		X	X	X	

Continua na próxima página

Tabela 1 – *Continua na página anterior*

Zhao et al. (2011)	RGB/HSV	X	X		X			X		X
Lee et al. (2012)	RGB	X	X	X			X	X	X	X
Millan-Garcia et al. (2012)										
Habiboğlu et al. (2012)	RGB/HSI		X				X	X		X
Truong e Kim (2012)	RGB/CIE L*a*b	X	X		X	X		X		X
Labati et al. (2013)	RGB/YUV	X		X		X		X		X
Yu et al. (2013)	RGB/HSI	X	X			X	X	X		X
Wang et al. (2013)	RGB/HSI	X	X	X				X		X
Barpoutis et al. (2014)	HSV	X	X	X	X		X	X		X
Dimitropoulos et al. (2014)	RGB	X	X		X		X	X		X
Gomes et al. (2014)	RGB/HSI/YCbCr	X	X		X					X
Memarzadeh e Mohammadi (2015)	RGB	X	X							X
Foggia et al. (2015)	RGB/YUV	X	X			X		X		X
Chino et al. (2015)	RGB/YCbCr				X			X		X
Zaidi et al. (2015)	RGB/YCbCr	X								X
Toulouse et al. (2016)	RGB/HSI/YCbCr/ CIE L*a*b/YUV							X		X
Qureshi et al. (2016)	RGB	X	X			X			X	X
Chi et al. (2017)	RGB	X	X					X		X
Han et al. (2017)	RGB/HSI//YUV	X								X
Mahmoud e Ren (2018)	RGB/YCbCr	X								X
Gong et al. (2019)	RGB/HSI	X		X		X		X	X	X
Mahmoud e Ren (2019)	CIE L*a*b	X		X				X		X
Khondaker et al. (2020)	YUV			X						X
Torabian et al. (2021)	RGB	X		X	X			X		X
Khalil et al. (2021)	RGB/CIE L*a*b	X			X				X	X



Tabela 2 – Metodologias recentes usando redes neurais convolucionais (CNN) na detecção de fogo e fumaça.

<b>Autor</b>	<b>Metodologia</b>	<b>Acurácia/ Taxa de detec.</b>	<b>Detec. Fogo</b>	<b>Detec. Fumaça</b>
Zhang et al. (2020)	CNN (AlexNet)	99,3%		X
Hu e Lu (2018)	CNN (ConvNet)	97,0%		X
Yin et al. (2019)	CNN (6 camadas)	95,0%		X
Xu et al. (2019)	CNN (9 camadas)	98,1%		X
Luo et al. (2018)	CNN (9 camadas)	99,8%		X
Zhong et al. (2018)	CNN (9 camadas)	97,6%	X	
Filonenko et al. (2017)	CNN (4 camadas)	91,4%		X
Yuan et al. (2019)	CNN (14 camadas)	97,3%		X
Muhammad et al. (2018b)	CNN (GoogleNet)	94,4%	X	
Polednik (2015)	CNN (CaffeNet)	92,0%	X	
Zhang et al. (2016)	CNN (AlexNet)	84,8%	X	
Muhammad et al. (2018a)	CNN (AlexNet)	94,4%	X	
Yin et al. (2017)	CNN (14 camadas)	97,2%		X
NAMOZOV e CHO (2018)	CNN (12 camadas)	96,2%	X	X
Lin et al. (2019b)	CNN (13 camadas)	95,2%		X
Tao et al. (2016)	CNN (AlexNet)	99,4%		X
Frizzi et al. (2016)	CNN (9 camadas)	98,1%	X	X
Jadon et al. (2019)	CNN (14 camadas)	93,9%	X	X

Fonte: Autoria própria.

## 2.1 Comentários finais

Esse capítulo apresentou os principais trabalhos de detecção de fogo e fumaça da literatura, desde métodos que utilizam a extração de características a partir de regras simples até o uso de redes neurais convolucionais. De maneira geral, a extração das características em uma imagem ou sequência de imagens acarreta um custo para o especialista que precisa conhecer muito bem o processo de detecção, enquanto as CNN's apresentam arquiteturas complexas com uso de memória e computação.

Por mais que as metodologias usando CNNs potencializem a detecção de fogo e fumaça, como pode ser visto na Tabela 2, ainda é preciso desenvolver novos métodos para auxiliar na detecção, de tal forma, que seja possível diminuir a taxa de falsos positivos dos modelos existentes. Dando passos nessa direção, esse trabalho propõe uma abordagem para a detecção de fogo a partir de sinais de vídeos por meio da análise de séries temporais. Essa abordagem será utilizada em conjunto com um detector espacial, que utiliza aprendizagem profunda, objetivando diminuir a taxa de falsos positivos na detecção de fogo.

## Capítulo 3

# Fundamentação Teórica

Este capítulo fornece o referencial teórico necessário para a compreensão da metodologia empregada e das contribuições realizadas neste trabalho. Sendo assim, os subsídios para extrair as séries temporais dos *pixels* de um vídeo são descritos, bem como, o cálculo das diferenças temporais. Além disso, são brevemente descritos os algoritmos de aprendizado de máquina e as métricas de avaliação de desempenho que serão usados na abordagem que será proposta.

### 3.1 Representação de vídeos em séries temporais

O termo imagem refere-se a uma função de intensidade luminosa bidimensional, denotada por  $f(c, l)$ , em que o valor ou amplitude de  $f$  nas coordenadas espaciais  $(c, l)$  dá a intensidade (brilho) da imagem naquele ponto (*pixel*) (Gonzalez e Woods, 2008). Como a luz é uma forma de energia,  $f(c, l)$  deve ser positiva e finita, isto é,

$$0 < f(c, l) < \infty. \quad (3.1)$$

Uma imagem digital monocromática é formada por vários *pixels*, sua menor unidade, e pode ser definida matematicamente como uma matriz bidimensional de números reais de tamanho  $(M, N)$ :

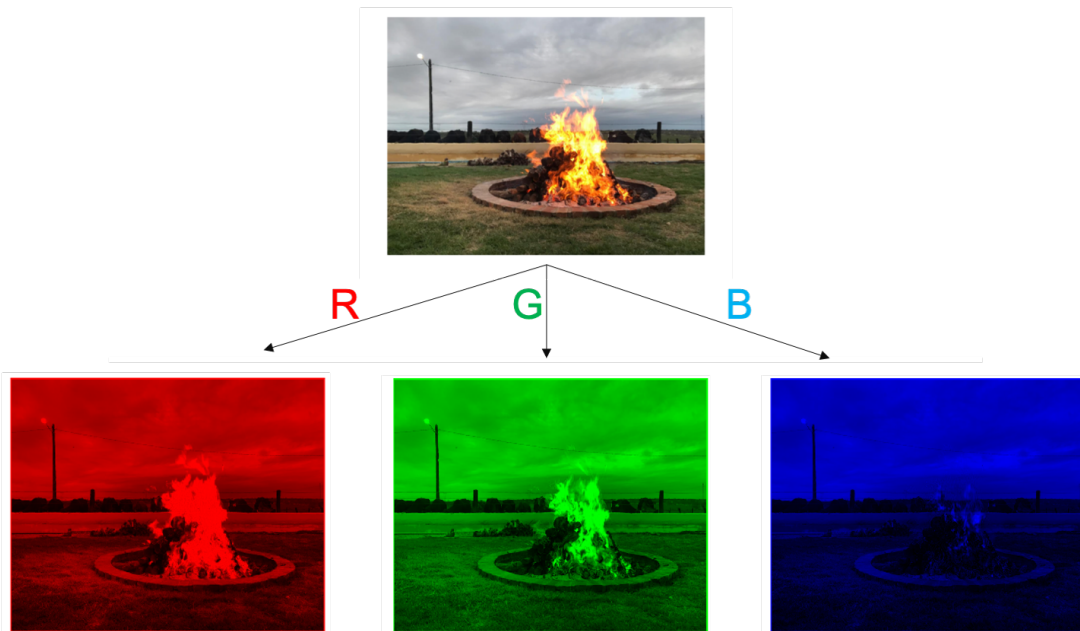
$$f(c, l) = \begin{bmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, N) \\ f(2, 1) & f(2, 2) & \cdots & f(2, N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix}, \quad (3.2)$$

sendo que  $l$  e  $c$  correspondem às coordenadas de cada *pixel* que compõe a imagem, linha e coluna, respectivamente. O valor da função bidimensional  $f(c, l)$  para um determinado *pixel* com coordenadas  $l_0, c_0$ , denotado por  $f(c_0, l_0)$ , é chamado de intensidade ou nível de cinza do *pixel* desta imagem. A intensidade pode variar de 0 a 255. No caso de imagens

coloridas no espaço de cores RGB, a representação é bem parecida com a equação 3.2, porém, existem três camadas, e cada *pixel* é representado por três intensidades de cores. As intensidades são representadas cada uma por uma cor, sendo, vermelho, verde e azul, variando de 0 a 255 para imagens de 8 bits, totalizando 24 bits.

Em uma imagem, cada *pixel* recebe um valor que corresponde à intensidade luminosa naquele ponto, referente a uma escala de tons de cinza ou brilho. No caso de imagens coloridas, cada *pixel* é formado por um conjunto de valores. É o que ocorre com imagens no padrão de cores RGB que é constituída por uma função  $f(c, l)$ , ou valor, para cada componente ou canal, R (red - vermelho), G (green - verde) e B (blue - azul), respectivamente (Gonzalez e Woods, 2008; Filho e Neto, 1999). As Figuras 5 e 6 representam uma imagem no espaço de cores RGB.

Figura 5 – Imagem de uma fogueira e sua composição pelos três canais: R, G e B.



Fonte: Autoria própria.

Considerando que a Figura 5 tem resolução igual a 640x360, ela pode ser representada por uma matriz com 360 linhas, 640 colunas e uma terceira dimensão que é a réplica da primeira matriz, sendo uma para cada componente de cor. Sendo assim, esta imagem é representada por uma matriz de dimensão  $[360, 640, 3]$ . O número total de *pixels* é igual a 230.400, e cada um deles é constituído pelas cores básicas R, G e B. Por meio de uma única imagem, não é possível representá-la usando séries temporais, pois não existe nenhuma informação temporal, isto é, a imagem é estática e contém apenas informações espaciais. Diferentemente do que se tornaria a análise de um vídeo, por este apresentar informações temporais.

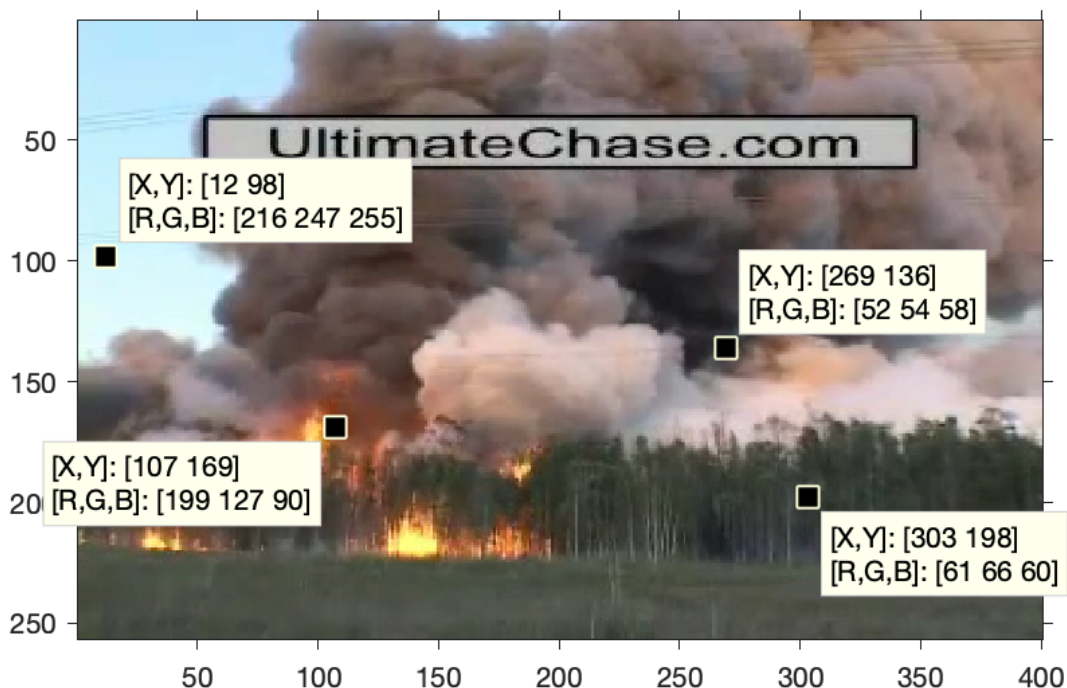
A Figura 6 mostra em detalhes alguns *pixels* estrategicamente selecionados: i)

de fogo, ii) de fumaça, iii) da mata, e iv) do céu. Esses *pixels* destacam as diferenças entre os padrões RGB em cada uma das seleções. Percebe-se que *pixels* de chama de fogo estão contidos no intervalo das cores vermelha e amarela do espaço de cores RGB, com intensidades de acordo com a seguinte regra (Thou-Ho Chen et al., 2004):

$$R > G > B. \quad (3.3)$$

No segundo caso, são destacadas as intensidades de cores RGB da fumaça e, devido a predominância da cor cinza os valores RGB são próximos  $R \approx G \approx B$ . Na terceira seleção, os valores RGB para os *pixels* da mata apresentam intensidades mais baixas e próximas entre si, o que é esperado tendo em vista a uniformidade predominante da cor verde escuro. Na última seleção percebe-se que os valores RGB do céu azul apresentam intensidades  $R < G < B$ .

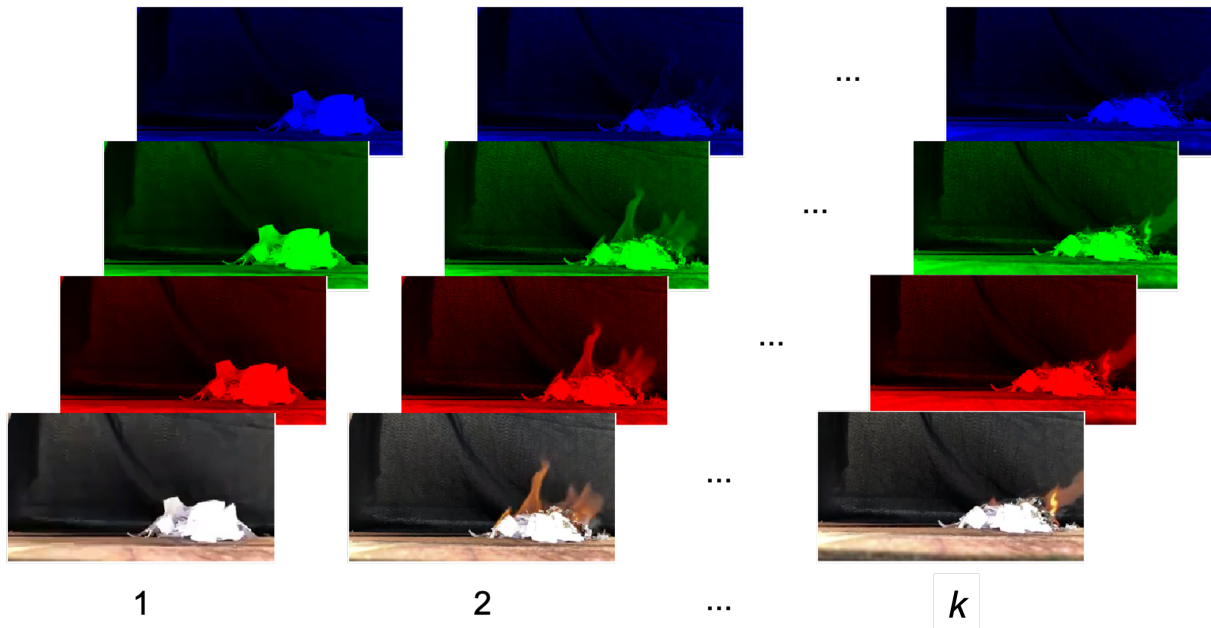
Figura 6 – Detalhes de *pixels* fogo, fumaça, mata e céu da imagem e sua composição pelas três intensidades de cores R, G e B.



Fonte: Autoria própria.

A representação de um vídeo é bem parecida com a de uma imagem. No entanto, em um vídeo são processadas várias imagens ao longo de um determinado tempo, ou seja, um vídeo é composto por diversas imagens igualmente espaçadas no tempo. Cada uma destas imagens é conhecida como quadro ou *frame*. O número de *frames* por segundo (FPS) indica a taxa do vídeo. Quanto maior essa taxa, maiores as informações temporais contidas no vídeo. A Figura 7 ilustra um vídeo com  $k$  quadros pertencentes aos espaços de cores RGB. Isto é, para cada quadro existem três informações de cores: vermelha, verde e azul (RGB).

Figura 7 – Representação de um vídeo com as componentes de cores RGB e comprimento de  $k$  quadros.



Fonte: Autoria própria.

Formalmente, uma série temporal é qualquer conjunto de observações ordenadas no tempo. Elas podem ser discretas ou contínuas (Morettin e Tolo, 2006). Dessa maneira, a representação das informações de um vídeo por meio de séries temporais é formada pelas informações das composições de cores do vídeo ao longo do tempo, isto é, ao longo dos quadros. A resolução da Figura 6 é igual a  $640 \times 360$ , resultando em 230.400 *pixels*. Existe pois, um total de 230.400 séries temporais neste vídeo, cujo comprimento temporal é  $k$ . O comprimento temporal está relacionado ao número de quadros que são exibidos por segundo (FPS). Logo, se um vídeo possui taxa de 25 FPS e tem a duração de 42 segundos, logo contém 1050 quadros. Sendo assim, a série temporal de um *pixel* pertencente às coordenadas  $c$  e  $l$  é definida por:

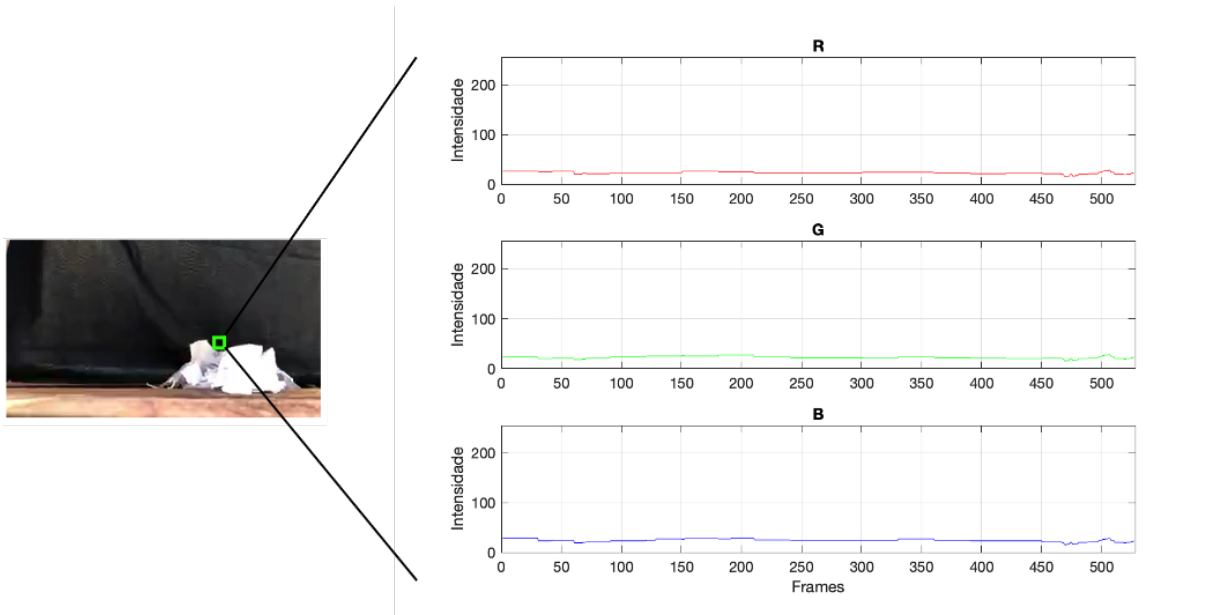
$$\mathbf{x}_{(c,l)}[k] = [f_{(c,l)}[1], f_{(c,l)}[2], \dots, f_{(c,l)}[k]]. \quad (3.4)$$

Ou seja, a série temporal de um *pixel* em um vídeo representa as intensidades de cores ao longo dos  $k$  quadros. Caso o vídeo seja composto por imagens coloridas RGB, cada amostra da série temporal  $\mathbf{x}_{(c,l)}[k]$  é representada por três intensidades, uma para cada componente de cor. Analogamente, a equação 3.4 pode ser utilizada para representar as componentes dos outros espaços de cores.

As Figuras 8 e 9 mostram quadros de dois vídeos gravados no mesmo cenário: um sem fogo e o outro com fogo, respectivamente. Os dois vídeos têm o comprimento de 525 quadros, 21 segundos, taxa de 25 FPS e resolução igual a  $640 \times 360$  *pixels*. Desta maneira, cada vídeo possui 230.400 séries temporais, contendo 525 amostras para cada

uma das componentes de cores. Considerando um *pixel* de cada vídeo na mesma posição  $(c, l) = (215, 430)$ , suas séries temporais são mostradas nas Figuras 8 e 9.

Figura 8 – Séries temporais  $x_{(215,430)} [k]$  de um *pixel* RGB em um cenário que não contém fogo. Total de  $k = 525$  frames.



Fonte: Autoria própria.

As regiões destacadas nas Figuras 8 e 9 são as mesmas para possibilitar analisar o comportamento temporal ao longo dos quadros de cada uma das séries. Na Figura 8 percebe-se que ao longo dos 525 quadros (21 segundos), praticamente não existem oscilações bruscas no espaço de cores RGB.

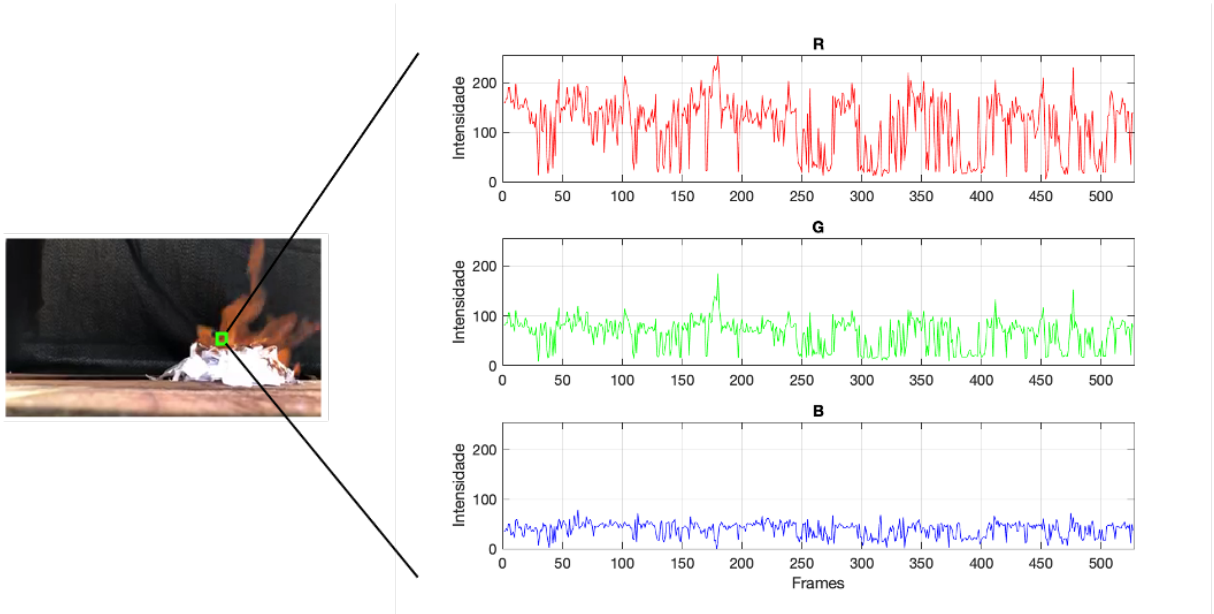
Já na Figura 9, existe fogo no cenário ao longo de todo o vídeo (21 segundos). Nota-se claramente a diferença deste conjunto de séries temporais RGB quando comparadas às séries da Figura 8. Oscilações bruscas no espaço de cores RGB acontecem devido ao movimento natural das chamas, que provocam alternância entre as cores do fogo e as cores do plano de fundo em cada um dos quadros do vídeo. É possível observar também que as intensidades das séries temporais de fogo das camadas RGB obedecem à regra definida na equação 3.3, isto é, a intensidade de cor do canal R é maior que G, que por sua vez, é maior que B.

## 3.2 Diferenças Temporais

A segmentação de vídeos baseada em movimento é uma tarefa de pré-processamento muito importante em diversas aplicações da visão computacional e do processamento de vídeos, tais como sistemas de vigilância, rastreamento de objetos, codificação de vídeos, recuperação de informações, análise de vídeos, dentre outros (Verstockt et al., 2009).



Figura 9 – Séries temporais  $x_{(430,215)} [k]$  de um *pixel* RGB em um cenário que contém fogo. Total de  $k = 525$  frames.



Fonte: Autoria própria.

Diversos trabalhos utilizam as diferenças temporais para detecção de movimento (Töreyin et al., 2006; Lee et al., 2012; Yu et al., 2013; Yang et al., 2008). No entanto, nesses trabalhos, os autores consideram a diferença entre um quadro representativo do plano de fundo e computam a diferença entre os quadros predecessores do vídeo.

O método de Diferenças Temporais para análise de movimento leva em conta o primeiro plano de cada imagem em relação à estimativa do plano de fundo. A segmentação de movimentos em vídeos resulta em um novo vídeo, com os objetos do primeiro plano destacados. Esse método é implementado comparando um quadro padrão (contendo o plano de fundo estático) e quadros obtidos no momento  $k$ . Para analisar o movimento, é computada a diferença entre o quadro padrão ( $f_{default}$ ) e cada um dos quadros em instantes  $k$  diferentes:

$$D[f(k)] = f(k) - f_{default}. \quad (3.5)$$

$D[f(k)]$  é a imagem resultante da diferença que mostra apenas alguma intensidade para as localizações dos *pixels* que foram alteradas entre os dois quadros. Para melhorar a segmentação, um valor limite é imposto nesta imagem resultante

$$[D[f(k)] - D[f(k-1)]] > t_h. \quad (3.6)$$

Isso significa que as diferenças nas intensidades de cor dos *pixels* das imagens são filtradas com base no valor limite  $t_h$ . A precisão desta abordagem depende da velocidade do movimento da cena.

De forma similar, nesta seção, é mostrada as Diferenças Temporais, representada pela primeira diferença de uma série temporal de um *pixel*, ou de uma região específica. Dessa maneira, é possível analisar em quais quadros existem as mudanças mais significativas. Essas informações podem ser úteis para analisar eventos quando existir apenas mudanças que venham a ser consideradas significativas, como, por exemplo, fogo, fumaça ou movimento atípico em uma região específica.

Analogamente à detecção de movimento em imagens de um vídeo, o método de Diferenças Temporais aplicado nas séries temporais resulta em séries de diferenças que apresentam valores iguais a zero para objetos estáticos e valores diferentes de zero para objetos que se movimentam ao longo dos quadros de um vídeo. A Diferença Temporal de uma série temporal de um determinado *pixel* localizado na posição  $(c, l)$  é dada por

$$\Delta(c, l) = x_{(c,l)}[k] - x_{(c,l)}[k - 1]. \quad (3.7)$$

Diversos trabalhos usam a componente R do espaço de cores RGB ou a luminância Y do espaço YUV para detecção de movimento em vídeos (Ebert e Shipley, 2009; Qiu et al., 2012). O uso de uma componente específica, como por exemplo a componente R ou a Y, se dá por ela apresentar maior intensidade nas regiões de fogo/chama quando comparada as outras componentes que formam os seus respectivos espaços de cores Foggia et al. (2015); Han et al. (2017).

### 3.3 Aprendizado de máquina

A equação 3.4 permite extrair séries temporais de regiões de interesse em vídeos e, a partir daí, compor um banco de dados que, por sua vez, pode ser utilizado para classificar algum evento de interesse, como, por exemplo, o fogo.

O aprendizado de máquina, ou um algoritmo de aprendizado de máquina, ou ainda o termo em inglês *machine learning*, é um processo computacional que usa dados de entrada para realizar previsões ou classificações. Esses algoritmos são capazes de alterar ou adaptar automaticamente sua arquitetura por meio da repetição para que se tornem cada vez melhores na realização da tarefa desejada. O processo de adaptação é chamado de treinamento, no qual amostras de dados de entrada são fornecidas juntamente com os resultados desejados. O algoritmo então se configura de forma otimizada para que possa, não apenas produzir o resultado desejado quando apresentado com as entradas de treinamento, mas também generalizar para produzir o resultado almejado a partir de dados novos e não vistos anteriormente. Esse treinamento é a parte de “aprendizagem” do aprendizado de máquina. O treinamento não se limita a uma adaptação inicial e pode se aperfeiçoar a cada conjunto de novos dados apresentados para o algoritmo (Bonaccorso, 2017; Molnar, 2020; Rättsch, 2004). Existem muitas maneiras pelas quais um algoritmo computacional pode se adaptar em resposta ao treinamento (Carleo et al., 2019):



- os dados de entrada podem ser selecionados e ponderados para fornecer os resultados mais decisivos;
- o algoritmo pode ter parâmetros numéricos variáveis que são ajustados por meio de otimização iterativa;
- o algoritmo pode organizar uma rede de possíveis caminhos computacionais para obter resultados ótimos;
- o algoritmo pode determinar distribuições de probabilidade a partir dos dados de entrada e usá-los para prever resultados.

O algoritmo pode aprender a reconhecer os padrões de maneira supervisionada ou não supervisionada. Na aprendizagem supervisionada, deve, por óbvio, existir um supervisor, que é dado pelo registro dos valores das variáveis de saída, ou variáveis-objetivo, que são as variáveis que se deseja prever a partir dos dados existentes. Essas variáveis devem ser escolhidas de modo a representar a resposta para algum problema que se deseja resolver (Sathya e Abraham, 2013). Como resultado, obtém-se um modelo que descreva o conjunto de dados utilizados e espera-se que ele permita prever o comportamento da saída para novas entradas. Os principais algoritmos de aprendizagem supervisionada criam modelos que permitem a classificação, para variáveis categóricas, e a regressão, para variáveis contínuas. A classificação consiste em atribuir um rótulo para a saída a partir de determinada entrada.

