



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**  
**DOUTORADO EM ENGENHARIA ELÉTRICA**

**JEFFERSON SILVA ALMEIDA**

**EDGEFIRESMOKE: A NOVEL LIGHTWEIGHT CNN MODEL FOR REAL-TIME VIDEO FIRE-SMOKE  
DETECTION**

**FORTALEZA**

**2023**

JEFFERSON SILVA ALMEIDA

EDGEFIRESMOKE: A NOVEL LIGHTWEIGHT CNN MODEL FOR REAL-TIME VIDEO FIRE-SMOKE  
DETECTION

Defesa de tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará.  
Área de Concentração: Engenharia Elétrica

Orientador: Prof. Dr. Victor Hugo Costa de Albuquerque

Coorientador: Prof. Dr. Fabrício Gonzalez Nogueira

FORTALEZA

2023



Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A448e Almeida, Jefferson Silva.

EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire-Smoke  
Detection / Jefferson Silva Almeida. – 2023.

77 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de  
Pós-Graduação em Engenharia Elétrica, Fortaleza, 2023.

Orientação: Prof. Dr. Victor Hugo Costa de Albuquerque.

Coorientação: Prof. Dr. Fabrício Gonzalez Nogueira.

1. Incêndio Florestal. 2. Redes Neurais Convolucionais. 3. Internet das Coisas. I. Título.

CDD 621.3

---

JEFFERSON SILVA ALMEIDA

EDGEFIRESMOKE: A NOVEL LIGHTWEIGHT CNN MODEL FOR REAL-TIME VIDEO FIRE-SMOKE  
DETECTION

Defesa de tese apresentada ao Programa de  
Pós-Graduação em Engenharia Elétrica do  
Centro de Tecnologia da Universidade Federal  
do Ceará. Área de Concentração: Engenharia  
Elétrica

Aprovada em: 15/09/2023

BANCA EXAMINADORA

---

Prof. Dr. Victor Hugo Costa de  
Albuquerque (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Fabrício Gonzalez  
Nogueira (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Bismark Claire Torrico  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Carlos Mauricio Jaborandy de Mattos  
Dourado Junior  
Instituto Federal do Ceará (IFCE)

---

Prof. Dr. Renato William Rodrigues de Souza  
Instituto Federal do Ceará (IFCE)

À Deus, pela vida, pelo alimento, pela família, pela moradia, e por sua eterna misericórdia. À meus familiares: meus avós, minha mãe, meu irmão. À minha família, esposa e filha que tanto amo.

## AGRADECIMENTOS

À Deus por me permitir ter êxito na vida acadêmica e trilhar um caminho exemplar.

À Universidade Federal do Ceará (UFC) e ao PPGEE pela oportunidade de estudar nesta renomada instituição de ensino superior.

Ao Prof. Dr. Victor Hugo Costa de Albuquerque, meu orientador, e coordenador do Biomedical Data Analytics Research Group (BioDatA), por me orientar em minha tese de doutorado, pelas sugestões valiosas e por acreditar em meu potencial como jovem estudante pesquisador.

Ao Prof. Dr. Fabrício Gonzalez Nogueira, meu co-orientador, e coordenador do Grupo de Pesquisa em Automação e Robótica (GPAR), pelo suporte material e compreensão nos momentos difíceis enfrentados durante a pandemia de COVID-19.

Aos membros da banca, pela disponibilidade em participar da defesa de qualificação, pelas sugestões de melhoria e pelo compromisso e disponibilidade em participar da defesa de tese.

Aos colegas de laboratório na UFC e IFCE, pela troca de ideias ao longo da vida acadêmica.

Ao laboratório no IFCE, a qual fiz parte por muitos anos, pela oportunidade de ingresso no mestrado acadêmico, na qual fui bolsista FUNCAP; pela oportunidade de ensinar e ao mesmo tempo desenvolver habilidades profissionais; e pela oportunidade de contribuição nos projetos de pesquisa científica e desenvolvimento tecnológico que participei.

A minha família e meus familiares, que nos momentos de minha ausência dedicados aos estudos, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente.

Agradeço a todos os professores que me orientaram em todos os níveis de ensino e pelo apoio ao ingressar na pós-graduação.

Aos colegas Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, egressos do curso de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará.

E à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo financiamento da pesquisa de doutorado via bolsa de estudos demanda social UFC. Além de todo o suporte financeiro para que eu pudesse me dedicar aos estudos e garantir o básico necessário de conforto a minha família.

“Quando passares pelas águas estarei contigo, e quando pelos rios, eles não te submergirão; quando passares pelo fogo, não te queimarás, nem a chama arderá em ti.”

(Isaías, capítulo 43, versículo 2)

## RESUMO

Os incêndios florestais podem ter impactos graves tanto no ambiente como nas comunidades humanas. Podem causar erosão do solo, perda de habitat e biodiversidade, bem como a liberação de dióxido de carbono e outros poluentes na atmosfera. Além disso, podem causar danos a propriedades, deslocamento de moradores e colocar em risco os bombeiros e outros socorristas. Os incêndios florestais também podem contribuir para as alterações climáticas, libertando carbono armazenado na atmosfera e alterando os ecossistemas. Neste trabalho, foi proposto um novo algoritmo capaz de monitorar pequenas áreas de reserva florestal através de *streaming* de vídeo em tempo real. O algoritmo proposto poderá complementar os meios existentes de monitorização e vigilância florestal e fornecerá soluções eficazes enfrentadas na monitorização por satélite. Este trabalho apresenta uma melhoria do método EdgeFireSmoke, de autoria própria, e utiliza uma rede neural artificial juntamente com o método de aprendizagem profunda. O algoritmo EdgeFireSmoke++, proposto nesta tese, foi capaz de detectar incêndios florestais com 95,41% de acurácia, 95,49% de precisão, 95,38% de Recall e 95,41% de F1-score. O EdgeFireSmoke++ registrou os melhores resultados nos experimentos computacionais, no qual obteve excelentes taxas de FPS em ambas as câmeras avaliadas, registrando 33 FPS na câmera 1 e 40 FPS na câmera 2. Para seu funcionamento, o algoritmo proposto mostrou-se bastante leve, em termos computacionais, podendo trabalhar em uma CPU de 4 núcleos, 2.1GHz, com consumo médio de 540MB de memória RAM. Nos experimentos computacionais o algoritmo proposto foi superior em relação aos demais métodos da literatura avaliados.

**Palavras-chave:** Incêndio Florestal, Redes Neurais Convolucionais, Internet das Coisas.

## ABSTRACT

Forest fires can have severe impacts on both the environment and human communities. They can cause soil erosion, loss of habitat and biodiversity, as well as the release of carbon dioxide and other pollutants into the atmosphere. In addition, they can cause damage to properties, displacement of residents, and put firefighters and other responders at risk. Forest fires can also contribute to climate change by releasing stored carbon into the atmosphere and altering ecosystems. In this work, we propose a novel algorithm capable of monitoring small areas of forest reserve environment through video streaming in real-time. It will complement the existing means of forest monitoring and surveillance and provide effective solutions faced in satellite-based monitoring. The proposed algorithm is an improvement of the EdgeFireSmoke method and uses an artificial neural network together with a deep learning method. The proposed EdgeFireSmoke++ algorithm was able to detect forest fires with 95.41% accuracy, 95.49% precision, 95.38% Recall and 95.41% F1-score. The proposed algorithm recorded the best FPS rates of the HD IP camera at 33 FPS and with the USB VGA camera at 40 FPS. For its operation, the proposed algorithm proved to be quite light, being able to work on a CPU with 4 cores, 2.1GHz, with an average consumption of 540MB of RAM memory. This test was superior to the methods evaluated in the literature.

**Keywords:** Forest Fire. Convolutional Neural Networks. Internet of Things.

## SUMÁRIO

<b>1</b>	<b>CAPÍTULO I: INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>A problemática</b>	<b>11</b>
<b>1.1.1</b>	<b><i>Motivação</i></b>	<b>15</b>
<b>1.1.2</b>	<b><i>Objetivos</i></b>	<b>18</b>
<b>1.2</b>	<b>Trabalhos publicados</b>	<b>18</b>
<b>1.2.1</b>	<b><i>Organização da Tese</i></b>	<b>18</b>
<b>2</b>	<b>CAPÍTULO II: FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
<b>2.1</b>	<b>A origem das CNNs</b>	<b>20</b>
<b>2.2</b>	<b>Camada Convolutiva e os Campos Receptivos</b>	<b>21</b>
<b>2.3</b>	<b>Filtros e Pesos da Camada Convolutiva</b>	<b>22</b>
<b>2.4</b>	<b>Representação dos Mapas de Características</b>	<b>23</b>
<b>2.5</b>	<b>Camada <i>Pooling</i></b>	<b>24</b>
<b>2.6</b>	<b>Arquiteturas CNN</b>	<b>24</b>
<b>2.7</b>	<b>Conclusão deste Capítulo</b>	<b>25</b>
<b>3</b>	<b>CAPÍTULO III: METODOLOGIA</b>	<b>26</b>
<b>3.1</b>	<b>O algoritmo <i>EdgeFireSmoke</i></b>	<b>26</b>
<b>3.1.1</b>	<b><i>Aquisição das imagens</i></b>	<b>27</b>
<b>3.1.2</b>	<b><i>Fatores Climatológicos</i></b>	<b>28</b>
<b>3.1.3</b>	<b><i>Arquitetura da RNA para reconhecimento de imagem florestal</i></b>	<b>29</b>
<b>3.1.3.1</b>	<b><i>Transformação de cores</i></b>	<b>29</b>
<b>3.1.3.2</b>	<b><i>Histograma e sua Normalização</i></b>	<b>30</b>
<b>3.1.3.3</b>	<b><i>Comparação de Histogramas</i></b>	<b>31</b>
<b>3.1.3.4</b>	<b><i>Rede Neural Artificial</i></b>	<b>31</b>
<b>3.1.4</b>	<b><i>Arquitetura CNN para detecção de incêndios florestais</i></b>	<b>33</b>
<b>3.1.5</b>	<b><i>Índice do Risco de Incêndio Florestal (IRIF)</i></b>	<b>37</b>
<b>3.1.6</b>	<b><i>Visualização de Dados em tempo real (IoT-IHM)</i></b>	<b>40</b>
<b>3.1.7</b>	<b><i>Datasets utilizados para treinar o algoritmo</i></b>	<b>40</b>
<b>3.1.7.1</b>	<b><i>Dataset 1: Imagens florestais</i></b>	<b>40</b>
<b>3.1.7.2</b>	<b><i>Dataset 2: Imagens aéreas de incêndio florestal</i></b>	<b>41</b>
<b>3.1.8</b>	<b><i>Treinamento do Algoritmo</i></b>	<b>41</b>
<b>3.1.9</b>	<b><i>Pseudocódigo</i></b>	<b>42</b>
<b>3.1.10</b>	<b><i>Métricas de avaliação</i></b>	<b>43</b>
<b>3.1.10.1</b>	<b><i>Acurácia</i></b>	<b>43</b>
<b>3.1.10.2</b>	<b><i>Sensibilidade</i></b>	<b>44</b>
<b>3.1.10.3</b>	<b><i>Precisão</i></b>	<b>45</b>
<b>3.1.10.4</b>	<b><i>F1-score</i></b>	<b>45</b>
<b>3.1.11</b>	<b><i>Raiz Quadrada do Erro Médio (RMSE)</i></b>	<b>46</b>



3.1.11.1	<b>Novas métricas: CCGM e CPTR</b> . . . . .	46
3.2	<b>Sugestão de aplicação real</b> . . . . .	47
3.3	<b>Conclusão deste Capítulo</b> . . . . .	48
4	<b>CAPÍTULO IV: RESULTADOS E DISCUSSÃO</b> . . . . .	49
4.1	<b>Análise Qualitativa</b> . . . . .	49
4.2	<b>Análise Quantitativa</b> . . . . .	59
4.3	<b>Análise do Índice de Risco de Incêndio Florestal</b> . . . . .	65
4.4	<b>Interface de Visualização de Dados (IoT-IHM)</b> . . . . .	68
4.5	<b>Conclusão deste Capítulo</b> . . . . .	70
5	<b>CAPÍTULO V: CONCLUSÃO</b> . . . . .	71
	<b>REFERÊNCIAS</b> . . . . .	73

## 1 CAPÍTULO I: INTRODUÇÃO

Neste capítulo inicial, serão apresentadas as informações relevantes que permitirá compreender o contexto e a abordagem proposta nesta tese. Abordaremos uma visão geral do tema, a sua importância e relevância para o meio ambiente e a sociedade, além de listar os objetivos e contribuições a ser alcançados, explicando de forma clara o escopo do estudo.

### 1.1 A problemática

Um estudo realizado pela Organização Meteorológica Mundial (OMM) revelou que o planeta Terra teve em 2023 o trimestre mais quente já registrado. O mês de Agosto de 2023 foi cerca de 1,5°C mais quente do que a média pré-industrial entre 1850 e 1900 (OMM, 2023).

Com base neste cenário global, segundo a Organização das Nações Unidas (ONU), as ondas de calor extremo, agravadas por incêndios florestais e pela poeira do deserto, têm impacto significativo na qualidade do ar, na saúde humana e no ambiente (ONU, 2023).

Há três anos a BBC Brasil publicou uma matéria alertando que a ocorrência de incêndios florestais registrados em nosso planeta, em 2020, foi a maior em escala e em emissões de  $CO_2$  das últimas duas décadas (BBC, 2020). Uma estimativa feita pela Organização Mundial da Saúde (OMS) mostrou que os incêndios florestais e atividades vulcânicas afetaram cerca de 6,2 milhões de pessoas entre 1998-2017. Para o *United Nations Environment Programme* (UNEP), o clima favorável a incêndios florestais perigosos está previsto para piorar nas próximas décadas (UNEP, 2022a).

Os incêndios florestais contribuem para a poluição do ar devido à emissão de fumaça, gases e podem causar sérios danos à saúde, incluindo problemas respiratórios, afetando a saúde vascular e até causando a morte. Incêndios florestais podem interromper o transporte, as comunicações, os serviços de energia e gás e o abastecimento de água. Eles também levam à deterioração da qualidade do ar e à perda de propriedades, safras, recursos, animais e vidas.

Nos Estados Unidos, o Centro Nacional de Informação Ambiental (NOAA) registrou um total de 58 mil focos de incêndios em 2020. Na Europa, de acordo com os Relatórios Anuais de Incêndios feitos pelo Sistema Europeu de Informação sobre Incêndios Florestais (EFFIS), ocorreram aproximadamente 65 mil incêndios em 2019. O monitoramento por satélite realizado pelo Instituto Nacional de Pesquisas Espaciais (INPE) a partir de dados coletados com o satélite americano AQUA, desenvolvido em parceria entre Japão e Brasil, registrou um total de 223 mil focos de incêndio no Brasil em 2020 (INPE, 2021). Este mesmo satélite registrou no ano de 2021 a ocorrência de 184 mil focos de incêndio, sendo esta marca ultrapassada em 2022, no qual registrou um total de 201 mil focos de incêndio florestal (INPE, 2023).

Se considerarmos as estatísticas de incêndio em uma escala global, esses números são significativamente elevados. Hoje, embora o monitoramento de incêndios florestais em todo o mundo, por satélites, seja útil para grandes áreas e regiões geográficas específicas, não existem métodos de videomonitoramento em tempo real acessíveis para pequenas áreas. Para o *United Nations Environment Programme* (UNEP), o número de incêndios florestais poderá aumentar em 50% até 2100 e os governos

não estão preparados para a piora deste cenário (UNEP, 2022b). Na Figura 1 é mostrado o registro da brigada combatendo um incêndio florestal.

Figura 1 – Incêndio no Parque Nacional da Canastra ocorrido em 2018.



Fonte: Instituto Chico Mendes de Conservação da Biodiversidade (ICMBio)

No Brasil, sabe-se que o monitoramento de áreas verdes feito a partir do espaço apresenta algumas limitações, pois este tipo de monitoramento foi projetado para grandes áreas e regiões remotas (IBAMA, 2022). Um fator limitante é que os satélites em órbitas polares podem produzir apenas um conjunto de imagens diariamente, enquanto os satélites geoestacionários disponíveis produzem imagens de hora em hora, o que dificulta o monitoramento em tempo real. Como consequência, o monitoramento de algumas áreas do país nem sempre acontece em tempo real. Atualmente, alguns incêndios florestais não conseguem ser detectados por esses meios devido a diversos fatores naturais do meio ambiente e limitações das tecnologias empregadas no monitoramento, como os identificados pelo INPE nas situações a seguir:

- incêndio em áreas com menos de 30 metros quadrados;
- fogo no solo denso da floresta, sem afetar as copas das árvores;
- nuvens cobrindo uma região;
- incêndios florestais rápidos ocorrendo entre as imagens de satélite capturadas;
- fogo atrás de uma montanha ou fora do campo de visão do satélite;
- falta de precisão na localização do fogo.

Este trabalho propõe uma ferramenta alternativa, que utiliza inteligência artificial para auxiliar no monitoramento inteligente por vídeo de pequenas áreas de reserva florestal. Desta forma, espera-se que o algoritmo proposto seja capaz de superar pelo menos três dos problemas listados acima: 1) Detecção de incêndio em pequenas áreas até 30 metros quadrados; 2) Detecção de incêndios florestais rápidos que formam pontos cegos para satélites; 3) Detecção de incêndio fora do campo de

visão dos satélites devido a nuvens cobrindo a região.

Na literatura, observamos um número significativo de trabalhos que envolvem Aprendizagem de Máquina (*Machine Learning*) e Aprendizagem Profunda (*Deep Learning*) para detecção de incêndios florestais. Os autores (LI *et al.*, 2022a; TAHIR *et al.*, 2022) defendem o uso de estratégias de detecção de objetos baseada em *Deep Learning*, e consideram que esses algoritmos são eficientes, pois identificam o objeto de interesse por meio de caixas delimitadoras (*bounding boxes*). Entretanto, esses algoritmos apresentam elevado custo computacional, com grande dependência de processamento em placas gráficas dedicadas (GPU). Em outras abordagens, as Redes Neurais Convolucionais têm sido utilizadas, por apresentarem ótimo desempenho com imagens digitais. Especialmente as CNNs leves (*lightweight CNNs*), apresentam uma menor dependência de GPU em relação aos métodos citados anteriormente (SILVA *et al.*, 2022; RACHANA *et al.*, 2022; CHATRAGADDA *et al.*, 2022; OLIVER *et al.*, 2020).

O uso de câmeras RGB especiais ou convencionais, como câmeras térmicas e infravermelhas, também tem sido investigado, pois aumentam a possibilidade de detecção de incêndios em condições adversas de visibilidade (TLIG *et al.*, 2022; SADI *et al.*, 2021; LI *et al.*, 2021). As limitações mais proeminentes das câmeras térmicas e infravermelhas incluem distância de detecção limitada a algumas dezenas de metros e alto custo de aquisição.