O algoritmo de aprendizagem não-supervisionado encontra seu próprio caminho na fase de treinamento a partir dos dados de entrada. O aprendizado não-supervisionado não necessita que os dados usados para treino conttenham os rótulos ou classificação. Por meio dele, busca-se encontrar o padrão entre as diferentes amostras e separar as que possuem as mesmas características (El Naqa et al., 2015).

Um terceiro tipo de aprendizado de máquina é o aprendizado semi-supervisionado, em que parte dos dados é rotulada e outras partes não o são. Nesse cenário, variáveis rotuladas podem ser usadas para auxiliar no aprendizado de variáveis não rotuladas (Molnar, 2020).

Para avaliar o modelo obtido, normalmente se separa uma parte dos exemplos existentes para realizar testes, de preferência não os utilizando durante o processo de treinamento para evitar o sobre-ajuste, ou *overfitting*. Esse fenômeno ocorre quando o modelo descreve muito bem o conjunto utilizado para treinamento, mas tem baixa capacidade de prever novas entradas. É importante ressaltar que os dados, de forma inerente, apresentam desvios causados por fatores aleatórios que os algoritmos devem minimizar a influência desses desvios na geração dos modelos.

Levando-se em consideração o desenvolvimento deste trabalho, as séries temporais que são extraídas de vídeos, exibidas na equação 3.4, são classificadas como pertencentes às regiões de fogo ou não pertencentes a esta região. Para treinar o modelo, são usadas milhares de séries temporais rotuladas como “fogo” e “não-fogo”. A partir das características das séries temporais classificadas como “fogo”, o algoritmo aprende a rotular o que é “fogo” e “não-fogo”. Esse tipo de modelo é denominado de classificação. Esta é uma aplicação que pode ser feita com modelos supervisionados. Neste trabalho, para classificar as séries temporais extraídas de vídeos, são usados seis classificadores diferentes:

- Naïve;
- regressão logística;
- árvore de decisão;
- análise discriminante linear;
- máquina de vetores de suporte (SVM);
- $K$ -vizinhos mais próximos.

Os classificadores utilizados para os testes são brevemente descritos e referenciados, como também a metodologia de validação utilizada.

### 3.3.1 Árvore de decisões

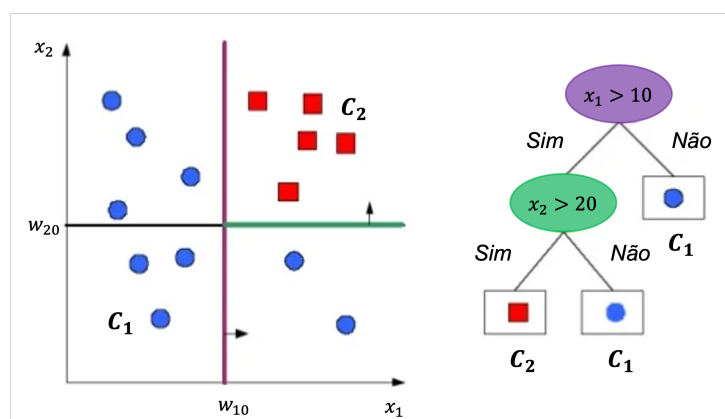
A árvore de decisões é um algoritmo de aprendizado supervisionado que pode ser usado para classificar dados com base em um modelo construído em dados de treinamento (Su e Zhang, 2006).

A árvore de decisões consiste em um mapa dos possíveis resultados de uma série de escolhas. Essas escolhas estão relacionadas a diversas decisões tomadas com base no banco de dados utilizado. Sua representação consiste em três partes principais (Safavian e Landgrebe, 1991):

- Raiz: é onde a primeira decisão deve ser tomada. A partir dela o algoritmo é direcionado a outras decisões, de acordo com a escolha realizada nessa questão.
- Nós: são todas as decisões apresentadas na árvore.
- Folhas: local onde a classificação é realizada, ou seja, são os resultados da árvore de decisões.

Sendo assim, nesta estrutura, cada nó interno corresponde a um teste sobre um dado atributo, cada ramo descendente representa uma possibilidade para esse teste, e cada folha, contendo a classe respectiva às instâncias por ela classificadas, é a decisão obtida após testar os atributos de forma sequencial. O caminho percorrido ao longo dos raiz e nós, corresponde a uma regra de classificação, resultando em uma classe.

Figura 10 – Árvore de decisões composta pela raiz, nó e folhas. Com base nos dados apresentados a árvore é capaz de classificar se uma amostra pertence à classe  $C_1$  ou  $C_2$ .



Fonte: Adaptado de [Safavian e Landgrebe \(1991\)](#).

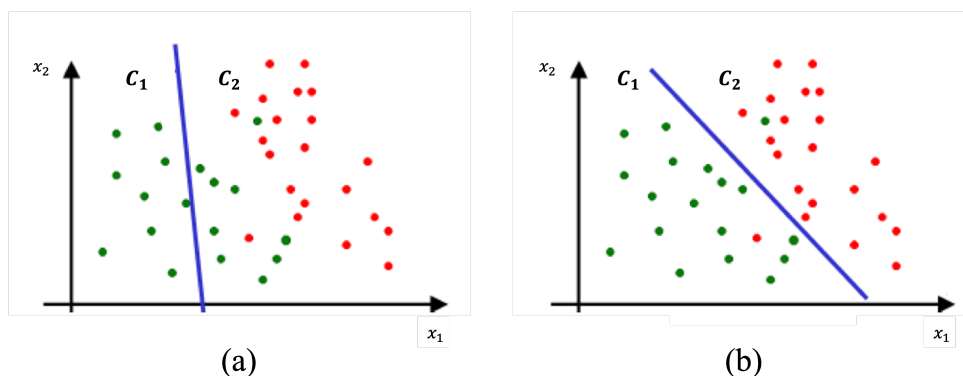
A Figura 10 ilustra uma árvore de decisões. A primeira condição é considerada a raiz. A segunda condição e, caso haja outras condições, são definidas como nós. Caso as condições sejam satisfeitas, o dado apresentado a esse classificador pertence à classe  $C_1$  ou  $C_2$ , que são denominadas como folhas.

O uso das Árvore de Decisões na detecção de fogo pode ser visto em [Stojanova et al. \(2006\)](#); [Maksimović e Vujovic \(2013\)](#); [Suklabaidya e Das \(2021\)](#) e [Abid \(2020\)](#).

### 3.3.2 Análise Discriminante Linear (LDA)

A análise discriminante é uma técnica da estatística multivariada utilizada para discriminar e classificar objetos. Segundo [Li e Yuan \(2005\)](#) é uma técnica da estatística multivariada que estuda a separação de objetos de uma população em duas ou mais classes. A discriminação ou separação é a primeira etapa, sendo a parte exploratória da análise e consiste em se procurar características capazes de serem utilizadas para alocar objetos em diferentes grupos previamente definidos. A classificação ou alocação pode ser definida como um conjunto de regras que serão usadas para alocar novos objetos ([Johnson et al., 2014](#)). O problema da discriminação entre dois ou mais grupos, visando posterior classificação, foi inicialmente abordado por [Fisher \(1936\)](#), enquanto que, [Wei et al. \(2013\)](#) utilizam esta técnica para realizar a detecção de fogo.

Figura 11 – Classes,  $C_1$  e  $C_2$ , separadas por uma função discriminante linear. (a) Função discriminante linear com alto erro e (b) função discriminante linear otimizada com o erro minimizado.



Fonte: Adaptado de Johnson et al. (2014).

Conforme está representado na Figura 11, a análise de discriminante linear consiste em obter funções matemáticas capazes de classificar uma observação  $x$  em uma das classes  $C_1$  ou  $C_2$ , com base em características observadas. Busca-se minimizar a probabilidade de má classificação, isto é, minimizar a probabilidade de classificar erroneamente uma observação em uma classe, quando realmente pertence a outra.

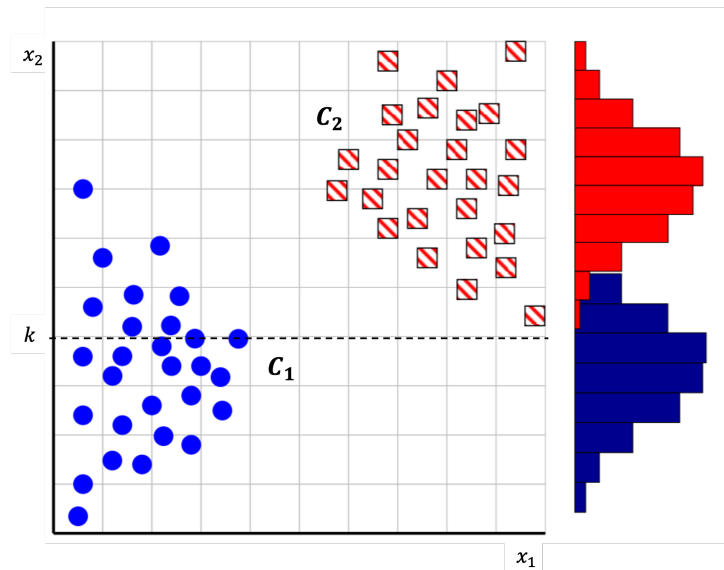
### 3.3.3 Classificador Naïve

O classificador Naïve Bayes é um classificador supervisionado probabilístico simples baseado na aplicação do teorema de Bayes (da estatística Bayesiana) com suposições de independência fortes. Ele descreve a probabilidade de um evento, baseado em um conhecimento prévio que pode estar relacionado ao evento. O classificador é denominado ingênuo (*naïve*) por assumir que as características são condicionalmente independentes, ou seja, que a informação de um evento não é informativa sobre nenhum outro (Koller e Friedman, 2009).

O teorema de Bayes é usado para determinar a probabilidade de uma hipótese quando o conhecimento prévio está disponível. Depende de probabilidades condicionais. Dados dois eventos A e B, o teorema de Bayes afirma que

$$P(C_1|C_2) = \frac{P(C_2|C_1) \times P(C_1)}{P(C_2)}, \quad (3.8)$$

dado que  $P(C_1|C_2)$  é a probabilidade a *posteriori*, ou seja, é a probabilidade do evento  $C_1$  ocorrer dado que o evento  $C_2$  ocorreu.  $P(C_2|C_1)$  é a probabilidade de verossimilhança, ou seja, do evento  $C_2$  ocorrer dado o evento  $C_1$ .  $P(C_1)$  é a probabilidade a *priori* e  $P(C_2)$  é a probabilidade marginal.

Figura 12 – Classificação usando Naïve para separar duas classes  $C_1$  ou  $C_2$ .

Fonte: Adaptado de [Koller e Friedman \(2009\)](#).

Considere duas classes,  $C_1$  e  $C_2$ . Ao estudar as características de várias amostras dessas classes, é possível discernir alguns padrões e probabilidades.

O classificador Naïve Bayes é uma técnica para realizar tal classificação. O comprimento da antena é um recurso usado para classificar um inseto em uma das duas classes. Suponha que para um valor de  $x_2$  que seja menor que um determinado valor  $k$ , o classificador Naïve classifica essa amostra por ela apresentar a possibilidade maior de pertencer à classe  $C_1$  do que a  $C_2$ , por exemplo (Figura 12). O uso do classificador Naïve aplicado na detecção de fogo pode ser encontrado em [Krstinić et al. \(2009\)](#); [Borges e Izquierdo \(2010\)](#) e [Maheen e Aneesh \(2019\)](#).

### 3.3.4 Regressão logística

A Regressão Logística tem sido utilizada na estatística há muitos anos e, recentemente, tornou-se objeto de estudo da comunidade de aprendizagem de máquina ([Hosmer e Lemeshow, 2000](#)). É uma técnica recomendada para situações em que a variável dependente é de natureza dicotômica ou binária. Quanto às variáveis independentes, tanto podem ser categóricas ou não. A regressão logística é um recurso que permite estimar a probabilidade associada à ocorrência de determinado evento em face de um conjunto de variáveis explanatórias.

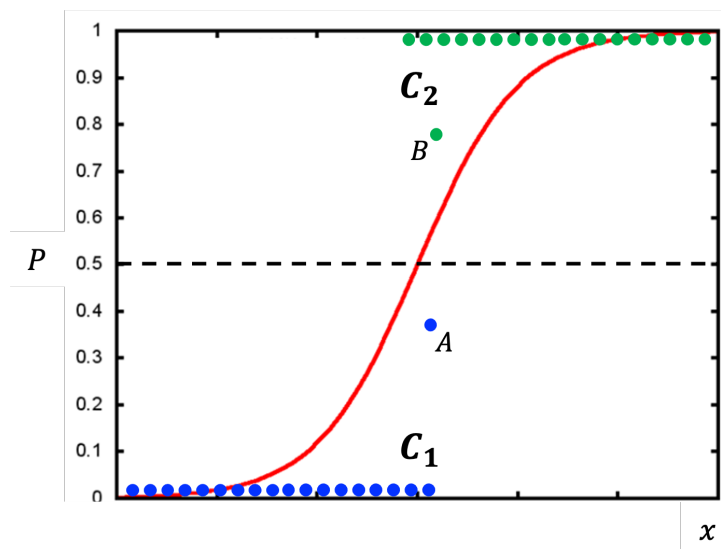
O modelo simplificado de regressão logística, no qual se tem a probabilidade de sucesso dada por:

$$P(x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x))}, \quad (3.9)$$

as probabilidade  $P$  vão resultar em valores entre 0 e 1. Os coeficientes  $\beta_0$  e  $\beta_1$  são estimados

a partir do conjunto de dados, pelo método da verossimilhança, em que se encontra uma combinação de coeficientes que maximiza a probabilidade de a amostra ter sido observada. A curva logística tem um comportamento probabilístico no formato da letra “S”, o que é uma característica da regressão logística (Figura 13).

Figura 13 – Regressão logística separando duas classes  $C_1$  ou  $C_2$ . Em função do valor limiar indicado pela linha tracejada, as amostras representadas pelo ponto A e B são classificadas como pertencentes à classe  $C_1$  e  $C_2$ , respectivamente.



Fonte: Adaptado de Hosmer e Lemeshow (2000).

O uso da classificação utilizando a regressão logística aplicada na detecção de fogo pode ser vista com mais detalhes em Chang et al. (2013); Catry et al. (2009) e De Vasconcelos et al. (2001).

### 3.3.5 $K$ -vizinhos mais próximos ( $Knn$ )

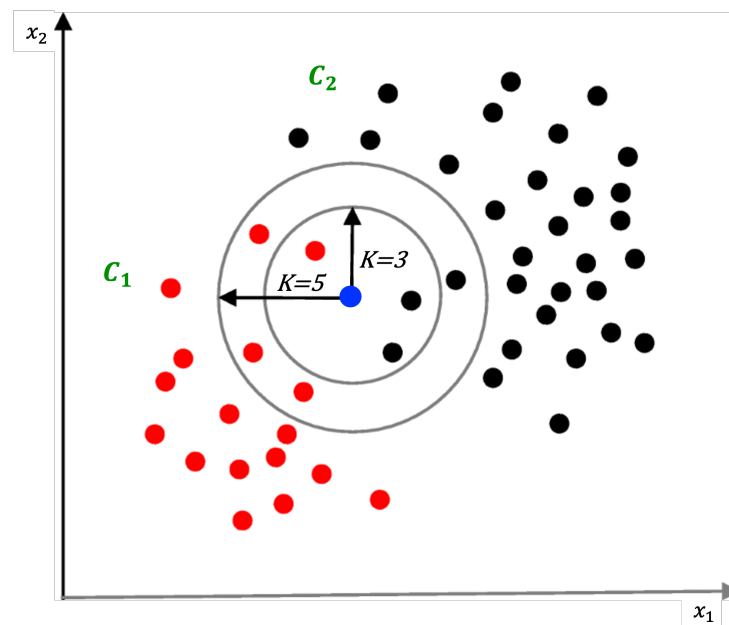
O algoritmo  $K$ -Nearest Neighbors ( $Knn$ ) é um dos algoritmos mais populares para reconhecimento de padrões (Suguna e K.Thanushkodi, 2010). É um algoritmo não paramétrico, sendo primeiramente descrito por E. Fix e L. L. Hodges em 1951 conforme citado em Abbott (2014). A categoria em que o método se insere é chamada de métodos de aprendizado baseados em instâncias. Esses métodos diferem de outros utilizados para aprendizado de máquina porque não constroem modelos que representam o conjunto de treinamento durante o aprendizado, mas simplesmente armazenam os dados para serem consultados na etapa de classificação.

$Knn$  é um algoritmo que classifica novos dados com base em uma medida de similaridade entre seus “vizinhos” mais próximos, ou seja, aqueles que têm características semelhantes às suas. Ele pressupõe que itens semelhantes estão próximos uns dos outros, então, tenta encaixar o dado em questão nos conjuntos de seus vizinhos. O valor de “ $K$ ” é

o número de vizinhos mais próximos considerados para a amostra poder ser classificada. Quanto menor a distância entre os vizinhos mais próximos, maior a chance de serem mais semelhantes. A seleção de valores diferentes para  $K$  pode gerar resultados de classificação diferentes para o mesmo objeto de amostra.

A Figura 14 apresenta uma ilustração de como o algoritmo  $K$ nn funciona para classificar uma nova amostra. Para  $K = 3$ , a nova amostra é classificada como  $C_1$ ; porém, foi classificada como  $C_2$  quando  $K = 5$ .

Figura 14 – Algoritmo  $K$ nn. Quando  $k = 3$ , a amostra azul (ao centro dos círculos) é classificada como pertencente à classe  $C_2$ , pois recebe mais votos dessa classe. Já, quando  $K = 5$ , a mesma amostra azul é classificada como  $C_1$ , pois se assemelha a mais amostras desta classe.



Fonte: Adaptado de [Abbott \(2014\)](#).

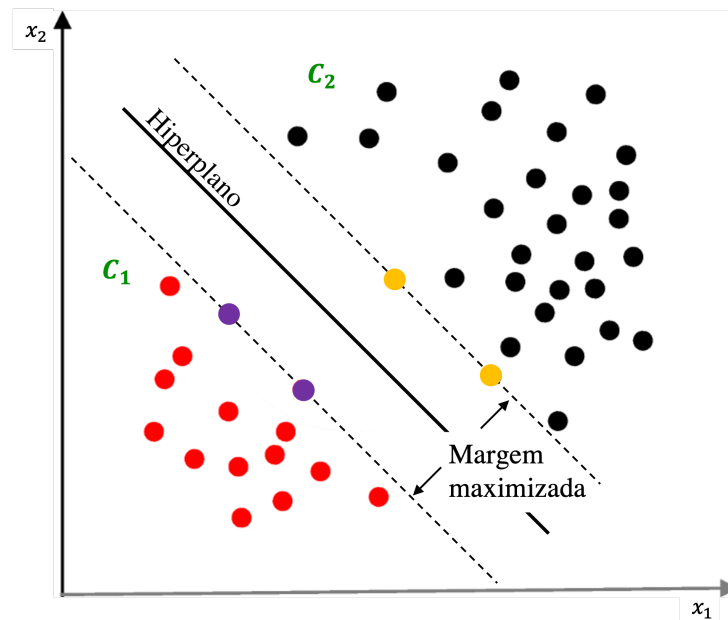
[Yuan et al. \(2014\)](#); [Zivkovic e Van der Heijden \(2006\)](#) e [Wu et al. \(2019\)](#) utilizam o classificador  $K$ nn para detectarem fogo.

### 3.3.6 Máquinas de Vetores Suporte (SVM)

As Máquinas de Vetores de Suporte, do inglês *Support Vector Machine* (SVM), também conhecido como Máquina de Suporte Vetorial, foi elaborado com o estudo proposto por [Boser et al. \(1992\)](#). Trata-se de um algoritmo de aprendizado supervisionado, cujo objetivo é classificar determinado conjunto de pontos de dados que são mapeados para um espaço de características multidimensionais usando uma função kernel, que por sua vez, é uma abordagem utilizada para classificar problemas. Nessa abordagem, o limite de decisão no espaço de entrada é representado por um hiperplano em dimensão superior no espaço ([Drucker et al., 1997](#); [Saradhi et al., 2005](#)).

De maneira geral, o objetivo do SVM é encontrar uma linha de separação, mais comumente chamada de hiperplano entre dados de duas classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes (Figura 15). Os vetores de suporte são pontos de dados que estão mais próximos do hiperplano e influenciam sua posição e orientação. Por meio desses vetores de suporte, é possível maximizar a margem do classificador, na medida em que a exclusão dos vetores de suporte alterará a posição do hiperplano. Esses são os pontos que dão suporte, ou seja, ajudam a criar um modelo de SVM.

Figura 15 – Hiperplano de uma máquina de suporte vetorial (SVM). Os vetores de suporte são pontos de dados que estão mais próximos do hiperplano e influenciam sua posição e orientação. O uso desses vetores de suporte maximiza a margem do classificador, na medida em que a exclusão dos vetores de suporte alterará a posição do hiperplano.



Fonte: Adaptado de [Saradhi et al. \(2005\)](#).

O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte. Isso faz do SVM um classificador linear binário não probabilístico. Dado um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra. Na prática, um algoritmo de otimização é usado para encontrar os valores dos coeficientes que maximizam a margem ([Saradhi et al., 2005](#)).

Exemplos do uso da detecção de fogo por meio do classificador SVM podem ser vistos em [Ko et al. \(2009\)](#); [Zhao et al. \(2011\)](#); [Mahmoud e Ren \(2019\)](#); [Russo et al. \(2018\)](#) e [Suklabaidya e Das \(2021\)](#).



### 3.4 Validação cruzada

Normalmente, antes de treinar um modelo, separa-se de forma aleatória o conjunto de dados em treinamento e validação, gerando incerteza estatística intrínseca ao processo aleatório de composição dos conjuntos. Uma alternativa para minimizar esse problema são procedimentos baseados no princípio de repetição, em que o modelo é treinado e avaliado em várias divisões arbitrárias da base de dados. Kohavi (1995) apresenta algumas formas de particionamento dos dados, dentre eles o método *k-fold*.

A validação cruzada *K-fold* (*K-fold cross-validation*) é uma técnica para avaliar a capacidade de generalização de um modelo a partir de um conjunto de dados de treinamento. Busca-se estimar o quão preciso é, na prática, esse modelo treinado, ou seja, o seu desempenho para um novo conjunto de dados, chamado de conjunto de dados de teste.

O método de validação cruzada *k-fold* consiste em dividir o conjunto total de dados em  $k$  subconjuntos, mutuamente exclusivos, do mesmo tamanho e, a partir daí, um subconjunto é utilizado para validação e os  $k - 1$  restantes são utilizados para estimação dos parâmetros, fazendo-se o cálculo da acurácia do modelo. Esse processo é realizado  $k$  vezes até que todo o conjunto passe pelo processo de treino e validação gerando uma acurácia de avaliação média para o modelo. A Figura 16 ilustra o esquema realizado pelo método *k-fold*.

Figura 16 – Esquema do método de validação cruzada *k-fold* com  $k = 5$ . O conjunto de dados é dividido em  $k$  partes iguais.



Fonte: Adaptado de Venâncio (2021b).

### 3.5 Aprendizado de máquina profundo - YOLO

Aprendizado profundo é um subconjunto do aprendizado de máquina que, por sua vez, é derivado da grande área Inteligência Computacional (Hunt, 2014). O processo de

aprendizado é profundo porque a estrutura das redes neurais artificiais consiste em várias camadas: de entrada, saída e oculta. Cada camada contém unidades que transformam os dados de entrada em informações que a próxima camada pode usar para executar uma determinada tarefa de detecção, por exemplo (Ongsulee, 2017).

YOLO (*You Only Look Once*) é um método de detecção de objetos de passada única (*single pass*), que utiliza aprendizado profundo para extrair as características (*features*) espaciais de uma imagem. O YOLO é um algoritmo bastante utilizado na detecção de objetos (Redmon et al., 2016). A ideia fundamental do YOLO assume que, se uma rede possui informações suficientes para determinar se um objeto está ou não presente em uma imagem, é plausível acreditar que a rede também possui informações para encontrar a localização do objeto na imagem (Muhammad et al., 2018a; Govil et al., 2020; Huang et al., 2016).

O algoritmo YOLO usa uma única rede neural convolucional profunda (CNN), cuja arquitetura é chamada de Darknet, para prever simultaneamente probabilidades das classes e caixas delimitadoras em imagens. A versão original do YOLO foi proposta por Redmon et al. (2016), posteriormente aprimorada em outras duas versões (YOLOv2 e YOLOv3) de código aberto desenvolvido na linguagem C. Posteriormente, Bochkovskiy et al. (2020) desenvolveram a quarta versão do YOLO (YOLOv4). Outras versões do YOLO podem ser vistas em Muhammad et al. (2019a) e Jadon et al. (2019).

A Figura 17 sintetiza o funcionamento do algoritmo YOLO. Primeiramente o YOLO redimensiona a imagem de entrada para  $N \times N$  *pixels* e a fragmenta em uma grade de células de dimensão  $S \times S$  não sobrepostas. Cada uma das  $S^2$  células é responsável por fazer a predição de  $B$  caixas delimitadoras de objetos, juntamente com pontuações de confiança para cada caixa  $C$  probabilidades de classe. O valor de confiança para a caixa delimitadora e a predição da classe são combinados em uma pontuação final, que indicará a probabilidade dessa caixa conter um objeto específico. Cada célula é responsável por detectar a existência de apenas um objeto. Se o centro desse objeto estiver dentro da célula, essa célula é responsável por detectá-lo. O algoritmo YOLO, que é usado neste trabalho, foi proposto em Bochkovskiy et al. (2020) e, em Venâncio (2021a), é configurado para detectar duas classes de interesse (fogo e fumaça), sendo assim, cada célula  $i$  representada por:

$$\mathbf{c}_i = [c_{i1}, c_{i2}, \mathbf{b}_{i1}, \dots, \mathbf{b}_{iB}], \quad (3.10)$$

sendo  $j$ -th caixa delimitadora da  $i$ -th célula representada por  $b_{ij} = [p_{ij}, x_{ij}, y_{ij}, h_{ij}, w_{ij}]$ , em que  $p_{ij}$  é a probabilidade da caixa  $j$  conter um objeto,  $x_{ij}$  e  $y_{ij}$  são as coordenadas do centro da caixa  $j$ ,  $h_{ij}$  e  $w_{ij}$  são, respectivamente, a altura e a largura da caixa  $j$ , e  $c_{i1}$  e  $c_{i2}$  são as probabilidades de o objeto dentro da célula  $i$  ser da classe 1 (fumaça) ou 2 (fogo), respectivamente.

Se mais de um valor de confiança estiver acima de um certo limiar  $t_{conf}$ , a caixa

delimitadora será definida conforme o maior valor de confiança  $s_{ij}$ , bem como a classe será definida conforme o maior valor de probabilidade de classe  $c_i$  (Redmon et al., 2016). Todavia, se o número de células  $S^2$  for suficientemente grande, as caixas delimitadoras de células diferentes estarão muito próximas umas das outras e poderão ser atribuídas ao mesmo objeto.

De modo a evitar esse cenário, o YOLO emprega uma técnica conhecida como supressão não-máxima (NMS - *non-maximum suppression*) após a saída da rede. Essa operação seleciona a caixa com maior probabilidade de detecção  $p_{ij}$  e elimina as caixas atribuídas ao mesmo objeto que mais a sobrepõem. Desconsiderando a caixa selecionada anteriormente, o processo se repete até que todas as caixas restantes sejam processadas.

Figura 17 – Processo de detecção do algoritmo YOLO: a imagem de entrada é dividida em uma grade de células  $S \times S$  (à esquerda). O modelo atribui uma confiança acima de  $t_{conf}$  às caixas (ao centro). Como ainda são muitas caixas para um mesmo objeto, aquela com maior confiança é selecionada por meio da supressão não-máxima (à direita).



Fonte: Adaptado de Saponara et al. (2021).

## 3.6 Avaliação de desempenho

Uma tarefa importante é a de quantificar a qualidade dos modelos de aprendizado de máquina obtidos. Esta tarefa é essencial na obtenção da generalização desejável. A escolha de métricas quantitativas, que permitam representar o real potencial dos modelos propostos, é imprescindível (Castro e Braga, 2011; Venâncio, 2021b).

O desempenho dos modelos propostos neste trabalho é avaliado pela capacidade de esses modelos atribuírem as classes corretas às séries temporais. As medidas frequentemente usadas para esta tarefa são descritas abaixo e são detalhadas em Hossin e M.N (2015).

### 3.6.1 Acurácia

Em geral, a métrica usada na avaliação e seleção de modelos de classificação é a acurácia (ou taxa de acerto) estimada em relação a um dado conjunto de teste. A acurácia

consiste em comparar os valores de saída obtidos com o modelo treinado, utilizando os exemplos de teste como entrada, com os valores de saída existentes nesses exemplos. A taxa de acerto, ou *Acurácia*, diz quanto o modelo que está sendo avaliado acertou das previsões possíveis, e é dada por:

$$Acc = \frac{VP + VN}{VP + FP + FN + VN}, \quad (3.11)$$

sendo VP o Verdadeiro Positivo, que é a série temporal detectada como "fogo" e é realmente "fogo". VN é o Verdadeiro Negativo, ou seja, é a série temporal que não é fogo e é classificada como "não-fogo". FP, Falso Positivo, é a série temporal classificada como "fogo", mas ela não é fogo.

### 3.6.2 Medida $F_1$ ( $F_1$ -score)

A medida  $F_1$  é uma média harmônica calculada com base na precisão e na revocação. Pode ser obtida com base na equação:

$$F_1 = 2 \times \frac{r \times p}{r + p}, \quad (3.12)$$

sendo que,  $r$  é conhecido como revogação (ou *recall*). Essa métrica é a taxa de séries temporais detectadas corretamente dentre todas as séries de interesse, ou seja, ela avalia a capacidade do método de detectar, com sucesso, resultados classificados como positivos, sendo definida como:

$$r = \frac{VP}{VP + FN}. \quad (3.13)$$

Por outro lado, a precisão ( $p$ ) é a proporção de objetos detectados corretamente dentre todos os objetos detectados pelo modelo. Pode-se dizer ainda que, por meio da precisão, se avalia a quantidade de séries temporais positivas sobre a soma de todos os valores positivos:

$$p = \frac{VP}{VP + FP}. \quad (3.14)$$

É interessante ressaltar que uma média simples não é recomendada nesse contexto, visto que não penaliza valores extremos. Por exemplo, um modelo com precisão 0 e revocação 1 teria média 0,5. Já o  $F_1$  para esse mesmo caso é 0.

## 3.7 Comentários finais

Este capítulo apresentou os conceitos essenciais para que seja possível entender a metodologia que será apresentada no Capítulo 4, composta por um detector espacial e por uma análise de séries temporais. Para isso, princípios básicos do algoritmo de aprendizado profundo, o YOLO, foram introduzidos. Também foram discutidos os classificadores de

aprendizado de máquina que serão utilizados para classificar as séries temporais das regiões de interesse pré-selecionadas pelo YOLOv4. Estes conceitos serão fundamentais para a interpretação dos resultados que serão explicitados no Capítulo 5.

Vale ressaltar que, neste trabalho, não contempla comparar as diversas arquiteturas existentes usadas para detectar as características espaciais. Detectores espaciais baseados em CNN têm alto desempenho na detecção de incêndio (Schmidhuber, 2015), mas por sua própria natureza, eles são incapazes de detectar padrões temporais (Gaur et al., 2020). Nesse sentido, este trabalho utiliza a rede YOLOv4 parametrizada e treinada em (Venâncio et al., 2021). Os autores compararam outras arquiteturas, mas o YOLOv4 mostrou eficiência e foi capaz de detectar todos os eventos positivos de incêndio, atingindo 100% da taxa de verdadeiro positivo, considerando o conjunto de dados do sistema *Apaga o Fogo* (CEMIG, UFMG, Gaia, RaroLabs e UFVJM, 2021).

# Capítulo 4

## Metodologia

Neste capítulo são descritos os conjuntos de dados dos vídeos empregados nas etapas de treinamento, validação e teste dos classificadores utilizando aprendizado de máquina. Logo em seguida, é apresentada a metodologia utilizada para detecção de fogo em sinais de vídeo por meio da análise de séries temporais. Para isso, são necessárias duas etapas. Na primeira, é realizada a detecção a partir de características espaciais de uma região de interesse onde possivelmente pode haver fogo. Na segunda etapa, é proposto um procedimento para confirmar se realmente há fogo na região de interesse a partir da classificação das séries temporais pertencentes a essa região.

### 4.1 Conjunto de dados

Como será visto, o sistema de detecção de incêndio proposto consiste em um estágio de extração de características estáticas e um estágio de extração de características dinâmicas. Por isso, requer dois conjuntos de dados uma para cada estágio. A primeira base de dados é composta por imagens e a outra composta por vídeos.

O estágio de detecção espacial utiliza aprendizado profundo e foi proposto em [Venâncio \(2021a\)](#) e [Venâncio \(2021b\)](#). Para treinar e validar esse modelo, o autor utilizou a base de dados D-Fire ([Gaia, solutions on demand, 2018](#)). Essa base de dados foi projetada especificadamente para tarefas de classificação e localização de incêndios.

A base de dados D-Fire é composta por 21.527 imagens com seus respectivos rótulos, ou seja, identificadores de classe e coordenadas normalizadas da caixa delimitadora. A função desta caixa é delimitar a região onde há eventos de fogo e/ou fumaça. Ao todo, 1.164 imagens contêm apenas fogo, 5.867 contêm apenas fumaça, 4.658 contêm fogo e fumaça e 9.838 contêm exemplos negativos, ou seja, nem fogo nem fumaça. A maioria das imagens tem mais de uma ocorrência de uma determinada classe. Em imagens com eventos de fumaça, há uma média de 1,13 caixas delimitadoras rotuladas como fumaça por imagem. Nas imagens que possuem eventos de fogo, por sua vez, há uma média de 2,52

caixas delimitadoras rotuladas como incêndio por imagem. Essa diferença ocorre porque a coleta de imagens priorizou incêndios de média e longa distância, em que é possível visualizar vários exemplos com múltiplos focos de incêndio. No total, existem 11.865 caixas delimitadoras rotuladas como fumaça e 14.692 rotuladas como fogo.

Em relação ao segundo estágio, análise de séries temporais, foi necessária uma busca exaustiva na internet por vídeos contendo fogo. Geralmente, coletar vídeos da internet não é uma tarefa fácil, pois a maioria possui direitos autorais e de privacidade, além da possibilidade de alteração de sua qualidade. Para treinar e validar a abordagem proposta, alguns vídeos foram coletados de trabalhos científicos: Foggia et al. (2015); A. Enis Cetin (2021); Muhammad et al. (2018a); Ultimate Chase (2019). Além disso, foram gerados vídeos de incêndio e realizadas simulações de incêndio controlado em áreas verdes. Para as etapas de treinamento, validação e teste foram selecionados ao todo 87 vídeos. A Figura 18 apresenta um quadro de cada um dos vídeos do conjunto de dados de treinamento e validação. Para treinar e validar a abordagem de análise temporal, foram usados 35 (trinta e cinco) vídeos contendo principalmente regiões de mata e incêndio.

Na etapa de teste, foram escolhidos vídeos de câmeras de vigilância da plataforma *Apaga o Fogo!* (CEMIG, UFMG, Gaia, RaroLabs e UFVJM, 2021). A plataforma *Apaga o Fogo!* utiliza câmeras para monitorar, em tempo real, as matas e florestas da região metropolitana de Belo Horizonte, detectando automaticamente os focos de incêndio. Todas as câmeras estão conectadas a um servidor localizado no Centro de Estudos da Fala, Acústica, Linguagem e Música (CEFALA) da UFMG por meio de um enlace de internet por uma rede privada virtual (VPN - *virtual private network*), o que promove um gerenciamento eficaz dos vídeos coletados. Os vídeos são detectados e salvos no servidor a uma taxa de 2 FPS e 30 segundos de duração.

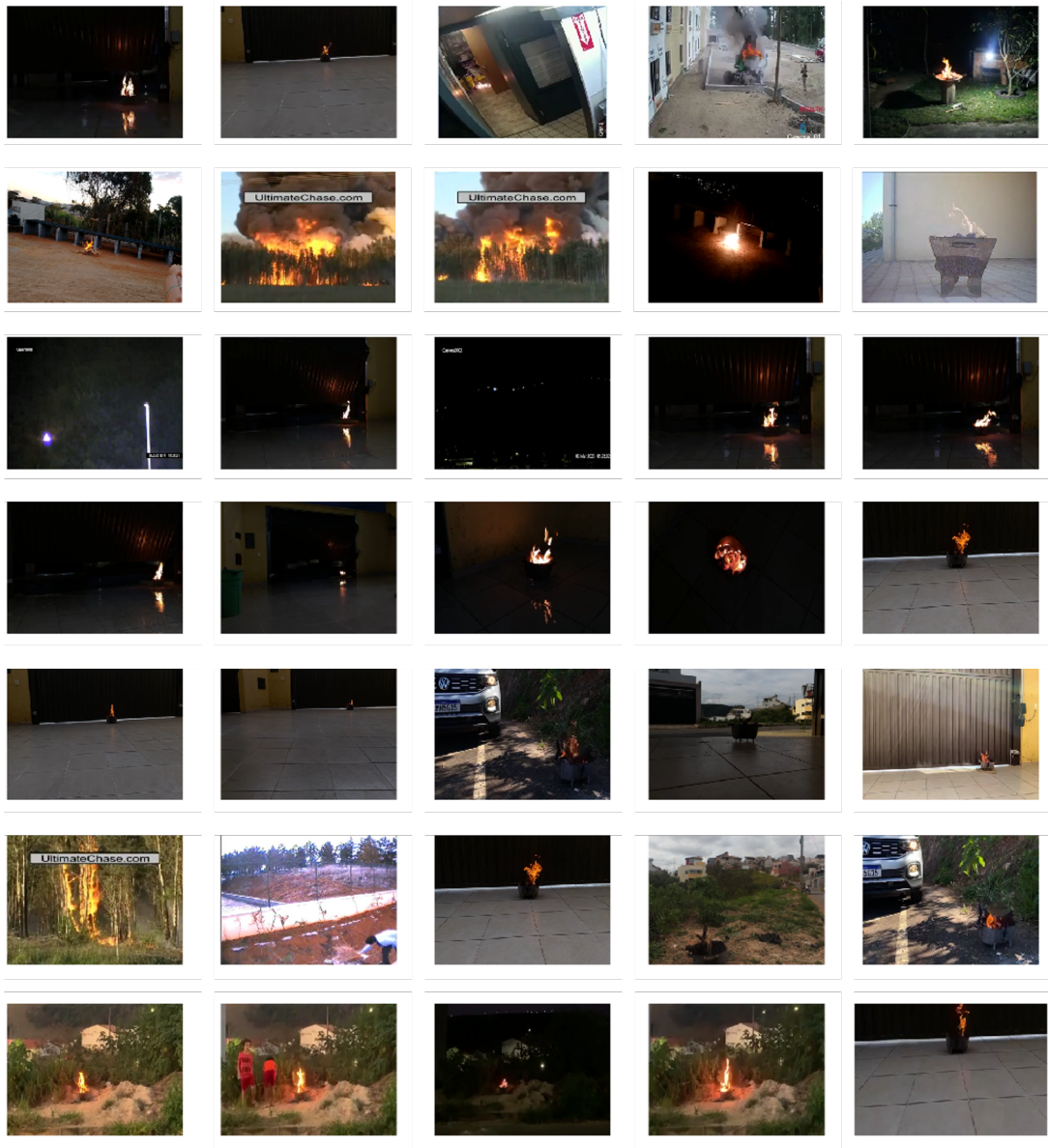
No total, foram separados 52 vídeos<sup>1</sup> para a etapa de testes. Esses vídeos foram submetidos à detecção espacial. O YOLO resolve a grande maioria de dados reais envolvendo a detecção de fogo, por isso foi realizada uma busca exaustiva no servidor do *Apaga o Fogo*. Não são muitos os casos de falsos positivos, daí foram selecionados 26 vídeos que apresentam a detecção de eventos parecidos com o fogo (falsos positivos). Para ter uma base de dados balanceada, foram selecionados 26 vídeos contendo eventos de fogo (verdadeiros positivos). Ou seja, 26 foram classificados equivocadamente como fogo, pois, na realidade, não possuem este evento. Os 26 vídeos restantes foram detectados como verdadeiros positivos, isto é, o sistema detectou fogo em uma situação de evento real. A Figura 19 mostra alguns quadros de vídeos do conjunto de dados de teste.

Os quadros dos vídeos apresentados na Figura 19 representam uma parte dos 52 vídeos que formam o conjunto de dados de teste e mostram 15 vídeos contemplando

<sup>1</sup> Os vídeos usados no treinamento e validação e teste estão disponíveis em : [https://drive.google.com/drive/folders/1HHpxW\\_908VZJ8wRi-D0pCwoZd2nPKj11?usp=share\\_link](https://drive.google.com/drive/folders/1HHpxW_908VZJ8wRi-D0pCwoZd2nPKj11?usp=share_link).



Figura 18 – Quadros dos 35 vídeos que compõem o conjunto de dados utilizados na etapa de treinamento e validação.

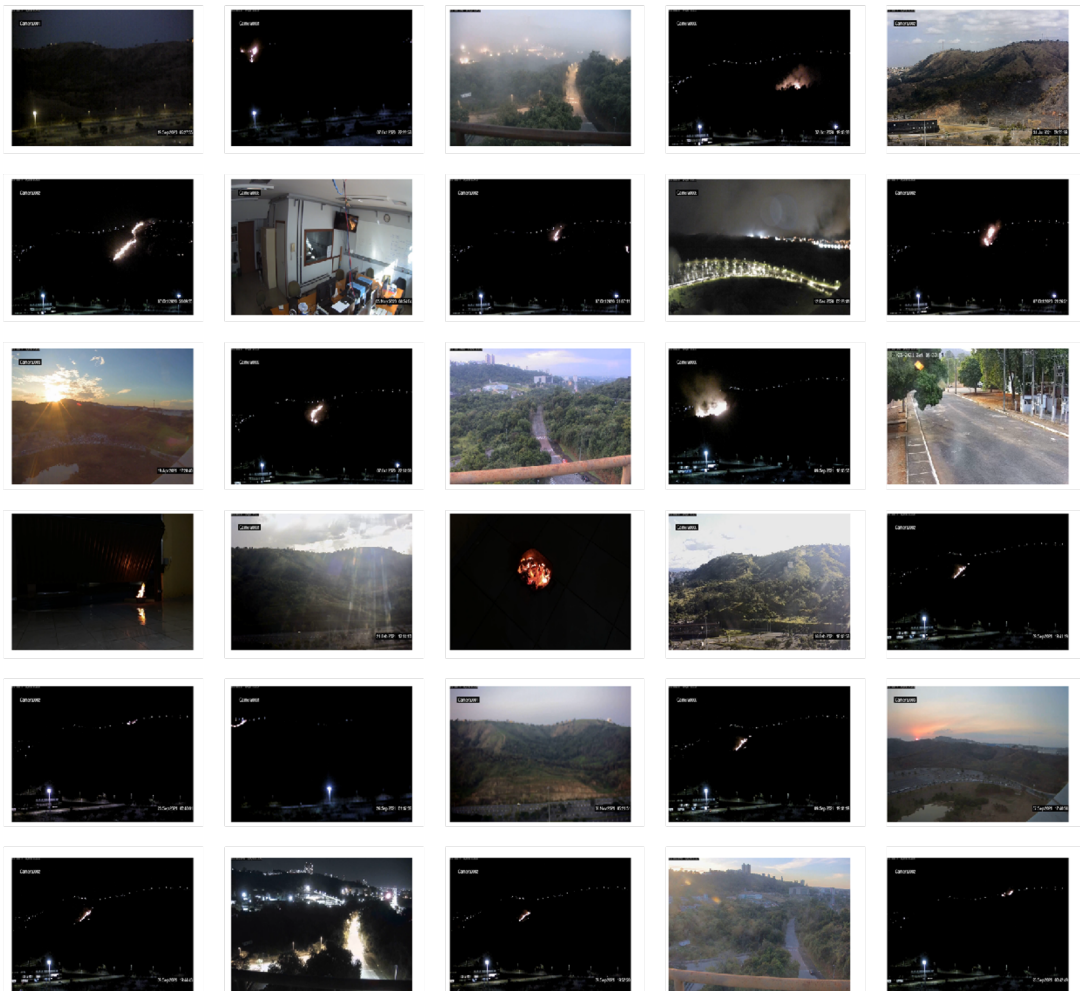


Fonte: Autoria própria.

cenários sem fogo e 15 cenários com fogo. Esses vídeos foram selecionados para que fosse possível investigar como a abordagem utilizando análise de séries temporais colabora na correção das detecções equivocadas realizadas pelo sistema de detecção espacial, e não para avaliar o desempenho do detector espacial. A avaliação do desempenho do detector espacial aplicado na detecção de fogo é realizada em [Venâncio \(2021b\)](#) e [Venâncio et al. \(2022\)](#).



Figura 19 – Quadros de 30 dos 52 vídeos que compõem o conjunto de dados utilizados na etapa de teste.



Fonte: Autoria própria.

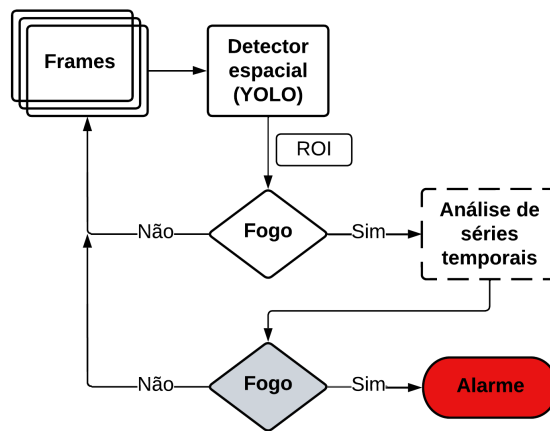
## 4.2 Abordagem proposta

O sistema para detecção de fogo proposto é ilustrado na Figura 20 e é constituído por dois estágios sequenciais:

- (i) detector espacial de fogo;
- (ii) análise das séries temporais.

De acordo com o diagrama mostrado na Figura 20, o detector espacial monitora vídeos. Caso haja alguma detecção, é selecionada uma região de interesse que possa ter o indício de incêndio. A região selecionada é, então, avaliada por meio da análise das séries temporais. As seções seguintes detalham este procedimento de detecção.

Figura 20 – Sistema para detecção de fogo composto por dois estágios sequenciais: (i) detector e (ii) análise de séries temporais.



Fonte: Autoria própria.

#### 4.2.1 Região de interesse (ROI)

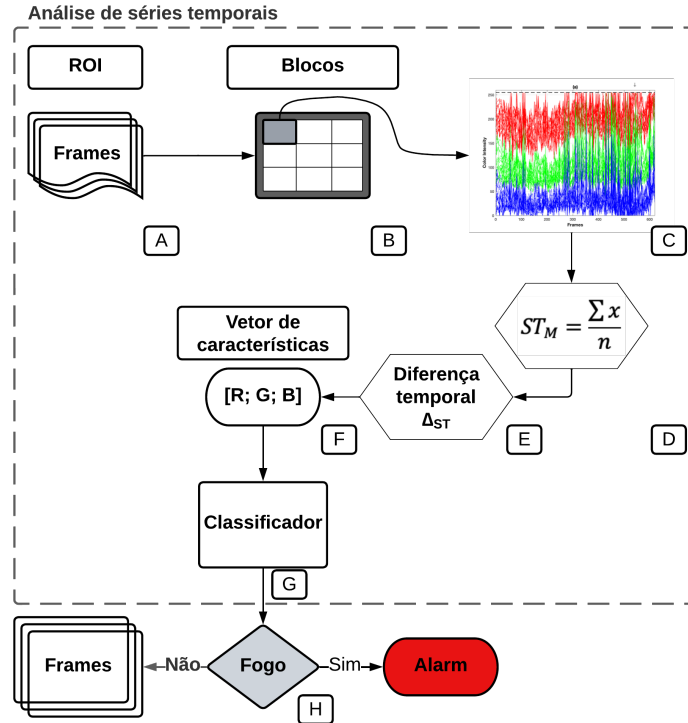
Inicialmente, uma sequência de quadros de um vídeo é utilizada como entrada para o detector. O primeiro estágio (i) tem o objetivo de identificar e localizar eventos de fogo em uma sequência de imagens. Esse detector é baseado no algoritmo de detecção YOLO. Caso não haja nenhuma detecção de fogo, o procedimento é reiniciado e uma nova entrada de quadros é realizada. Caso haja alguma detecção de fogo, o algoritmo YOLO, descrito na seção 3.5, delimita a região do evento com uma caixa delimitadora. Essa caixa delimitadora é indicada no diagrama da Figura 20 como ROI, ou seja, região de interesse, termo do inglês *Region Of Interest*. É possível haver mais de uma ROI em uma sequência de quadros de um vídeo e, ao longo dessa sequência, a área da região pode variar, acompanhando o tamanho da forma do evento detectado.

Uma vez confirmada a detecção de um evento no primeiro estágio, no estágio subsequente (ii), realiza-se a análise das séries temporais dos sinais de vídeo. Na Figura 20, o bloco do diagrama referente à análise de séries temporais (destacado com linha tracejada) é constituído por um processo multi-passos mostrado em detalhes na Figura 21.

A ROI definida pelo detector espacial contempla algum evento de incêndio, seja ele verdadeiro positivo ou falso positivo e, na maioria das vezes, apresenta a sua área bem menor que a dimensão total de um quadro. Normalmente, o detector espacial é capaz de detectar algum foco de incêndio ainda no início e a aplicação da AST numa região com dimensões reduzidas proporciona um processamento computacional menor do que em toda a área de um quadro. O vídeo da ROI é usado como entrada para o processo multi-passos da AST (A).

A Figura 22 mostra um exemplo de um quadro cuja ROI foi detectada pelo YOLO. O quadro em questão tem resolução de  $720 \times 1280$  pixels, enquanto a ROI possui

Figura 21 – Análise de séries temporais (AST): processo multi-passos extrai as séries temporais de sinais de vídeo, rotula-as em duas classes (fogo/não-fogo) e as classifica usando aprendizado de máquina.



Fonte: Autoria própria.

a dimensão de  $45 \times 61$  *pixels*. Dessa maneira, ao invés de analisar o contexto temporal de um vídeo composto por quadros de dimensão  $720 \times 1280$  *pixels*, a análise é realizada apenas na região que, neste exemplo, possui uma área quase 330 vezes menor. Assim, de acordo com a Figura 21, a sequência de quadros composta pela ROI detectada pelo YOLO é utilizada como entrada para a AST. A ROI, detectada pelo YOLO, pode variar sua dimensão ao longo dos quadros, no entanto, para a análise temporal é considerada apenas a primeira delimitação realizada pelo detector espacial.

#### 4.2.2 Blocos

O segundo passo da abordagem proposta (B) é dividir as regiões delimitadas pelo detector espacial em blocos de dimensão  $N \times N$ . Neste trabalho,  $N$  assume valores iguais a:

$$N = 1, 2, 4, 8, 16, 32. \quad (4.1)$$

Variar o tamanho do bloco ( $N$ ) permite analisar o comportamento da abordagem proposta em relação à capacidade de auxiliar na detecção do fogo.

Para exemplificar, considere  $N$  sendo igual ao tamanho  $1 \times 1$  *pixel*, ou seja,  $N = 1$ .

Figura 22 – Região de interesse (falso positivo) detectada pelo algoritmo YOLO em um quadro de um vídeo com dimensão de  $1280 \times 720$  *pixels*. Ampliação da ROI com dimensão igual a  $45 \times 61$  *pixels*.



Fonte: Autoria própria.

Nesse caso, o bloco contém apenas um único *pixel*, que pode ser representado por uma série temporal a partir da equação 3.4. Para elucidar ainda mais, considere agora um vídeo composto por imagens no espaço de cores RGB com resolução de  $320 \times 240$  *pixels*, que totalizam 76.800 *pixels* para cada uma das componentes (veja a Figura 23). Todos esses *pixels* gerariam o mesmo número de séries temporais para  $N = 1$ , tornando o processamento uma tarefa árdua. Se, então se considerar dividir esta imagem em blocos com dimensão  $N = 16$ , por exemplo, cada um dos 300 blocos gerados conteriam exatamente 256 ( $16 \times 16$  *pixels*) séries temporais.

Caso uma imagem, ou uma sequência de imagens, possua resolução que não seja múltipla do valor da dimensão do bloco ( $N$ ), é realizada uma extensão das bordas. A extensão é preenchida por zeros, técnica conhecida como *zero padding*, permitindo que a imagem resultante tenha dimensões divisíveis pelo fator desejado. A Figura 24 mostra um exemplo desta técnica. Para esse exemplo, a imagem apresentada possui resolução  $290 \times 210$  *pixels*, e para que seja possível dividi-la em blocos de dimensão  $N = 16$  é preciso realizar a extensão das bordas com preenchimento por zeros. A imagem resultante passa a possuir resolução  $304 \times 224$  *pixels*, gerando 266 blocos. Nota-se que a cor das bordas estendidas possui a cor preta.

Figura 23 – Quadro de um vídeo dividido em blocos de tamanho  $N = 16 \times 16$  *pixels*. Em destaque (cor verde) um bloco pertencente à região de fogo. A dimensão da imagem é de  $320 \times 240$  *pixels* e possui ao todo 300 blocos, cada um com 256 *pixels*.



Fonte: Autoria própria.

### 4.2.3 Série temporal média

Considerando que cada bloco possui  $n$  séries temporais, para representá-lo por uma única série temporal, basta tomar a média das séries temporais:

$$\overline{ST}_{Bloco} = \frac{1}{n} \sum_{i=1}^n x_i[k], \quad (4.2)$$

cada bloco passaria a ser representado por uma série temporal para cada componente RGB. Isto é, no exemplo dado, ao invés de representar o vídeo por 76.800 séries temporais, agora é possível representá-lo por apenas 300 séries temporais se  $N = 16$ . Esse mesmo raciocínio é válido para os outros valores de  $N$  representados na expressão 4.1. Na equação 4.2  $n$  é o número total de séries temporais do bloco. No caso de blocos  $16 \times 16$ ,  $n$  é igual a 256 séries temporais.  $\overline{ST}_{Bloco}$  é representada por três intensidades de cores, uma para cada componente de cor, conforme é mostrado na Figura 25.

Apesar das séries temporais apresentadas na Figura 25 serem bastante parecidas entre si, computar a média permite reter informações de uma região de fogo mesmo



Figura 24 – Quadro de um vídeo dividido em blocos de tamanho  $N = 32 \times 32$  pixels. A dimensão original ( $290 \times 210$  pixels) da imagem não é divisível por  $N$ , acarretando em uma extensão das bordas da imagem usando zeros (*zero padding*). A imagem resulta em uma resolução  $304 \times 224$  pixels e possui ao todo 266 blocos.



Fonte: Autoria própria.

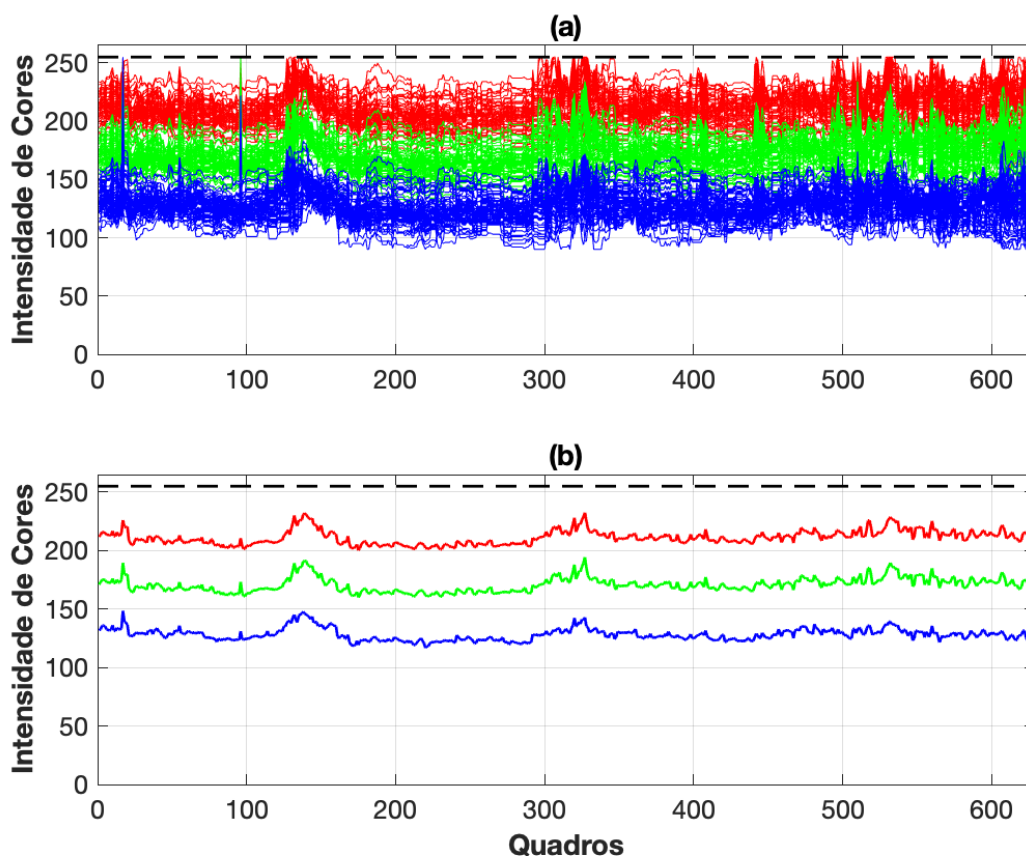
havendo a saturação de algumas séries da componente vermelha. Dessa maneira as séries temporais conseguem reter informações das baixas intensidades, sem haver perda das informações com altas intensidades.

#### 4.2.4 Diferenças Temporais

Comumente, vídeos de câmeras de vigilância necessitam ser armazenados e transmitidos, o que embora, a princípio, pareça simples, é deveras oneroso. Transmitir vídeos captados em um lugar ermo, tal qual uma floresta, com infraestrutura mínima, é um desafio. Da mesma maneira, armazenar uma quantidade grande de vídeos torna-se um problema, pois demanda equivalente capacidade de armazenamento. O uso de baixa taxa de quadros é uma solução simples que permite gerar vídeos com tamanhos reduzidos, cooperando, assim, tanto na transmissão quanto no armazenamento.

A base de dados disponível pelo sistema *Apaga o Fogo* apresenta vídeos com taxa

Figura 25 – (a) Séries temporais ( $\mathbf{x}$ ) de um bloco com tamanho  $N = 8$  pertencente à região de fogo.  $N = 8$  representa um total de  $n = 64$  séries temporais para cada componente R, G e B. (b) Séries temporais médias ( $\overline{ST}_{Bloco}$ ) do bloco para as componentes R, G e B. A linha tracejada representa o valor máximo da intensidade de cor, 255.