Uma técnica que tem sido extensivamente investigada usando CNNs é a Transferência de Aprendizagem (*Transfer Learning*). Esta técnica consiste em avaliar o uso de CNNs previamente treinadas para detectar incêndios florestais com imagens totalmente diferentes dos padrões originais de aprendizagem (AGARWAL; JHA, 2021). Portanto, neste trabalho, avaliaremos esta técnica com os principais métodos de CNNs modernos mais leves encontrados na literatura, como VGG16 e VGG19 (SIMONYAN; ZISSERMAN, 2014), MobileNetV2 (SANDLER *et al.*, 2018), InceptionV3 (SZEGEDY *et al.*, 2016), Xception (CHOLLET, 2017) e DenseNet121 (HUANG *et al.*, 2017). É apresentado na Tabela 1 um resumo de artigos populares selecionados da literatura que usam CNNs para detecção de incêndio. A seleção desses métodos foi considerada incluindo os métodos apresentados na Tabela 1. A partir dos trabalhos enumerados, na Tabela 1, foi possível identificar as lacunas de cada trabalho e propor uma nova solução ao problema em questão com a adição de uma etapa de detecção de ambiente florestal e de informações meteorológicas.

Com base na experiência dos autores no domínio prescrito, houve trabalhos sobre detecção de fumaça em vários ambientes que foram desenvolvidos usando modelos inteligentes de computação na borda (MUHAMMAD *et al.*, 2019c) e modelos de aprendizado profundo (KHAN *et al.*, 2019; KHAN *et al.*, 2021). Para que o leitor compreenda, os modelos de computação na borda (*edge computing*) podem ser entendidos como algoritmos que são leves, em termos computacionais, o suficiente para serem executados localmente em dispositivos embarcados, sem a dependência de enviar dados brutos a servidores mais robustos. Além disso, soluções de monitoramento de incêndio energeticamente eficientes também foram apresentadas pelos autores com redes inteligentes (MUHAMMAD *et al.*, 2020b) e detecção de incêndio em ambientes IoT (CHAUHAN *et al.*, 2023). Além destes, outros trabalhos notáveis sobre sumarização de vídeo, um processo de encurtamento de um vídeo selecionando quadros-chave ou

Tabela 1 – Principais observações da literatura e suas características. Siglas: IF - Incêndio Florestal; AF - Ambiente Florestal; CD - Caixas Delimitadoras; IHM - Interface Homem Máquina

Algoritmos	Deteção de IF	Deteção de AF	Desenha CD	Comunicação via Internet	Variáveis Meteorológicas	Visualização com IHM
<b>EdgeFireSmoke++</b> (Algoritmo Proposto)	Sim	Sim	Não	Sim	Sim	Sim
<b>SRN-Yolo</b> (LI <i>et al.</i> , 2022a)	Não	Sim	Sim	Não	Não	Sim
<b>YoloV5</b> (TAHIR <i>et al.</i> , 2022)	Não	Sim	Sim	Não	Não	Sim
<b>EdgeFireSmoke</b> (SILVA <i>et al.</i> , 2022)	Sim	Sim	Não	Sim	Não	Não
<b>VGG16</b> (SIMONYAN; ZISSERMAN, 2014) (KHAN <i>et al.</i> , 2019)	Não	Sim	Não	Não	Não	Não
<b>VGG19</b> (CHATRAGADDA <i>et al.</i> , 2022) (SIMONYAN; ZISSERMAN, 2014)	Não	Sim	Não	Não	Não	Não
<b>MobileNetV2</b> (SANDLER <i>et al.</i> , 2018) (MUHAMMAD <i>et al.</i> , 2019c)	Não	Sim	Não	Não	Não	Não
<b>InceptionV3</b> (AGARWAL; JHA, 2021) (SZEGEDY <i>et al.</i> , 2016)	Não	Sim	Não	Não	Não	Não
<b>Xception</b> (CHOLLET, 2017)	Não	Não	Não	Não	Não	Não
<b>DenseNet121</b> (CHATRAGADDA <i>et al.</i> , 2022) (HUANG <i>et al.</i> , 2017)	Não	Sim	Não	Não	Não	Não

Fonte: Próprio Autor.

partes de vídeos que capturam os pontos principais do vídeo, foram desenvolvidos e predominantemente destinados a sumarização de recursos eficientes (MUHAMMAD *et al.*, 2019a), sumarização de baixo custo (MUHAMMAD *et al.*, 2019b) e aplicativos IloT *multiview* (HUSSAIN *et al.*, 2019) voltados à área de Internet Industrial das Coisas. Muito recentemente, um sistema automático de extinção de incêndios foi desenvolvido pelos autores (JAGATHEESAPERUMAL *et al.*, 2023). Esses trabalhos servem de inspiração para o desenvolvimento da nova abordagem *EdgeFireSmoke++* proposta neste trabalho.

Atualmente, tem-se dado ênfase aos algoritmos leves, pois requerem menor consumo de energia, o que é essencial para sistemas embarcados integrados com veículos aéreos não-tripulados (VANTs), e para processamento na CPU (YUAN *et al.*, 2023; LI *et al.*, 2022b; WANG *et al.*, 2021; FAN; PEI, 2021). Além do mais, o uso de informações meteorológicas para prever o risco de incêndio florestal também tem sido investigado (PARK *et al.*, 2022; REDDY; KALYANASUNDARAM, 2022; OMAR *et al.*, 2021). Nesta pesquisa será avaliado a possibilidade de utilização de dados climáticos em conjunto com as CNNs modernas mais leves.

O algoritmo proposto nesta tese é o aperfeiçoamento de um primeiro artigo, de autoria própria, publicado no *IEEE Transactions on Industrial Informatics* (SILVA *et al.*, 2022). Ao contrário do trabalho anterior que propôs um modelo para embarcar em torres vigilância e veículos aéreos não-tripulados, o algoritmo atual já publicado no *Expert Systems with Applications* (ALMEIDA *et al.*, 2023) é destinado ao uso por câmeras de videomonitoramento com conexão à internet, que podem ser instaladas em torres de vigilância.

A principal diferença aqui apresentada foi a adição de uma Rede Neural Artificial (RNA), que antecede a CNN proposta por (SILVA *et al.*, 2022). Além disso, com o aprimoramento na visualização dos dados, apresentamos uma Interface Homem Máquina (IHM) que coleta os dados meteorológicos a partir da internet para calcular e exibir o risco de incêndio florestal em tempo real em determinadas regiões monitoradas.

A RNA realizará o pré-processamento, fazendo a triagem das imagens de entrada e detecção da ausência ou presença de um ambiente florestal. Enquanto a CNN classificará as imagens da floresta em quatro grupos: 1) Área verde, 2) Área queimada, 3) Área de incêndio florestal e 4) Área de neblina. Assim, tornará o algoritmo proposto mais confiável e robusto em suas previsões. O novo método

que chamamos de *EdgeFireSmoke++* (ALMEIDA *et al.*, 2023) tem recursos aprimorados e melhorias em relação à sua versão anterior (SILVA *et al.*, 2022).

As principais contribuições deste trabalho foram assim resumidas:

1. Proposição de uma nova base de dados com imagens reais de incêndio florestal;
2. Inclusão da etapa de pré-processamento com base na estratégia proposta originalmente por (SILVA *et al.*, 2022) para detecção de incêndio em ambiente florestal.
3. Utilização de variáveis meteorológicas para o cálculo do risco de incêndios florestais com base na inferência (LOURENÇO *et al.*, 1997).
4. Implementação de Interface Humano-Máquina (IHM) para visualização das detecções e escala de cores dos riscos de incêndio florestal.
5. Integração com IoT usando a API OpenWeather para coleta de dados meteorológicos de um servidor web.
6. Melhorias no processamento de vídeo observadas em tempo real com qualidade HD e taxa de quadros compatível com câmeras IP de 30 FPS.
7. Proposição de novas métricas de avaliação de desempenho: CCGM e CPTR.
8. Solução para a dependência de placa gráfica dedicada (GPU) presente no modelo *EdgeFireSmoke* inicial, proposto em um primeiro estágio desta pesquisa (SILVA *et al.*, 2022).
9. Melhorias no algoritmo *EdgeFireSmoke*, em um segundo estágio da pesquisa (ALMEIDA *et al.*, 2023), no qual alcançou-se compatibilidade para processamento de tempo real em CPU.

### **1.1.1 Motivação**

As ocorrências de incêndios florestais e queimadas têm se mostrado de extrema relevância desde o seu monitoramento por satélite iniciado no Brasil em 1999. Estes eventos tem acontecido em todos os biomas no território nacional, causando impactos negativos na fauna, na flora, na saúde das pessoas e no clima, afetando assim o equilíbrio do meio ambiente.

Atualmente, sabemos que esta problemática tem abrangência global. Além do Brasil e vizinhos da América do Sul, outros países como Austrália e Estados Unidos tem sofrido com os impactos causados por esses eventos.

De acordo com dados estatísticos do INPE, as ocorrências de incêndios florestais tem sido frequentes ao longo dessas duas décadas monitoradas no Brasil. Entre 2002 e 2010 foram registradas as maiores ocorrências anuais de focos ativos de incêndio, conforme pode se visto na série histórica mostrada na Figura 2. Em 2021 foram registrados um total de 184 mil focos ativos de incêndio. Em 2022 foi registrado um número maior, totalizando 200 mil focos ativos de incêndio, valor próximo da média histórica anual brasileira já registrada de aproximadamente 219 mil focos ativos.

O gráfico da Figura 3 nos mostra que os incêndios no Brasil se intensificam no segundo semestre entre os meses de Julho à Dezembro, e atinge o maior número de focos no mês de Setembro. Este período é o ideal para ações de prevenção e combate aos incêndios florestais por meio de brigadas de incêndio e ações preditivas como o monitoramento por satélite, videomonиторamento terrestre, e a mais recente estratégia de monitoramento em tempo real utilizando VANTs.

Este aumento significativo na ocorrência de incêndios acontece no período de estiagem de chuvas. Portanto, o clima favorece a propagação de incêndios florestais pelos seguintes motivos (MAUGER *et al.*, 2009):

- Ar seco;
- Céu com poucas nuvens;
- Vegetação ressecada.

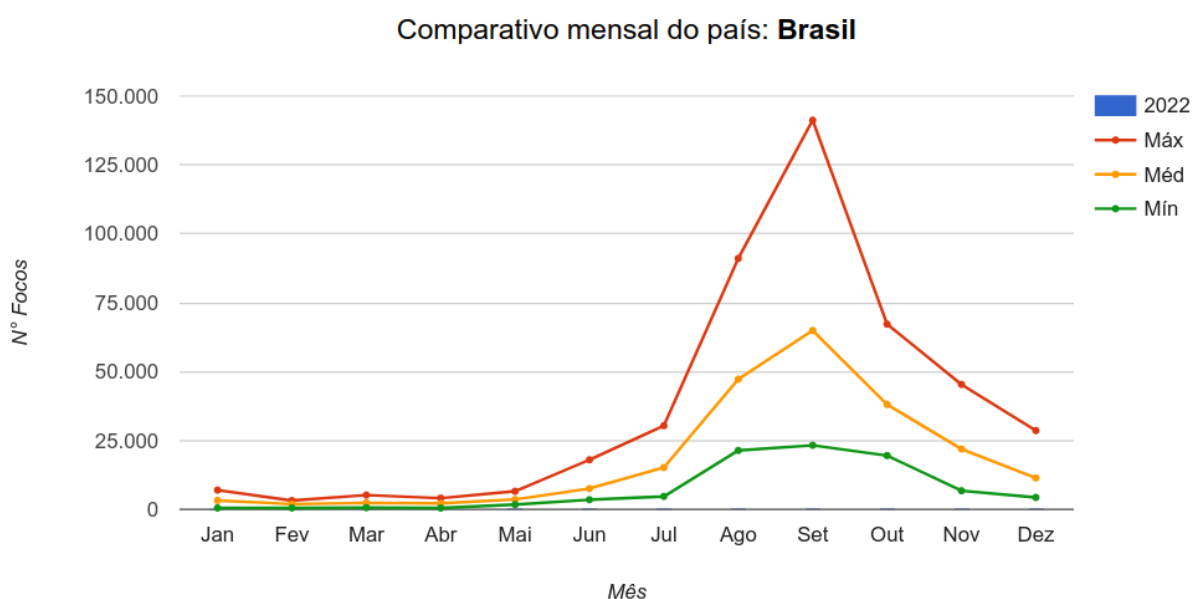
Por isso, nesse período de seca as ações de monitoramento, prevenção e combate a incêndios devem ser intensificadas. Com base na Figura 4, série histórica por tipo de bioma registrada

Figura 2 – Série histórica do total de focos ativos detectados pelo satélite de referência, no período de 1998 até 08/09/2023.



Fonte: Portal Queimadas (INPE), Acesso: <https://queimadas.dgi.inpe.br/queimadas/porta1>

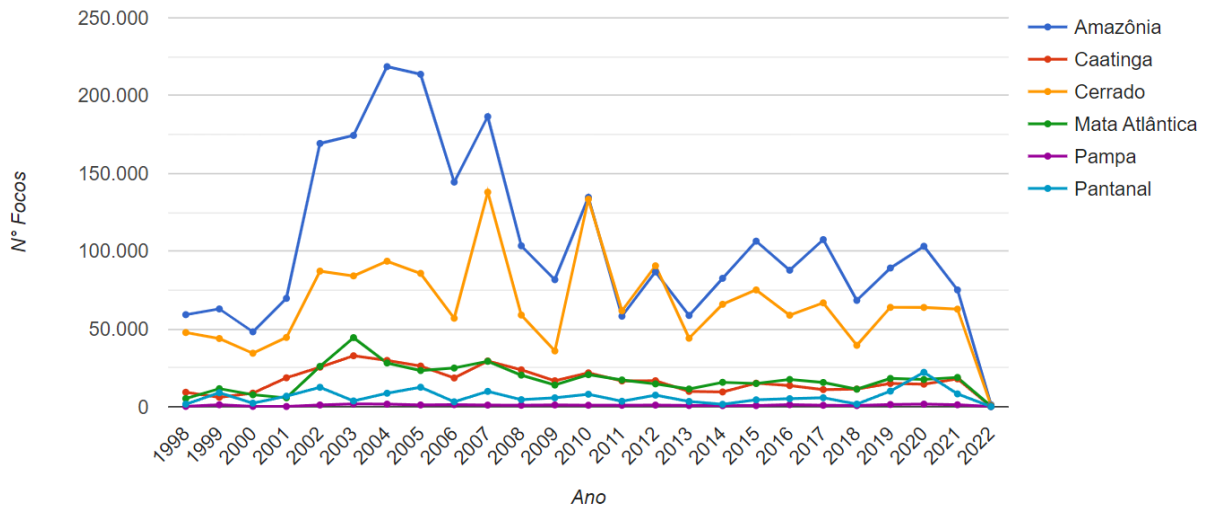
Figura 3 – Comparativo dos dados do ano corrente com os valores máximos, médios e mínimos, no período de 1998 até 04/01/2022.



Fonte: Portal Queimadas (INPE), Acesso: <https://queimadas.dgi.inpe.br/queimadas/porta1>

pelo INPE, percebemos que nas últimas duas décadas os biomas mais afetados por incêndios são a Amazônia e o Cerrado, seguido dos outros biomas com menor ocorrência.

Figura 4 – Focos de incêndio por bioma e ano, no período de 1998 até 01/02/2022.  
Série histórica de focos por bioma e ano



Fonte: Portal Queimadas (INPE), Acesso: <https://queimadas.dgi.inpe.br/queimadas/portal>

Os canais de comunicação vêm noticiando os impactos causados por estes incêndios, no qual sabemos que atingem não somente o meio ambiente, mas diretamente ao próprio ser humano. Na Tabela 2, temos um resumo das principais notícias registradas no período de 2015 à 2023 por importantes agências de notícias brasileiras e internacionais, com destaque nos anos de 2020 e 2022.

Tabela 2 – Eventos marcantes registrados por importantes agências de notícias no Brasil.

Ano	Notícia na Mídia	Fonte
2023	Queimadas no Brasil: Amazônia e Cerrado concentram 95% da área destruída em 2022	Exame
2023	Mais de 16 milhões de hectares foram queimados no Brasil em 2022	AgênciaBrasil, EBC
2022	Brasil queimou área maior que Pernambuco nos nove primeiros meses de 2022	MapBiomas
2021	Brasil já registra mais de 100 mil focos de incêndio em 2021	ClimaTempo
2021	5 biomas brasileiros registram piores queimadas de 2021 em agosto	CNN Brasil
2021	Queimadas e desmatamento estão transformando Amazônia em fonte de carbono	National Geographic Brasil
2020	Inpe confirma aumento de quase 200% em queimadas no Pantanal entre 2019 e 2020	Agência Câmara de Notícias
2020	Queimadas na Amazônia em 2020 passam número registrado em todo o ano de 2019	G1 Notícias
2020	Quase 3 bilhões de animais foram afetados por incêndios na Austrália	CNN Internacional
2019	Incêndios na Amazônia afetam crianças e custam R\$ 1,5 milhão ao SUS	AgênciaBrasil, EBC
2018	Incêndio mais letal da história da Califórnia deixa mais de 40 mortos	G1 Notícias
2017	Brasil termina 2017 com um número recorde de queimadas desde 1999	Imazon
2016	Brasil já teve 53 mil queimadas em 2016; alta de 65% é motivo de alarme	UOL Notícias
2015	Queimadas crescem 27,5% no país em 2015, mostra Inpe	AgênciaBrasil, EBC

Fonte: Próprio Autor.

Diante desta problemática relevante e frequente ao longo de várias décadas no Brasil, decidimos abordar esta causa nesta tese, pesquisar soluções, alternativas ao uso de satélites espaciais, visando suprir as dificuldades existentes, e elaborando ferramentas úteis que possam vir a ser utilizadas no monitoramento e prevenção dos incêndios florestais.



### 1.1.2 Objetivos

O principal objetivo desta tese é criar e avaliar por meio de simulações computacionais e experimentos de laboratório, um novo algoritmo de detecção de incêndios florestais por visão computacional, dotado de uma estratégia inovadora utilizando uma rede neural convolucional, informações meteorológicas e câmeras IP com acesso via internet para a detecção e estimativa do risco de de incêndio florestal.