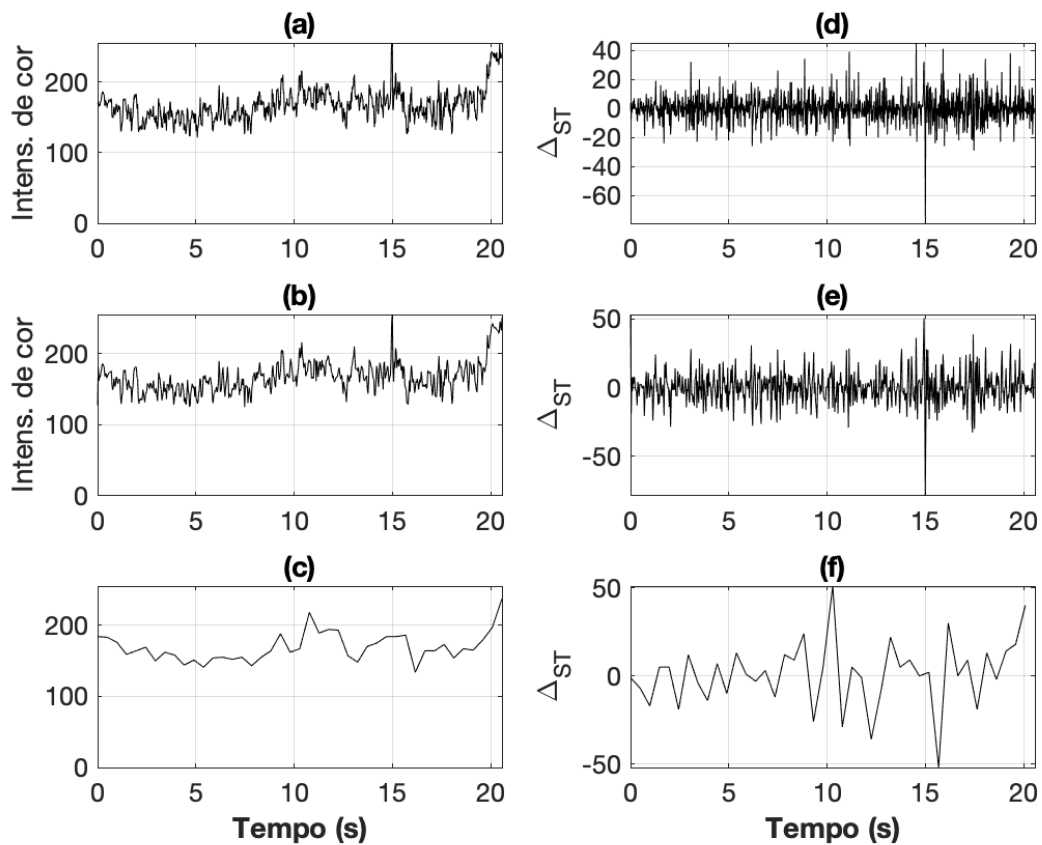
Fonte: Autoria própria.

de 2 quadros por segundo e duração de 30 segundos, resultando em um total de 60 quadros por vídeo. Ao invés de computar a primeira diferença, uma outra maneira de analisar as séries temporais amostradas e 2 FPS seria observar o domínio da frequência, aplicando por exemplo a transformada de Fourier. No entanto, as chamas de fogo apresentam oscilações, devido a cintilação, maiores que 12 Hz. Tendo em vista esta evidência e considerando o teorema da amostragem (Aguirre, 2007), idealmente seria necessário gravar um vídeo com pelo menos duas vezes a máxima frequência do sinal. Sendo assim, a gravação de vídeos para sistemas de detecção de fogo deveria considerar uma taxa superior a taxa de Nyquist, ou seja, 24 quadros por segundo. Diante da impossibilidade de se realizar uma análise espectral, pois esta exigiria uma amostragem acima da taxa de Nyquist, torna-se necessário buscou-se outras formas de se utilizar a informação temporal contida nos sinais. Uma primeira ideia seria explorar as diferenças entre as funções densidade de probabilidade dos sinais estimadas a partir dos seus histogramas. Essa possibilidade foi tentada com

resultados apenas razoáveis. Resultados muito melhores foram obtidos por meio da análise das diferenças temporais conforme detalhado a seguir.

A redução na taxa é eficaz para reduzir o tamanho do vídeo, porém o movimento de fogo, ao longo dos quadros, perde a fluidez natural. Considerando as séries temporais pertencentes a uma região de fogo, essas apresentam um menor número de oscilações na intensidade de cores se comparadas a uma série temporal de um vídeo com taxa alta. Esse fato pode ser observado na Figura 26. Computar a Diferença Temporal (3.7) das séries temporais ajuda a contornar essa situação.

Figura 26 – (a), (b) e (c) são séries temporais ( $\mathbf{x}$ ) da componente R pertencentes à região de fogo com diferentes taxas: 60 FPS, 30 FPS e 2 FPS, respectivamente. (d), (e) e (f) apresentam as respectivas diferenças temporais ( $\Delta_{ST}$ ).



Fonte: Autoria própria.

A Figura 26 (a), (b) e (c) mostra três séries temporais da mesma região de fogo de vídeos com taxas diferentes, respectivamente, 60, 30 e 2 FPS. Suas respectivas diferenças temporais,  $\Delta_{ST}$  (equação 3.7), são mostradas em (d), (e) e (f).

A Figura 26 permite realizar duas análises: a primeira em relação às séries temporais e a segunda em relação às diferenças temporais. No que tange às séries temporais, (a), (b) e (c), pode-se dizer que, quanto maior a taxa, maior quantidade de informações,



pois tem-se um número maior de amostras. À medida que a taxa diminui, a quantidade de oscilações também diminui e a intensidade das oscilações tende a ser menor. Na análise das diferenças temporais, (d), (e) e (f), a quantidade de oscilações também diminui quando há decaimento na taxa, porém, as diferenças temporais conseguem manter as intensidades das oscilações mesmo quando a taxa diminui. As diferenças temporais realçam as oscilações que ocorrem nas regiões de fogo, em razão à cintilação natural apresentada por uma chama de fogo.

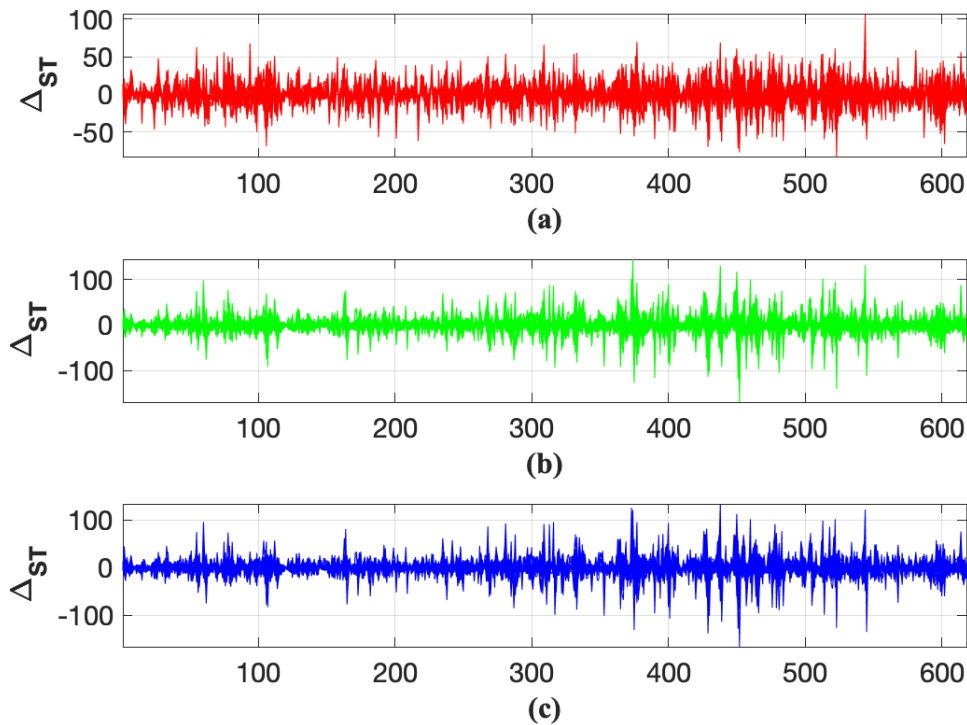
#### 4.2.5 Classificação das séries temporais usando aprendizado de máquina

Computar as diferenças temporais das séries temporais permite realçar, muito mais que usar simplesmente as séries temporais, as variações que ocorrem entre a chama de fogo e o plano de fundo. Isto corrobora para que a aplicação dos classificadores que utilizam aprendizado de máquina sejam mais eficientes. Para determinar se a região delimitada pelo detector espacial possui ou não fogo foram comparados seis classificadores diferentes, conforme descrito na seção 3.3.

Devido ao fato do treinamento ser supervisionado, as séries temporais foram rotuladas em “fogo” e “não fogo”. Cada um dos blocos da ROI possui três diferenças temporais, uma para cada componente do espaço de cores R, G e B. A Figura 27 mostra as diferenças temporais das três componentes R, G e B típicas de uma região de fogo. As séries temporais selecionadas e rotuladas no conjunto de trinta e cinco vídeos de treinamento foram selecionadas e rotuladas manualmente de regiões de fogo, mata, luzes artificiais e cores parecidas com o fogo. Estas séries compõem o vetor de entrada dos classificadores. As séries de fogo foram selecionadas principalmente nas regiões das bordas do fogo. E as séries não pertencentes às regiões de fogo, rotuladas como “não fogo”, foram selecionadas nas regiões que compõem os planos de fundo dos vídeos, como, por exemplo, florestas, céu, estacionamentos e luzes. O conjunto de dados de treinamento composto pelas diferenças temporais é igualmente dividido, isto é, as classes são equilibradas.

As três diferenças temporais apresentadas na Figura 27 foram extraídas de uma região de fogo e representam as componentes do espaço RGB de um bloco. Esse tipo de série de diferenças temporais exemplifica as séries que compõem os vetores de entrada utilizados para o treinamento dos classificadores. Na prática as séries temporais usadas como entrada possuem um total de 60 quadros, gravados a uma taxa de 2 quadros por segundo. Para cada um dos classificadores foram usados diferentes conjuntos de entradas para verificar a influência do tamanho do bloco na classificação das séries temporais analisadas. Todos os classificadores foram treinados usando otimização de hiperparâmetros, de forma que os resultados obtidos foram os melhores para os conjuntos de dados propostos. A escolha ótima dos parâmetros de cada classificador teve como critério de parada o número de

Figura 27 – Diferenças temporais ( $\Delta_{ST}$ ) das componentes (a) vermelha (R), (b) verde (G) e (c) azul (B) de uma região de fogo. Estas diferenças temporais representam cada um dos blocos e compõem o vetor de entrada para os classificadores. Ao todo são mostradas 64 séries, o que representa um bloco de tamanho  $8 \times 8$  *pixels*.



Fonte: Autoria própria.

iterações igual a trinta (30), uma vez que os parâmetros ótimos dos modelos foram obtidos em números menores de iteração. Conforme descrito na seção 3.4, o método utilizado para particionamento dos dados de treinamento e validação foi a validação cruzada *5-fold*.

O modelo ajustado para cada classificador foi testado usando um novo conjunto de dados contendo 52 (cinquenta e dois) vídeos. A etapa de teste dos classificadores permite verificar o seu desempenho frente a vídeos não apresentados aos modelos na fase de treinamento e validação.

Seguindo o fluxo na Figura 21, caso o classificador que está sendo analisado detecte fogo, um alarme é disparado para avisar a presença do evento. Caso o classificador não detecte fogo no vídeo apresentado no sistema proposto, o sistema volta à etapa inicial e aguarda uma nova detecção de evento a ser realizada pelo detector espacial.

### 4.3 Comentários finais

Neste capítulo foram apresentados os dois conjuntos de dados que serão usados para treinar/validar e, respectivamente, testar a metodologia proposta. A composição desses conjuntos de dados foi uma tarefa árdua, tendo em vista a escassez de vídeos estáticos e sem direitos autorais na internet. Do conjunto de dados que será usado para treinamento e validação no Capítulo 5, 26 foram de autoria própria. Ao todo são utilizados 87 vídeos, sendo 35 para treinamento e validação e 52 para a etapa de testes. Já os vídeos que compõem o conjunto de dados de teste foram, em sua maioria, coletados da plataforma *Apaga o Fogo!*. Todos os vídeos possuem duração de 30 segundos e taxa de 2 quadros por segundo.

A abordagem proposta para detectar fogo em ROIs de vídeos, a partir das diferenças temporais, foi apresentada e o processo é ilustrado nas Figuras 20 e 21.

# Capítulo 5

## Resultados

Os resultados dos procedimentos descritos no Capítulo 4 são reportados aqui. Primeiramente, uma breve descrição dos conjuntos de dados de treinamento, validação e teste é realizada. Os principais resultados das etapas de treinamento e validação são apresentados. A partir do conjunto de dados de teste, é realizado um comparativo do desempenho de seis classificadores diferentes. Além disso, são realizados testes alterando-se a dimensão dos blocos entre os valores de  $N = 1, 2, 4, 8, 16, 32$ . O tempo computacional e o desempenho dos algoritmos submetidos a diferentes tamanhos de blocos são discutidos. A partir do conjunto de dados de teste, os resultados da AST são comparados com o método proposto em [Venâncio et al. \(2021\)](#). Os autores apresentam uma abordagem na qual primeiro extraem as características espaciais usando CNN e depois analisam as variações temporais das caixas ao longo dos quadros. Os autores consideram as distâncias do centroide das caixas delimitadoras entre os quadros subsequentes. Se as variações forem significativas, a detecção de incêndio é confirmada. Essa abordagem é chamada de Análise Temporal (TA).

Levando-se em consideração a Figura 21, todo o procedimento da análise das séries temporais foi realizado usando uma máquina Intel(R) Xeon(R) Gold 6140 de 18 núcleos e 36 threads operando a 2,30 GHz, 256 GB de RAM, RAID 5 com SSDs de 1TB e Placa de Vídeo NVIDIA Quadro P5000 com 2560 cores CUDA e 16GB GDDR5X, executando o sistema operacional Ubuntu 18.04.3 LTS e a ferramenta de compilação CUDA 10.1, V10.1.243.

### 5.1 Conjuntos de dados atribuídos aos tamanhos dos blocos $N$

Conforme já discutido, vide Figura 21, os quadros das regiões de interesse são divididos em blocos de tamanhos  $N$  de acordo com a expressão 4.1. Dessa maneira, para cada tamanho de bloco  $N$ , foi criado um conjunto específico de dados a ser usado para

treinamento e validação dos classificadores descritos na seção 3.3. Na Tabela 3, estão discriminadas as quantidades de séries temporais selecionadas para cada tamanho de bloco  $N$ . O tamanho do bloco é avaliado com o objetivo de se obter a melhor relação entre custo computacional e desempenho dos classificadores.

Tabela 3 – Conjuntos de dados usados para o treinamento dos classificadores. Quantidade de séries temporais ( $\mathbf{x}$ ) utilizadas para cada dimensão  $N$  do bloco e quantidade de diferenças temporais ( $\Delta_{\text{ST}}$ ).

$N$	$1 \times 1$	$2 \times 2$	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$	Total
Nº de $\mathbf{x}$	10.560	10.560	22.080	76.800	168.960	675.840	964.800
Nº de $\Delta_{\text{ST}}$	10.560	2.640	1.380	1.200	660	660	17.100

Fonte: Autoria própria.

Para treinamento e validação dos classificadores foram destinados 35 (trinta e cinco) vídeos (Figura 18) e, de acordo com a Tabela 3, foram selecionadas, desses vídeos, mais de 960.000 séries temporais ( $\mathbf{x}[k]$ ) para compor os conjuntos de dados para os tamanhos de blocos  $N$ . As séries temporais são divididas igualmente em duas classes e rotuladas em “fogo” e “não fogo”. Na terceira linha, apresenta-se a quantidade de diferenças temporais ( $\Delta_{\text{ST}}$ ), definida pela equação 3.7, computada a partir das séries temporais médias de cada bloco.

Para testar os modelos é utilizado outro conjunto de dados composto por 52 (cinquenta e dois) vídeos, conforme descrito na seção 4.1. A quantidade de séries temporais que são utilizadas na etapa de teste depende da dimensão da ROI, que é definida pelo detector espacial em cada um dos vídeos. Evidente que cada vídeo apresenta um evento detectado e as dimensões das ROIs são diferentes. Sabendo disso, são utilizadas, no total, 472.710 séries temporais para a etapa de teste dos algoritmos que utilizam aprendizado de máquina.

A próxima seção apresenta os resultados da classificação das séries temporais usando aprendizado de máquina. Ao final, é realizada uma comparação entre os melhores algoritmos de aprendizado de máquina, levando-se em consideração o tamanho do blocos.

## 5.2 Classificação das séries temporais usando aprendizado de máquina

Uma vez definidos os conjuntos de dados de treinamento para cada um dos diferentes tamanhos dos blocos,  $N$ , esses conjuntos são utilizados como vetores de entrada para cada um dos classificadores empregados neste trabalho. Todos os classificadores foram treinados usando validação cruzada com 5 divisões no conjunto de dados, ou seja, *5-fold*.

O tempo de processamento da etapa de treinamento é mostrado na Tabela 4 e a acurácia da validação na Figura 28.

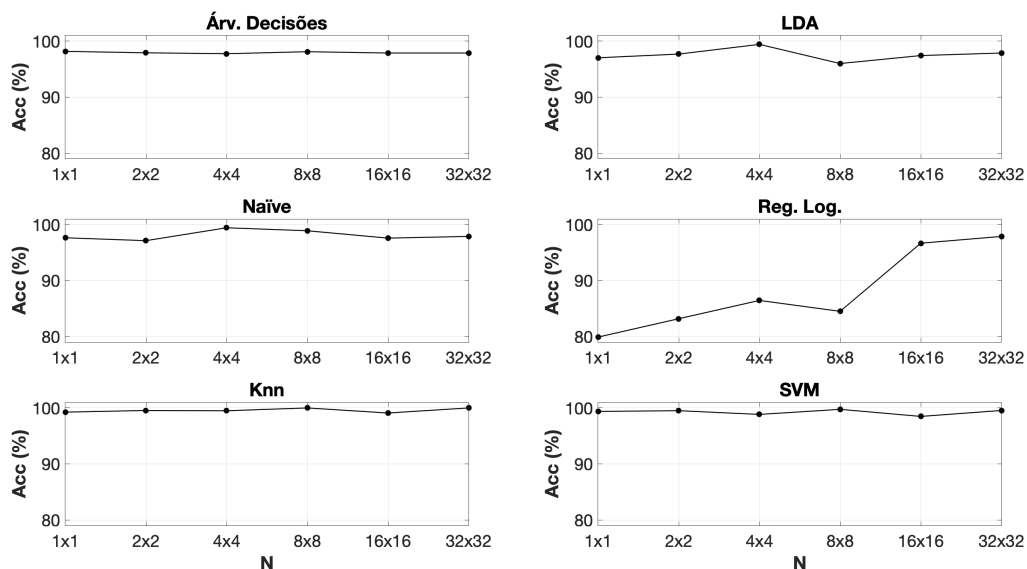
Tabela 4 – Tempo de processamento (em segundos) da etapa de treinamento dos modelos para cada um dos tamanhos dos blocos  $N$ .

$N$	$1 \times 1$	$2 \times 2$	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$
Árvore de decisões	145,7	83,2	66,3	51,9	40,6	31,1
LDA	115,7	94,9	83,7	73,6	68,9	65,8
Naïve	4.808,8	213,6	180,4	149,9	121,2	116,9
Regressão logística	8,5	1,2	1,2	1,5	1,8	1,5
$knn$	396,3	145,1	68,8	58,7	76,6	53,1
SVM	3.629,3	494,1	391,2	294,7	140,6	136,1

Fonte: Autoria própria.

Na Tabela 4, os tempos de treinamento são apresentados em segundos. Quanto menor o tamanho do bloco  $N$  maior é o tempo de treinamento. Para cada tipo de classificador existem diversos parâmetros que impactam a acurácia. Todos os classificadores tiveram seus parâmetros otimizados de tal maneira que os resultados obtidos são os melhores para o conjunto de treinamento proposto. A escolha ótima dos parâmetros de cada classificador teve como critério de parada o número de iterações igual a 30. Os modelos Naïve e SVM possuem os maiores tempos de treinamento, atingindo, respectivamente, 4.808,8 e 3.629,2 segundos quando  $N = 1$ .

Figura 28 – Acurácias da etapa de validação dos classificadores para cada um dos tamanhos de blocos  $N$ . Classificadores: (a) Árvore de decisões, (b) LDA, (c) Naïve, (d) Regressão Logística, (e)  $Knn$  e (f) SVM.



Fonte: Autoria própria.

Observando a Figura 28, pode-se comparar as acurácias na etapa da validação de cada um dos classificadores para os diferentes tamanhos de blocos  $N$ . Para  $N = 1$  o modelo de regressão logística obteve a menor acurácia, atingindo 80,0% (Figura 28(d)), enquanto que a maior acurácia foi do classificador SVM, 99,4%. Para  $N = 2$  a menor e maior acurácias são 88,4% e 99,7% que foram obtidas pelos modelos de regressão logística e  $Knn$ , respectivamente. A acurácia da validação dos dados quando  $N = 4$  atingiu 86,5% para a regressão logística e 99,6% para o modelo  $Knn$ . Quando  $N = 8$ , a regressão logística atinge a menor acurácia, 84,0% e os modelos  $Knn$  e SVM atingem 100%. Para  $N = 16$ , os valores mínimo e máximo da acurácia são, respectivamente, 97,1% (árvore de decisões) e 96,6% (SVM). E, para  $N = 32$ , a menor acurácia é igual a 97,0% (árvore de decisões) e 99,7% ( $Knn$ ).

Concluídas as etapas de treinamento e validação, os classificadores são submetidos a um novo conjunto de vídeos: o conjunto de testes.

### 5.3 Teste dos classificadores

O conjunto de testes é constituído por 52 (cinquenta e dois) novos vídeos que foram usados para avaliar o desempenho dos classificadores. Todos os 52 vídeos foram detectados pelo detector espacial, YOLO. Em 50% deles (26 vídeos), a presença de fogo foi detectada, sendo que realmente havia fogo, isto é, são verdadeiros positivos. Por outro lado, para os outros 26 vídeos, o detector avaliou a existência de fogo, sem, contudo, havê-lo, ou seja, são detecções falsas, falsos positivos. O objetivo da análise das séries temporais é verificar se é possível diminuir a taxa de falsos positivos, sem que haja decréscimo na taxa de verdadeiros positivos.

Tabela 5 – Tempo médio ( $\pm$  desvio padrão) em segundos da AST para processar um vídeo do conjunto de teste para cada classificador considerando o tamanho do bloco  $N$ .

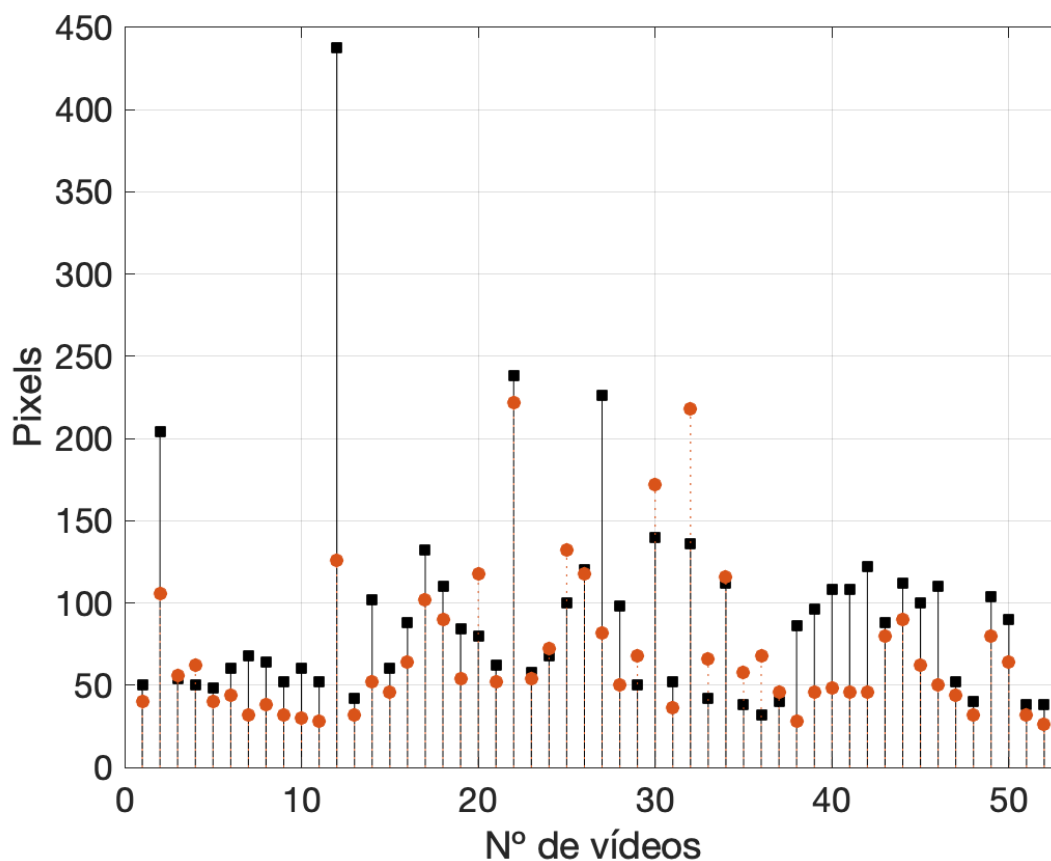
<b>N</b>	<b>Árv. decisões</b>	<b>LDA</b>	<b>Naïve</b>	<b>Reg. Log.</b>	<b>Knn</b>	<b>SVM</b>
<b>1 × 1</b>	31±45	37±54	370±516	57±79	95±132	38±52
<b>2 × 2</b>	8,2±12	9,5±13,6	9,3±12	15±20	13±17	9,0±12
<b>4 × 4</b>	2,1±2,8	2,5±3,4	2,6±3,6	3,9±5,4	3,2±4,7	2,2±3,3
<b>8 × 8</b>	0,59±0,84	0,61±0,84	0,68±0,91	1,0±1,28	0,76±0,96	0,54±0,68
<b>16 × 16</b>	0,19±0,22	0,21±0,25	0,21±0,25	0,30±0,36	0,23±0,26	0,19±0,22
<b>32 × 32</b>	0,10±0,09	0,10±0,09	0,10±0,10	0,13±0,13	0,11±0,10	0,10±0,09

Fonte: Autoria própria.

A Tabela 5 mostra os tempos médios, em segundos, da etapa de teste dos 52 vídeos. Tendo em vista que a AST é realizada apenas na ROI, que foi definida pelo detector espacial, cada vídeo apresenta um tempo específico, pois o tempo da análise é proporcional ao tamanho da ROI. Para constar, a média das dimensões das ROIs é  $93 \times 69$  pixels. A

maior dimensão definida pelo detector é igual a  $438 \times 126$  e, a menor, a  $38 \times 26$  *pixels* (veja a Figura 29). Nota-se também que, quanto maior for o tamanho dos blocos menor será o tempo de processamento de um vídeo pela AST. Sendo assim, para exemplificar, o processamento da AST de um vídeo cujo tamanho dos blocos é igual a  $N = 1$ , é, em média, igual a 31,06 segundos quando o classificador aplicado é a árvore de decisões. Para esse mesmo classificador e  $N = 32$ , o tempo de processamento médio de um vídeo é de 0,10 segundos.

Figura 29 – Dimensão das ROIs de cada um dos vídeos usados na etapa de testes. (—■) representa a largura e (-.-●) a altura das ROIs selecionadas pelo detector espacial do conjunto de vídeos de teste.



Fonte: Autoria própria.

A Figura 29 mostra a dimensão (largura  $\times$  altura) em *pixels* de cada uma das ROIs dos vídeos usados na etapa de teste dos classificadores. A linha preta (—■) representa a largura e a linha vermelha (-.-●) a altura da ROI. Em média a dimensão da região de interesse, delimitada pelo detector espacial, é aproximadamente 9 vezes menor que a dimensão total de um vídeo<sup>1</sup>. A título de comparação, considere o vídeo com a maior resolução, isto é, o vídeo que possui a maior dimensão da ROI, vídeo número 12 (localizado

<sup>1</sup> Os vídeos obtidos da plataforma *Apaga o Fogo!* possuem a dimensão  $1280 \times 720$  *pixels*.



no maior pico) e o que possui a menor ROI, vídeo número 52. Se para realizar o teste desses vídeos for utilizado o classificador SVM para os diferentes tamanhos dos blocos  $N$ , tem-se os tempos de processamento em segundos, conforme é mostrado na Tabela 6. Como já dito, quanto menor for o  $N$  maior o tempo de processamento da abordagem, vez que a ROI possui um maior número de séries temporais. Sendo assim, para  $N = 1$  a menor ROI apresenta um tempo de processamento mais de 50 vezes menor que a maior ROI. Vale ressaltar que os vídeos possuem ao todo 60 quadros e duração de 30 segundos. Por outro lado, se se considerar a maior resolução da ROI, o tempo de processamento do maior valor do tamanho do bloco,  $N = 32$ , é mais de 600 vezes menor que para  $N = 1$ .

Tabela 6 – Tempo de processamento em segundos da maior e menor ROI dos vídeos de teste usando o classificador SVM.

Dimensão da ROI / N	Tempo (s)					
	1 × 1	2 × 2	4 × 4	8 × 8	16 × 16	32 × 32
<b>438 × 126</b>	252,2	59,1	16,2	3,4	1,1	0,41
<b>38 × 26</b>	4,7	1,2	0,26	0,09	0,04	0,03

Fonte: Autoria própria.

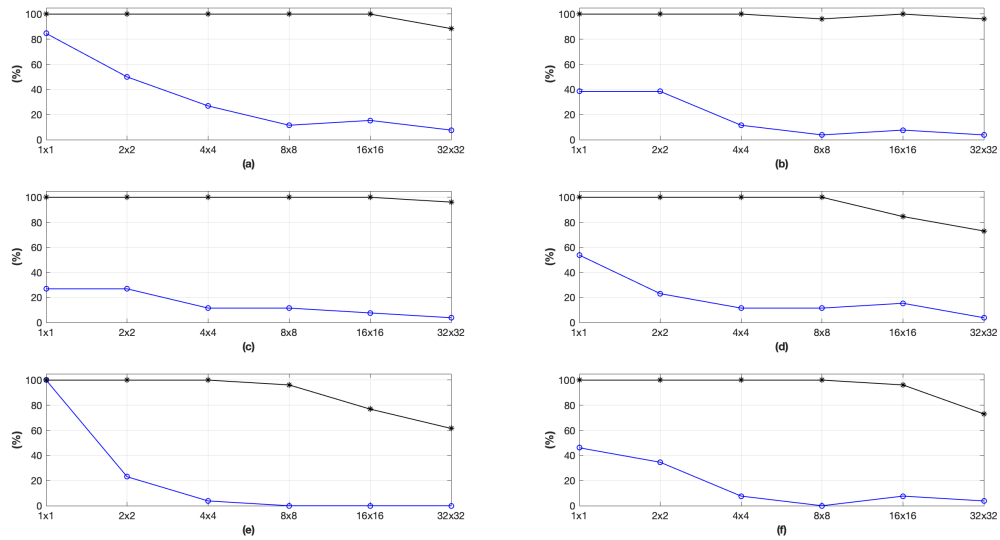
O conjunto de testes composto pelos 52 vídeos foi usado para estimar o impacto da AST na detecção de fogo. A Figura 30 mostra a relação entre o tamanho do blocos e o desempenho de cada um dos classificadores na etapa de teste.

Na Figura 30, podem ser vistas as taxas de detecção usando a AST. Para simplificar o exame dessa figura, considere analisar isoladamente uma subfigura e as análises das outras serão análogas. Sendo assim, de acordo com a Figura 30(f), que se refere ao classificador SVM, pode-se observar que a taxa de verdadeiros positivos (TVP) é igual a 100% quando  $N = 1, 2, 4, 8$  e para  $N = 16, 32$  essa taxa é menor que 100%. Uma redução na taxa de verdadeiros positivos não é desejável, pois, haver fogo em um vídeo e o classificador não ser capaz de detectá-lo pode ser devastador. Por outro lado, objetiva-se diminuir ao máximo a taxa de falsos positivos (TFP). Para  $N = 8$  a TFP é igual 0%. Ou seja, dos 26 vídeos que foram detectados erroneamente com fogo pelo detector espacial, a AST foi capaz de corrigir o diagnóstico e classificá-los como vídeos que não possuem vestígio de incêndios.

Na Tabela 7, são mostrados os desempenhos dos resultados obtidos para os classificadores em todos os tamanhos de blocos. Nela, são destacados em negrito os principais resultados, levando-se em consideração não apenas a acurácia, mas a capacidade do modelo em não diminuir a taxa de verdadeiros positivos. Em casos em que há igualdade no desempenho, prevalece o modelo que tem o maior tamanho do bloco  $N$ , pois, de acordo com a Tabela 5, quanto maior o bloco, menor será o tempo de processamento.

Em relação ao classificador árvore de decisões, quando  $N = 8$ , a acurácia atingiu 94,23%. A AST com este classificador foi capaz de reconhecer todos os vídeos que realmente

Figura 30 – Taxas de detecção (%) dos classificadores para os diferentes tamanhos de blocos usando os vídeos do conjunto de teste. (-\*) é a taxa de verdadeiros positivos (TVP) e (-o) a taxa de falsos positivos (TFP). Classificadores: (a) Árvore de decisões, (b) LDA, (c) Naïve, (d) Regressão Logística, (e) *Knn* e (f) SVM.



Fonte: Autoria própria.

apresentavam fogo e diminuir 88,46% a taxa de falsos positivos, uma vez que, seis vídeos do conjunto não foram detectados corretamente.

Tabela 7 – Desempenho do método AST na detecção de fogo. Para cada classificador foi destacado em negrito o tamanho do bloco ( $N$ ) que apresentou as melhores métricas.

	N	VP	FP	FN	VN	TVP	TFP	TFN	TVN	Acc (%)	Recall	Precision	F1-Score
Árvore de Decisões	1 × 1	26	22	0	4	100,00	84,62	0,00	15,38	57,69	1,00	0,54	0,70
	2 × 2	26	13	0	13	100,00	50,00	0,00	50,00	75,00	1,00	0,67	0,80
	4 × 4	26	7	0	19	100,00	26,92	0,00	73,08	86,54	1,00	0,79	0,88
	<b>8 × 8</b>	<b>26</b>	<b>3</b>	<b>0</b>	<b>23</b>	<b>100,00</b>	<b>11,54</b>	<b>0,00</b>	<b>88,46</b>	<b>94,23</b>	<b>1,00</b>	<b>0,90</b>	<b>0,95</b>
	16 × 16	26	4	0	22	100,00	15,38	0,00	84,62	92,31	1,00	0,87	0,93
	32 × 32	23	2	3	24	88,46	7,69	11,54	92,31	90,38	0,88	0,92	0,90
LDA	1 × 1	26	10	0	16	100,00	38,46	0,00	61,54	80,77	1,00	0,72	0,84
	2 × 2	26	10	0	16	100,00	38,46	0,00	61,54	80,77	1,00	0,72	0,84
	4 × 4	26	3	0	23	100,00	11,54	0,00	88,46	94,23	1,00	0,90	0,95
	8 × 8	25	1	1	25	96,15	3,85	3,85	96,15	96,15	0,96	0,96	0,96
	<b>16 × 16</b>	<b>26</b>	<b>2</b>	<b>0</b>	<b>24</b>	<b>100,00</b>	<b>7,69</b>	<b>0,00</b>	<b>92,31</b>	<b>96,15</b>	<b>1,00</b>	<b>0,93</b>	<b>0,96</b>
	32 × 32	25	1	1	25	96,15	3,85	3,85	96,15	96,15	0,96	0,96	0,96
Naïve	1 × 1	26	7	0	19	100,00	26,92	0,00	73,08	86,54	1,00	0,79	0,88
	2 × 2	26	7	0	19	100,00	26,92	0,00	73,08	86,54	1,00	0,79	0,88
	4 × 4	26	3	0	23	100,00	11,54	0,00	88,46	94,23	1,00	0,90	0,95
	8 × 8	26	3	0	23	100,00	11,54	0,00	88,46	94,23	1,00	0,90	0,95
	<b>16 × 16</b>	<b>26</b>	<b>2</b>	<b>0</b>	<b>24</b>	<b>100,00</b>	<b>7,69</b>	<b>0,00</b>	<b>92,31</b>	<b>96,15</b>	<b>1,00</b>	<b>0,93</b>	<b>0,96</b>
	32 × 32	25	1	1	25	96,15	3,85	3,85	96,15	96,15	0,96	0,96	0,96
Regressão Logística	1 × 1	26	14	0	12	100,00	53,85	0,00	46,15	73,08	1,00	0,65	0,79
	2 × 2	26	6	0	20	100,00	23,08	0,00	76,92	88,46	1,00	0,81	0,90
	4 × 4	26	3	0	23	100,00	11,54	0,00	88,46	94,23	1,00	0,90	0,95
	<b>8 × 8</b>	<b>26</b>	<b>3</b>	<b>0</b>	<b>23</b>	<b>100,00</b>	<b>11,54</b>	<b>0,00</b>	<b>88,46</b>	<b>94,23</b>	<b>1,00</b>	<b>0,90</b>	<b>0,95</b>
	16 × 16	22	4	4	22	84,62	15,38	15,38	84,62	84,62	0,85	0,85	0,85
	32 × 32	19	1	7	25	73,08	3,85	26,92	96,15	84,62	0,73	0,95	0,83
Knn	1 × 1	26	26	0	0	100,00	100,00	0,00	0,00	50,00	1,00	0,50	0,67
	2 × 2	26	6	0	20	100,00	23,08	0,00	76,92	88,46	1,00	0,81	0,90
	4 × 4	<b>26</b>	<b>1</b>	<b>0</b>	<b>25</b>	<b>100,00</b>	<b>3,85</b>	<b>0,00</b>	<b>96,15</b>	<b>98,08</b>	<b>1,00</b>	<b>0,96</b>	<b>0,98</b>
	8 × 8	25	0	1	26	96,15	0,00	3,85	100,00	98,08	0,96	1,00	0,98
	16 × 16	20	0	6	26	76,92	0,00	23,08	100,00	88,46	0,77	1,00	0,87
	32 × 32	16	0	10	26	61,54	0,00	38,46	100,00	80,77	0,62	1,00	0,76
SVM	1 × 1	26	12	0	14	100,00	46,15	0,00	53,85	76,92	1,00	0,68	0,81
	2 × 2	26	9	0	17	100,00	34,62	0,00	65,38	82,69	1,00	0,74	0,85
	4 × 4	26	2	0	24	100,00	7,69	0,00	92,31	96,15	1,00	0,93	0,96
	<b>8 × 8</b>	<b>26</b>	<b>0</b>	<b>0</b>	<b>26</b>	<b>100,00</b>	<b>0,00</b>	<b>0,00</b>	<b>100,00</b>	<b>100,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>
	16 × 16	25	2	1	24	96,15	7,69	3,85	92,31	94,23	0,96	0,93	0,94
	32 × 32	19	1	7	25	73,08	3,85	26,92	96,15	84,62	0,73	0,95	0,83

Fonte: Autoria própria.

Para  $N = 16$ , a acurácia do modelo LDA atingiu 96,15%, pois conseguiu classificar todos os vídeos positivos e detectou corretamente 24 vídeos com falsos positivos, diminuindo a TFP para 7,69%. Assim como a LDA, o classificador Naïve apresenta acurácia de 96,15%, uma vez que diminuiu a taxa de falsos positivos para 7,69% sem depreciar a taxa de verdadeiros positivos.

Para  $N = 8$ , a acurácia do modelo de regressão logística atingiu 94,23%, pois conseguiu classificar todos os vídeos positivos e detectou corretamente 23 vídeos com falsos positivos, diminuindo a TFP para 11,54%. Apesar da acurácia ser a mesma quando  $N = 4$ , nesse caso o tempo de processamento é, aproximadamente, 4,26 vezes maior, se comparado ao  $N = 8$  (veja a Tabela 5).

O classificador Knn apresentou acurácia para  $N = 4$ , igual a 98,08% e manteve a taxa de verdadeiros positivo em 100%. Para  $N = 8$  o classificador não foi capaz de classificar um vídeo onde realmente há fogo, diminuindo, assim, a taxa de verdadeiros positivos para 96,15%. O mesmo acontece quando  $N = 16, 32$ , reduzindo a TVP para

76,92% e 61,54%, respectivamente.

O SVM atingiu acurácia de 100,00% quando o tamanho do bloco era igual  $8 \times 8$ . O modelo foi capaz de classificar corretamente os 26 vídeos que possuem fogo, como também, todos os vídeos que possuem falsos positivos.

Os resultados da AST destacados em negrito apresentados na Tabela 7 são comparados com o detector espacial, YOLOv4, sem qualquer contexto temporal e, também, com a abordagem YOLOv4+AT (análise temporal), baseada na variação das distâncias euclidianas entre os centroides das caixas delimitadoras (ROI's) detectadas pelo YOLOv4. Ambas as abordagens foram aplicadas à detecção de incêndios em Venâncio et al. (2021).

Tabela 8 – Métricas do conjunto de testes: impacto da AST na detecção de fogo.

Abordagens	TVP	TFP	Acc (%)	F1-Score	Atraso
YOLO (Venâncio et al., 2021)	100,00	100,00	50,00	0,67	3,72
YOLO+AT (Venâncio et al., 2021)	84,62	11,54	86,54	0,86	+0,15
YOLO+AST <sub>ArrDec</sub> <sub>8×8</sub>	100,00	11,54	94,23	0,95	+0,59
YOLO+AST <sub>LDA</sub> <sub>16×16</sub>	100,00	7,69	96,15	0,96	+0,21
YOLO+AST <sub>Naive</sub> <sub>16×16</sub>	100,00	7,69	96,15	0,96	+0,21
YOLO+AST <sub>RegLog</sub> <sub>8×8</sub>	100,00	11,54	94,23	0,95	+0,68
YOLO+AST <sub>Knn</sub> <sub>4×4</sub>	100,00	3,85	98,08	0,98	+3,24
<b>YOLO+AST<sub>SVM</sub><sub>8×8</sub></b>	<b>100,00</b>	<b>0,00</b>	<b>100,00</b>	<b>1,00</b>	+0,54

Fonte: Autoria própria.

Na primeira linha da Tabela 8, o detector espacial, YOLOv4, avaliou corretamente todos os 26 vídeos que contêm eventos de incêndio (TVP=100%). Por outro lado, os vídeos que não possuem fogo, mas possuem eventos semelhantes ao fogo, foram classificados incorretamente (TFP=100%). O tempo médio para o YOLOv4 detectar possíveis eventos de incêndio foi de 3,72 segundos, com desvio padrão igual a 1,04.

As linhas subsequentes da Tabela 8 apresentam abordagens com contexto temporal. YOLOv4+AT reduziu a taxa de falso positivos em 88,46%, porém reduziu taxa de verdadeiros positivos em 15,38%, resultando em uma precisão de 86,54%. Reduzir a taxa de verdadeiros positivos é indesejável, uma vez que, caso haja incidência de fogo, a abordagem deve gerar o alarme para que providências sejam tomadas. Acrescentar AT ao processo de detecção implica, em média, em um acréscimo de 0,15 segundos (desvio padrão de  $\pm 0,05$ ).

Por outro lado, as abordagens que utilizam a AST são capazes de manter a TVP na sua totalidade, garantindo, a plena detecção dos eventos com ocorrência de incêndio. Estes resultados estão representados nas seis últimas linhas da tabela.

Destacado em negrito na Tabela 8, apresenta-se a abordagem AST usando o classificador SVM, considerando o tamanhos blocos  $N = 8$ , aplicada na sequência do

detector espacial, representada por YOLOv4+AST<sub>SVM\_8x8</sub>. Essa abordagem apresentou a menor taxa de falsos positivos, conseguindo classificar corretamente todos os 26 vídeos contendo alarmes falsos. Dessa forma, a abordagem atingiu 100,00% de acurácia.

Em relação ao custo computacional, a AST apresentou tempo médio de processamento igual a 0,54 segundos para o classificador SVM. Por outro lado, a abordagem de contexto temporal (TA), descrita em Venâncio et al. (2021), apresenta tempo médio de 0,15 segundos, sendo 3,6 vezes mais rápida. Vale destacar que a metodologia apresentada para a AST não utilizou processamento paralelo.

As Figuras 31(a) e 32(a) mostram as ROIs delimitadas pelo detector espacial em dois vídeos de teste. A primeira figura representando um falso positivo e a segunda, um verdadeiro positivo. Na Figura 31(a), percebe-se que o detector espacial interpreta um reflexo do sol na lente da câmera como sendo um possível foco de incêndio. Realmente, quando esse vídeo é analisado, o reflexo aliado ao movimento natural das folhas da árvore no plano de fundo, tornam essa região muito parecida com uma região de fogo. Em virtude dessas características, na análise das séries temporais desta região, quando  $N = 1, 2$ , o classificador SVM identifica movimentos nas séries temporais parecidos com os que ocorrem nas bordas da região de fogo, proporcionado pelo movimento das folhas da árvore no plano de fundo. No entanto, quando  $N = 4, 8, 16, 32$ , a abordagem é capaz de classificar a região corretamente, ou seja, sem a incidência de fogo.

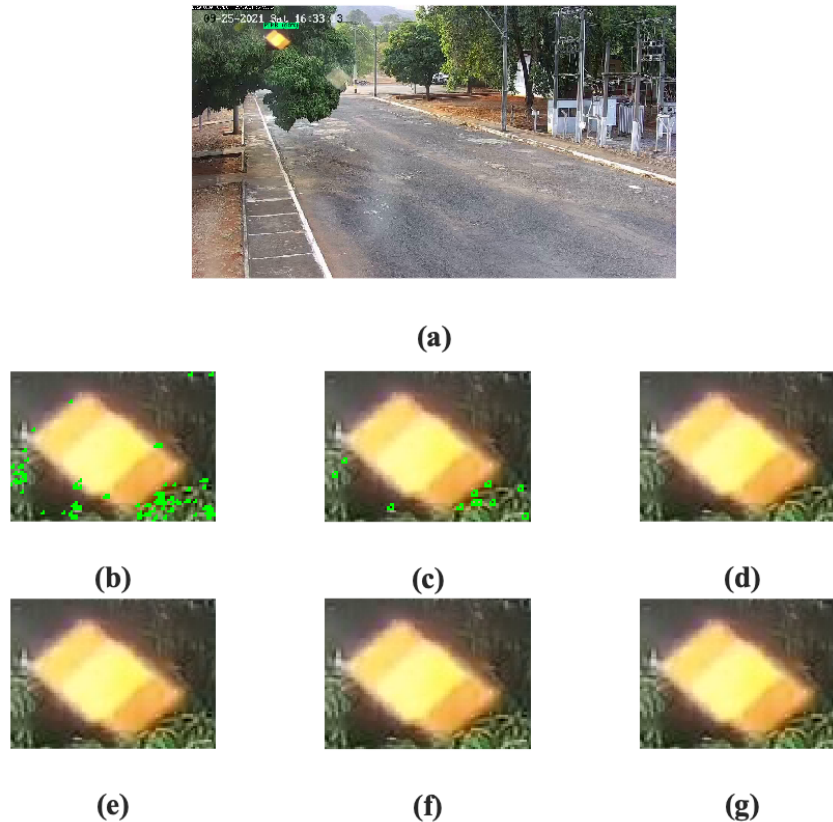
Por outro lado, a Figura 32 mostra a ROI de um vídeo contendo fogo e detectado pelo algoritmo YOLO. Para  $N = \{1, 2\}$  o classificador SVM tende a detectar toda a extensão do fogo contido na ROI. Quando  $N = \{4, 8\}$  o classificador detecta a maioria da região do fogo. Para  $N = 16$ , a abordagem proposta detecta os blocos que apresentam as maiores variações do movimento das chama de fogo. Para  $N = 32$  não foi possível detectar nenhum evento de incêndio.

Várias outras figuras mostrando as detecções das regiões com grandes possibilidades de haver fogo podem ser vistas no capítulo Apêndice A.

## 5.4 Comentários finais

Neste capítulo foram apresentados os experimentos conduzidos para validação de uma abordagem robusta e computacionalmente factível utilizando séries temporais para detectar incêndios, sendo capaz de ser realizada em computadores pessoais sem alto poder computacional. A abordagem AST é utilizada como pós-processamento do detector espacial (YOLO). O algoritmo YOLO detecta a região de interesse e a AST realiza a análise temporal apenas da região de interesse e não de todo cenário que compõe o vídeo analisado. Essa abordagem permite que a região analisada pela AST seja, em média, nove vezes menor que dimensão total de um vídeo.

Figura 31 – Quadro de um vídeo de teste que apresenta um evento parecido com fogo (falso positivo). (a) Detecção da ROI pelo detector espacial, YOLO, com possibilidade de ser incêndio. Ampliação da ROI e análise das séries temporais (AST) usando o classificador SVM para diferentes tamanhos de blocos: (b)  $N = 1$ , (c)  $N = 2$ , (d)  $N = 4$ , (e)  $N = 8$ , (f)  $N = 16$  e (g)  $N = 32$ . Os blocos da cor verde foram identificados como fogo pelo classificador SVM.

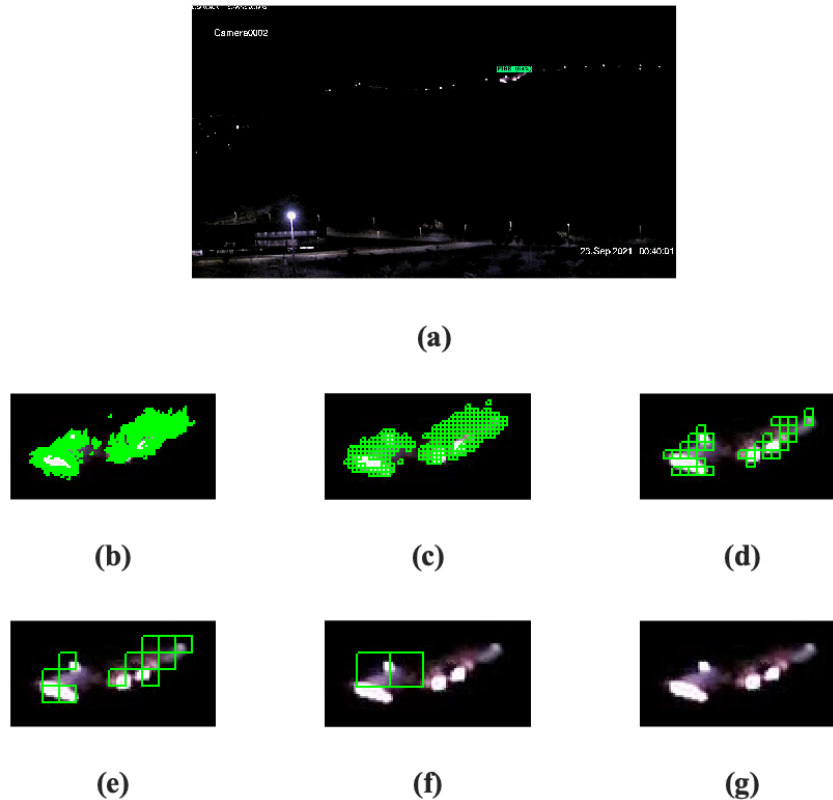


Fonte: Autoria própria.

A AST foi analisada a partir de seis algoritmos de aprendizado de máquina. Além disso, avaliou-se a resultados para tamanhos de blocos diferentes. Os resultados das combinações entre algoritmos e tamanhos dos blocos foi apresentada nas Tabelas 5 e 7, em que são mostrados os tempos e o desempenho. Tamanhos de blocos maiores implicam menores tempos computacionais para analisar as séries temporais. Os classificadores com tamanhos de blocos menores apresentam falsas detecções devido a cores semelhantes com a do fogo e ao movimento das folhas, quando o plano de fundo é composto por árvores. A junção de cores do fogo e movimento do plano de fundo gera detecções erradas, pois o algoritmo interpreta essas características como sendo uma imitação da cintilação natural das chamas de fogo. Esse problema tende a ser suprimido quando o valor de  $N$  aumenta, conforme pode ser visto na Figura 31.

Vale ressaltar também que a AST pode ser aplicada em vídeos com diferentes

Figura 32 – Quadro de um vídeo de teste que apresenta incêndio em estágio inicial (verdadeiro positivo). (a) Detecção da ROI contendo fogo em ambiente noturno pelo detector espacial, YOLO. Ampliação da ROI e análise das séries temporais (AST) usando o classificador SVM para diferentes tamanhos de blocos: (b)  $N = 1$ , (c)  $N = 2$ , (d)  $N = 4$ , (e)  $N = 8$ , (f)  $N = 16$  e (g)  $N = 32$ . Os blocos da cor verde foram identificados como fogo pelo classificador SVM.



Fonte: Autoria própria.

dimensões. Isso proporciona trabalhar em conjunto com diversos sistemas de detecção e monitoramento sem que seja necessário redimensionar os vídeos.

A partir do classificador SVM e com o tamanho do bloco igual a  $8 \times 8$  pixels, foi possível classificar corretamente 100,00% do conjunto de vídeos de teste.

Um ponto muito importante a ser discutido é o custo computacional para processar o detector espacial e adicionalmente o AST. Em relação ao detector espacial, seja ele o YOLOv4, ou qualquer outra abordagem que utilize aprendizado profundo, o custo computacional é uma característica inerente ao sistema de detecção de incêndio. Isto é, sempre haverá um custo computacional para analisar a presença de fogo em um vídeo. O YOLOv4 tem bom desempenho, mas paga um alto custo computacional na etapa de treinamento. O uso de sistemas mais leves melhora o custo computacional ao custo de

uma queda de desempenho.

Dessa maneira, o uso de um método computacionalmente mais leve para detecção da ROI pode ser um problema, devido ao potencial aumento de casos de falsos negativos. Neste caso, é possível que o uso da AST compense pelo menos parcialmente essa queda de desempenho identificando os casos de falsos positivos. No entanto, o importante não é utilizar um detector espacial muito rápido, mas sim um que seja capaz de detectar corretamente os verdadeiros eventos de incêndios (VP), mesmo que a taxa de falsos positivos aumente. E dessa forma, o uso da AST pode compensar a queda de desempenho proporcionada pela alta taxa de falsos positivos.

Em relação ao tempo de processamento causado pela abordagem temporal apresentada em [Venâncio et al. \(2021\)](#), apresentou um tempo de processamento aproximadamente 3,6 vezes menor que a AST. No entanto, o ganho que o AST proporciona para o sistema de detecção é bastante significativo, pois reduz em até 100,00% o número de falsos alarmes, levando-se em consideração o conjunto de testes utilizado. Em média, a AST acarreta em um atraso de apenas 0,54 segundos, o que não impede a detecção de fogo em tempo real.



# Capítulo 6

## Conclusão

Os métodos baseados em redes neurais convolucionais aplicados na detecção automática de fogo são os mais notáveis e têm sido bastante explorados nos últimos anos. Devido à sua capacidade de detectar e localizar as regiões de fogo em uma imagem a partir das características espaciais, esses métodos têm ganhado preferência em relação aos métodos clássicos de extração das características baseados em regras. No entanto, as redes neurais convolucionais exigem um grande conjunto de imagens para o treinamento e um alto poder computacional. Conforme mostrado no Capítulo 2, por mais que essa metodologia tenha um grande potencial em detectar automaticamente o fogo, ainda assim apresenta limitações em detectar imagens onde há determinados objetos que se parecem ou que tenham comportamento parecido com o fogo, gerando falsos alarmes.

Diante das dificuldades que os métodos mais avançados de detecção automática de fogo têm apresentado, a abordagem apresentada neste estudo vem ao encontro do esforço para se tentar diminuir a taxa de falsos alarmes. Sendo assim, o método da Análise de séries Temporais (AST) é utilizado para detecção de fogo em vídeos a partir do comportamento temporal. A premissa básica é que o fogo, diferentemente de objetos estáticos, apresenta movimento característico, denominado cintilação. Uma vez que um sistema de detecção automático delimite uma região de interesse, analisar o comportamento temporal dessa região é muito importante. A AST realiza essa análise temporal a partir de séries temporais dos *pixels*. No Capítulo 3 foi apresentado como um vídeo pode ser representado por séries temporais.

Para que a AST pudesse ser implementada foi necessário constituir dois conjuntos de dados compostos por vídeos: um utilizado para treinamento/validação dos modelos de aprendizado de máquina e outro para realizar o teste desses modelos.

A abordagem AST foi aplicada às regiões de interesse selecionadas pelo detector espacial YOLO que é baseado em uma rede neural convolucional. Os resultados foram expostos no Capítulo 5. Essas regiões de interesse foram divididas em blocos. Então, foram agrupados conjuntos de séries temporais de mesmo tamanho, formando seis diferentes

conjuntos. Essa abordagem se mostrou necessária para investigar a relação da assertividade do método *versus* o custo computacional.

Demonstrou-se que, quanto menor o tamanho do bloco, maior é seu tempo de processamento. Por outro lado, os blocos menores apresentam as maiores taxas de falsos positivos, apesar de conseguirem manter as taxas de verdadeiros positivos.

Tamanhos de blocos maiores resultam em menores tempo de processamento. A taxa de falsos positivos tende a diminuir, enquanto a taxa de verdadeiros positivos apresenta decréscimo.

O ponto ótimo é obtido quando o tamanho do bloco  $N$  é igual a  $8 \times 8$  *pixels* utilizando o classificador SVM, que proporciona à AST manter a taxa de verdadeiros positivos, diminuir a taxa de falsos negativos e adicionar uma média de apenas 0,54 segundos de processamento na detecção do evento.

A abordagem proposta permite uma redução na totalidade da taxa de falsos positivos sem diminuir a taxa de verdadeiros positivos (100%) a um baixo custo computacional, resultando em um atraso da ordem de centenas de milissegundos. Esses resultados não impedem o uso da AST em sistemas reais de detecção de fogo. O AST foi capaz de aumentar a acurácia do sistema de detecção de 50% para 100%. A abordagem mostrou-se robusta a luzes artificiais, luzes de veículos e objetos que possuem cores parecidas com o fogo. Além disso, a AST é capaz de trabalhar com diferentes resoluções de vídeo. Ela é robusta à saturação de cores em regiões de incêndio e também pode ser aplicada a vídeos com taxas superiores a 2 fps. No entanto, quanto maior a dimensão da ROI, maior o custo computacional. Da mesma forma, trabalhar com vídeos que possuem alta taxa de quadros ( $>2$  fps) também torna a abordagem computacionalmente mais cara.

Por outro lado, vídeos gravados por câmeras que operam com movimento horizontal, vertical ajuste da distância focal não são capazes de serem classificados usando AST devido ao fato de que não irá haver o monitoramento do mesmo ponto, impossibilitando a análise das características temporais. Além disso, é importante garantir que câmeras estáticas não apresentem oscilações provocadas por ventos, ao custo de comprometer a análise temporal.

## 6.1 Trabalhos futuros

Como trabalho futuro será interessante analisar o uso das séries temporais aplicadas à detecção de fumaça. Poder-se-á investigar se as séries temporais das regiões de fumaça possuem características próprias que permitam ser identificadas por um classificador simples, assim como ocorreu para as séries temporais de fogo. Detectar fumaça é mais complexo que o fogo dadas as características da fumaça, como, por exemplo, o formato, o

movimento e as diversas tonalidades que pode possuir variando de branca a preta. Por essas razões, a taxa de falsos positivos na detecção da fumaça costuma ser alta e o uso da AST poderá ajudar a diminuí-la.

Outro ponto interessante será implementar a AST usando processamento paralelo, para que ela seja capaz de lidar com ROIs de dimensões maiores e taxas de vídeos mais elevadas mantendo o tempo de execução computacional. Além disso, será possível avaliar quadros completos, e não apenas a ROI, permitindo identificar falsos negativos, caso estes não sejam detectados pelo detector espacial.

A análise de séries temporais mostrou-se uma forma eficiente de se incorporar informações não utilizadas pelo algoritmo YOLO usado para classificação de padrões espaciais contidos em imagens.

# Referências

- A. Enis Cetin (2021). Computer vision based fire detection dataset. <http://signal.ee.bilkent.edu.tr/VisiFire/>.
- Abbott, D. (2014). *Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst*. Wiley Publishing, 1 edition.
- Abid, F. (2020). A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology*, 57:559—590.
- Aguirre, L. A. (2007). *Introdução à Identificação de Sistemas—Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 3a edition.
- Alkhatib, A. A. A. (2014). A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, 10(3):597368.
- Arrue, B. C., Ollero, A., e De Dios, J. M. (2000). An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intelligent Systems and Their Applications*, 15(3):64–73.
- Avazov, K., Mukhiddinov, M., Makhmudov, F., e Cho, Y. I. (2022). Fire detection method in smart city environments using a deep-learning-based approach. *Electronics*, 11(1).
- Barmpoutis, P., Dimitropoulos, K., e Grammalidis, N. (2014). Smoke detection using spatio-temporal analysis, motion modeling and dynamic texture recognition. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 1078–1082.
- Barnich, O. e Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724.
- Beutling, A. (2009). *Modelagem do comportamento do fogo com base em experimentos laboratoriais e de campo*. PhD thesis, Universidade Federal do Paraná.
- Bhogal, G. S. e Rawat, A. K. (2020). Analysis on smoke detection techniques. In *Smart Intelligent Computing and Applications*, pages 163–180. Springer.
- Bochkovskiy, A., Wang, C.-Y., e Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

- Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- Borges, P. V. K. e Izquierdo, E. (2010). A probabilistic approach for vision-based fire detection in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(5):721–731.
- Borges, P. V. K., Mayer, J., e Izquierdo, E. (2008). Efficient visual fire detection applied for video retrieval. In *2008 16th European Signal Processing Conference*, pages 1–5. IEEE.
- Bortolini, D. E., Santos, A. L., Magalhães, H. A., Yehia, H. C., Vieira, D. A. G., e Nascimento, C. A. M. (2014). Detecção de incêndios utilizando persistência espaço-temporal e segmentação por cor em vídeo. In *Anais do XX Congresso Brasileiro de Automática*.
- Boser, B. E., Guyon, I. M., e Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT92*, New York, NY, USA. Association for Computing Machinery.
- Bu, F. e Gharajeh, M. S. (2019). Intelligent and vision-based fire detection systems: A survey. *Image and Vision Computing*, 91:103803.
- Buhus, E. R. e Apatean, A. (2017). State of the art: Methods for video based fire detection via stationary camera. *Acta Technica Napocensis*, 58(1):6.
- Cappellini, V., Mattii, L., e Mecocci, A. (1989). An intelligent system for automatic fire detection in forests. In *Third International Conference on Image Processing and its Applications, 1989.*, pages 563–570.
- Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., e Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4).
- Castro, C. L. d. e Braga, A. P. (2011). Aprendizado supervisionado com conjuntos de dados desbalanceados. *Sba: Controle & Automação Sociedade Brasileira de Automática*, 22:441 – 466.
- Catry, F. X., Rego, F. C., Bação, F. L., e Moreira, F. (2009). Modeling and mapping wildfire ignition risk in portugal. *International Journal of Wildland Fire*, 18(8):921–931.
- Celik, T. (2010). Fast and efficient method for fire detection using image processing. *ETRI Journal*, 32(6):881–890.
- Celik, T., Demirel, H., Ozkaramanli, H., e Uyguroglu, M. (2007a). Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, 18(2):176–185.