Os objetivos específicos que devem ser alcançados nesta pesquisa, são:

- Propor uma Rede Neural Convolucional (CNN) para Detecção de Incêndio Florestal;
- Otimizar a rede CNN proposta nesta tese para funcionamento com CPUs de 4 núcleos com consumo de até 1GB de memória RAM;
- Avaliar o método em uma base de dados da literatura;
- Avaliar diferentes métodos de detecção de incêndios por imagem que utilizam CNNs com a mesma base de dados e processamento na nuvem, local e embarcado;
- Investigar o desempenho do algoritmo proposto em acurácia e tempo de processamento;
- Realizar simulações computacionais e experimentos de laboratório com o algoritmo proposto e os presentes no estado da arte que obedecem a premissa de reduzido custo computacional;
- Apresentar uma solução eficiente para detecção e alerta do risco de incêndio florestal utilizando videomonitoramento;

## 1.2 Trabalhos publicados

Durante esta pesquisa de doutorado foram escritos dois artigos científicos sobre a temática abordada. Estes artigos foram submetidos, aceitos e publicados. Portanto, esta tese encontra-se fundamentada nestes dois artigos científicos:

1. **Almeida, J. S.**, Senthil Kumar Jagatheesaperumal, Fabrício G. Nogueira, e Victor Hugo C. de Albuquerque (2023). **EdgeFireSmoke++: A novel lightweight algorithm for real-time forest fire detection and visualization using internet of things-human machine interface**. Expert Systems with Applications. (Qualis A1 em Eng. IV, Fator de Impacto 8.665).
2. **Almeida, J. S.**, Huang, C., Nogueira, F., Bhatia, S., and Victor Hugo C. de Albuquerque (2022). **EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire-Smoke Detection**. IEEE Transactions on Industrial Informatics. (Qualis A1 em Eng. IV, Fator de Impacto 11.648).

### 1.2.1 Organização da Tese

Esta tese encontra-se organizada em cinco capítulos. Antes de finalizar este capítulo introdutório, aproveito para lhe agradecer, querido leitor, pela dedicação em ler esta tese, pois ao chegar nesta página acredito que o texto tenha despertado a sua curiosidade para saber mais detalhes desta pesquisa.

A Introdução, Capítulo 1, é responsável por situar o leitor no contexto da pesquisa, forne-

cendo uma visão geral do tema, sua importância e relevância. Além disso, a introdução apresenta os objetivos da pesquisa e a problemática abordada, definindo claramente o escopo do estudo.

O capítulo 2, Fundamentação Teórica, é de extrema importância para a compreensão de aspectos construtivos do novo modelo CNN proposto nesta tese. A seção detalha a origem das Redes Neurais Convolucionais e seus principais componentes estruturais, além de apresentar uma arquitetura típica de CNN.

No capítulo 3, Metodologia, são descritos os métodos e procedimentos utilizados na pesquisa. A seção detalha a abordagem adotada, as técnicas de coleta de dados, materiais utilizados e as etapas de análise. A metodologia é exposta de forma clara e precisa, permitindo que outros pesquisadores possam replicar o estudo.

O capítulo 4, Resultados, é onde os achados da pesquisa são apresentados de maneira organizada e analisados em relação aos objetivos propostos. Nesta seção, são utilizadas tabelas, gráficos e outros recursos para ilustrar e interpretar os dados coletados. Neste capítulo é enfatizado a relevância dos resultados obtidos em relação ao contexto da pesquisa.

Por fim, o capítulo 5, Conclusão, sintetiza os principais resultados e conclusões da pesquisa. Nesta seção, é feita uma análise crítica dos resultados à luz dos objetivos e da literatura relevante. Além disso, a conclusão destaca as contribuições do estudo para o campo de pesquisa, discute possíveis limitações e sugere áreas para futuras investigações.

## 2 CAPÍTULO II: FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresentaremos informações fundamentais para a compreensão desta tese, tais como as origens das CNNs, como é a arquitetura básica e seus principais componentes. Veremos conceitos importantes para implementá-las. Por fim, comentaremos sobre os primeiros modelos CNNs criados que serviram de fundamentação para as redes neurais convolucionais mais avançadas e a proposta nesta pesquisa.

### 2.1 A origem das CNNs

As Redes Neurais Convolucionais (CNNs do inglês: *Convolutional Neural Networks*) emergiram nos anos 1980, época em que deu-se início ao estudo do córtex visual do cérebro humano para aplicação no reconhecimento de imagens por computadores.

A partir dos anos 2000, com o aumento significativo do poder computacional, as CNNs conseguiram alcançar um desempenho sobre-humano em algumas tarefas visuais complexas. Atualmente, elas estão sendo utilizadas em todos os campos da ciência. Há registros de aplicações na área médica, na indústria, na segurança, na economia e em muitos outros setores da sociedade.

Na segunda metade do século XX, dois importantes cientistas, David H. Hubel e Torsten Wiesel, realizaram uma série de experimentos com gatos entre os anos 1958 e 1959 e alguns anos depois com macacos (HUBEL, 1959; HUBEL; WIESEL, 1959). Motivados pelo entendimento do córtex visual humano, estes cientistas proporcionaram aprendizados relevantes sobre a estrutura do córtex visual.

Em 1981 eles receberam o prêmio Nobel de Fisiologia e Medicina pelo trabalho de pesquisa realizado e pela importante contribuição à ciência. Em resumo, eles mostraram que uma grande parte dos neurônios no córtex visual possuem um pequeno *campo receptivo local*, que reagem apenas a estímulos visuais localizados e uma região limitada do campo visual (veja o exemplo na Figura 5, no qual são representados por círculos tracejados os campos receptivos de cinco neurônios). O campo visual pode ser formado pela sobreposição de vários campos receptivos.

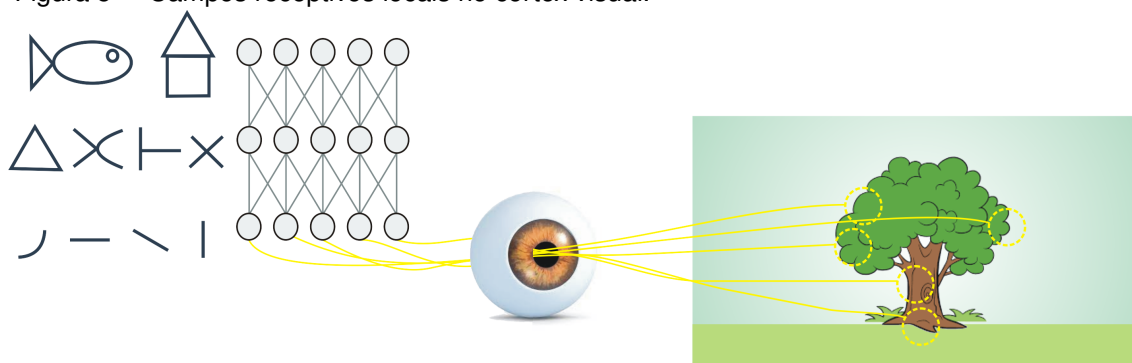
Os achados relevantes incluem o fato de alguns neurônios reagir a apenas a imagens nas linhas verticais, enquanto outros reagem a linhas em diferentes orientações. Eles notaram que existem neurônios que reagem a padrões mais complexos e possuem campos receptivos maiores, que caracterizam-se por conter combinações de dois ou mais campos receptivos de nível inferior.

A partir destes achados, chegaram a conclusão de que os neurônios de nível superior são fundamentados nas saídas dos neurônios próximos de nível inferior. Em qualquer área do campo visual, esta arquitetura é capaz de detectar todos os tipos de padrões complexos.

No ano de 1980 esses estudos do córtex visual serviram de inspiração para a criação do *neocognitron* (LECUN *et al.*, 1998), que aos poucos evoluiu para o que hoje chamamos de Redes Neurais Convolucionais.

Em 1998 foi criada por Yann LeCun, Léon Bottou, Yoshua Bengio e Patrick Haffner, a famosa arquitetura LeNet-5, amplamente utilizada para reconhecer números em cheques manuscritos. Este foi

Figura 5 – Campos receptivos locais no córtex visual.



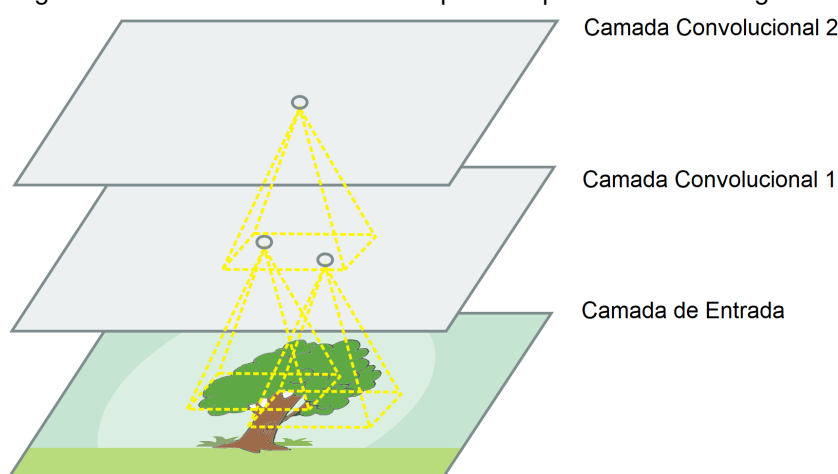
Fonte: Próprio Autor, adaptado de Géron (2019).

um importante marco no campo da visão computacional. Adiante, discutiremos sobre essa arquitetura e seus blocos de construção, como camadas totalmente conectadas, funções de ativação sigmóide, camadas convolucionais e camadas de *pooling*.

## 2.2 Camada Convolutiva e os Campos Receptivos

A camada convolutiva é o componente mais importante de uma CNN. Os neurônios na primeira camada convolutiva estão conectados apenas a pixels em seus campos receptivos (veja Figura 6). Cada neurônio na segunda camada convolutiva está conectado apenas a neurônios dentro de uma área delimitada retangular na primeira camada. Desta forma, esta arquitetura possibilita que a rede tenha foco em características de baixo nível na primeira camada oculta e, em seguida, os reúna em características de nível superior de maneira a repetir essa lógica nas camadas posteriores. Em imagens do mundo real essa estrutura sequencial é comum e constitui a razão para o ótimo desempenho das CNNs no reconhecimento de imagens.

Figura 6 – Camadas CNN com campos receptivos locais retangulares.

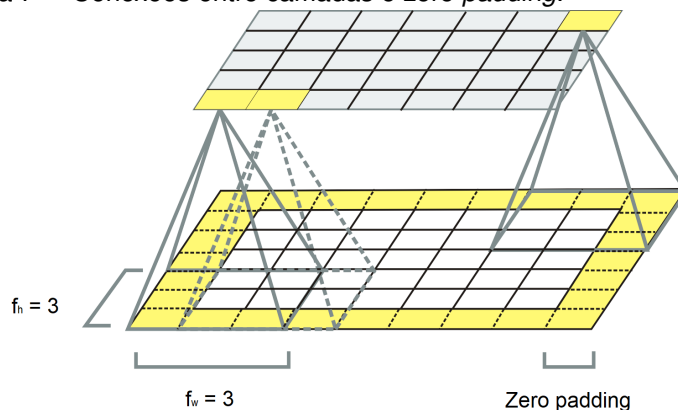


Fonte: Próprio Autor, adaptado de Géron (2019).

Na ilustração a seguir da Figura 7, a altura e largura do campo receptivo são representadas por  $f_h$  e  $f_w$ . Observe que um neurônio de endereço na linha  $i$ , coluna  $j$ , de uma determinada camada está conectado às saídas dos neurônios na camada anterior localizada nas linhas  $i$  a  $i + f_h - 1$ , e nas

colunas  $j$  a  $j + f_w - 1$ . Conforme mostrado no diagrama, uma prática comum é a de adicionar zeros ao redor das entradas para que uma camada tenha as mesmas dimensões em altura e largura da camada anterior. Este processamento é chamado de *zero padding*.

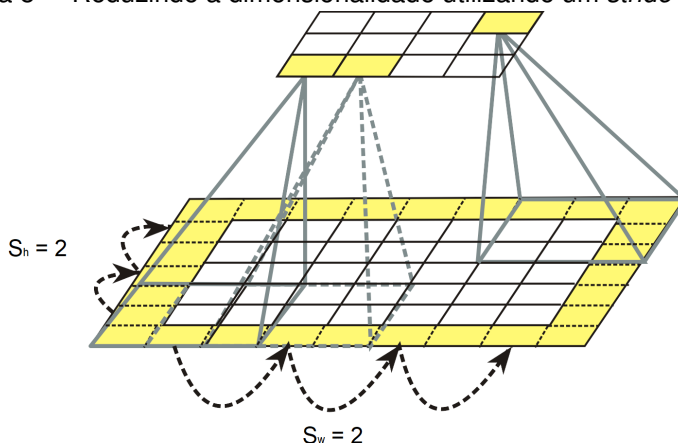
Figura 7 – Conexões entre camadas e *zero padding*.



Fonte: Próprio Autor, adaptado de Géron (2019).

Na Figura 8, um segundo recurso interessante apresentado é o espaçamento entre os campos receptivos, que permite conectar uma grande camada de entrada a uma camada significativamente menor. Por definição, a distância entre dois campos receptivos consecutivos recebe o nome de *stride*. No diagrama apresentado, uma camada de entrada com dimensão  $5 \times 7$ , preenchida com zeros, é conectada a uma camada  $3 \times 4$  usando campos receptivos  $3 \times 3$  e um *stride* de 2. Na linha  $i$ , coluna  $j$  está localizado um neurônio na camada superior, no qual está ligado às saídas dos neurônios na camada anterior localizada nas linhas  $i \times s_h$  a  $i \times s_h + f_h - 1$ , e nas colunas  $j \times s_w$  a  $j \times s_w + f_w - 1$ , em que  $s_h$  e  $s_w$  são os *strides* verticais e horizontais.

Figura 8 – Reduzindo a dimensionalidade utilizando um *stride* 2.



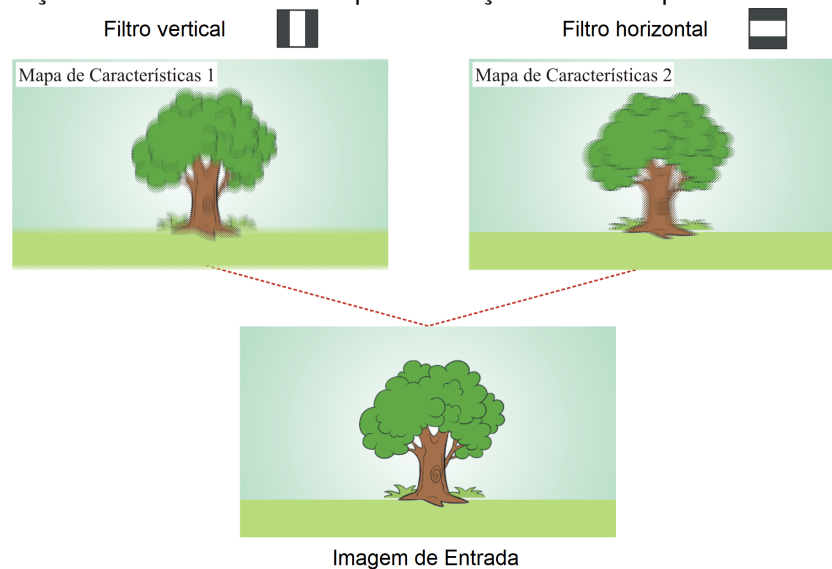
Fonte: Adaptado de Géron (2019).

### 2.3 Filtros e Pesos da Camada Convolutiva

Os pesos de um neurônio podem ser representados por uma imagem com dimensões idênticas a um campo receptivo. Por definição, os conjuntos possíveis de pesos são chamados de

filtros ou *kernels de convolução*. Logo, peso e filtro são o mesmo parâmetro da rede. Por sua vez, os filtros contém padrões geométricos que serão procurados na imagem de entrada. Geralmente o filtro é representado por uma matriz  $p \times q$ , e possuem dimensões ímpares de tamanho  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ , e assim por diante. No exemplo da Figura 9 foram aplicados os filtros de linha vertical e horizontal  $7 \times 7$  na imagem de entrada. Como resultado, as linhas brancas da imagem de entrada são aprimoradas no sentido da linha branca contida no filtro, enquanto o restante é desfocado. Desta forma, para cada filtro será obtido um mapa de características, no qual para  $n$  filtros obter-se-á um total de  $n$  mapas de características. Assim, os mapas de características contém as áreas de uma imagem mais similar ao filtro. Durante o treinamento, uma CNN encontra os filtros mais úteis para sua tarefa e aprende a combiná-los em padrões mais complexos, como por exemplo uma cruz, resultante da combinação entre o filtro vertical e horizontal.

Figura 9 – Aplicação de dois filtros distintos para obtenção de dois mapas de características.



Fonte: Próprio Autor, adaptado de Géron (2019).

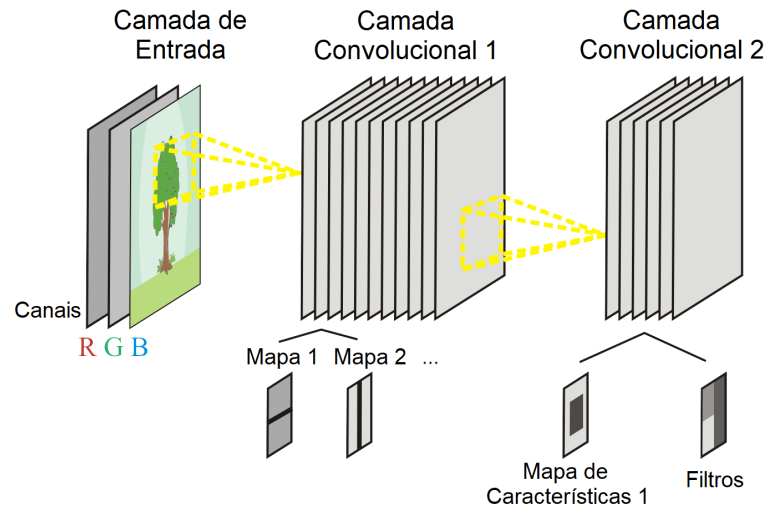
## 2.4 Representação dos Mapas de Características

A camada convolucional é composta por vários mapas de tamanhos iguais, sendo melhor representada em 3D, por meio do empilhamento exemplificado na Figura 10.

Essa camada aplica simultaneamente vários filtros às suas entradas, tornando-a capaz de detectar muitas características em qualquer região em suas entradas.

As imagens de entrada são compostas de múltiplas subcamadas, cada uma referente a um canal de cor. Nas imagens RGB, temos três canais: vermelho, verde e azul. Porém existem outros sistemas de cor que podem vir a ser utilizados, como o escala de cinza de único canal.

Figura 10 – Imagem de entrada RGB e camadas de convolução com vários mapas de características.