- Celik, T., Ozkaramanli, H., e Demirel, H. (2007b). Fire and smoke detection without sensors: Image processing based approach. In *2007 15th European Signal Processing Conference*, pages 1794–1798. IEEE.
- CEMIG, UFMG, Gaia, RaroLabs e UFVJM (2021). Apaga o fogo.
- Çetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habibo?lu, Y. H., Töreyn, B. U., e Verstockt, S. (2013). Video fire detection - Review.
- Çetin, A. E., Merci, B., Günay, O., Uğur Töreyn, B., e Verstockt, S. (2016). *Methods and techniques for fire detection : signal, image and video processing perspectives*. Elsevier.
- Chang, Y., Zhu, Z., Bu, R., Chen, H., Feng, Y., Li, Y., Hu, Y., e Wang, Z. (2013). Predicting fire occurrence patterns with logistic regression in heilongjiang province, china. *Landscape Ecology*, 28(10):1989–2004.
- Che-Bin Liu e Ahuja, N. (2004). Vision based fire detection. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 134–137.
- Chen, J., He, Y., e Wang, J. (2010). Multi-feature fusion based fast video flame detection. *Building and Environment*, 45(5):1113 – 1122.
- Chen, T., Yin, Y., Huang, S., e Ye, Y. (2006). The smoke detection for early fire-alarming system base on video processing. In *2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 427–430.
- Chi, R., Lu, Z.-M., e Ji, Q.-G. (2017). Real-time multi-feature based fire flame detection in video. *IET Image Processing*, 11(1):31–37.
- Chino, D. Y., Avalhais, L. P., Rodrigues, J. F., e Traina, A. J. (2015). Bowfire: detection of fire in still images by integrating pixel color and texture analysis. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 95–102. IEEE.
- Chunyu, Y., Jun, F., Jinjun, W., e Yongming, Z. (2010). Video fire smoke detection using motion and color features. *Fire technology*, 46(3):651–663.
- Company, S. C. B. (2010). *Stirling: a book on steam for engineers*. Kessinger Publishing.
- Cui, Y., Dong, H., e Zhou, E. (2008). An early fire detection method based on smoke texture analysis and discrimination. In *2008 Congress on Image and Signal Processing*, volume 3, pages 95–99.
- d. Breejen, E., Breuers, M. G., Cremer, F., Kemp, R. A. W., de Roos, M. A., Schutte, K., Jan S. de Vries Schutte, K., e De Vries, J. (1998). Autonomous forest fire detection. In on Fire, C. e Meteorology, F., editors, *International Conference on Forest Fire Research*.

- De Vasconcelos, M. P., Silva, S., Tome, M., Alvim, M., e Pereira, J. C. (2001). Spatial prediction of fire ignition probabilities: comparing logistic regression and neural networks. *Photogrammetric engineering and remote sensing*, 67(1):73–81.
- Dedeoğlu, N., Töreyn, B. U., Gudukbay, U., e Çetin, A. E. (2005). Real-time fire and flame detection in video. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii/669–ii/672 Vol. 2.
- Dimitropoulos, K., Barmpoutis, P., e Grammalidis, N. (2014). Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE transactions on circuits and systems for video technology*, 25(2):339–351.
- Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., e Vapnik, V. (1997). Support vector regression machines. In Mozer, M. C., Jordan, M. I., e Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press.
- Dunnings, A. J. e Breckon, T. P. (2018). Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1558–1562.
- Ebert, J. e Shipley, J. (2009). A computer vision based method for fire detection in color videos. *International journal of imaging*, 2(S09):22–34.
- El Naqa, I., Li, R., e Murphy, M. J. (2015). *What Is Machine Learning?*, pages 3–11. Springer International Publishing, Cham.
- Elsevier (2022). Compendex.
- Filho, O. e Neto, H. (1999). *Processamento digital de imagens*. BRASPORT.
- Filonenko, A., Kurnianggoro, L., e Jo, K.-H. (2017). Smoke detection on video sequences using convolutional and recurrent neural networks. In Nguyen, N. T., Papadopoulos, G. A., Jędrzejowicz, P., Trawiński, B., e Vossen, G., editors, *Computational Collective Intelligence*, pages 558–566, Cham. Springer International Publishing.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7:179–188.
- Foggia, P., Saggese, A., e 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*, 25(9):1545–1556.
- Foo, S. Y. (1996). A rule-based machine vision system for fire detection in aircraft dry bays and engine compartments. *Knowledge-Based Systems*, 9(8):531–540.

- Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.-M., Moreau, E., e Fnaiech, F. (2016). Convolutional neural network for video fire and smoke detection. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 877–882.
- Gaia, solutions on demand (2018). D-Fire: an image data set for fire and smoke detection. <https://git.io/JONna>. Acessado em fevereiro de 2021.
- Gaur, A., Singh, A., Kumar, A., e Kapoor, K. (2020). Video flame and smoke based fire detection algorithms: A literature review. *Fire Technology*, 56:1943–1980.
- Genovese, A., Labati, R. D., Piuri, V., e Scotti, F. (2011). Wildfire smoke detection using computational intelligence techniques. In *2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings*, pages 1–6.
- Gomes, P., Santana, P., e Barata, J. (2014). A vision-based approach to fire detection. *International Journal of Advanced Robotic Systems*, 11(9):149.
- Gomez-Rodriguez, F., Pascual-Pena, S., Arrue, B., e Ollero, A. (2002). Smoke detection using image processing. In *Proceedings of the IV International Conference on Forest Fire Research*.
- Gong, F., Li, C., Gong, W., Li, X., Yuan, X., Ma, Y., Song, T., e Versaci, M. (2019). A real-time fire detection method from video with multifeature fusion. *Intell. Neuroscience*, 2019.
- Gonzalez, R. C. e Woods, R. E. (2008). *Digital image processing*. Prentice Hall, Upper Saddle River, N.J.
- Govil, K., Welch, M. L., Ball, J. T., e Pennypacker, C. R. (2020). Preliminary results from a wildfire detection system using deep learning on remote camera images. *Remote Sensing*, 12(1).
- Gubbi, J., Marusic, S., e Palaniswami, M. (2009). Smoke detection in video using wavelets and support vector machines. *Fire Safety Journal*, 44(8):1110 – 1115.
- Gunay, O., Tasdemir, K., Töreyn, B. U., e Çetin, A. E. (2010). Fire detection in video using lms based active learning. *Fire technology*, 46(3):551–577.
- Gómez-Rodríguez, F., Arrue, B., e Ollero, A. (2003). Smoke monitoring and measurement using image processing. application to forest fires. *Proceedings of SPIE - The International Society for Optical Engineering*, 5094.
- Habiboğlu, Y. H., Günay, O., e Çetin, A. E. (2012). Covariance matrix-based fire and flame detection method in video. *Machine Vision and Applications*, 23(6):1103–1113.



- Hamad, A. e Tsumura, N. (2012). Background subtraction based on time-series clustering and statistical modeling. *Optical Review*, 19.
- Hamme, D. V., Veelaert, P., Philips, W., e Teelen, K. (2010). Fire detection in color images using markov random fields. In *International conference on advanced concepts for intelligent vision systems*, pages 88–97. Springer.
- Han, X.-F., Jin, J., Wang, M.-J., Jiang, W., Gao, L., e Xiao, L.-P. (2017). Video fire detection based on gaussian mixture model and multi-color features. *Signal, Image and Video Processing*, 11(8):1419–1425.
- Healey, G., Slater, D., Lin, T., Drda, B., e Goedeke, A. D. (1993). A system for real-time fire detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–606.
- Hornig, W.-B. e Peng, J.-W. (2001). Real-time fire detection from video: a preliminary report. In *14th IPPR Conference on Computer Vision, Graphics and Image Processing*.
- Hornig, W.-B. e Peng, J.-W. (2008). A fast image-based fire flame detection method using color analysis. *Tamkang Journal of Science and Engineering*, 11:273–285.
- Hosmer, D. W. e Lemeshow, S. (2000). *Applied logistic regression*. John Wiley and Sons.
- Hossin, M. e M.N, S. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5:01–11.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., e Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.
- Hu, Y. e Lu, X. (2018). Real-time video fire smoke detection by utilizing spatial-temporal convnet features. *Multimedia Tools and Applications*, 77.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., e Murphy, K. (2016). Speed/accuracy trade-offs for modern convolutional object detectors.
- Hunt, E. B. (2014). *Artificial intelligence*. Academic Press.
- INPE (2022). Programa Queimadas - Instituto Nacional de Pesquisas Espaciais. <https://queimadas.dgi.inpe.br/queimadas/portal>. Acessado em outubro de 2022.
- Jadon, A., Omama, M., Varshney, A., Ansari, M. S., e Sharma, R. (2019). Firenet: A specialized lightweight fire & smoke detection model for real-time iot applications.

- Johnson, R. A., Wichern, D. W., et al. (2014). *Applied multivariate statistical analysis*, volume 6. Pearson London, UK:.
- Khalil, A., Ur Rahman, S., Alam, F., e Khalil, I. (2021). Fire detection using multi color space and background modeling. *Fire Technology*, 57:1221—1239.
- Khondaker, A., Khandaker, A., e Uddin, J. (2020). Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique. *The International Arab Journal of Information Technology*, 17:947–953.
- Kim, D. e Wang, Y. (2009). Smoke detection in video. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 5, pages 759–763.
- Ko, B., Cheong, K.-H., e 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., e Nam, J.-Y. (2009). Fire detection based on vision sensor and support vector machines. *Fire Safety Journal*, 44(3):322 – 329.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95*, page 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kolesov, I., Karasev, P., Tannenbaum, A., e Haber, E. (2010). Fire and smoke detection in video with optimal mass transport based optical flow and neural networks. In *2010 IEEE International Conference on Image Processing*, pages 761–764. IEEE.
- Koller, D. e Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.
- Krizhevsky, A., Sutskever, I., e Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84—90.
- Krstinić, D., Stipaničev, D., e Jakovčević, T. (2009). Histogram-based smoke segmentation in forest fire detection system. *Information technology and control*, 38(3).
- Labati, R. D., Genovese, A., Piuri, V., e Scotti, F. (2013). Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):1003–1012.
- Lecun, Y., Bengio, Y., e Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

- Lee, B. e Han, D. (2007). Real-time fire detection using camera sequence image in tunnel environment. In Huang, D.-S., Heutte, L., e Loog, M., editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pages 1209–1220, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Lee, C.-Y., Lin, C.-T., Hong, C.-T., Su, M.-T., et al. (2012). Smoke detection using spatial and temporal analyses. *International Journal of Innovative Computing, Information and Control*, 8(7):4749–4770.
- Li, M. e Yuan, B. (2005). 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532.
- Li, S., Yan, Q., e Liu, P. (2020). An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Transactions on Image Processing*, 29:8467–8475.
- Lin, G., Zhang, Y., Xu, G., e Zhang, Q. (2019a). Smoke detection on video sequences using 3d convolutional neural networks. *Fire Technology*, 55(5):1827–1847.
- Lin, G., Zhang, Y., Xu, G., e Zhang, Q. (2019b). Smoke detection on video sequences using 3d convolutional neural networks. *Fire Technology*, 55.
- Lisboa, A., Venâncio, P., Neto, A., Magalhães, H., Do Nascimento, C., e Yehia, H. C. (2020). Monitoramento de incêndios por meio de redes e linhas aéreas de energia elétrica. *Revista da Universidade Federal de Minas Gerais*, 26:66–81.
- Luo, Y., Zhao, L., Liu, P., e Huang, D. (2018). Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimedia Tools and Applications*, 77.
- Mahdipour, E. e Dadkhah, C. (2014). Automatic fire detection based on soft computing techniques: review from 2000 to 2010. *Artificial intelligence review*, 42(4):895–934.
- Maheen, J. B. e Aneesh, R. P. (2019). Machine learning algorithm for fire detection using color correlogram. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, volume 1, pages 1411–1418, India.
- Mahmoud, M. e Ren, H. (2018). Forest fire detection using a rule-based image processing algorithm and temporal variation. *Mathematical Problems in Engineering*, 2018:1–8.
- Mahmoud, M. A. I. e Ren, H. (2019). Forest fire detection and identification using image processing and svm. *Journal of Information Processing Systems*, 15(1):159–168.
- Majid, S., Alenezi, F., Masood, S., Ahmad, M., Gündüz, E. S., e Polat, K. (2022). Attention based cnn model for fire detection and localization in real-world images. *Expert Systems with Applications*, 189:116114.

- Maksimović, M. e Vujovic, V. (2013). Comparative analysis of data mining techniques applied to wireless sensor network data for fire detection. *JITA - Journal of Information Technology and Applications (Banja Luka) - APEIRON*, 6.
- Maksymiv, O., Rak, T., e Peleshko, D. (2017). Real-time fire detection method combining adaboost, lbp and convolutional neural network in video sequence. In *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pages 351–353.
- Mao, W., Wang, W., Dou, Z., e Li, Y. (2018). Fire recognition based on multi-channel convolutional neural network. *Fire Technology*, 54:531–554.
- Marbach, G., Loepfe, M., e Brupbacher, T. (2006). An image processing technique for fire detection in video images. *Fire Safety Journal*, 41(4):285 – 289. 13th International Conference on Automatic Fire Detection, Duisburg, Germany.
- Matlani, P. e Shrivastava, M. (2018). A survey on video smoke detection. In Mishra, D. K., Nayak, M. K., e Joshi, A., editors, *Information and Communication Technology for Sustainable Development*, pages 211–222, Singapore. Springer Singapore.
- Memarzadeh, B. e Mohammadi, M. A. (2015). Fire detection using multi criteria image processing technique in video sequences. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 16(1):136–144.
- Mengxin, L., Xu, W., Xu, K., Fan, J., e Hou, D. (2013). Review of fire detection technologies based on video image. *Journal of Theoretical and Applied Information Technology*, 49:700–707.
- Millan-Garcia, L., Sanchez-Perez, G., Nakano, M., Toscano-Medina, K., Perez-Meana, H., e Rojas-Cardenas, L. (2012). An early fire detection algorithm using ip cameras. *Sensors*, 12(5):5670–5686.
- Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.
- Morettin, P. e Tolo, C. (2006). *Análise de séries temporais*. ABE - Projeto Fisher. Edgard Blucher.
- Muhammad, K., Ahmad, J., e Baik, S. W. (2018a). Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288:30–42. Learning System in Real-time Machine Vision.
- Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., e Baik, S. W. (2019a). Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(7):1419–1434.

- Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., e Baik, S. W. (2018b). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, 6:18174–18183.
- Muhammad, K., Khan, S., Elhoseny, M., Hassan Ahmed, S., e Wook Baik, S. (2019b). Efficient fire detection for uncertain surveillance environment. *IEEE Transactions on Industrial Informatics*, 15(5):3113–3122.
- NAMOZOV, A. e CHO, Y. (2018). An efficient deep learning algorithm for fire and smoke detection with limited data. *Advances in Electrical and Computer Engineering*, 18:121–128.
- Nguyen, M. D., Vu, H. N., Pham, D. C., Choi, B., e Ro, S. (2021). Multistage real-time fire detection using convolutional neural networks and long short-term memory networks. *IEEE Access*, 9:146667–146679.
- Noda, S. e Ueda, K. (1994). Fire detection in tunnels using an image processing method. In *Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference*, pages 57–62.
- Ojo, J. e Oladosu, J. A. (2014). Video-based smoke detection algorithm: A chronological survey. *Journal of Computer Engineering and Intelligent System, USA*, 5(7):38–50.
- Ojo, J. A. e Oladosu, J. A. (2018). Effective smoke detection using spatial-temporal energy and weber local descriptors in three orthogonal planes (wld-top). *Journal of Computer Science & Technology*, 18.
- Ongsulee, P. (2017). Artificial intelligence, machine learning and deep learning. In *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, pages 1–6, Bangkok, Thailand. IEEE.
- Petra, M., Abu Bakar, M., De Silva, L., e Mohamed Umar, M. (2017). State of the art of smoke and fire detection using image processing. *International Journal of Signal and Imaging Systems Engineering*, 10:22.
- Phillips, W., Shah, M., e da Vitoria Lobo, N. (2002). Flame recognition in video. *Pattern Recognition Letters*, 23(1):319 – 327.
- Piccinini, P., Calderara, S., e Cucchiara, R. (2008). Reliable smoke detection in the domains of image energy and color. In *2008 15th IEEE International Conference on Image Processing*, pages 1376–1379.
- Polednik, B. T. (2015). Detection of fire in images and video. In *Student Conference of Innovations, Technology and Science*.

- Qiu, T., Yan, Y., e Lu, G. (2012). An autoadaptive edge-detection algorithm for flame and fire image processing. *IEEE Transactions on Instrumentation and Measurement*, 61(5):1486–1493.
- Quintiere, J. (2016). *Principles of Fire Behavior*. CRC Press.
- Qureshi, W. S., Ekpanyapong, M., Dailey, M. N., Rinsurongkawong, S., Malenichev, A., e Krasotkina, O. (2016). Quickblaze: early fire detection using a combined video processing approach. *Fire technology*, 52(5):1293–1317.
- Rafiee, A., Dianat, R., Jamshidi, M., Tavakoli, R., e Abbaspour, S. (2011). Fire and smoke detection using wavelet analysis and disorder characteristics. In *2011 3rd International Conference on Computer Research and Development*, volume 3, pages 262–265.
- Rasbash, D. e Drysdale, D. (1982). Fundamentals of smoke production. *Fire Safety Journal*, 5(1):77 – 86.
- Rätsch, G. (2004). A brief introduction into machine learning. In *Proceedings of Chaos Communication Congress 21C3 Creative Commons*, Alemanha.
- Redmon, J., Divvala, S., Girshick, R., e Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.
- Rinsurongkawong, S., Ekpanyapong, M., e Dailey, M. (2012). Fire detection for early fire alarm based on optical flow video processing. In *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 1–4.
- Rodríguez, F. G., Arrue, B. C., e Ollero, A. (2003). Smoke monitoring and measurement using image processing. application to forest fires. *Proceedings of SPIE - The International Society for Optical Engineering*, 5094.
- Ruas, G. I. S. (2009). Sistema visual de segmentação de objetos robusto à iluminação para monitoramento automático de tráfego de veículos terrestres. Master's thesis, Universidade de Brasília.
- Russo, A. U., Deb, K., Tista, S. C., e Islam, A. (2018). Smoke detection method based on lbp and svm from surveillance camera. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pages 1–4.
- Saeed, F., Paul, A., Kumar, K., e Nayyar, A. (2020). Convolutional neural network based early fire detection. *Multimedia Tools and Applications*, 79.
- Safavian, S. R. e Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.

- Saponara, S., Elhanashi, A., e Gagliardi, A. (2021). Real-time video fire/smoke detection based on cnn in antifire surveillance systems. *Journal of Real-Time Image Processing*, 18:1–12.
- Saradhi, V., Karnik, H., e Mitra, P. (2005). A decomposition method for support vector clustering. In *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, pages 268–271, Chennai, India.
- Sathya, R. e Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2).
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.
- Seito, A., Gill, A., Silva, S., Pannoni, F., Ono, R., Carlo, U., e Silva, V. (2008). *A segurança contra incêndio no Brasil*. Projeto Editora.
- Shahid, M., Virtusio, J. J., Wu, Y.-H., Chen, Y.-Y., Tanveer, M., Muhammad, K., e Hua, K.-L. (2022). Spatio-temporal self-attention network for fire detection and segmentation in video surveillance. *IEEE Access*, 10:1259–1275.
- Sharma, J., Granmo, O.-C., Goodwin, M., e Fidje, J. (2017). Deep convolutional neural networks for fire detection in images. In *Communications in Computer and Information Science*, pages 183–193.
- Simonyan, K. e Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.
- So, A. T. P. e Chan, W. L. (1994). A computer vision based and fuzzy logic aided security and fire detection system. *Fire Technology*, 30(3):341–356.
- Son, G., Park, J.-S., Yoon, B.-W., e Song, J.-G. (2018). Video based smoke and flame detection using convolutional neural network. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 365–368.
- St-Charles, P., Bilodeau, G., e Bergevin, R. (2015). Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373.
- Stojanova, D., Panov, P., Kobler, A., Džeroski, S., e Taškova, K. (2006). Learning to predict forest fires with different data mining techniques. In *Conference on data mining and data warehouses (SiKDD 2006)*, Ljubljana, Slovenia, pages 255–258.

- Su, J. e Zhang, H. (2006). A fast decision tree learning algorithm. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1, AAAI'06*, page 500–505. AAAI Press.
- Subbiah, G., Abhishek, C., e Akshayanat, C. (2020). Machine vision based fire detection techniques: A survey. *Fire Technology*, 57.
- Suguna, N. e K.Thanushkodi (2010). An improved k-nearest neighbor classification using genetic algorithm. *International Journal of Computer Science Issues*, 7.
- Suklabaidya, S. e Das, I. (2021). *Framing Fire Detection System of Higher Efficacy Using Supervised Machine Learning Techniques*, pages 117–127. Springer Singapore, Singapore.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., e Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.
- Tao, C., Zhang, J., e Wang, P. (2016). Smoke detection based on deep convolutional neural networks. In *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIIICII)*, pages 150–153.
- Thou-Ho Chen, Cheng-Liang Kao, e Sju-Mo Chang (2003). An intelligent real-time fire-detection method based on video processing. In *IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, 2003. Proceedings.*, pages 104–111.
- Thou-Ho Chen, Ping-Hsueh Wu, e Yung-Chuen Chiou (2004). An early fire-detection method based on image processing. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 3, pages 1707–1710 Vol. 3.
- Torabian, M., Pourghassem, H., e Mahdavi-Nasab, H. (2021). Fire detection based on fractal analysis and spatio-temporal features. *Fire Technology*, 57(5):2583–2614.
- Töreyn, B. U. e Çetin, A. E. (2007). Online detection of fire in video. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–5. IEEE.
- Töreyn, B. U., Dedeoğlu, Y., e Çetin, A. E. (2005). Wavelet based real-time smoke detection in video. In *2005 13th European Signal Processing Conference*, pages 1–4.
- Töreyn, B. U., Dedeoğlu, Y., e Çetin, A. E. (2006). Contour based smoke detection in video using wavelets. In *2006 14th European Signal Processing Conference*, pages 1–5.
- Töreyn, B. U., Dedeoğlu, Y., Gudukbay, U., e Çetin, A. E. (2006). Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27:49–58.



- Töreyn, B. U., Dedeoğlu, Y., e Çetin, A. E. (2005). Flame detection in video using hidden markov models. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–1230.
- Toulouse, T., Rossi, L., Celik, T., e Akhloufi, M. (2016). Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. *Signal, Image and Video Processing*, 10(4):647–654.
- Truong, T. X. e Kim, J.-M. (2012). Fire flame detection in video sequences using multi-stage pattern recognition techniques. *Engineering Applications of Artificial Intelligence*, 25(7):1365 – 1372. Advanced issues in Artificial Intelligence and Pattern Recognition for Intelligent Surveillance System in Smart Home Environment.
- Ultimate Chase (2019). Fire Video Stock Footage Page. [http://ultimatechase.com/Fire\\_Video.htm](http://ultimatechase.com/Fire_Video.htm). Acessado em fevereiro de 2019.
- Venâncio, P., Lisboa, A., e Barbosa, A. (2022). An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications*.
- Venâncio, P. V. A. B. (2021a). Hyperparameter tuning of Convolutional Neural Networks implemented in the Darknet framework. <https://git.io/JLpE8>.
- Venâncio, P. V. A. B., Rezende, T. M., Lisboa, A. C., e Barbosa, A. V. (2021). Fire detection based on two-dimensional convolutional neural network and temporal analysis. In *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE.
- Venâncio, P. V. A. B. d. (2021b). Um sistema automático de detecção de incêndios baseado em aprendizado profundo para dispositivos de baixo poder computacional. Master’s thesis, Universidade Federal de Minas Gerais - UFMG.
- Verstockt, S. (2011). *Multi-modal video analysis for early fire detection*. PhD thesis, Ghent University.
- Verstockt, S., Lambert, P., Van de Walle, R., Merci, B., e Sette, B. (2009). State of the art in vision-based fire and smoke detection. In Luck, H. e Willms, I., editors, *International Conference on Automatic Fire Detection, 14th, Proceedings*, volume 2, pages 285–292. University of Duisburg-Essen. Department of Communication Systems.
- Vicente, J. e Guillemant, P. (2002). An image processing technique for automatically detecting forest fire. *International Journal of Thermal Sciences*, 41(12):1113–1120.

- Vijayalakshmi, S. R. e Muruganand, S. (2017). Smoke detection in video images using background subtraction method for early fire alarm system. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pages 167–171.
- Wang, H., Finn, A., Erdinc, O., e Vincitore, A. (2013). Spatial-temporal structural and dynamics features for video fire detection. In *2013 IEEE workshop on applications of computer vision (WACV)*, pages 513–519. IEEE.
- Wang, S., He, Y., Yang, H., Wang, K., e Wang, J. (2017). Video smoke detection using shape, color and dynamic features. *Journal of Intelligent & Fuzzy Systems*, 33(1):305–313.
- Wei, Y., Jie, L., Jun, F., e Yongming, Z. (2013). Color model and method for video fire flame and smoke detection using Fisher linear discriminant. *Optical Engineering*, 52(2):027205.
- Wen-Bing Horng, Jian-Wen Peng, e Chih-Yuan Chen (2005). A new image-based real-time flame detection method using color analysis. In *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.*, pages 100–105.
- Wu, H., Wu, D., e Zhao, J. (2019). An intelligent fire detection approach through cameras based on computer vision methods. *Process Safety and Environmental Protection*, 127:245–256.
- Xie, Y., Zhu, J., Cao, Y., Zhang, Y., Feng, D., Zhang, Y., e Chen, M. (2020). Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features. *IEEE Access*, 8:81904–81917.
- Xiong, Z., Caballero, R., Wang, H., Finn, A. M., Lelic, M. A., e Peng, P.-Y. (2007). Video-based smoke detection: possibilities, techniques, and challenges. In *IFPA, fire suppression and detection research and applications - a technical working conference (SUPDET)*, Orlando, FL.
- Xu, G., Zhang, Y., Zhang, Q., Lin, G., Wang, Z., Jia, Y., e Wang, J. (2019). Video smoke detection based on deep saliency network. *Fire Safety Journal*, 105:277–285.
- Xu, Z. e Xu, J. (2007). Automatic fire smoke detection based on image visual features. In *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pages 316–319.
- Yamagishi, H. e Yamaguchi, J. (1999). Fire flame detection algorithm using a color camera. In *MHS'99. Proceedings of 1999 International Symposium on Micromechatronics and Human Science (Cat. No.99TH8478)*, pages 255–260.
- Yamagishi, H. e Yamaguchi, J. (2000). A contour fluctuation data processing method for fire flame detection using a color camera. In *2000 26th Annual Conference of the*

- IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, volume 2, pages 824–829 vol.2.
- Yang, H., Jang, H., Kim, T., e Lee, B. (2019). Non-temporal lightweight fire detection network for intelligent surveillance systems. *IEEE Access*, 7:169257–169266.
- Yang, J., Chen, F., e Zhang, W. (2008). Visual-based smoke detection using support vector machine. In *2008 Fourth International Conference on Natural Computation*, volume 4, pages 301–305.
- Yasmin, R. (2009). Detection of smoke propagation direction using color video sequences. *International Journal of Soft Computing*, 4(1):45–48.
- Ye, W., Zhao, J., Wang, S., Wang, Y., Zhang, D., e Yuan, Z. (2015). Dynamic texture based smoke detection using surfacelet transform and hmt model. *Fire Safety Journal*, 73:91 – 101.
- Yin, M., Lang, C., Li, Z., Feng, S., e Wang, T. (2019). Recurrent convolutional network for video-based smoke detection. *Multimedia Tools Appl.*, 78(1):237—256.
- Yin, Z., Wan, B., Yuan, F., Xia, X., e Shi, J. (2017). A deep normalization and convolutional neural network for image smoke detection. *IEEE Access*, 5:18429–18438.
- Yu, C., Mei, Z., e Zhang, X. (2013). A real-time video fire flame and smoke detection algorithm. *Procedia Engineering*, 62:891 – 898. 9th Asia-Oceania Symposium on Fire Science and Technology.
- Yuan, F., Zhang, L., Wan, B., Xia, X., e Shi, J. (2019). Convolutional neural networks based on multi-scale additive merging layers for visual smoke recognition. *Machine Vision and Applications*, 30.
- Yuan, J., Hu, H. B., Yuan, W., Jia, Y., e Zhang, Y. M. (2014). Early image fire-detection based on maximum margin criterion. In *Mechanical Engineering and Instrumentation*, volume 526 of *Applied Mechanics and Materials*, pages 324–329. Trans Tech Publications Ltd.
- Zaidi, N. I. b., Lokman, N. A. A. B., Daud, M. R. B., Achmad, H., e Chia, K. A. (2015). Fire recognition using rgb and ycbcr color space. *ARPJ Journal of Engineering and Applied Sciences*, 10(21):9786–9790.
- Zaman, T., Hasan, M., Ahmed, S., e Ashfaq, S. (2018). Fire detection using computer vision. In *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 356–359.

- Zhang, F., Qin, W., Liu, Y., Xiao, Z., Liu, J., Wang, Q., e Liu, K. (2020). A dual-channel convolution neural network for image smoke detection. *Multimedia Tools and Applications*, 79.
- Zhang, Q., Xu, J., Xu, L., e 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. Atlantis Press.
- Zhang, Z., Zhao, J., Zhang, D., Qu, C., Ke, Y., e Cai, B. (2008). Contour based forest fire detection using fft and wavelet. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 760–763.
- Zhao, J., Zhang, Z., Han, S., Qu, C., Yuan, Z.-Y., e Zhang, D. (2011). Svm based forest fire detection using static and dynamic features. *Comput. Sci. Inf. Syst.*, 8(3):821–841.
- Zhong, Z., Wang, M., Shi, Y., e Gao, W. (2018). A convolutional neural network-based flame detection method in video sequence. *Signal, Image and Video Processing*, 12.
- Zivkovic, Z. e Van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780.

# Apêndice A

## Detecções do conjunto de teste

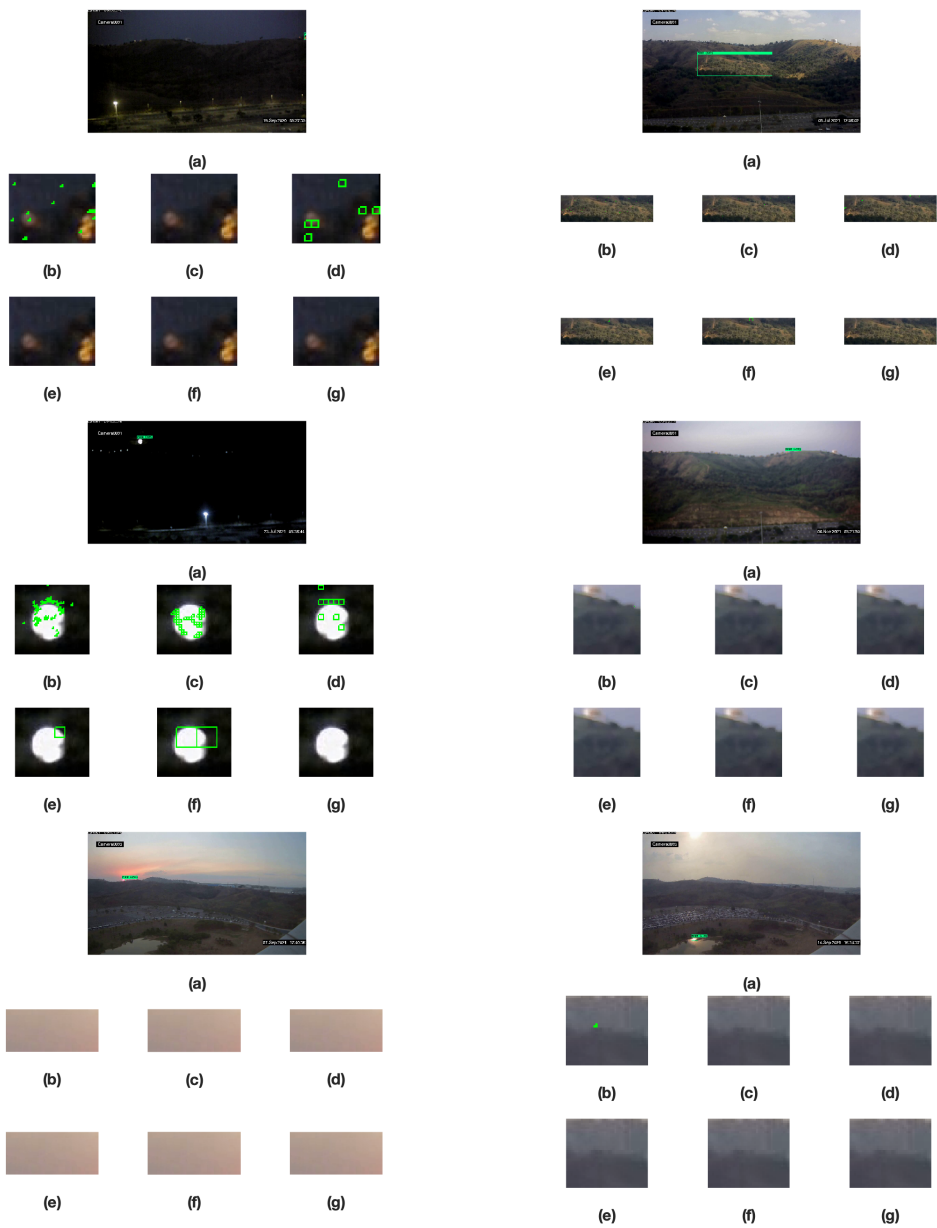
As detecções realizadas pelo método AST a partir de cada um dos classificadores usados são mostradas nesta seção. Devido à quantidade de vídeos usados no conjunto de teste, 52 vídeos, serão exibidas as detecções de apenas 30 vídeos para cada classificador, sendo que 15 são falsos positivos e o restante são verdadeiros positivos. Todas as as detecções realizadas estão disponíveis online para análise <sup>1</sup>.

As figuras mostradas representam quadros dos vídeos de conjunto de testes. Estas figuras são constituídas por sete sub-figuras, sendo que a primeira, (a), apresenta a detecção da ROI realizada pelo detector espacial, YOLO e as sub-figuras subsequentes mostram os resultados da aplicação usando a AST para diferentes tamanhos de blocos: (b)  $N = 1$ , (c)  $N = 2$ , (d)  $N = 4$ , (e)  $N = 8$ , (f)  $N = 16$  e (g)  $N = 32$ .

---

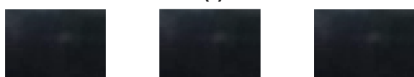
<sup>1</sup> [https://drive.google.com/drive/folders/1HHpxW\\_908VZJ8wRi-D0pCwoZd2nPKj11?usp=share\\_link](https://drive.google.com/drive/folders/1HHpxW_908VZJ8wRi-D0pCwoZd2nPKj11?usp=share_link)

# A.1 Árvore de decisões





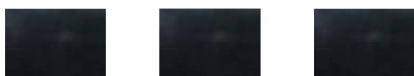
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



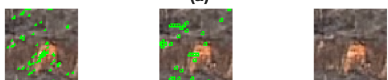
(e)

(f)

(g)



(a)



(b)

(c)

(d)



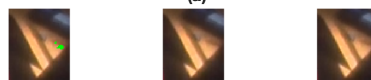
(e)

(f)

(g)



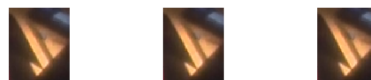
(a)



(b)

(c)

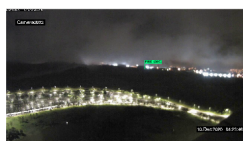
(d)



(e)

(f)

(g)



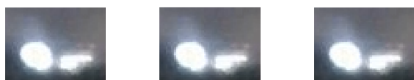
(a)



(b)

(c)

(d)



(e)

(f)

(g)



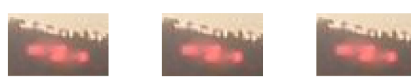
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



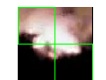
(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)

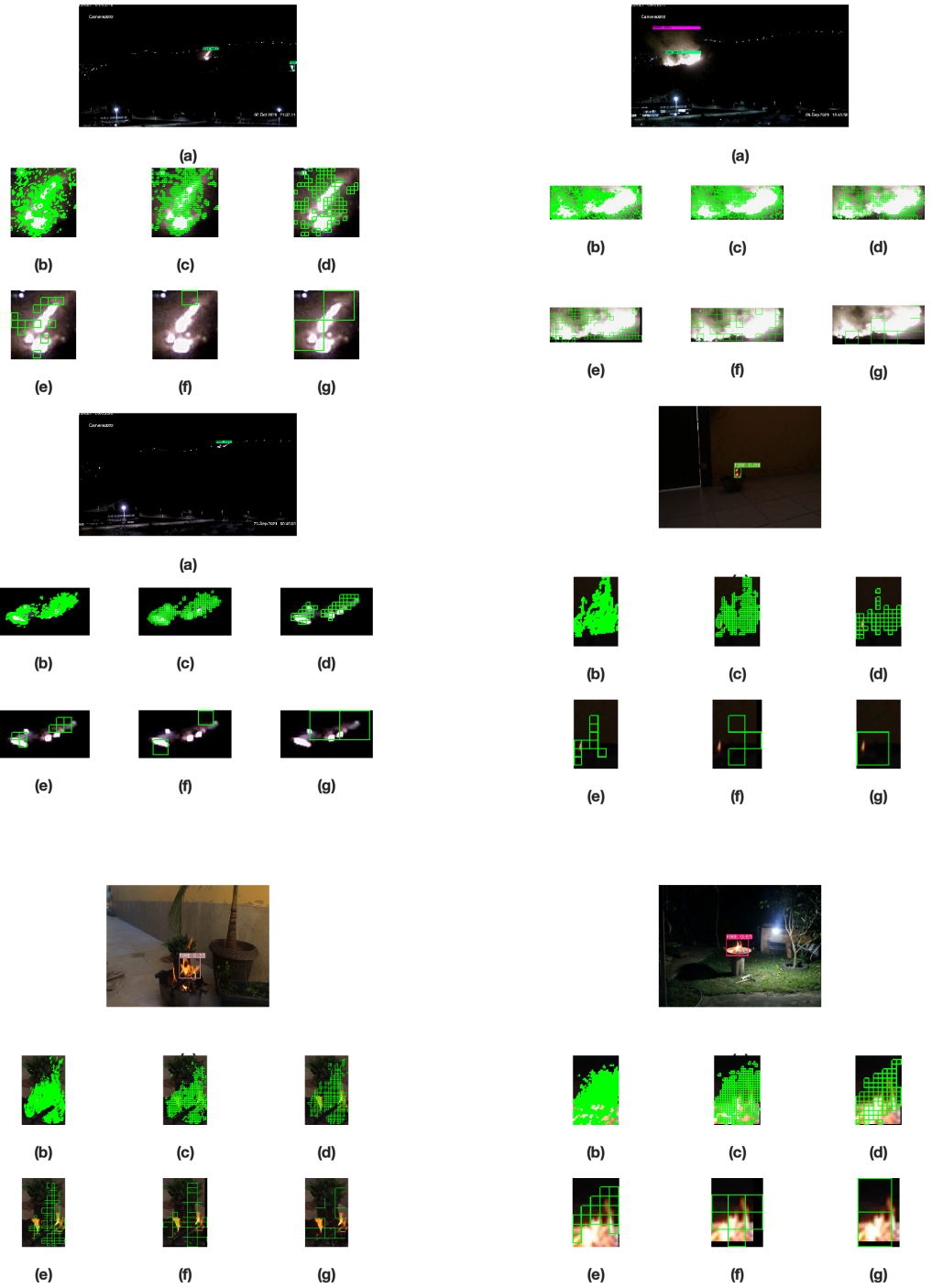


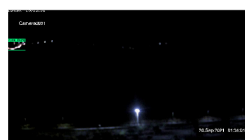
(f)



(g)







(a)



(b)



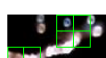
(c)



(d)



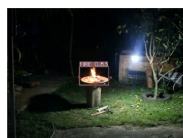
(e)



(f)



(g)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



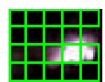
(b)



(c)



(d)



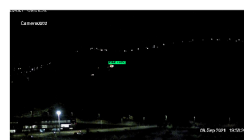
(e)



(f)



(g)



(a)



(b)



(c)



(d)



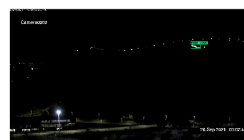
(e)



(f)



(g)



(a)



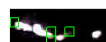
(b)



(c)



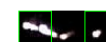
(d)



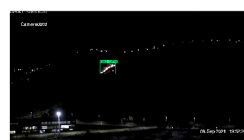
(e)



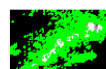
(f)



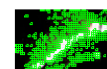
(g)



(a)



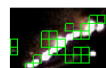
(b)



(c)



(d)



(e)

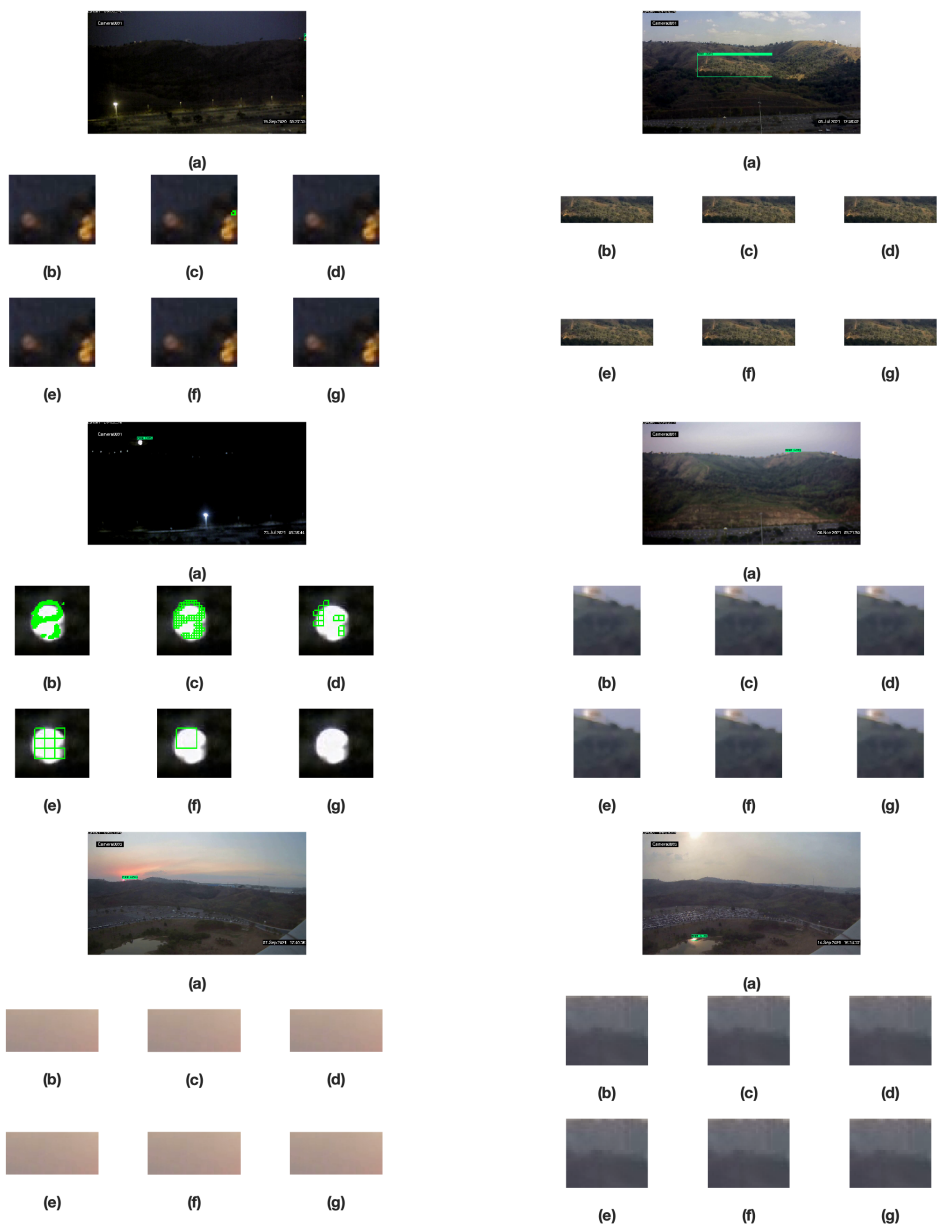


(f)



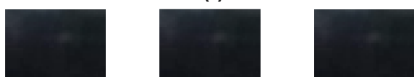
(g)

## A.2 LDA





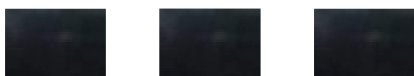
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



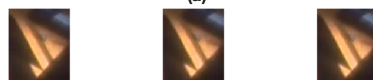
(e)

(f)

(g)



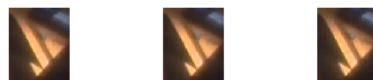
(a)



(b)

(c)

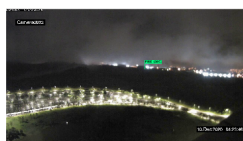
(d)



(e)

(f)

(g)



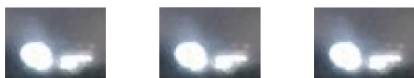
(a)



(b)

(c)

(d)



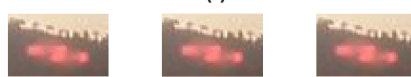
(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

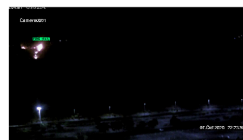
(d)



(e)

(f)

(g)



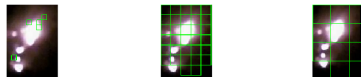
(a)



(b)

(c)

(d)



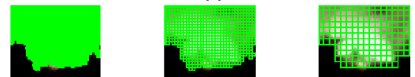
(e)

(f)

(g)



(a)



(b)

(c)

(d)



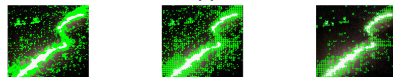
(e)

(f)

(g)



(a)



(b)

(c)

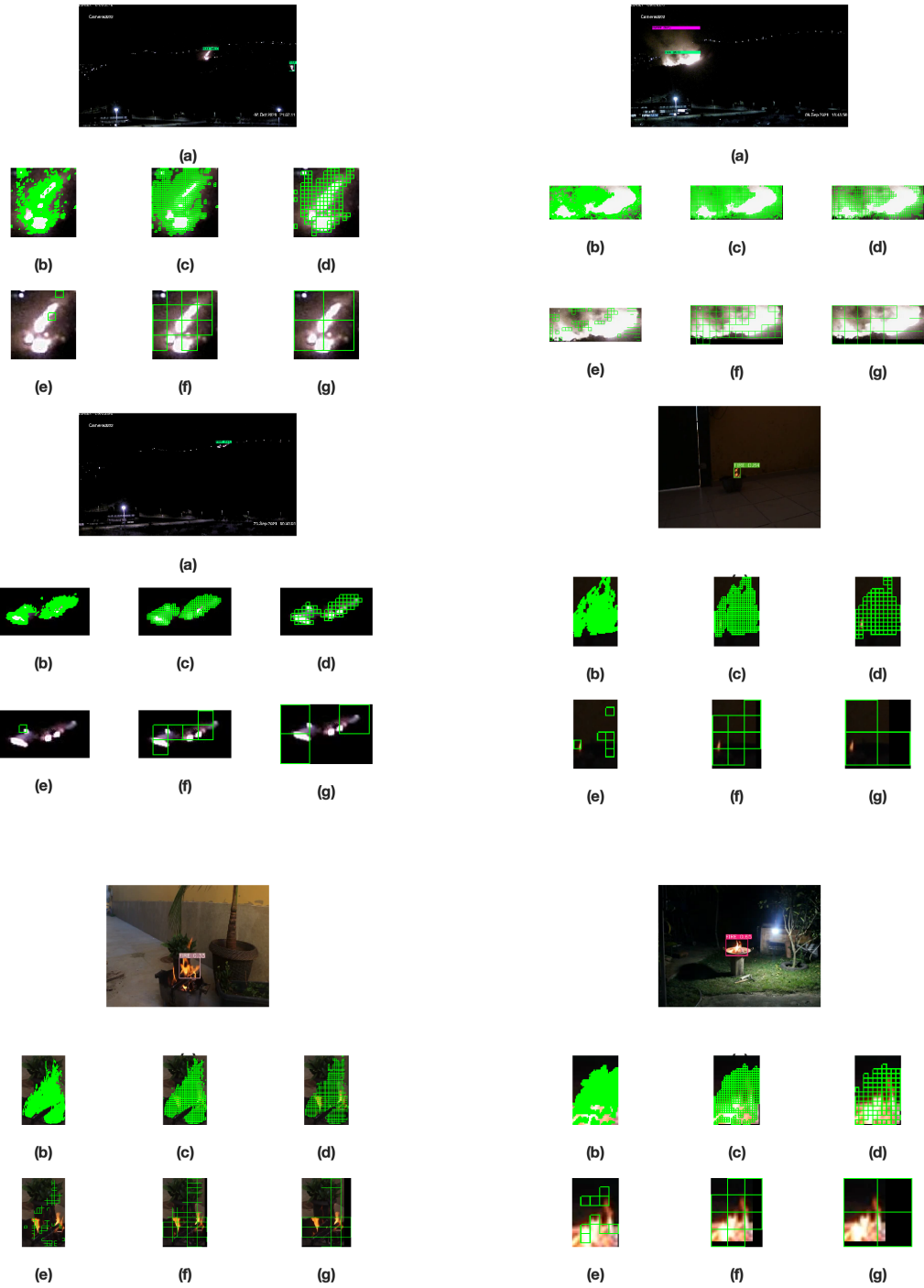
(d)

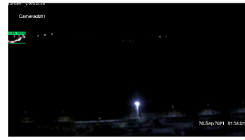


(e)

(f)

(g)





(a)



(b)

(c)

(d)



(e)

(f)

(g)



(b)

(c)

(d)



(e)

(f)

(g)



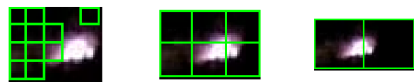
(a)



(b)

(c)

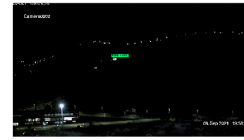
(d)



(e)

(f)

(g)



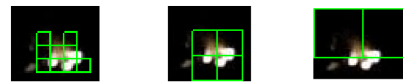
(a)



(b)

(c)

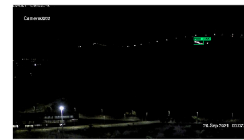
(d)



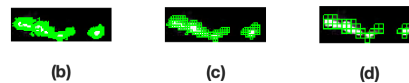
(e)

(f)

(g)



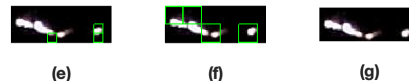
(a)



(b)

(c)

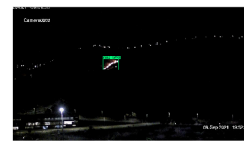
(d)



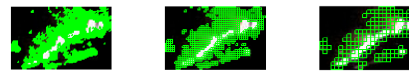
(e)

(f)

(g)



(a)



(b)

(c)

(d)

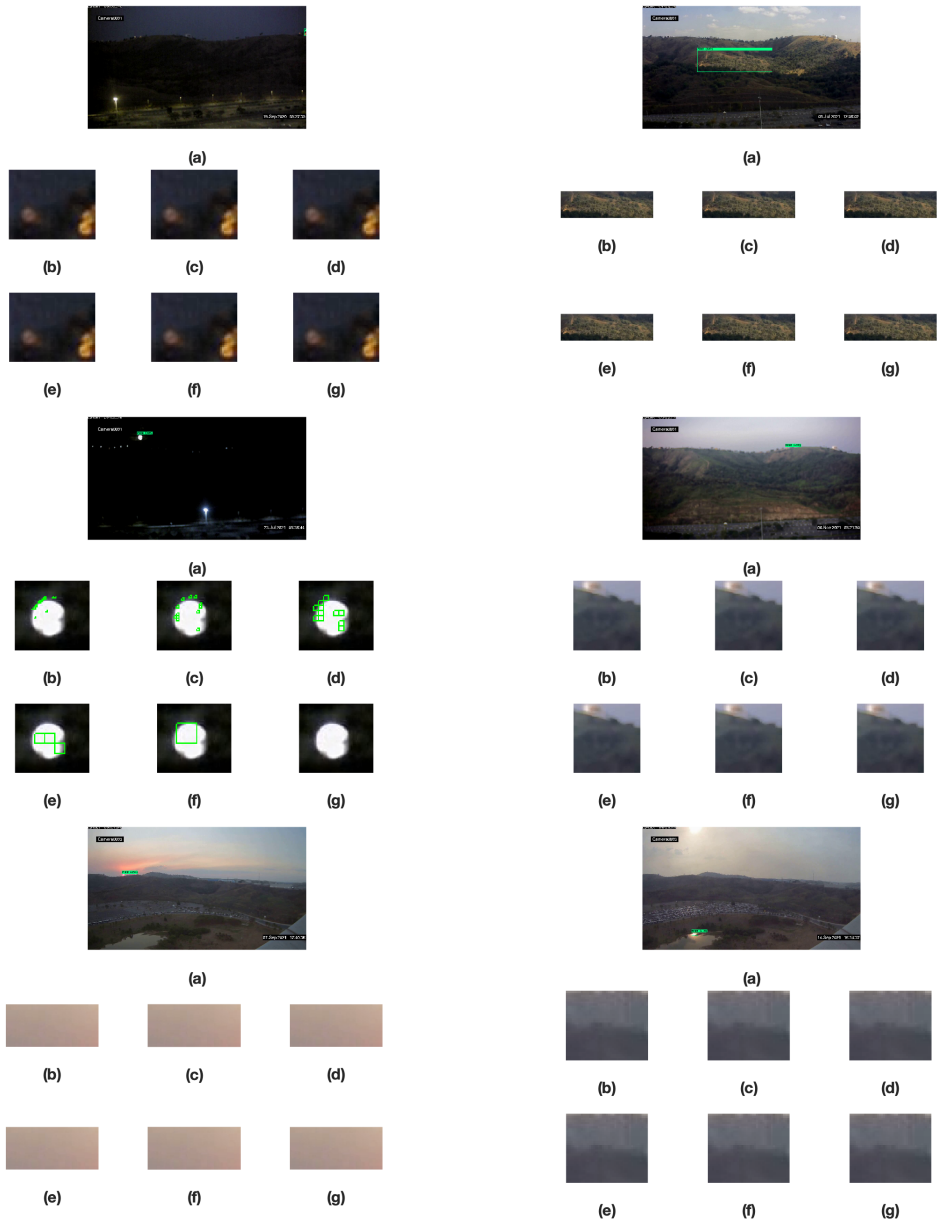


(e)

(f)

(g)

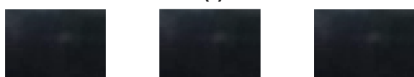
### A.3 Naïve







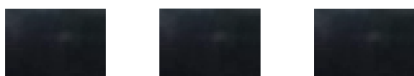
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



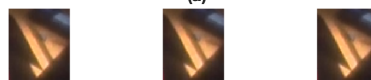
(e)

(f)

(g)



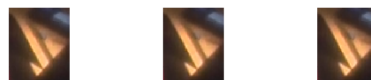
(a)



(b)

(c)

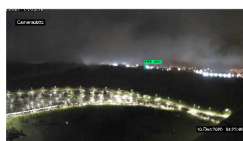
(d)



(e)

(f)

(g)



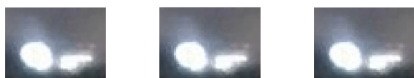
(a)



(b)

(c)

(d)



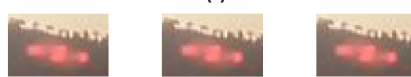
(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



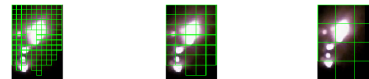
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



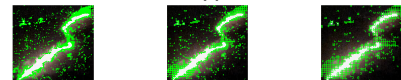
(e)

(f)

(g)



(a)



(b)

(c)

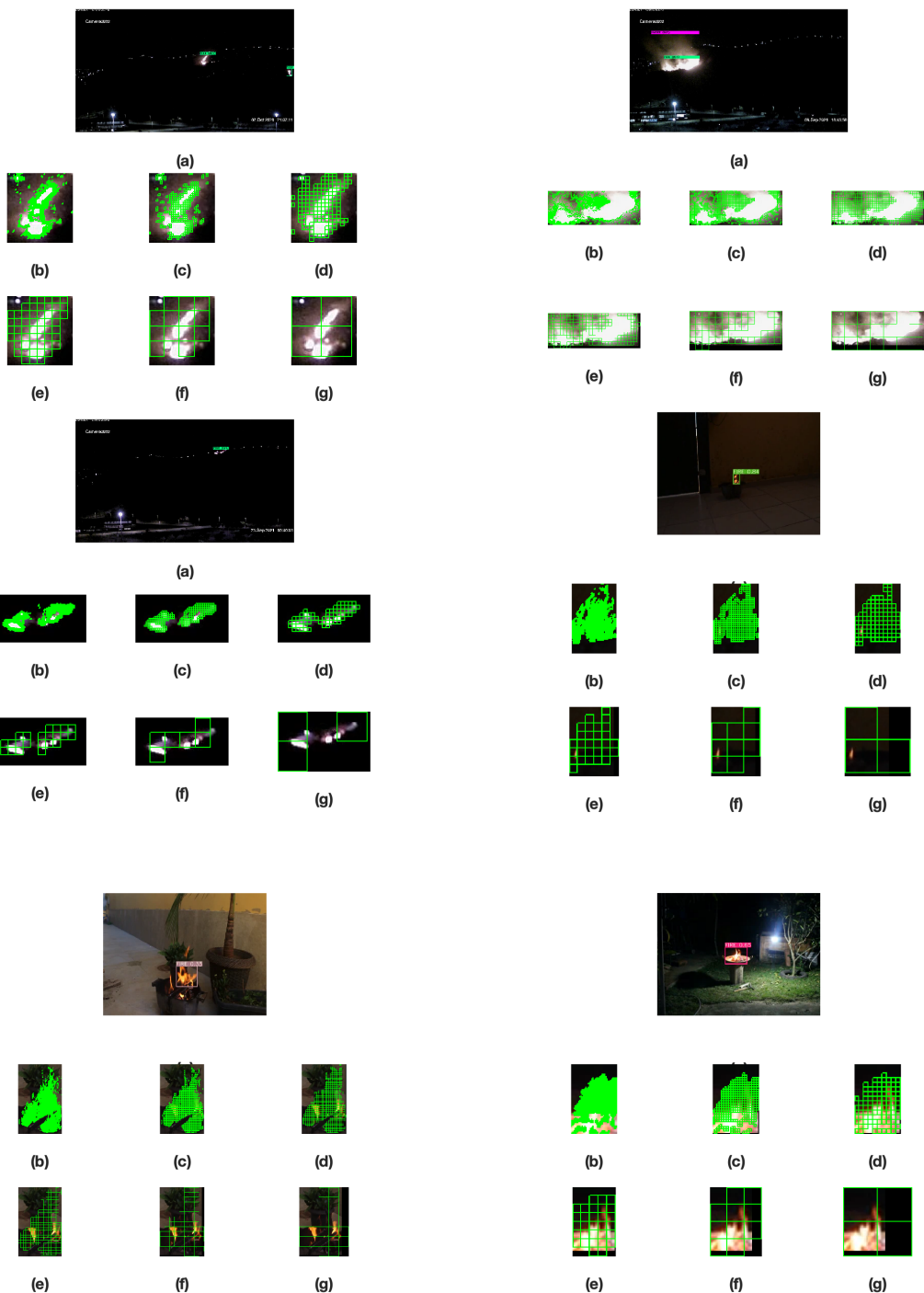
(d)

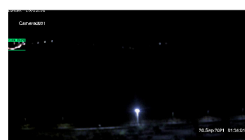


(e)