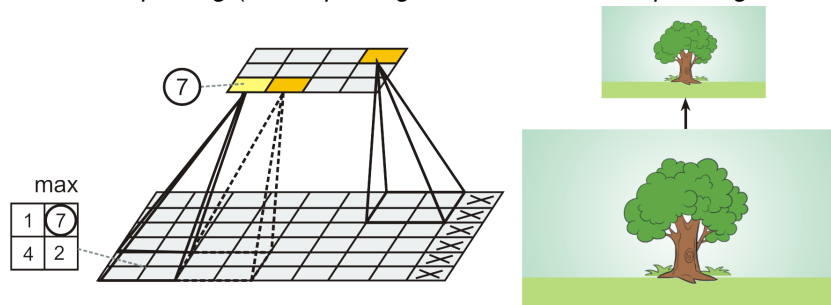
Fonte: Próprio Autor, adaptado de Géron (2019).

## 2.5 Camada Pooling

A camada *pooling* tem por objetivo subamostrar a imagem de entrada. Seus principais efeitos são: redução da carga computacional, menor consumo de memória e redução no número de parâmetros.

Um neurônio de *pooling* não tem pesos. As funções de agregação mais comuns utilizadas são a função máximo ou média. No exemplo da Figura 11 utilizamos um *pooling kernel*  $2 \times 2$ , um *stride* de valor 2 e nenhum preenchimento. Perceba que apenas o valor máximo de entrada em cada kernel chega à camada seguinte e as outras entradas são descartadas.

Figura 11 – Camada *max pooling* (*kernel pooling*  $2 \times 2$ , *stride* 2, sem *padding*).



Fonte: Próprio Autor, adaptado de Géron (2019).

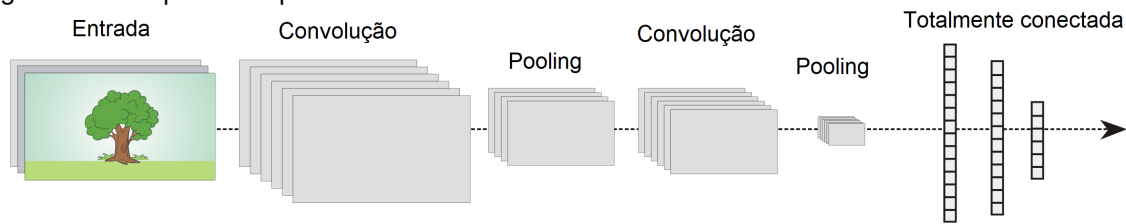
É preciso cautela com o uso desta camada, pois a utilização de um simples kernel  $2 \times 2$ , com incremento de 2, gera uma saída duas vezes menor em ambas as direções (vertical e horizontal), descartando 75% de informação da imagem de entrada.

## 2.6 Arquiteturas CNN

Nas arquiteturas CNN típicas algumas camadas convolucionais são empilhadas, alternando entre camadas *pooling* e assim por diante, como apresentado no exemplo da Figura 12. Após a última camada *pooling* geralmente tem-se uma rede neural totalmente conectada, com uma ou mais camadas

ocultas e uma camada de saída responsável por gerar probabilidades estimadas de classe.

Figura 12 – Arquitetura típica de uma CNN.



Fonte: Próprio Autor, adaptado de Géron (2019).

Variantes desta arquitetura convencional foram desenvolvidas ao longo dos anos, promovendo avanços surpreendentes. Entre as principais arquiteturas CNN criadas está a LeNet-5, que foi proposta por LeCun *et al.* (1998) e foi originalmente avaliada em um problema de reconhecimento de dígitos manuscritos por imagem. Uma outra arquitetura muito conhecida foi a AlexNet, proposta por Krizhevsky *et al.* (2012), possui semelhanças com a LeNet-5, porém, têm mais camadas convolucionais e maior profundidade. A GoogLeNet, criada por Szegedy *et al.* (2014), permitiu obter melhorias significativas em relação aos métodos mencionados anteriormente, chegando a ser a vencedora na competição *ImageNet Large Scale Visual Recognition Challenge 2014* (ILSVRC) (ILSVRC2014, 2014). Além destas já mencionadas, foi criada por He *et al.* (2016) a Residual Network, conhecida por ResNet, que tem muitas semelhanças com o modelo GoogLeNet.

As arquiteturas CNNs mais recentes criadas a partir destas precursoras, que abordaremos nesta pesquisa, utilizaram-nas como fundamento, fazendo melhorias contínuas em prol da busca pelo menor erro ao classificar padrões específicos em imagens digitais. Entre estas CNNs, pode-se mencionar a VGG-16 e VGG-19 (SIMONYAN; ZISSERMAN, 2014; HUANG *et al.*, 2017), a MobileNetV2 (SANDLER *et al.*, 2018), a InceptionV3 (AGARWAL; JHA, 2021), a Xception (CHOLLET, 2017) e a DenseNet121 (CHATRAGADDA *et al.*, 2022). Em outros trabalhos mais recentes, voltados à área de detecção de fumaça e incêndios por imagem, pode-se destacar a EdgeFireSmoke (SILVA *et al.*, 2022), Edge-Intelligence (MUHAMMAD *et al.*, 2020a) e a Energy-Efficient (KHAN *et al.*, 2019).

Por fim, um problema comum percebido com o uso destas CNNs foi o elevado consumo de memória RAM exigido pelas camadas convolucionais, especialmente durante o treinamento. Sabe-se que uma rede com várias camadas de convolução é capaz de exigir facilmente um elevada capacidade de memória. Por isso, é importante que as redes sejam otimizadas para a execução de determinadas tarefas, pois assim é possível construir redes menos complexas e com menor demanda de memória para o seu funcionamento.

## 2.7 Conclusão deste Capítulo

Este capítulo apresentou a teoria fundamental do processo de construção e funcionamento das redes neurais convolucionais. Além do mais, apresentamos a arquitetura típica de uma CNN e as abordagens mais recentes da literatura utilizadas como fundamentação neste estudo. No próximo capítulo, veremos sobre a metodologia empregada nesta tese.



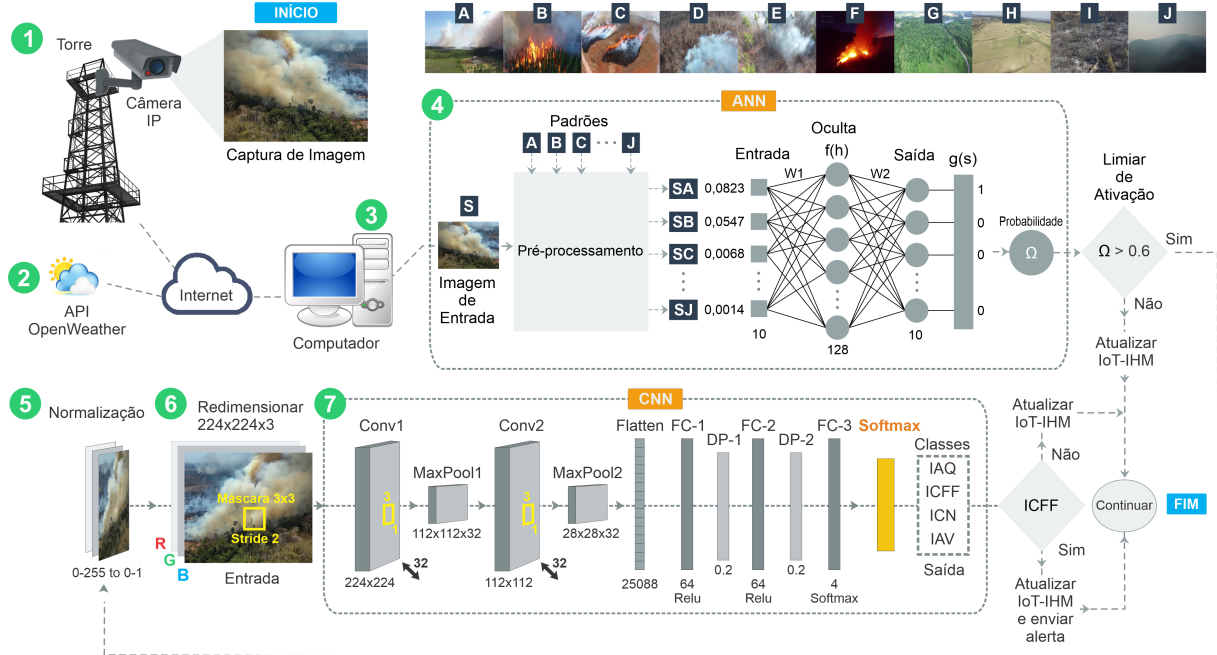
### 3 CAPÍTULO III: METODOLOGIA

Neste capítulo, será explicado os materiais e métodos empregados nesta tese. Será detalhado sobre o algoritmo de detecção de incêndio florestal proposto e as suas várias etapas, que envolvem: a aquisição das imagens, fatores climatológicos, pré-processamento, a arquitetura de rede neural artificial desenvolvida e as redes neurais convolucionais avaliadas, os experimentos computacionais realizados, a estimativa do risco e incêndio, além de aspectos relacionados ao conjunto de dados utilizado e escolha das métricas de avaliação.

#### 3.1 O algoritmo *EdgeFireSmoke*

O algoritmo de detecção de incêndio florestal, ilustrado na Figura 13, foi elaborado em dois níveis principais, na qual utilizou-se técnicas de processamento de imagem digital, aprendizagem de máquina e redes neurais convolucionais para formular o novo algoritmo proposto. No primeiro nível, empregou-se uma Rede Neural Artificial, RNA (no inglês, ANN: *Artificial Neural Network*) com a finalidade de detectar a ausência ou presença de ambiente florestal após a captura de uma imagem digital. No segundo nível, construiu-se a arquitetura de uma nova Rede Neural Convolucional (do inglês, CNN: *Convolutional Neural Network*) capaz de identificar quatro situações distintas de ocorrência de incêndios.

Figura 13 – Modelo conceitual proposto e ilustrado em sete etapas. (1) Representa a forma de captura das imagens; (2) Dados meteorológicos obtidos a partir da internet; (3) Computador representando a máquina para execução do algoritmo; (4) RNA para triagem das imagens de entrada; (5) Normalização de imagens; (6) Redimensionamento de imagem para processamento com a CNN; (7) CNN proposta por (SILVA *et al.*, 2022).



Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Vale comentar que na Figura 13, etapa 4, a caixa de pré-processamento representa as técnicas de transformação de cores, histograma e sua normalização e a comparação dos histogramas. O

algoritmo proposto nesta pesquisa originou-se da junção entre os métodos ANN e CNN. Por fim, adiante discutiremos os aspectos técnicos e metodológicos avaliados.

### **3.1.1 Aquisição das imagens**

A captura de imagens é uma etapa relevante no mecanismo de detecção e classificação de incêndio florestal por imagem. A depender da resolução do sensor, de aspectos construtivos e tempo de captura necessário pela câmera, pode-se ter uma influência significativa no desempenho almejado pelo algoritmo de visão computacional.

As câmeras convencionais com sensor CCD e CMOS, como as presentes em smartphones e câmeras DSLR, são amplamente utilizadas para capturar imagens de incêndios florestais em diferentes cenários. Essas imagens são valiosas para avaliar a extensão do fogo, sua proximidade a áreas habitadas e outras informações essenciais. Pesquisas recentes, como a que realizamos nesta tese (ALMEIDA *et al.*, 2023), demonstram a importância da aquisição de imagens em tempo real por meio de câmeras convencionais para monitorar incêndios.

As câmeras térmicas detectam a radiação infravermelha emitida por objetos, permitindo a visualização de fontes de calor, como focos de incêndio, mesmo em condições de fumaça densa ou no escuro. Estas câmeras têm desempenhado um papel crucial na detecção precoce de incêndios. Artigos, como o de Sousa *et al.* (SOUSA *et al.*, 2020), destacam a integração de câmeras térmicas com sistemas de monitoramento para identificar incêndios em estágios iniciais.

As câmeras de visão noturna são particularmente úteis para a detecção de incêndios florestais durante a noite ou em condições de visibilidade reduzida. A capacidade de capturar imagens em ambientes de baixa luminosidade é fundamental. Essas câmeras são frequentemente usadas em combinação com sistemas de alerta (MOHNISH *et al.*, 2022). Estudos, como o de Agirman *et al.* (AGIRMAN; TASDEMIR, 2022), exploram o uso de imagens noturnas na detecção de incêndios florestais.

A aquisição de imagens de satélite é essencial para monitorar grandes áreas afetadas por incêndios florestais. Essas imagens fornecem uma visão panorâmica que permite avaliar a extensão do incêndio e sua proximidade a áreas habitadas. Pesquisas, como o estudo de Bin Azami *et al.* (AZAMI *et al.*, 2021), discutem como imagens de satélite podem ser processadas para identificar focos de incêndio e avaliar seu impacto ambiental.

Os drones são uma ferramenta versátil na aquisição de imagens digitais em áreas de difícil acesso. Eles podem ser equipados com câmeras convencionais, câmeras térmicas ou câmeras de visão noturna para capturar imagens de incêndios florestais de ângulos únicos. O estudo de Nguyen *et al.* (NGUYEN *et al.*, 2021) explora o uso de drones na captura de imagens para monitorar incêndios florestais em tempo real.

Nesta pesquisa optou-se por avaliar dois tipos de câmeras distintas. As especificações são apresentadas na Tabela 3.

O primeiro modelo avaliado foi uma câmera IP com conexão WiFi, resolução HD e acesso remoto via protocolo RTSP. Este modelo de câmera é utilizado em aplicações de segurança e video-monitoramento em ambientes internos. Possui lente fixa de 2,8mm, sensor 1/4" HD *Progressive Scan*

Tabela 3 – Descrição dos sensores utilizados para a aquisição de imagens.

	<b>Modelo</b>	<b>Resolução</b>	<b>FOV</b>	<b>FPS</b>	<b>Sensor</b>	<b>Conexão</b>
Cam1	IP Câmera Intelbras Mibo IC3	1280x720	111°	30	1/4	WiFi, RTSP
Cam2	Câmera VGA genérica de PC	640x480	60°	30	1/4	USB, Serial

Fonte: Próprio Autor.

CMOS, Ângulo de visão horizontal de 111° (Diagonal) / 92° (Horizontal), Alcance IR infravermelho de 10 metros, Velocidade do obturador automático, e suporte aos protocolos de comunicação IPv4, TCP/IP, DHCP, Onvif, ICMP e RTSP. Além do mais, é compatível com os padrões Wi-Fi IEEE 802.11b, 802.11g, 802.11n, com frequências 2,4 GHz a 2,4835 GHz, segurança 64 / 128 bit WEP, WPA / WPA2, WPA-PSK / WPA2-PSK, taxa de transmissão 11b: 11 Mbps, 11g: 54 Mbps, 11n: 150 Mbps, ganho da antena 1,7 dBi e compressão de vídeo H.264. A segunda câmera utilizada foi uma câmera USB genérica de computador com resolução VGA, sensor de imagem CMOS colorido de 1/4", resolução 640x480, formato de vídeo no sistema de cores RGB com 24bits, interface de comunicação com protocolo USB 2.0, alcance do foco manual de 3m, compressão de imagem embutida, equilíbrio de branco automático e balanço automático de cor.

### 3.1.2 Fatores Climatológicos

Os dados meteorológicos são fundamentais para que consigamos calcular o índice de deflagração e propagação de incêndios florestais, pois são eles que permitem condicionar o estado de secura dos combustíveis (LOURENÇO *et al.*, 1997). No estudo feito por (LOURENÇO, 1991), verificou-se que temperatura e umidade relativa do ar são variáveis chave para a eclosão de incêndios, enquanto que o vento é o responsável pela rápida propagação. Em um estudo recente feito pelos autores Gomes *et al.* (GOMES *et al.*, 2020), os pesquisadores avaliaram a distribuição dos focos de calor no entorno de rodovias federais no estado do Ceará. Com base nos dados meteorológicos utilizados e a análise dos resultados do índice de risco de incêndio de Angstrom, a pesquisa confirmou que o segundo semestre é o período que apresenta condições atmosféricas favoráveis a ignição e propagação rápida de incêndios em vegetação, principalmente no interior do estado do Ceará.

Na grande maioria, os estudos baseiam-se em informações meteorológicas registradas em *datasets* disponibilizados publicamente. No entanto, nesta tese, será avaliada a ideia de que as informações do clima possam ser advindas de uma API na internet em tempo real. Será avaliada a utilidade da API e se será suficiente para este monitoramento, de forma a suprir a falta de uma rede de estações meteorológicas locais em cada região. Por isso, será utilizada a API *OpenWeather*, uma plataforma gratuita que fornece dados meteorológicos de todo o planeta em tempo real. Portanto, as informações que utilizadas serão:

- Temperatura do ar, em °C;
- Umidade relativa do ar, em %;
- Velocidade dos ventos, em km/h.

### 3.1.3 Arquitetura da RNA para reconhecimento de imagem florestal

Uma fragilidade no método proposto por (SILVA *et al.*, 2022) foi a falta de uma etapa de pré-processamento, capaz de identificar se a imagem de entrada trata-se realmente de um ambiente florestal. A RNA será a responsável por classificar a imagem de entrada em dois grupos: ausência ou presença de ambiente florestal. Dessa forma, reduziremos erros posteriores na etapa de detecção com a CNN.

A sequência de operações matemáticas e de processamento de imagens envolvidas nas etapas de pré-processamento são descritas nas seções a seguir.

#### 3.1.3.1 Transformação de cores

A primeira técnica utilizada no método proposto é a transformação de cores de uma imagem no padrão RGB para o espaço de cores HSV. As transformações de cores formuladas por Gonzalez *et al.* (GONZÁLEZ-MOLES *et al.*, 2021) são expressas de forma genérica pela equação 3.1:

$$g(x, y) = T[f(x, y)] \quad (3.1)$$

em que  $f(x, y)$  é uma imagem de entrada colorida,  $g(x, y)$  é uma imagem de saída após a transformação, e  $T$  representa um operador de transformação na imagem  $f$ .

O modelo de cores RGB (siglas do inglês: *Red, Green, Blue*) é o padrão mais comum encontrado em dispositivos digitais, como: monitores de computador, televisores e sensores de câmeras digitais. É um sistema que utiliza as três cores primárias: vermelho, verde e azul, que de forma aditiva originam as outras cores que conhecemos. Por isso, o modelo RGB foi inspirado neste órgão e escolhido como o mais adequado para dispositivos de captura e visualização de imagens digitais devido a sua capacidade de melhor sensibilizar o olho humano.

Eventualmente, pode ser necessário realizar a transformação de um sistema de cores para outro, como por exemplo a conversão de RGB para HSV (do inglês: *Hue, Saturation, Value*). Enquanto o modelo de cores RGB é orientado ao hardware e voltado apenas para a visualização de imagens digitais, o padrão HSV mostrou-se mais adequado para o processamento em determinadas situações.

A transformação  $T$  no espaço de cores RGB para HSV de uma imagem  $f$  de 8 bits pode ser feita após uma transformação de escala de intensidade dos pixels na imagem  $f$  na faixa de zero a um, separadamente por canal. Matematicamente, utilizam-se as seguintes expressões, sendo a Equação 3.2 para o cálculo do canal  $V$ , de valor:

$$V \leftarrow \max(R, G, B) \quad (3.2)$$

A segunda expressão, Equação 3.3, nos permite calcular o canal  $S$ , de saturação:

$$S \leftarrow \frac{V - \min(R, G, B)}{V} \quad (3.3)$$

A terceira expressão, Equação 3.4, é destinada ao cálculo do canal H, de matiz:

$$H \leftarrow \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)}, & \text{se } V = R \\ \frac{120+60(B-R)}{(V-\min(R,G,B))}, & \text{se } V = G \\ \frac{240+60(R-G)}{(V-\min(R,G,B))}, & \text{se } V = B \\ 0, & \text{se } R = G = B \end{cases} \quad (3.4)$$

Adicionalmente, é preciso considerar algumas particularidades desta transformação. Se o valor de  $H$  for menor que zero, calcular:

$$\text{Se } H < 0, \text{ entao } H \leftarrow H + 360 \quad (3.5)$$

De forma geral, englobando os três canais  $H$ ,  $S$  e  $V$ , os valores calculados na saída devem estar dentro desta faixa de valores:

$$0 \leq V \leq 1.0 \leq S \leq 1.0 \leq H \leq 360 \quad (3.6)$$

Para finalizar a transformação  $T$ , os valores para uma imagem  $f$  de 8 bits são convertidos por canal seguindo as expressões:

$$V \leftarrow 255V, S \leftarrow 255S, H \leftarrow \frac{H}{2} \text{ (para ajustar de 0 a 255)} \quad (3.7)$$

### 3.1.3.2 Histograma e sua Normalização

A segunda técnica utilizada no método proposto é o cálculo do histograma. O histograma de uma imagem digital que possua níveis de intensidade no intervalo  $[0, L - 1]$  é uma função discreta expressa pela Equação 3.8:

$$h(r_k) = n_k \quad (3.8)$$

onde,  $r_k$  representa o  $k$ -ésimo valor de intensidade e o número de pixels da imagem com intensidade  $r_k$  é designado por  $n_k$ .

Considerando uma imagem  $f$  de dimensões  $M \times N$ , em que  $M$  e  $N$  são a dimensão da largura e altura de  $f$ , costuma-se normalizar um histograma pela divisão entre cada um desses componentes pelo número total de pixels da imagem, encontrado pelo produto  $MN$ . Desta forma, a normalização de um histograma resulta na Equação 3.9:

$$p(r_k) = \frac{n_k}{MN}, \text{ to } k = 0, 1, 2, 3, \dots, L - 1. \quad (3.9)$$

Em outras palavras, pode-se dizer que  $p(r_k)$  é uma estimativa de probabilidade de ocorrência do nível de intensidade  $r_k$  em uma imagem  $f$ . Portanto, em um histograma normalizado a soma de todos os seus componentes é igual a 1.

### 3.1.3.3 Comparação de Histogramas

A terceira técnica utilizada foi a comparação de histogramas. Sejam  $H_1$  e  $H_2$  os histogramas de duas imagens  $f_1$  e  $f_2$ , respectivamente. A comparação entre estes dois histogramas pode ser feita utilizando uma medida de similaridade. Neste trabalho, utilizamos a medida de correlação entre duas imagens, que é expressa pela Equação 3.10:

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}} \quad (3.10)$$

onde,

$$\bar{H}_k = \frac{1}{2} \sum_J H_k(J) \quad (3.11)$$

em que  $N$  representa o número total de compartimentos (*bins*) utilizado no cálculo do histograma.

Desta forma, após calcular a comparação entre dois histogramas quaisquer, obtemos como resultado um único valor numérico que expressa em uma escala  $[-1, 0, +1, 0]$  o quão dois histogramas são similares, sendo  $-1$  nenhuma similaridade e  $+1$  máxima similaridade.

### 3.1.3.4 Rede Neural Artificial

A quarta técnica empregada neste trabalho foi a utilização de uma rede neural artificial (MURTAGH, 1991). As redes neurais artificiais (RNAs) têm motivação nos neurônios biológicos do cérebro humano. Foi avaliado nesta tese uma rede neural artificial do tipo *feedforward* com três camadas (BEBIS; GEORGIPOULOS, 1994), contendo a camada de entrada, uma única camada intermediária, e a camada de saída. A camada intermediária confere à RNA uma maior capacidade em termos computacionais na aproximação de funções contínuas. A camada intermediária incrementa a capacidade de aprendizado da rede neural quando comparada a rede *feedforward* de uma única camada. Por se tratar de uma rede neural de múltiplas camadas, para treiná-la com a técnica de aprendizado supervisionado, foi necessário utilizar-se da forma generalizada da regra Delta, conhecida por algoritmo de retropropagação de erros ou *Backpropagation* (RUMELHART *et al.*, 1986).

Foi utilizado neste trabalho a forma *off-line* de aprendizado supervisionado (WANG *et al.*, 2010), na qual os dados de treinamento são selecionados previamente e permanecem inalterados. Durante a aprendizagem, o modelo tem seus pesos sinápticos ajustados com o *feedback* advindo do erro mensurado na saída da rede. Após a RNA obter uma solução, os seus pesos sinápticos permanecem fixos.

Durante o dimensionamento da RNA para fins de classificação, levamos em consideração que a sua estrutura dependerá de vários fatores, como por exemplo a natureza e complexidade do

problema, dimensão dos dados de entrada, conhecimento prévio da problemática, e como os dados são representados, se por meio de características ou na sua forma original. Por isso, para modelar uma rede minimalista, objetivando funcionar com ótimo desempenho em *hardware* limitados, foi necessário investigar estes fatores a partir das imagens de incêndio florestal a serem utilizadas.

Será discutido agora sobre o treinamento das redes de múltiplas camadas (do inglês, *Multilayer Perceptron*, MLP) (HAYKIN, 1994; GARDNER; DORLING, 1998). Sabe-se que o erro obtido por redes de uma única camada é calculado através da diferença entre a saída desejada e a saída atual da rede. No entanto, para redes MLP esse procedimento pode ser utilizado somente na camada de saída, visto que não existem saídas desejadas para as suas camadas intermediárias.

A solução para o problema de estimar o erro das saídas intermediárias foi formulada em 1986 por (RUMELHART *et al.*, 1986), através do algoritmo *Backpropagation*. Este algoritmo utiliza o cálculo do gradiente descendente para estimar o erro das camadas intermediárias considerando o efeito que estas causam no erro da camada de saída. Desta forma, os pesos das camadas intermediárias são ajustados proporcionalmente pela retroalimentação do erro na saída através das conexões entre as camadas.

O emprego de múltiplas camadas contribui para uma transformação nos dados de entrada, de forma a torná-lo menos complexo ao longo das camadas intermediárias até um nível tratável na saída. Assim, um problema de natureza não-linear pode se transformar em uma abordagem linear na saída da MLP (HAYKIN, 1994).

Logo, uma rede MLP de uma camada intermediária se comporta por meio de duas transformações sucessivas, expressas matematicamente por  $H(x; w_H)$ , para a camada intermediária, e  $Y(H(x, w_H); w_s)$ , relativa à camada de saída, no qual  $w_H$  corresponde ao vetor de pesos da camada intermediária e  $w_s$  ao vetor de pesos da camada de saída.

Uma vez identificada a necessidade de utilizar uma rede neural de múltiplas camadas, torna-se necessária a determinação das funções de ativação da camada intermediária e de saída, e o número de neurônios em cada uma delas. Neste trabalho, utilizou-se a função de ativação *sigmoide* na camada intermediária e a função *argmax* na camada de saída (RASAMOELINA *et al.*, 2020).

A fim de produzir a saída  $h$  dos neurônios da camada intermediária a partir dos vetores de peso  $w1 = (w1_1, w1_2, \dots, w1_n)^T$  e de entrada  $x = (x_1, x_2, \dots, x_n)$  foi adotada uma função de ativação totalmente diferenciável, a função de ativação sigmoial:

$$g(h) = \frac{1}{1 + e^{-h}} \quad (3.12)$$

Na camada de saída foi empregada a função de ativação *argmax* de forma a gerar a saída  $y$  a partir dos vetores de peso  $w2 = (w2_1, w2_2, \dots, w2_n)^T$  e de saída da camada intermediária  $h = (h_1, h_2, \dots, h_n)$ . Esta função é utilizada para identificar o argumento máximo no vetor de saída  $y$ . Em outras palavras, a função *argmax* permite encontrar a classe com a maior probabilidade prevista, e pode ser expressa pela Equação 3.13:

$$\operatorname{argmax}([z_1, z_2, \dots, z_n]) = [y_1, y_2, \dots, y_n] = [0, 1, \dots, 0] \quad (3.13)$$

onde  $y_i = 1$  se é o único valor máximo no vetor  $\bar{z}$ .

Uma nova etapa foi adicionada na saída da rede para permitir um ajuste que chamaremos de limiar de ativação, representado por  $\Omega$ . Este limiar, ilustrado na Figura 13, permite estabelecer um valor de probabilidade na qual o algoritmo fornece uma saída confiável (KHALIFE *et al.*, 2023), e é expresso por:

$$\Omega = \begin{cases} 1, & \text{se } p \geq 0,6 \\ 0, & \text{se } p < 0,6 \end{cases} \quad (3.14)$$

onde  $p$  é o valor associado à classe com a maior probabilidade prevista.

O valor limite  $\Omega$  foi escolhido com base nos ensaios experimentais realizados em laboratório. Posteriormente, normalizou-se entre o intervalo de 0 a 1 para adequar as condições ambientais de implantação do modelo CNN. Esses limites normalizados contribuem significativamente para a redução da falsa detecção de incêndio. Na Tabela 4, é apresentado um resumo dos parâmetros configurados no modelo RNA.

Tabela 4 – Um resumo dos parâmetros da RNA.

Topologia	Taxa de aprendizado	Treinamento	Função de ativação
Três camadas (entrada, oculta, saída) 10/128/10	0,01	<i>Backpropagation</i>	Sigmoide na camada oculta e <i>Argmax</i> na saída

Fonte: Próprio Autor.

Por fim, na Tabela 5 é apresentado um breve resumo das etapas de pré-processamento e suas contribuições significativas que antecedem a rede neural convolucional que será abordada adiante.

Tabela 5 – Um breve resumo e descrição das etapas de pré-processamento.

<b>Transformação de cores</b>
Conversão da imagem de entrada no padrão RGB para HSV. O sistema de cores HSV provou ser mais adequado para processamento de imagens em situações de incêndio florestal.
<b>Histograma e sua Normalização</b>
Esta técnica permite transformar a imagem digital em uma representação estatística simplificada.
<b>Comparação de Histograma</b>
Esta técnica permite calcular a similaridade entre a imagem de entrada e os padrões de interesse. Sua saída gera um vetor de coeficientes para cada padrão comparado.
<b>Rede Neural Artificial (RNA)</b>
A aplicação desta ANN corrige alguns erros de classificação que surgem na etapa de comparação de histograma. Sua saída permite detectar a presença ou ausência de ambiente florestal.

Fonte: Próprio Autor.

### 3.1.4 Arquitetura CNN para detecção de incêndios florestais

A CNN proposta nesta tese e publicada recentemente por Silva *et al.* (2022) foi construída para ser utilizada com dispositivos de borda, onde o método *EdgeFireSmoke* foi otimizado computacionalmente para ser embarcado em GPUs básicas com mínimo de 1GB de RAM. Otimizou-se o



método *EdgeFireSmoke* removendo camadas convolucionais em excesso, com base na CNN proposta no trabalho de (KHAN *et al.*, 2019). O experimento computacional realizado para otimização da CNN foi a busca em grade (*Grid Search*), no qual foi avaliado o desempenho variando vários parâmetros, especificados na Tabela 6. Os melhores parâmetros encontrados estão identificados em itálico. A técnica de busca em grade tem sido utilizada em muitos trabalhos na literatura, a exemplo, Priyadarshini e Cotton (2021) aplicaram a técnica para encontrar os melhores parâmetros de uma CNN. Com base nos experimentos computacionais realizados nesta tese, percebeu-se que duas camadas convolucionais são o requisito mínimo para resolver este problema de 4 classes. Nos experimentos realizados, o número de filtros foi reduzido de 64 para 32 filtros em cada camada convolucional (SILVA *et al.*, 2022). Além disso, uma redução significativa no consumo de memória foi obtida removendo o excesso de camadas convolucionais, filtros em excesso e usando um *stride* igual a 4 na camada *maxpooling 2*. O método proposto possui o menor número de parâmetros. Além disso, pode ser vinculado a um sistema de CFTV IP para detectar incêndios e alertar os usuários pela internet.

Tabela 6 – Busca em grade pelos melhores parâmetros da CNN proposta por (SILVA *et al.*, 2022).

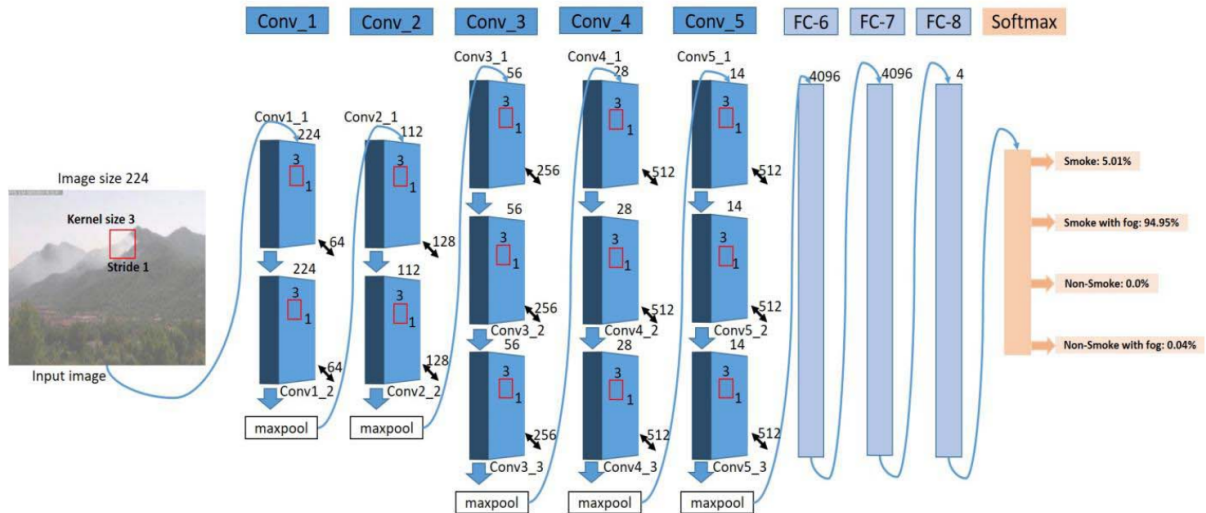
CNN	Parâmetros da CNN	Recursos extras
EdgeFireSmoke	Camadas convolucionais: 1, 2, 4, 8;	<i>Pooling (LECUN et al., 1998)</i>
	Filtros: 2, 4, 8, 16, 32, 64;	<i>Flatten (LECUN et al., 1998)</i>
	Neurônios: 8, 16, 32, 64, 128, 256, 512;	<i>Dropout (SRIVASTAVA et al., 2014)</i>

Fonte: Próprio Autor.

Os autores Khan *et al.* (2019), propuseram o método *EnergyEfficient* que é baseado no modelo VGG16, treinado a partir do ImageNet. A arquitetura é mostrada na Figura 14. O tamanho da imagem de entrada é fixado em 224x224x3 pixels. Cada imagem é passada por cinco camadas de convolução diferentes. A primeira convolução compreende duas camadas convolucionais com um tamanho de entrada de 224x 224 usando 64 *kernels* de tamanho 3x3 com *stride* 1. O resultado é então propagado para a camada de *maxpooling* com *kernel* 2x2 e *stride* 2 para obter as ativações máximas dos mapas de características. A segunda convolução consiste em duas camadas de convolução com um tamanho de entrada de 112x 112 seguido por *maxpooling*. A terceira convolução consiste em três camadas convolucionais com 256 filtros de tamanho 3x3 com *stride* 1 e um tamanho de entrada 56x56 seguido por *maxpooling*. A próxima convolução tem tamanho de entrada de 28x28. Uma pilha dessas camadas convolucionais é seguida por três camadas totalmente conectadas. A primeira e a segunda camada totalmente conectadas têm 4096 neurônios, enquanto a terceira camada totalmente conectada foi modificada de 1000 para 4 neurônios. Os autores usaram as técnicas de transferência de aprendizagem (*Transfer Learning*), muito utilizada por vários outros trabalhos (PAN; YANG, 2010) e fizeram o ajuste fino (*Fine Tuning*) (YOSINSKI *et al.*, 2014) para treinar este modelo. A última camada foi personalizada para as quatro classes do conjunto de dados. Este método é complexo e possui muitas camadas convolucionais, o que torna o processamento custoso em termos computacionais. Ao avaliar o propósito de uso desta rede neural convolucional foi percebido que o método está sobredimensionado. Originalmente, o modelo VGG16, assim como outros modelos mais complexos, foi pensado para resolver um problema originalmente de 1000 classes do ImageNet. Portanto, este foi um ponto abordado nesta

pesquisa que levou a pensar e criar um modelo de rede neural convolucional otimizado para um problema de até uma dezena de classes.

Figura 14 – Arquitetura original da Energy-Efficient.

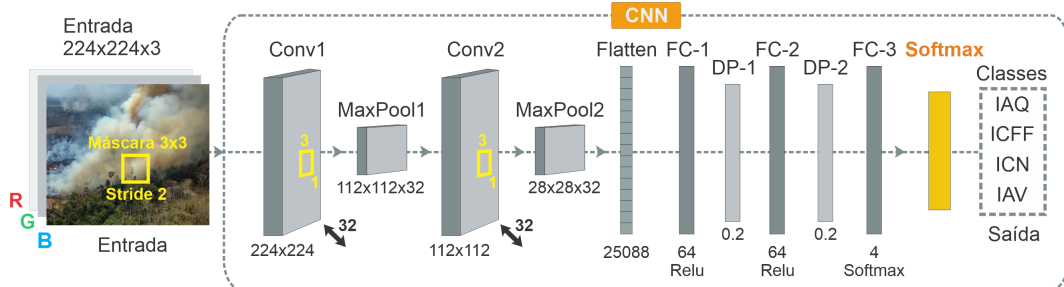


Fonte: Khan *et al.* (2019).