(f)

(g)





(a)



(b)

(c)

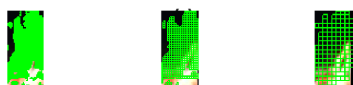
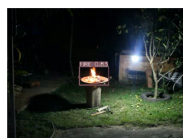
(d)



(e)

(f)

(g)



(b)

(c)

(d)



(e)

(f)

(g)



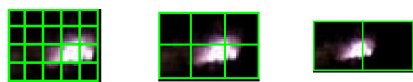
(a)



(b)

(c)

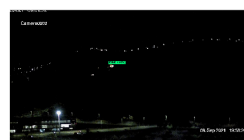
(d)



(e)

(f)

(g)



(a)



(b)

(c)

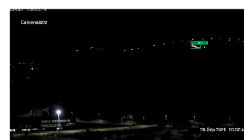
(d)



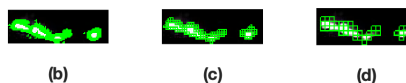
(e)

(f)

(g)



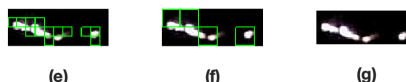
(a)



(b)

(c)

(d)



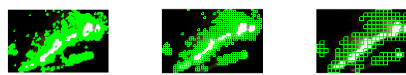
(e)

(f)

(g)



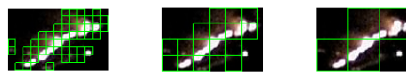
(a)



(b)

(c)

(d)

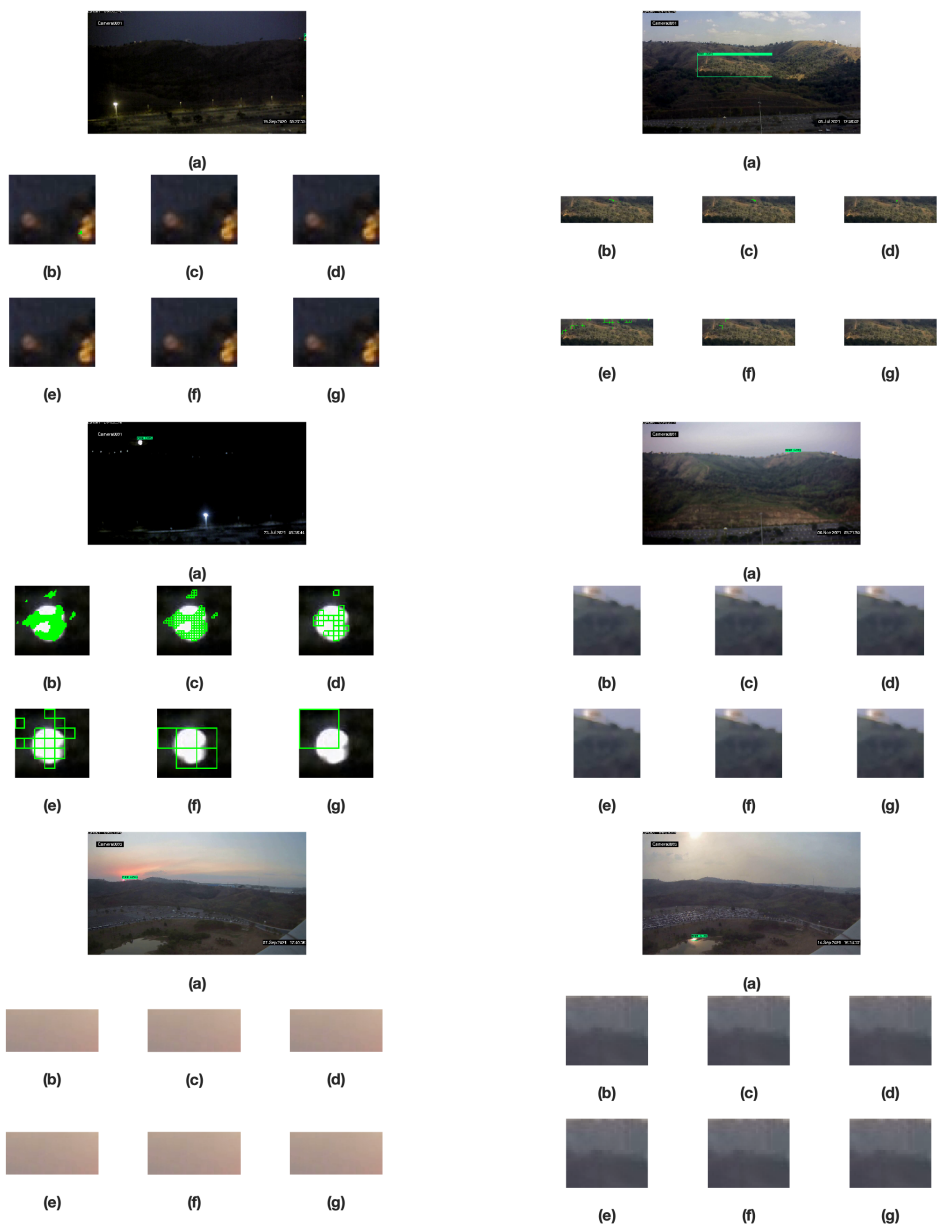


(e)

(f)

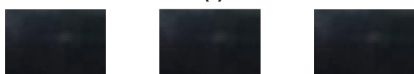
(g)

## A.4 Regressão logística





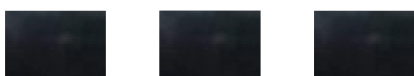
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



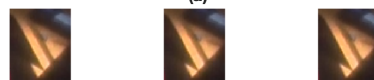
(e)

(f)

(g)



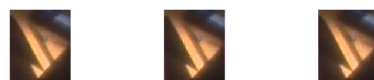
(a)



(b)

(c)

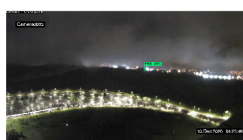
(d)



(e)

(f)

(g)



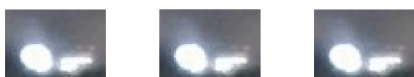
(a)



(b)

(c)

(d)



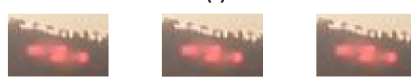
(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



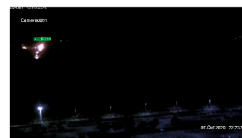
(e)



(f)



(g)



(a)



(b)



(c)



(d)



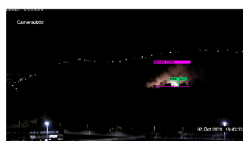
(e)



(f)



(g)



(a)



(b)



(c)



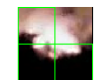
(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)

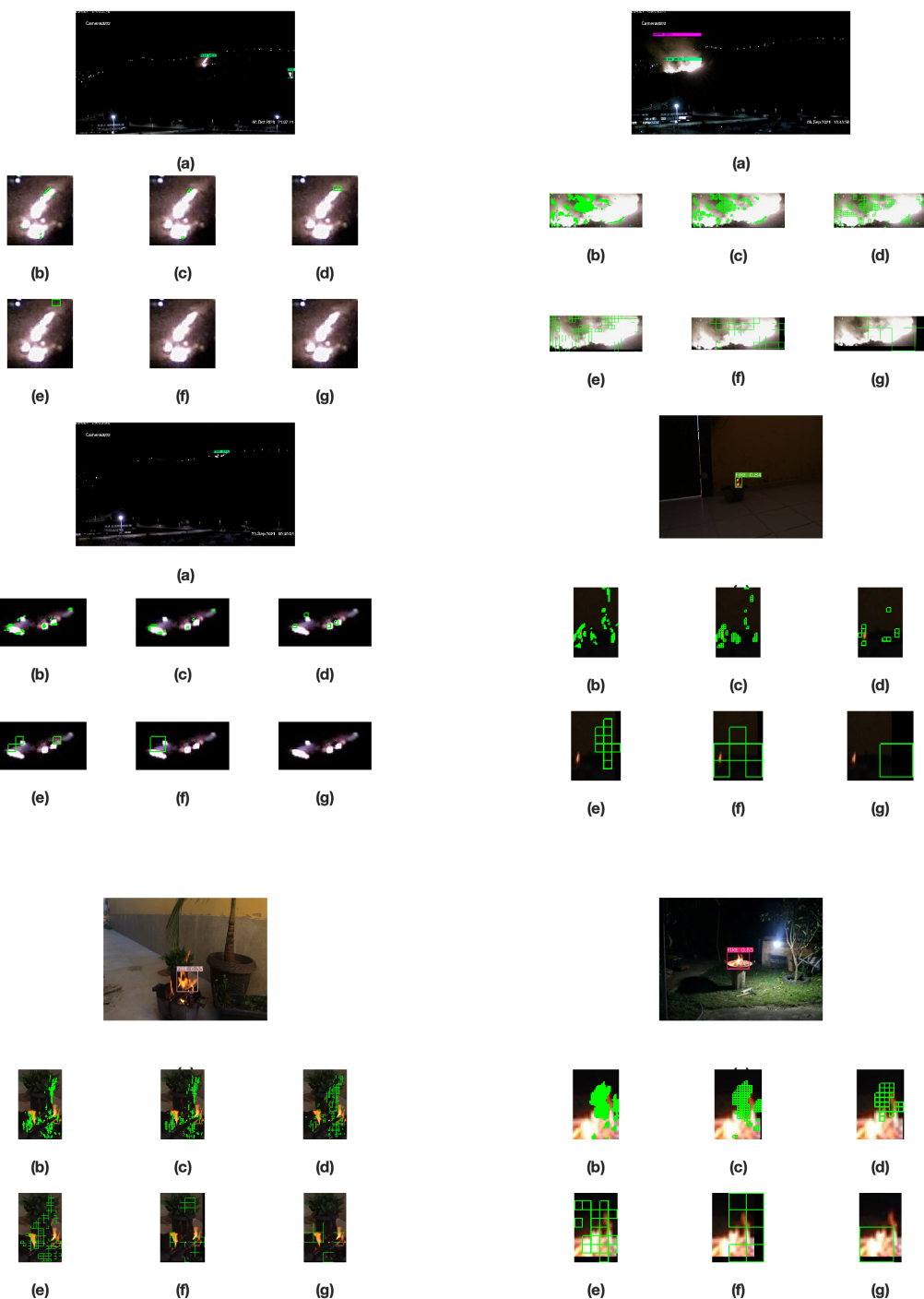


(f)

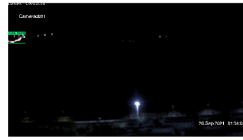


(g)





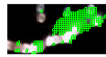




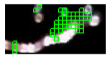
(a)



(b)



(c)



(d)



(e)



(f)



(g)



(b)



(c)



(d)



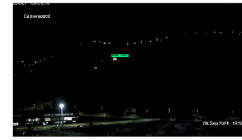
(e)



(f)



(g)



(a)



(b)



(c)



(d)



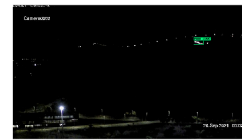
(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



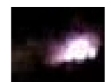
(c)



(d)



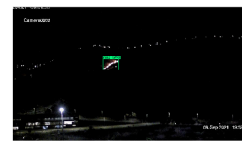
(e)



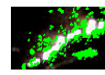
(f)



(g)



(a)



(b)



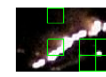
(c)



(d)



(e)

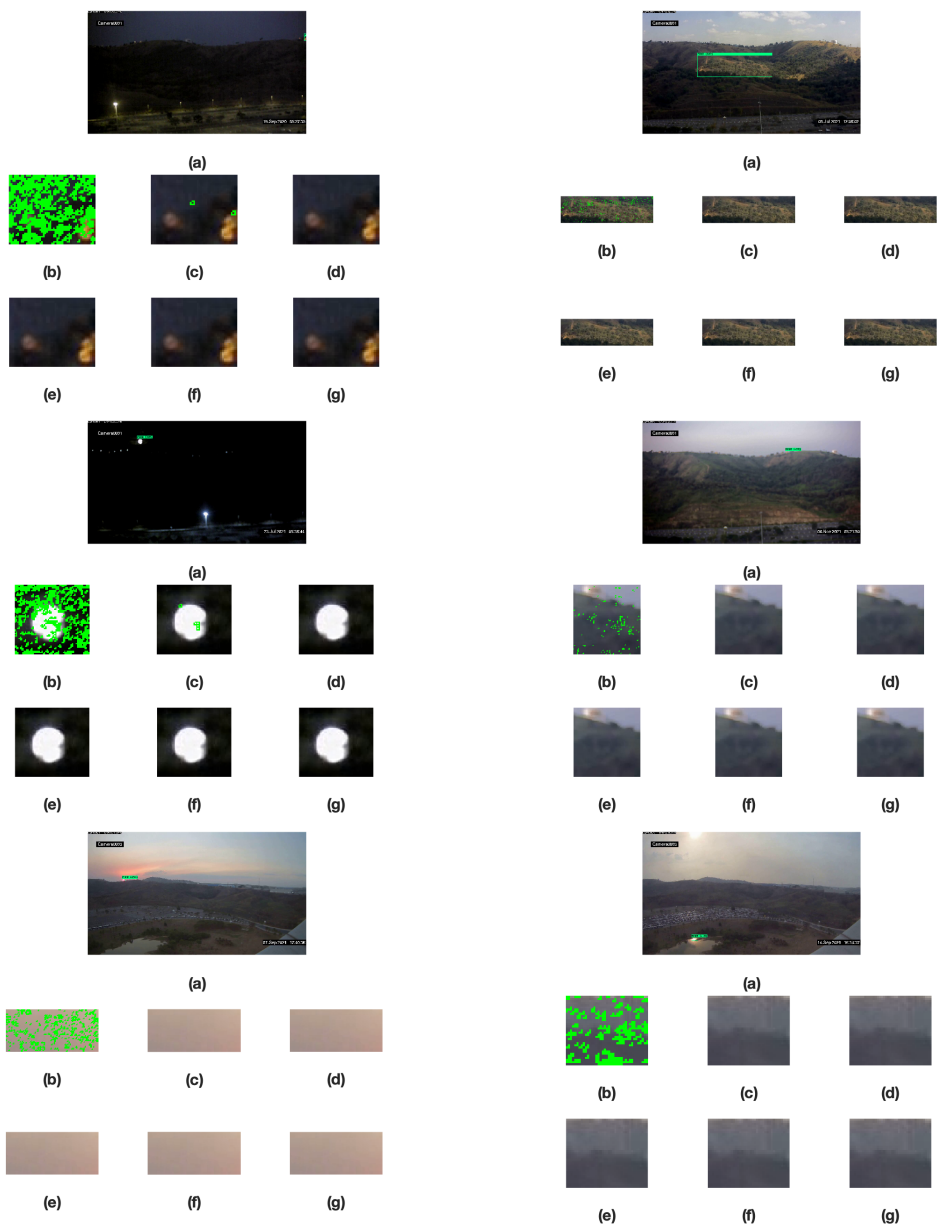


(f)



(g)

## A.5 $K$ -vizinhos mais próximos ( $Knn$ )





(a)



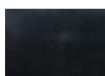
(b)



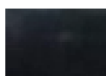
(c)



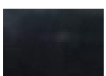
(d)



(e)



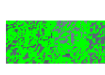
(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



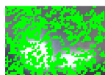
(f)



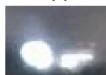
(g)



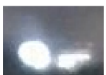
(a)



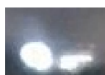
(b)



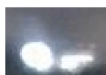
(c)



(d)



(e)



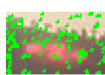
(f)



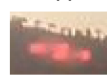
(g)



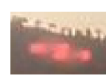
(a)



(b)



(c)



(d)



(e)



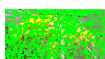
(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



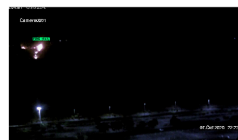
(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(a)



(b)



(c)



(d)



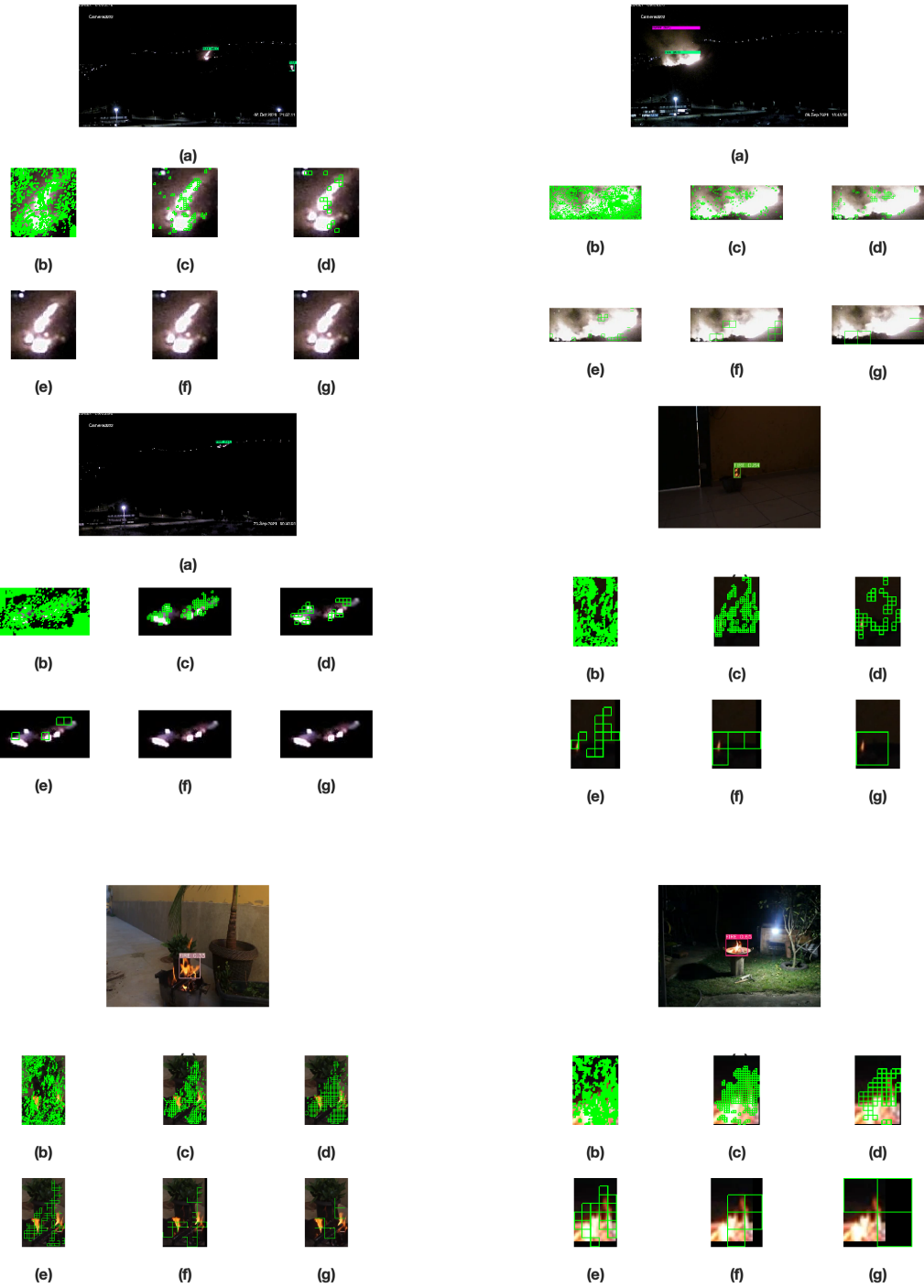
(e)

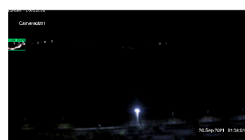


(f)



(g)





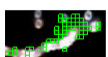
(a)



(b)



(c)



(d)



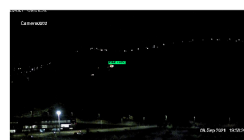
(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(b)



(c)



(d)



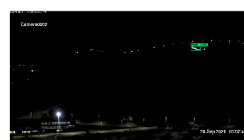
(e)



(f)



(g)



(a)



(b)



(c)



(d)



(e)



(f)



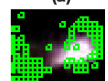
(g)



(a)



(b)



(c)



(d)



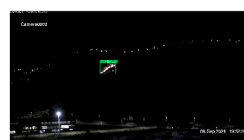
(e)



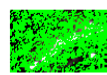
(f)



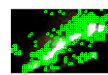
(g)



(a)



(b)



(c)



(d)



(e)

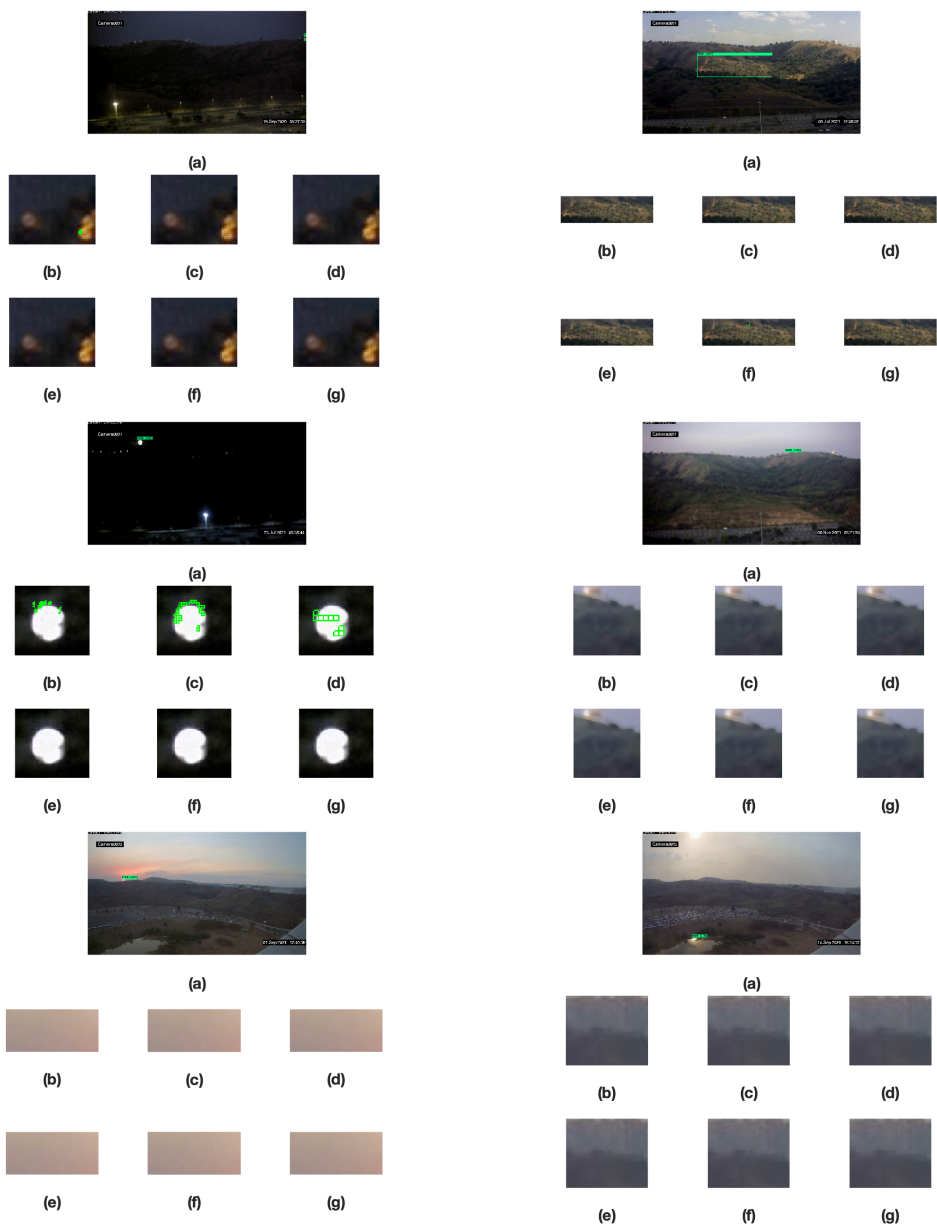


(f)



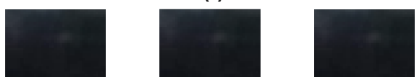
(g)

## A.6 SVM





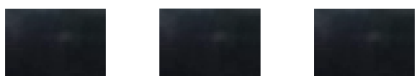
(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



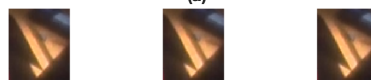
(e)

(f)

(g)



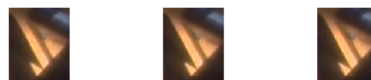
(a)



(b)

(c)

(d)



(e)

(f)

(g)



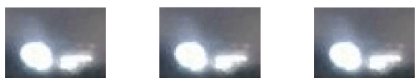
(a)



(b)

(c)

(d)



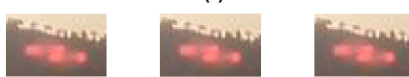
(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)





(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)



(a)



(b)

(c)

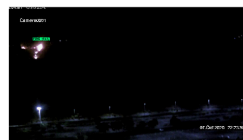
(d)



(e)

(f)

(g)



(a)



(b)

(c)

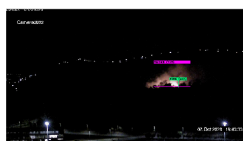
(d)



(e)

(f)

(g)



(a)



(b)

(c)

(d)



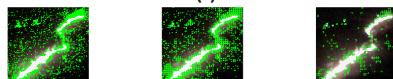
(e)

(f)

(g)



(a)



(b)

(c)

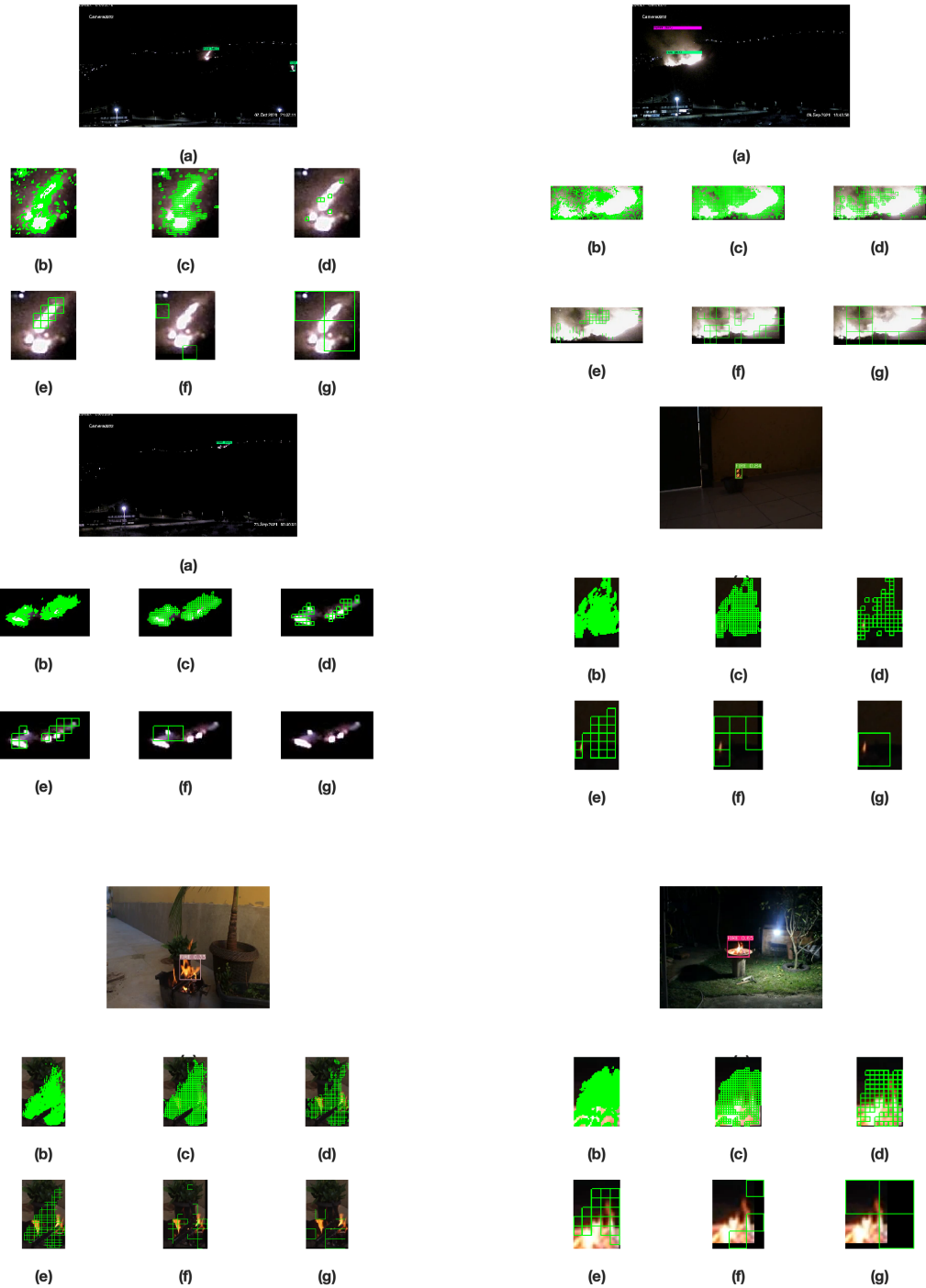
(d)

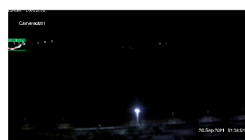


(e)

(f)

(g)





(a)



(b)

(c)

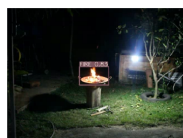
(d)



(e)

(f)

(g)



(b)

(c)

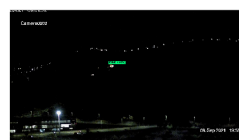
(d)



(e)

(f)

(g)



(a)



(b)

(c)

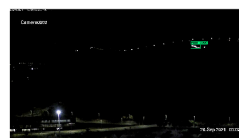
(d)



(e)

(f)

(g)



(a)



(b)

(c)

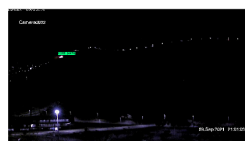
(d)



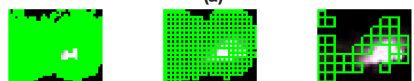
(e)

(f)

(g)



(a)



(b)

(c)

(d)



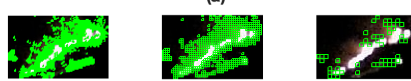
(e)

(f)

(g)



(a)



(b)

(c)

(d)



(e)

(f)

(g)