Na Figura 15 é apresentada a arquitetura CNN que foi dimensionada para o método EdgeFireSmoke proposto nesta tese, e a sua versão melhorada EdgeFireSmoke++. A rede CNN proposta tem duas camadas convolucionais com 32 filtros em cada camada. A primeira camada convolucional (CONV1) recebe a imagem de entrada e faz operações de convolução gerando 32 mapas de características. A camada *MaxPooling*1, originalmente utilizada por Lecun *et al.* (1998) e também chamada de *Subsampling*, reduz o tamanho original de sua entrada, neste caso, produzindo mapas de características de dimensão  $112 \times 112 \times 32$  e, conseqüentemente, contribui para a diminuição do consumo de memória e redução do custo computacional do modelo (SANTOS *et al.*, 2020). A segunda camada convolucional (CONV2), recebe esses mapas de características reduzidos e realiza operações convolucionais usando 32 filtros. A segunda camada *MaxPooling* é responsável por reduzir em dimensão para  $1/4$  os mapas de ativação vindos da camada anterior, que passam a ter  $28 \times 28 \times 32$  pixels. Por fim, a rede reorganiza na estrutura de uma matriz unidimensional todos os 32 mapas de características, que terão o tamanho de 25.088 valores. Os dados na camada vetorizada (*Flatten*) são os dados de entrada de uma rede neural densa. Melhor explicando, o processo de *flattening* refere-se a uma etapa de pré-processamento comum antes de conectar as camadas densas em uma rede neural convolucional (LECUN *et al.*, 1998). Esta rede neural densa tem uma camada de entrada, duas camadas ocultas totalmente conectadas com 64 neurônios (FC-1 e FC-2), função de ativação ReLu (KRIZHEVSKY *et al.*, 2017) e possui uma camada de saída com quatro neurônios (FC-3). Com base no trabalho de Srivastava *et al.* (2014) adicionamos ao modelo duas camadas *Dropout*, que desativam 20% dos neurônios em cada camada densa para prevenção do *overfitting* (DP-1 e DP-2). Na camada de saída foi usada a função de ativação *Softmax* (LUO *et al.*, 2020). O otimizador Adam (KINGMA; BA, 2014) foi usado durante o treinamento desta rede com uma taxa de aprendizagem de 0,01. A saída da rede neural convolucional treinada dependerá dos pesos carregados, que podem ser obtidos a partir do treinamento

realizado nos conjuntos de dados. O modelo foi implementado usando a biblioteca Tensorflow-CPU 2.0 e a linguagem Python 3.6.

Figura 15 – Modelo CNN compacto utilizado com o algoritmo proposto nesta tese.



Fonte: Próprio autor, originalmente publicado em (SILVA *et al.*, 2022).

Para chegar à arquitetura CNN proposta, realizamos vários experimentos computacionais a partir da configuração usada por (KHAN *et al.*, 2019). Nestes experimentos, reduzimos gradativamente o número de camadas convolucionais, filtros e modificamos o número de neurônios nas duas camadas ocultas usando múltiplos de dois, até atingir o valor mínimo possível sem nenhuma perda significativa de desempenho na detecção de incêndios florestais. Essa otimização foi projetada para construir uma CNN que necessite de menor consumo de memória RAM em comparação com os métodos da literatura. Além da busca em grade (*Grid Search*) outros experimentos computacionais foram realizados. Um resumo dos experimentos realizados neste estudo é apresentado a seguir:

### 1. Experimentos:

- **Simulação Computacional I:** Computação em nuvem, local e embarcado (**Artigo 1, Silva *et al.* (2022)**)
- **Simulação Computacional II:** Computação local (**Artigo 2, Almeida *et al.* (2023)**)

### 2. Recursos:

- Google Colab com GPU Tesla T4, 2560 núcleos, 16GB (Nuvem)
- PC com GPU GTX1650, 896 núcleos, 4GB (Local)
- Kit de desenvolvimento Jetson Nano, GPU 128 núcleos, 4GB (Embarcado)
- PC com CPU AMD Ryzen 5, 2,10GHz, 4 núcleos, 16GB RAM (Local)

No primeiro artigo científico publicado (SILVA *et al.*, 2022), foram realizados os experimentos computacionais descritos na Simulação Computacional I:

#### 1. Simulação Computacional I:

- Busca em grade, dimensionamento da EdgeFireSmoke;
- Treinamento das CNNs (EdgeFireSmoke, Energy-Efficient e Edge-Intelligence);
- Exportação dos modelos (.json e .h5);
- Avaliação embarcado nas GPUs (Tesla T4, GTX1650, Jetson Nano);
- Cálculo das métricas (Acurácia, F1-Score, Recall, Matriz de confusão, Tempo de processamento);
- Verificação do consumo de memória RAM;
- Medição da potência em Watts na GPU necessária para treino e teste;

No segundo artigo científico publicado e abordado com riqueza de detalhes nesta tese (ALMEIDA *et al.*, 2023), foram realizados os experimentos computacionais descritos na Simulação Computacional II:

### 1. Simulação Computacional II:

- Treinamento da RNA para pré-processamento;
- Treinamento das CNNs (EdgeFireSmoke, VGG-16, VGG-19, MobileNetV2, InceptionV3, Xception, DenseNet121);
- Avaliação na CPU AMD Ryzen 5, 2,1GHz, 4 núcleos, 16GB RAM;
- Acompanhamento das métricas (Acurácia, Precisão, Recall, F1-score, FPS, CCGM, CPTR);
- Integração com Dados Meteorológicos a partir da Internet;
- Cálculo do Risco de Incêndio e escala de cores;
- Saída de informações em uma IHM;

Nesta tese fizemos melhorias no algoritmo original da *EdgeFireSmoke* (SILVA *et al.*, 2022) mantendo o propósito de fazer a detecção de incêndio florestal por imagem. O novo algoritmo passou a contar com uma etapa adicional que irá garantir que a imagem a ser processada foi de fato capturada em um ambiente florestal. Assim, a ideia foi reduzir a ocorrência de eventuais falsos alarmes ao detectar incêndio em ambiente não-florestal. Além disso, avaliamos a rede neural artificial combinada com outras CNNs disponíveis na literatura (ALMEIDA *et al.*, 2023). Apresentamos na Tabela 7 um quadro informativo que contém as principais características das redes neurais convolucionais avaliadas neste trabalho.

Tabela 7 – Quadro resumo contendo as principais características das CNNs avaliadas na tese.

CNN	Características	Referência
EdgeFireSmoke	Possui 64 filtros, 2 camadas convolucionais, 2 camadas pooling, 1 camada flatten, Rede neural densa com 132 neurônios e 2 camadas ocultas, 2 camadas dropout 20%.	(SILVA <i>et al.</i> , 2022; ALMEIDA <i>et al.</i> , 2023)
VGG16	Tem 4224 filtros, 13 camadas convolucionais, 5 camadas pooling, rede neural densa com 8196 neurônios e 2 camadas ocultas.	(SIMONYAN; ZISSERMAN, 2014)
MobileNet V2	Possui 3 camadas convolucionais, 7 camadas de transformação e 1 camada pooling.	(SANDLER <i>et al.</i> , 2018)
Inception V3	Tem 92 camadas convolucionais, além de camadas de normalização, ativação, pooling e uma rede neural densa com 8196 neurônios e 2 camadas ocultas.	(AGARWAL; JHA, 2021)
Xception	Possui 4 camadas convolucionais principais, vários blocos de convolução separados, normalização, ativação, pooling e uma rede neural densa com 8196 neurônios e 2 camadas ocultas.	(CHOLLET, 2017)
DenseNet121	Tem 119 camadas convolucionais, vários blocos de normalização, ativação, pooling e uma rede neural densa com 8196 neurônios e 2 camadas ocultas.	(CHATRAGADDA <i>et al.</i> , 2022)
VGG19	Possui 16 camadas convolucionais, 5504 filtros, 5 camadas pooling e uma rede neural densa com 8196 neurônios e 2 camadas ocultas..	(HUANG <i>et al.</i> , 2017)

Fonte: Próprio Autor.

### 3.1.5 Índice do Risco de Incêndio Florestal (IRIF)

O cálculo do risco de incêndio florestal foi formulado por (LOURENÇO *et al.*, 1997), no qual chegaram a conclusão de que o risco de incêndio florestal depende de três variáveis meteorológicas: temperatura e umidade relativa do ar, pois estas contribuem para o estado de secura dos combustíveis e a deflagração do fogo; e a velocidade dos ventos, que contribui para a propagação do fogo sobre a vegetação. Matematicamente, este índice é expresso pela Equação 3.15:

$$IRIF = \frac{T}{U} + \frac{V}{100} \quad (3.15)$$

T - Temperatura do ar, em °C;

U - Umidade relativa do ar, em %;

V - Velocidade do vento, em km/h;

Adaptamos neste trabalho a nomenclatura definida por (LOURENÇO, 1991), que expressa por meio de intervalos uma escala do risco de deflagração e propagação de incêndio florestal, a qual chamaremos de Índice do Risco de Incêndio Florestal (IRIF). Este índice será útil para apresentar uma estimativa do risco de incêndio florestal, especificamente por meio de um retorno visual na interface humano-máquina, IHM (no inglês, HMI: *Human Machine Interface*).

Antes, para melhor compreensão, será apresentada a análise que foi realizada e que possibilitou definir as classes de risco de incêndio com base em situações reais de ocorrência de incêndio florestal. Nesta análise consideramos os dados do clima no Brasil registrados entre os anos de 2021 e 2022 pelo INMET. Os dados foram baixados no site do Banco de Dados Meteorológicos para Ensino e Pesquisa (BDMEP), no seguinte endereço: <https://portal.inmet.gov.br/dadoshistoricos>. Este banco de dados é muito utilizado por outros pesquisadores, por exemplo, foi empregado no trabalho de Lima *et al.* (2021) para avaliar os aspectos sazonais relacionados ao risco de incêndios florestais na região nordeste do Brasil. A primeira situação analisada e apresentada na Tabela 8 engloba o registro de cinco ocorrências de ausência de incêndio nas principais regiões do país. Os registros foram feitos por estações meteorológicas nas localidades especificadas mais próximas a região dos eventos. Enquanto na segunda situação analisada e apresentada na Tabela 9 abordamos o registro de cinco ocorrências de incêndio nessas mesmas regiões. Todas as situações reais dos incêndios utilizados na análise foram veiculadas em meios de comunicação e pelo menos uma gerou publicação científica, como no incêndio ocorrido no Parque Ambiental do Cocó (FILHO; ARAÚJO, 2021). Os eventos podem ser conferidos a seguir:

- **Evento 1 (E1):** Incêndio atinge Parna Jaú e ICMBio pede apoio para combater o fogo (AMAZONIA-REAL.COM.BR, 2021).
- **Evento 2 (E2):** Incêndio no Parque do Cocó teve causa involuntária, diz perícia (DIARIODONOR-DESTE.COM.BR, 2022).
- **Evento 3 (E3):** Incêndio em Chapada dos Guimarães (MT) já dura 5 dias e destruiu 4,6 mil hectares (G1.GLOBO.COM, 2021).
- **Evento 4 (E4):** Bombeiros combatem incêndio no Parque Estadual da Serra da Boa Esperança (G1.GLOBO.COM, 2022a).
- **Evento 5 (E5):** Incêndio atinge 500 hectares da Estação Ecológica do Taim, no Sul do RS (G1.GLOBO.COM, 2022b).

Após o estudo na base de dados do INMET com os casos reais e as descrições dos eventos dadas nos canais de comunicação, elaboramos a Tabela 10 que possibilitou estimar por uma

Tabela 8 – **Situações reais de normalidade**; Registro de dados do clima feitos pelo INMET no primeiro semestre, no período 2021-2022, antes da ocorrência dos eventos.

Região	Estação	Data	UTC	Localidade	T (°C)	U (%)	V (km/h)	IRIF
Norte	Manaus, AM	09/02/2022	13:00	Parque Nacional do Jaú	27,00	85,00	3,60	0,35
Nordeste	Fortaleza, CE	25/02/2021	13:00	Parque Ambiental do Cocó	30,00	67,00	14,40	0,59
Centro-Oeste	Cuiabá, MT	22/02/2021	13:00	Chapada dos Guimarães	29,00	65,00	32,40	0,77
Sudeste	Varginha, MG	05/02/2022	13:00	P. Est. da Serra da Boa Esperança	25,00	79,00	7,20	0,39
Sul	Rio Grande, RS	03/02/2022	13:00	Estação Ecológica do Taim	28,00	73,00	10,80	0,49

Fonte: Próprio Autor.

Tabela 9 – **Situações reais de incêndio florestal**; Registro de dados do clima feitos pelo INMET no segundo semestre, no período 2021-2022. As informações apresentadas fazem referência ao dia dos eventos, sempre às 13 horas UTC.

Região	Estação	Data	UTC	Localidade	T (°C)	U (%)	V (km/h)	IRIF
Norte	Manaus, AM	25/09/2022	13:00	Parque Nacional do Jaú [E1]	33,00	20,00	18,00	1,83
Nordeste	Fortaleza, CE	17/11/2021	13:00	Parque Ambiental do Cocó [E2]	31,00	18,00	14,40	1,87
Centro-Oeste	Cuiabá, MT	22/08/2021	13:00	Chapada dos Guimarães [E3]	33,00	9,00	21,60	3,88
Sudeste	Varginha, MG	25/10/2022	13:00	P. Est. da Serra da Boa Esperança [E4]	23,00	17,00	10,80	1,46
Sul	Rio Grande, RS	12/12/2022	13:00	Estação Ecológica do Taim [E5]	28,00	19,00	14,40	1,62

Fonte: Próprio Autor.

aproximação linear os intervalos para classificação do grau de risco de incêndio florestal.

Tabela 10 – Definição das classes de risco de incêndio florestal.

Risco	Grau de Risco	Intervalo	Escala de Cores
1	Reduzido	0 <IRIF <= 0.5	Azul
2	Moderado	0.5 <IRIF <= 1.0	Verde
3	Elevado	1.0 <IRIF <= 1.5	Amarelo
4	Muito Elevado	1.5 <IRIF <= 2.0	Laranja
5	Extremo	IRIF >2.0	Vermelho

Fonte: Próprio Autor, adaptada de Lourenço (1991).

Para que se entenda o grau de risco, explicamos brevemente adiante com base na teoria formulada por (LOURENÇO, 1991). (LOURENÇO, 1991).

1. **Risco Reduzido:** Podem surgir pequenos focos que normalmente não provocam incêndio.
2. **Risco Moderado:** Podem surgir focos e o fogo tem progressão lenta em área florestal verde. Em áreas secas pode progredir rapidamente com o vento.
3. **Risco Elevado:** Os materiais combustíveis inflamam-se rapidamente. Os incêndios podem ser incontrolláveis se não forem combatidos a tempo. Recomenda-se a vigilância aérea e estado de alerta.
4. **Risco Muito Elevado:** Após a ignição o fogo se estabelece rapidamente, com rápida propagação e aumento de intensidade. Sugere-se a ação de brigadas com intervenção em pontos estratégicos para conter o avanço do fogo.
5. **Risco Extremo:** Fácil ignição, propagação imediata com grande velocidade e intensidade. É uma situação de máxima gravidade, que deve envolver todos os meios de monitoramento e combate disponíveis.

É importante mencionar que os intervalos utilizados podem ser ajustados em trabalhos futuros após um estudo mais aprofundado em campo. Logo, sugerimos neste trabalho que as situações reais abordadas podem ser classificadas conforme apresentado na Tabela 11.

Tabela 11 – Estimativa do grau de risco de ocorrência de incêndios florestais nos casos abordados, considerando os mesmos locais em momentos diferentes antes dos eventos nas seguintes situações: (1) normalidade e (2) ocorrência de incêndio.

Situação	Semestre do ano	Localidade	IRIF	Grau de Risco	Escala de cores
Normalidade	Primeiro	Parque Nacional do Jaú	0,35	Reduzido	Azul
		Parque Ambiental do Cocó	0,59	Moderado	Verde
		Chapada dos Guimarães	0,77	Moderado	Verde
		Parque Estadual da Serra da Boa Esperança	0,39	Reduzido	Azul
		Estação Ecológica do Taim	0,49	Reduzido	Azul
Incêndio	Segundo	Parque Nacional do Jaú	1,83	Muito Elevado	Laranja
		Parque Ambiental do Cocó	1,87	Muito Elevado	Laranja
		Chapada dos Guimarães	3,88	Extremo	Vermelho
		Parque Estadual da Serra da Boa Esperança	1,46	Elevado	Amarelo
		Estação Ecológica do Taim	1,62	Muito Elevado	Laranja

Fonte: Próprio Autor.

### 3.1.6 Visualização de Dados em tempo real (IoT-IHM)

Para a visualizar as imagens em tempo real, necessitaremos de um algoritmo rápido, capaz de fazer o processamento em sincronismo com a taxa de *frames* por segundo (FPS) da câmera. Por isso, serão avaliados o método proposto e os métodos de *deep learning* clássicos da literatura. Foi utilizada a biblioteca OpenCV versão 4.5 com a linguagem Python 3.6.

A Interface Humano Máquina (IHM) teve o seu funcionamento incorporado a um dispositivo de Internet das Coisas (IoT). As duas fontes de informações principais são advindas da internet, sendo elas as imagens da câmera IP e os dados meteorológicos da API OpenWeather (<https://openweathermap.org/>). Portanto, utilizamos um conceito novo a qual chamaremos IoT-IHM, que significa uma interface de visualização de dados inteligente conectada à internet.

### 3.1.7 Datasets utilizados para treinar o algoritmo

Dois conjuntos de dados usados neste trabalho foram compilados e organizados pelos autores. Todas as imagens em ambos os conjuntos de dados foram capturadas em cenários reais de incêndio florestal. Neste trabalho usamos 5.209 imagens para treinar a RNA (*Dataset 1*) e 24.725 imagens para treinar cada CNN separadamente no mesmo dataset (*Dataset 2*). Uma breve descrição delas é dada abaixo.

#### 3.1.7.1 Dataset 1: Imagens florestais

Este conjunto de dados é um subconjunto do *Dataset 2*, no qual foi considerado um maior número de classes, especificamente dez classes. Amostras deste dataset podem ser vistas na Figura 13 e encontram-se nomeadas de A à J. Suas amostras foram importantes para treinar o modelo RNA e torná-lo capaz de reconhecer um ambiente florestal a partir da imagem digital de entrada. Desta forma, a RNA permitirá que o detector CNN tenha uma etapa de triagem que o antecede, a fim de melhorar seu desempenho na detecção de incêndio florestal por imagem. Ao todo são 4.516 imagens para treinamento, 693 imagens para validação e 1.085 imagens para teste. O formato das imagens é o *.jpeg* com resolução de  $512 \times 512$  pixels.

### 3.1.7.2 **Dataset 2: Imagens aéreas de incêndio florestal**

Um segundo conjunto de dados que possui imagens aéreas de incêndios florestais, construído por (SILVA *et al.*, 2022), foi utilizado para completar nosso modelo de visão computacional de detecção de incêndios. Neste conjunto de dados, vídeos públicos gravados por Veículos Aéreos Não-Tripulados (do inglês *Unmanned Aerial Vehicle, UAV*), com cenários de incêndio foram selecionados e processados para construir o novo conjunto de dados. Os vídeos gravados incluem fogo, fumaça, áreas queimadas e áreas verdes, principalmente em áreas florestais. No total, foram disponibilizados 93 vídeos no formato *.mp4*, com resolução HD de  $1280 \times 720$  pixels.

Os vídeos salvos foram convertidos em imagens *.jpeg*, e alguns *frames* foram armazenados e então divididos em quatro classes:

1. *Área queimada (IAQ)*: a classe IAQ possui 9.348 imagens capturadas após a ocorrência do incêndio. Em geral, observa-se que o chão é preto e as árvores estão sem folhas.
2. *Fogo-fumaça (ICFF)*: a classe ICFF possui 15.579 imagens capturadas durante a ocorrência de incêndio, geralmente mostrando fogo, fumaça e o local, como áreas de floresta e vegetação seca ou com pastagem. Observou-se que a fumaça era branca ou escura.
3. *Área com neblina (ICN)*: a classe ICN tem 9.762 imagens capturadas em ambientes com neblina.
4. *Área verde (IAV)*: por fim, temos 14.763 imagens na classe IAV, que inclui imagens de mata, vegetação seca e áreas de pastagem sem fogo, fumaça ou neblina.

A Figura 16 mostra exemplos de imagens do conjunto de dados 1. A Tabela 12 fornece uma descrição dos dados. Combinando esses conjuntos de dados, temos 49.452 amostras e mais de 93 ambientes externos diferentes a partir de 93 vídeos.

### 3.1.8 **Treinamento do Algoritmo**

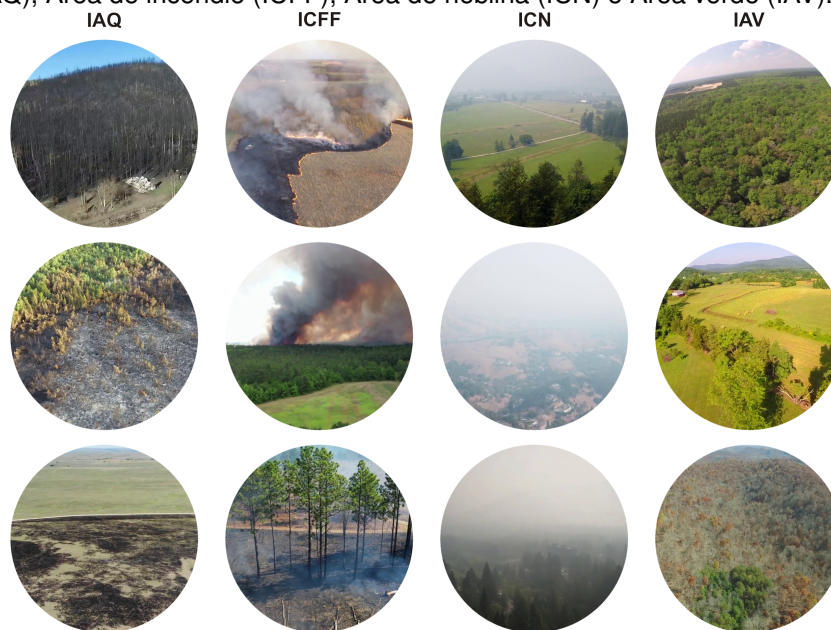
O algoritmo proposto nesta tese é composto pela junção de uma Rede Neural Artificial (RNA) e uma Rede Neural Convolucional (CNN) e foi treinado em duas etapas. Primeiro, treinamos a RNA com amostras do conjunto de dados 1, que foi particionado em três partes: treinamento, validação e teste. Usamos 500 épocas de treinamento para a RNA com uma taxa de aprendizado fixa de 0,01. No treinamento da RNA foi utilizado o algoritmo de retropropagação de erros (*Backpropagation*) proposto em 1986 (RUMELHART *et al.*, 1986). Ao final, obtivemos os vetores de peso  $w_1$  e  $w_2$ . Em seguida, procedemos ao treinamento das CNNs, separadamente, utilizando o conjunto de dados 2. Este segundo conjunto de dados foi particionado em três partes semelhantes ao *dataset* anterior. De acordo com (GÉRON, 2019), os pesos de um neurônio convolucional podem ser representados por uma pequena imagem, também chamada de filtros ou *kernels* de convolução, e como explicado pelo autor, durante o treinamento, uma CNN encontra os filtros mais úteis para sua tarefa e aprende a combiná-los em padrões mais complexos. Ao final do treinamento, obtivemos os modelos de cada CNN treinada com seus respectivos vetores de peso (arquivos *.h5* e *.json*). Para todas as CNNs fizemos o treinamento com 25 épocas de aprendizagem. As imagens foram redimensionadas para  $224 \times 224 \times 3$  pixels e normalizadas entre 0 e 1 antes do treinamento. Usamos um tamanho de lote 4 em todas as situações de



treinamento da CNN. Usamos o otimizador SGD (no inglês: *Stochastic Gradient Descent*) para treinar os modelos VGG16, Inception V3, Xception, DenseNet121 e VGG19. Para os modelos MobileNetV2 e com a EdgeFireSmoke usamos o otimizador Adam. Para treinar todas essas CNNs usamos uma taxa de aprendizado de 0,001. Aplicamos a técnica de transferência de aprendizado (do inglês: *Transfer Learning*) para treinar essas CNNs usando os pesos ImageNet originais, exceto para EdgeFireSmoke que foi treinada a partir de um modelo sem conhecimento prévio.

O código-fonte do método proposto e as bases de dados estão disponíveis no repositório<sup>1</sup>.

Figura 16 – Imagens representativas de incêndios florestais do conjunto de dados 2. Ele contém imagens reais de incêndios florestais e imagens *Hazy* reais. Marcadores: Área queimada (IAQ), Área de incêndio (ICFF), Área de neblina (ICN) e Área verde (IAV).



Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Tabela 12 – Descrição do conjunto de dados 2 com as classes: IAQ, ICFF, ICN e IAV.

Grupos	Imagens	Percentual (%)	IAQ	ICFF	ICN	IAV
Treinamento	9892	20	1870	3116	1953	2953
Validação	14833	30	2804	4673	2928	4428
Teste	24727	50	4674	7790	4881	7382
<b>Total</b>	<b>49452</b>	<b>100</b>	<b>9348</b>	<b>15579</b>	<b>9762</b>	<b>14763</b>

Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

### 3.1.9 Pseudocódigo

Sabemos que o pseudocódigo é uma forma de representação abstrata de algoritmos e permite expressar a lógica de um problema sem a necessidade de detalhes específicos de uma linguagem de programação. Sua importância na engenharia elétrica e na computação reside na sua capacidade de auxiliar no planejamento, na compreensão e na comunicação de algoritmos entre programadores.

<sup>1</sup> <https://sites.google.com/view/edgefiresmoke-plus/home>

Por meio do pseudocódigo, é possível estruturar a sequência de passos de um algoritmo de forma clara e concisa, facilitando a identificação de erros lógicos e a depuração do código. Além disso, o pseudocódigo serve como uma linguagem intermediária, abstraindo a complexidade da codificação em uma linguagem real e permitindo que diferentes programadores colaborem de forma mais eficiente. Sua flexibilidade e versatilidade tornam o pseudocódigo uma ferramenta indispensável na computação, auxiliando no desenvolvimento de algoritmos e programas de maneira mais eficiente e estruturada. Nesta tese elaboramos o pseudocódigo que representa a lógica de funcionamento do algoritmo proposto. Este pode ser visto na Tabela 13.

Tabela 13 – Pseudocódigo do algoritmo proposto e sua aplicação em tempo real.

EdgeFireSmoke
Início
1. Capturar imagem da câmera
2. Ler os dados meteorológicos a partir da internet
3. Detectar a ausência ou presença de ambiente florestal com a RNA
4. Se detectou ambiente florestal, siga ao passo seguinte, caso contrário, pule ao passo 6.
5. Classificar a imagem capturada com o modelo CNN
6. Calcular o Risco de Incêndio e atualizar a interface gráfica IoT-IHM
7. Emitir aviso, caso haja a presença de incêndio florestal.
8. Sair da aplicação caso a tecla 'q' esteja pressionada, caso contrário volte ao passo 1.
Fim

Fonte: Próprio Autor.

### 3.1.10 Métricas de avaliação

Neste trabalho, utilizou-se de sete métricas estatísticas para avaliar os algoritmos, a maior parte delas implementadas a partir da biblioteca *sci-kit learning*: Acurácia, Sensibilidade, Precisão, F1-score e RMSE; além das métricas CCGM e CPTR implementadas a mão e a partir da biblioteca nativa da linguagem Python. Essas métricas são comumente expressas como um valor entre 0 e 1, no entanto, foi conveniente converter os resultados obtidos a partir dessas métricas em uma representação percentual para facilitar a interpretação. Essa conversão permite uma compreensão mais intuitiva de desempenho dos modelos, pois é mais familiar para a maioria das pessoas. Adicionalmente foram mensurados o tempo de processamento por imagem e taxa de quadros por segundo (FPS). A seguir, explicaremos cada uma dessas métricas individualmente.

#### 3.1.10.1 Acurácia

A métrica acurácia é uma medida fundamental na avaliação de modelos e algoritmos em aprendizado de máquina e classificação. Ela representa a taxa de acertos do modelo em relação ao total de amostras avaliadas. A acurácia é amplamente utilizada como uma medida simples e intuitiva para verificar o desempenho de um modelo em relação à sua capacidade de fazer previsões corretas.

Segundo Mitchell *et al.* (2007), a acurácia é definida como a proporção de exemplos corretamente classificados em relação ao total de exemplos avaliados, e expressa em termos percentuais

pela Equação 3.16. É uma métrica que fornece uma visão geral do desempenho do modelo, permitindo uma rápida avaliação de sua capacidade de generalização.

$$Acc(\%) = \frac{VP + VN}{VP + FN + FP + VN} \times 100 \quad (3.16)$$

em que: VP: verdadeiros positivos; FN: falsos negativos; FP: falsos positivos; VN: verdadeiros negativos.

No entanto, é importante ressaltar que a acurácia pode não ser a métrica mais adequada em certos cenários, principalmente quando o conjunto de dados apresenta desequilíbrio de classes. Em situações em que as classes são desproporcionalmente representadas, a acurácia pode ser enganosa e não refletir de forma precisa a performance do modelo.

Em suma, a acurácia é uma métrica importante para avaliar o desempenho de modelos de aprendizado de máquina, fornecendo uma medida da taxa de acertos. No entanto, é necessário considerar suas limitações em casos de desequilíbrio de classes. É recomendável explorar outras métricas e aprofundar-se na literatura científica para uma avaliação mais abrangente e precisa dos modelos.

### 3.1.10.2 **Sensibilidade**

A métrica Sensibilidade, também conhecida como *Recall*, é uma medida importante na avaliação de modelos e algoritmos em aprendizado de máquina, especialmente em problemas de classificação. Ela representa a capacidade do modelo em identificar corretamente os exemplos positivos, ou seja, a proporção de positivos corretamente classificados em relação ao total de exemplos positivos presentes no conjunto de dados.

De acordo com Hastie *et al.* (2009), a Sensibilidade é calculada pela divisão dos verdadeiros positivos (exemplos positivos corretamente classificados) pela soma dos verdadeiros positivos e falsos negativos (exemplos positivos erroneamente classificados como negativos), sendo expressa pela Equação 3.17. É uma métrica que fornece informações importantes sobre a capacidade do modelo em detectar corretamente casos positivos.

$$Sensibilidade(\%) = \frac{VP}{VP + FN} \times 100 \quad (3.17)$$

É importante destacar que a Sensibilidade é particularmente útil em problemas em que a identificação de casos positivos é crucial e a ocorrência de falsos negativos pode ter implicações graves. Por exemplo, em um sistema de detecção de doenças, a Sensibilidade é relevante para identificar corretamente os casos de doença, garantindo que o tratamento seja fornecido adequadamente.

Em resumo, a Sensibilidade é uma métrica fundamental na avaliação de modelos de aprendizado de máquina, indicando a capacidade de identificar corretamente casos positivos. Sua aplicação é especialmente relevante em problemas em que a ocorrência de falsos negativos é indesejável.

### 3.1.10.3 Precisão

A métrica Precisão, também conhecida como Valor Preditivo Positivo, é uma medida importante na avaliação de modelos e algoritmos em aprendizado de máquina, especialmente em problemas de classificação. Ela representa a proporção de exemplos corretamente classificados como positivos em relação ao total de exemplos classificados como positivos pelo modelo.

Segundo Mitchell *et al.* (2007), a Precisão é calculada pela divisão dos verdadeiros positivos (exemplos positivos corretamente classificados) pela soma dos verdadeiros positivos e falsos positivos (exemplos negativos erroneamente classificados como positivos), e pode ser expressa matematicamente pela Equação 3.18. É uma métrica que fornece informações sobre a capacidade do modelo em fazer previsões positivas corretas.

$$Precisao(\%) = \frac{VP}{VP + FP} \times 100 \quad (3.18)$$

A Precisão é especialmente útil em problemas em que é essencial minimizar os falsos positivos, ou seja, evitar a classificação incorreta de exemplos negativos como positivos. Por exemplo, em um sistema de detecção de *spam* de e-mails, a Precisão é relevante para garantir que e-mails legítimos não sejam incorretamente classificados como *spam*.

Em resumo, a Precisão é uma medida essencial na avaliação de modelos de aprendizado de máquina, refletindo a capacidade de identificar corretamente casos positivos. Sua aplicação é particularmente relevante em problemas em que a ocorrência de falsos positivos é indesejável.

### 3.1.10.4 F1-score

A métrica F1-score, também conhecida por F-measure, F-score ou score F1, é uma medida importante na avaliação de modelos e algoritmos em aprendizado de máquina, especialmente em problemas de classificação. Ela combina a métrica Precisão e a métrica Sensibilidade para fornecer uma medida balanceada do desempenho do modelo.

O F1-score é calculado pela média harmônica entre a Precisão e a Sensibilidade (Equação 3.19). Ele fornece uma medida que considera tanto a capacidade do modelo em identificar corretamente os casos positivos (Sensibilidade) quanto a sua capacidade de evitar falsos positivos (Precisão). O resultado é uma métrica que busca o equilíbrio entre precisão e abrangência do modelo.

$$F1-score(\%) = 2 \times \frac{Precisao \times Sensibilidade}{Precisao + Sensibilidade} \times 100 \quad (3.19)$$

Hastie *et al.* (2009) abordam o F1-score como uma métrica amplamente utilizada na avaliação de modelos de aprendizado de máquina. Ele é particularmente útil em cenários onde tanto a precisão quanto a revocação (*Recall*) são importantes e devem ser levadas em consideração de maneira equilibrada.

Além disso, o artigo de Sokolova *et al.* (2006) é uma referência relevante, pois discute diferentes métricas de desempenho em classificação, incluindo o F1-score. O artigo apresenta vantagens

e desvantagens de várias métricas e fornece uma análise detalhada do F1-score, destacando sua utilidade na avaliação de modelos.

Em resumo, o F1-score é uma métrica valiosa para avaliar o desempenho de modelos de aprendizado de máquina, combinando Precisão e Sensibilidade em uma medida balanceada.

### 3.1.11 Raiz Quadrada do Erro Médio (RMSE)

A métrica RMSE (do inglês: *Root Mean Square Error*) é uma medida comumente utilizada para avaliar a precisão de modelos de regressão ou classificação, sendo amplamente aplicada em diversas áreas, como estatística, aprendizado de máquina e ciência de dados. O RMSE é uma medida de dispersão que quantifica a diferença média entre os valores previstos pelo modelo e os valores reais observados (BISHOP; NASRABADI, 2006).

O RMSE é calculado como a raiz quadrada do erro médio quadrático, representando a média das diferenças entre os valores previstos e os valores reais. A fórmula para o cálculo do RMSE é dada por (3.20):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (3.20)$$

Onde:

- $\hat{y}_i$  são os valores previstos pelo modelo;
- $y_i$  são os valores reais;
- $n$  é o número de amostras.

O RMSE também pode ser usado como uma métrica de otimização durante o processo de treinamento de redes neurais artificiais. Ao ajustar os parâmetros do modelo, é possível buscar a combinação que minimize o valor do RMSE, levando a um modelo mais preciso e ajustado aos dados.

Portanto, foi utilizada esta métrica especificamente para avaliar o treinamento da rede neural artificial (RNA), utilizada na etapa de pré-processamento do algoritmo proposto.

#### 3.1.11.1 Novas métricas: CCGM e CPTR

Nesta tese, foram propostas duas novas métricas que servirão de medida para avaliar a capacidade de generalização do modelo treinado e de processamento em tempo real. A primeira chamaremos de *Coefficiente da Capacidade de Generalização do Modelo* (CCGM), e será expressa pela Equação 3.23. Neste coeficiente quanto mais próximo do valor unitário, melhor será a capacidade de generalização do algoritmo treinado. Entende-se como capacidade de generalização, a capacidade do modelo em classificar corretamente amostras nunca vistas durante o treinamento.

$$CCGM = \frac{Acc(\%) \text{ no Teste}}{Acc(\%) \text{ no Treinamento}} \quad (3.21)$$

Em relação ao CCGM, podemos ainda levantar as seguintes hipóteses, Equação 3.22.

$$\left\{ \begin{array}{l} \text{Caso 1: } CCGM \leq 0,80 \\ \text{Caso 2: } 0,80 < CCGM < 0,95 \\ \text{Caso 3: } CCGM \geq 0,95 \end{array} \right. \quad (3.22)$$

- **Caso 1:** Pouca capacidade de generalização ou as amostras no conjunto de teste são mais complexas do que as amostras utilizadas no conjunto de treinamento.
- **Caso 2:** Intermediária capacidade de generalização ou as amostras do conjunto de teste têm média complexidade.
- **Caso 3:** Elevada capacidade de generalização ou as amostras no conjunto de teste são de baixa complexidade e muito semelhantes em relação as amostras usadas no treinamento.

A segunda métrica proposta foi chamada de *Coeficiente de Processamento em Tempo Real* (CPTR), e servirá de métrica para aferir a capacidade de um algoritmo processar imagens em tempo real.

$$CPTR = \frac{FPS \text{ do algoritmo}}{FPS \text{ da camera}} \quad (3.23)$$

Em relação ao CPTR, podemos levantar as seguintes situações, Equação 3.24.

$$\left\{ \begin{array}{l} \text{Caso 1: } CPTR < 1 \\ \text{Caso 2: } CPTR \geq 1 \end{array} \right. \quad (3.24)$$

- **Caso 1:** O algoritmo não consegue acompanhar o FPS da câmera, haverá perda de frames e o processamento em tempo real será prejudicado.
- **Caso 2:** O algoritmo consegue acompanhar o FPS da câmera, e portanto é capaz de realizar o processamento em tempo real, desde que garantidas as condições de estabilidade da conexão de internet e FPS nominal da câmera.

### 3.2 Sugestão de aplicação real

A abordagem proposta pode ser aplicada no mundo real usando imagens capturadas por câmeras de vigilância IP e até em drones com algumas modificações no algoritmo EdgeFireSmoke. A torre de vigilância poderá funcionar de forma estática com sua localização conhecida. O drone poderá fazer uma rota programada durante o monitoramento. As aplicações usando drones e torres têm como objetivo detectar incêndios florestais de pequena escala com até 30 metros quadrados de área.

O algoritmo EdgeFireSmoke++ proposto pode ser utilizado por uma rede de câmeras, por exemplo, interligadas a um sistema de supervisão em tempo real. A comunicação entre o dispositivo do cliente e o sistema de supervisão poderá ser feita pela rede internet. O dispositivo de monitoramento de vídeo deve possuir condições técnicas para mantê-lo em pleno funcionamento, que garanta o envio e recebimento de dados pela internet.

Além do mais, sugere-se o uso de painel solar para energizar a câmeras. Uma alternativa viável seria a utilização de câmeras IP para acesso remoto pela internet, dispensando o processamento local. A Figura 17 ilustra como seriam estas aplicações na prática.

Figura 17 – Ilustração dos sistemas de monitoramento de incêndio florestal: (a) torre de monitoramento autosuficiente com energia solar; (b) Veículo aéreo de vigilância.



Fonte: Próprio Autor.

### 3.3 Conclusão deste Capítulo

Neste capítulo, foram explorados de forma abrangente os materiais e métodos empregados nesta tese, com o objetivo de explicar o algoritmo proposto em detalhes, além de tornar clara a estratégia de avaliação seguida. Ao longo deste trabalho, foram investigadas técnicas de pré-processamento de imagem, uma arquitetura de rede neural artificial e as principais arquiteturas de redes neurais convolucionais compactas existentes na literatura, analisando sua relação com a detecção de incêndio florestal por imagem em tempo real. Por meio da avaliação de algoritmos de *Machine Learning* e *Deep Learning*, foram obtidos resultados significativos que contribuem para o conhecimento existente na área.

No próximo capítulo, será explorado os resultados obtidos, ampliando ainda mais a discussão sobre o tema. Será uma oportunidade para aprofundar a compreensão sobre os algoritmos avaliados e os seus desempenhos, além de avançar na direção de chegar a um único algoritmo que atenda a premissa de processamento em tempo real.

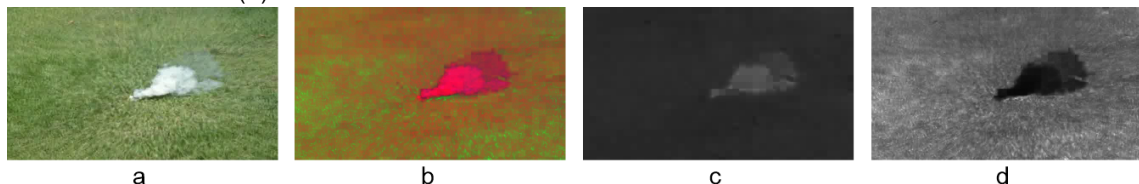
## 4 CAPÍTULO IV: RESULTADOS E DISCUSSÃO

Neste capítulo, são apresentados os resultados obtidos com o método proposto e as outras seis abordagens. Os métodos foram avaliados em um computador notebook com processador AMD Ryzen 5 3550H e 16 GB de RAM. Foi utilizada a linguagem Python 3.6, a biblioteca de aprendizagem profunda TensorFlow 2.0, e a biblioteca de visão computacional OpenCV versão 4.5.

### 4.1 Análise Qualitativa

Iniciaremos fazendo uma análise qualitativa. Primeiro, serão abordados alguns aspectos relevantes da rede neural artificial utilizada na etapa de pré-processamento. Adiante, aprofundaremos nos aspectos mais relevantes da rede neural convolucional. Neste momento, vamos iniciar apresentando um exemplo típico de princípio de incêndio em área verde com presença de fumaça, na qual temos como base a Figura 18, em que a imagem (a) encontra-se no sistema de cores RGB e (b) temos a mesma imagem convertida no sistema de cores HSV. É importante destacar que esta transformação de cores foi empregada apenas como pré-processamento antes da RNA. Sendo assim, a imagem de entrada na CNN manteve-se no sistema RGB. Em (c) e (d) temos a separação dos canais HS a partir da imagem HSV.

Figura 18 – Amostra de imagem nos sistemas de cores: (a) RGB contendo um princípio de incêndio em área verde e (b) HSV, após conversão. Em (c) temos apenas o canal H da imagem HSV anterior e (d) temos somente o canal S.



Fonte: Próprio Autor.

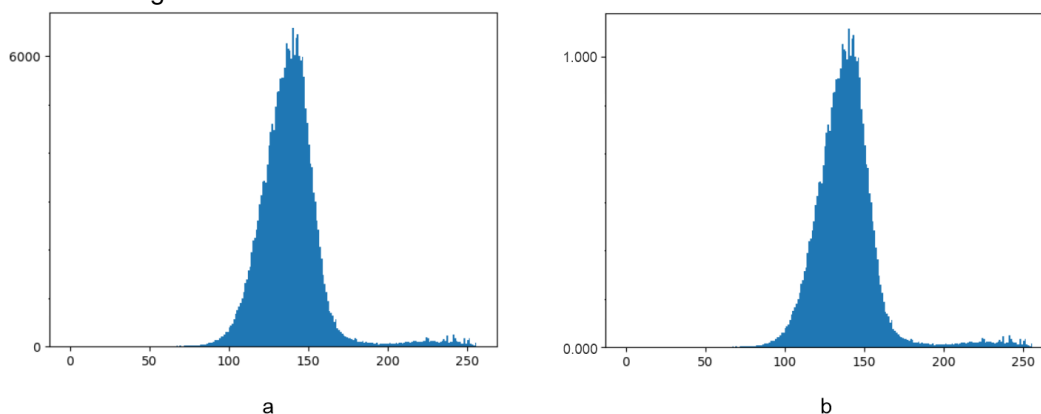
O cálculo do histograma foi explicado no capítulo anterior, no entanto, queremos mostrar agora o seu efeito prático. Os histogramas da Figura 19 (a) e (b) foram obtidos a partir do processamento dos canais HS, apresentados na Figura 18c e d. Logo, no pré-processamento utiliza-se de apenas dois canais da imagem de entrada HSV para calcular o histograma, na qual esse histograma é normalizado para posterior comparação. A normalização do histograma é muito importante para as etapas posteriores.

A comparação de histogramas é calculada a partir da medida de similaridade, que tem por base a medida da correlação entre duas imagens (histogramas). O resultado apresenta numericamente o quanto duas imagens são semelhantes entre si. Na Figura 20 apresentamos os padrões adotados como referência para a futura comparação de histogramas com a imagem de entrada. A comparação entre os padrões referência utilizados no algoritmo possibilitará a detecção de ambiente florestal.

A partir de agora, será feita uma importante análise para que seja possível compreender a funcionalidade e porque do emprego da rede neural artificial (RNA). Para isto, tomaremos por base as amostras da Figura 21, na qual faremos a comparação de histogramas obtidos a partir delas com os histogramas calculados a partir dos padrões da Figura 20. Pois bem, podemos observar na Tabela 14



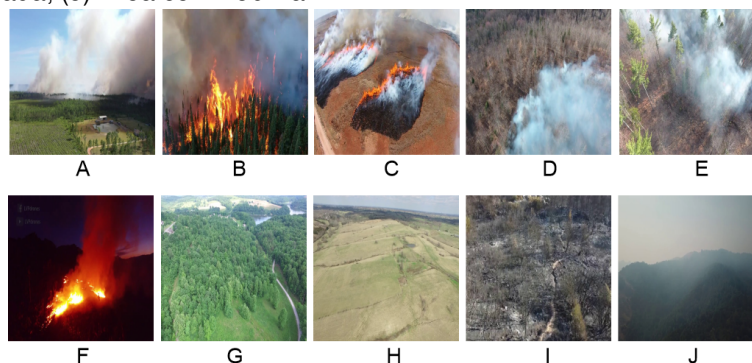
Figura 19 – Histograma obtido a partir da Figura 18c. Em (a) temos o histograma original, em (b) temos o histograma normalizado entre 0 e 1.



Fonte: Próprio Autor.

que eventualmente acontecem erros de comparação, que poderiam levar a inferências equivocadas pelo algoritmo. Por este motivo, foi inserida uma rede neural artificial (RNA) para corrigir este problema. A RNA foi treinada a partir das mesmas amostras de imagens e aprendeu a corrigir este problema, fazendo com que o algoritmo faça inferências corretas ao comparar os histogramas. Desta forma, foi possível fazer a detecção de ambiente florestal de forma rápida e sem muitos custos computacionais, deixando para a rede neural convolucional a tarefa mais computacionalmente custosa de classificar as amostras nas quatro classes já mencionadas.

Figura 20 – Padrões de referência utilizados na etapa de pré-processamento que antecede a RNA: (A-F) Incêndio florestal com presença de chama e/ou fumaça; (G-H) Área verde; (I) Área queimada; (J) Área com neblina.



Fonte: Próprio Autor.

Figura 21 – Imagens de Entrada usadas na análise apresentada nas Tabelas 14 à 17, contendo: (1) Incêndio florestal; (2) Área queimada; (3) Galpão Industrial; (4) Centro urbano.



Fonte: Próprio Autor.

Nas Tabelas 14 à 17 são apresentados quatro exemplos críticos de detecção de área

florestal e não-florestal. Em todos os exemplos foi considerado o limiar de ativação  $\Omega > 0,34$ . Este limiar pode ser ajustado na interface IHM. Esta etapa de detecção é importante, pois reduz as chances do algoritmo errar, caso seja empregado em ambientes não adequados a sua especialidade. Desta forma, foram analisados cada caso individualmente.

- **Exemplo 1** (Tabela 14): Temos uma situação em que o algoritmo recebe uma imagem de incêndio florestal e faz a comparação com os padrões de referência. A medida de correlação sugere equivocadamente que a imagem (1) é mais semelhante ao padrão (I), no entanto, a RNA percebe o erro e faz a correção. Após a correção, a imagem (1) passa a ser indicada corretamente pela função  $g(s)$  como semelhante ao padrão (A). Verifique nas Figuras 21 e 20 a semelhança entre as imagens. Em ambas temos a presença de fumaça em área verde. Como o limiar  $\Omega$  é maior que  $0,34$ , a rede infere que foi detectada uma área florestal.
- **Exemplo 2** (Tabela 15): Temos uma situação em que o algoritmo recebe uma imagem de área queimada e faz a comparação com os padrões de referência. A medida de correlação sugere corretamente que a imagem (2) é mais semelhante ao padrão (I); a RNA percebe o acerto e mantém a inferência. A imagem (2) passa a ser indicada corretamente pela função  $g(s)$  como semelhante ao padrão (I). Em ambas temos a presença de área queimada. Como o limiar  $\Omega$  é maior que  $0,34$ , a rede infere que foi detectada uma área florestal.
- **Exemplo 3** (Tabela 16): Temos uma situação em que o algoritmo recebe a imagem de um galpão industrial e faz a comparação com os padrões de referência. A medida de correlação sugere erroneamente que a imagem (3) é mais semelhante ao padrão (I); a RNA tenta corrigir o erro e infere semelhança ao padrão (J). A imagem (3) passa a ser indicada equivocadamente pela função  $g(s)$  como semelhante ao padrão (J). Como o limiar  $\Omega$  é menor que  $0,34$ , a rede infere corretamente que trata-se de uma área não-florestal.
- **Exemplo 4** (Tabela 17): Temos uma situação em que o algoritmo recebe a imagem de uma área urbana e faz a comparação com os padrões de referência. A medida de correlação sugere erroneamente que a imagem (4) é mais semelhante ao padrão (I); a RNA tenta corrigir o erro e mantém semelhança ao padrão (I). A imagem (4) passa a ser indicada equivocadamente pela função  $g(s)$  como semelhante ao padrão (I). Como o limiar  $\Omega$  é menor que  $0,34$ , a rede infere corretamente que trata-se de uma área não-florestal.

Tabela 14 – Exemplo 1: comparação de histograma entre a Figura 21-(1) e todos os padrões da Figura 20, de A à J, comparados um a um individualmente.

Imagem de Entrada	Padrão de Referência	Similaridade ( $d(H1, H2)$ )	Saída da RNA ( $\Omega$ )	Argmax ( $g(s)$ )
1	A	0,2410	<b>0,3698</b>	1
1	B	0,0903	-0,018	0
1	C	0,1771	0,0051	0
1	D	0,2601	0,0163	0
1	E	0,2200	0,2038	0
1	F	-0,0413	0,0204	0
1	G	0,0682	0,0662	0
1	H	0,0104	-0,0172	0
1	I	<b>0,5142</b>	0,3002	0
1	J	0,1251	0,0301	0

Fonte: Próprio Autor.

Tabela 15 – Exemplo 2: comparação de histograma entre a Figura 21-(2) e todos os padrões da Figura 20, de A à J, comparados um a um individualmente.

Imagem de Entrada	Padrão de Referência	Similaridade ( $d(H1, H2)$ )	Saída da RNA ( $\Omega$ )	Argmax ( $g(s)$ )
2	A	0,3046	0,0740	0
2	B	0,2300	0,1569	0
2	C	0,2209	0,0568	0
2	D	0,1976	0,3354	0
2	E	0,2798	0,0354	0
2	F	-0,0540	-0,0343	0
2	G	0,0079	0,1297	0
2	H	0,0687	0,0589	0
2	I	<b>0,5756</b>	<b>0,3560</b>	1
2	J	0,1958	-0,1458	0

Fonte: Próprio Autor.

Tabela 16 – Exemplo 3: comparação de histograma entre a Figura 21-(3) e todos os padrões da Figura 20, de A à J, comparados um a um individualmente.

Imagem de Entrada	Padrão de Referência	Similaridade ( $d(H1, H2)$ )	Saída da RNA ( $\Omega$ )	Argmax ( $g(s)$ )
3	A	0,1659	0,1203	0
3	B	0,1815	-0,0155	0
3	C	0,1144	0,0715	0
3	D	0,0857	0,1102	0
3	E	0,0903	0,1214	0
3	F	-0,0386	0,0079	0
3	G	-0,0383	0,1102	0
3	H	0,0064	0,0808	0
3	I	<b>0,2316</b>	0,0191	0
3	J	0,0526	<b>0,3287</b>	1

Fonte: Próprio Autor.

Tabela 17 – Exemplo 4: comparação de histograma entre a Figura 21-(4) e todos os padrões da Figura 20, de A à J, comparados um a um individualmente.

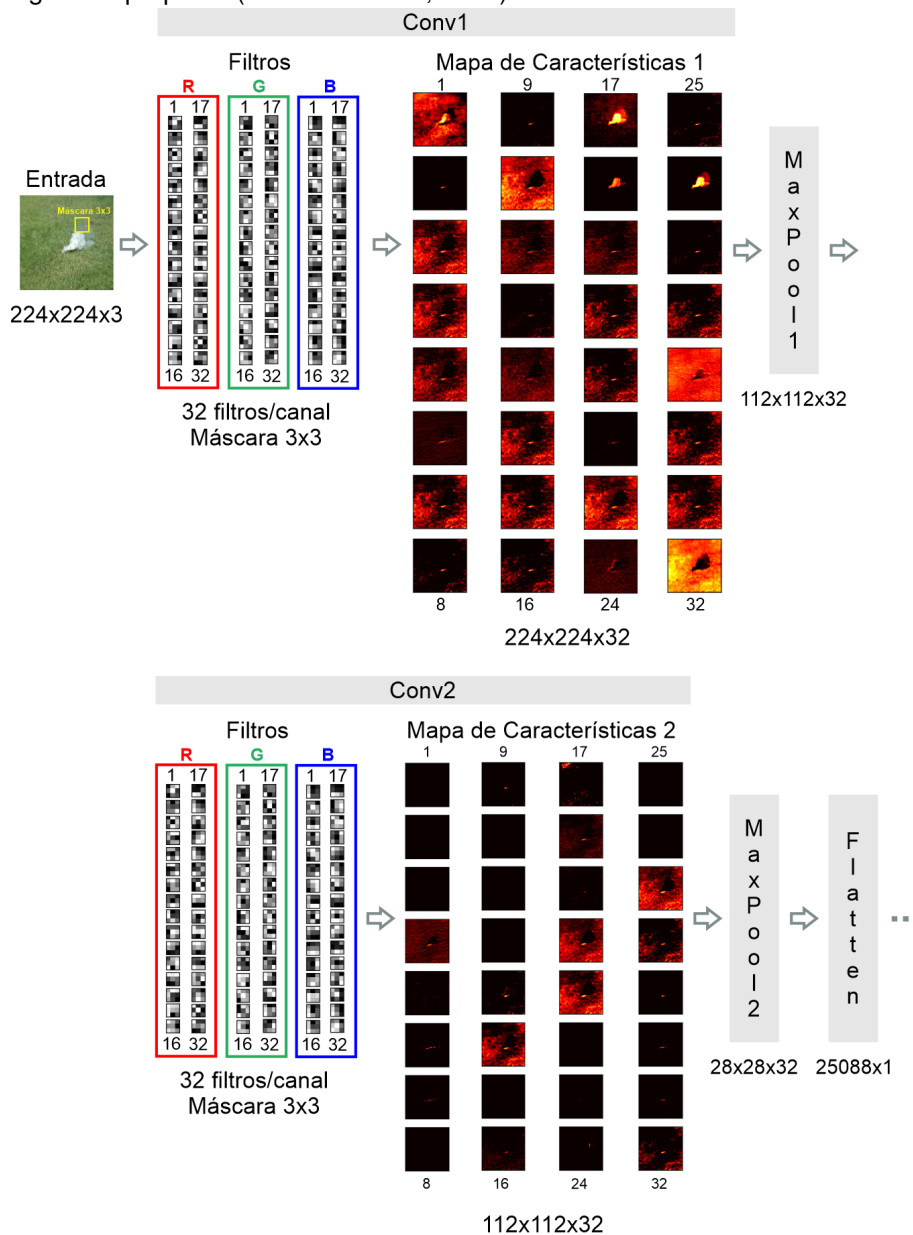
Imagem de Entrada	Padrão de Referência	Similaridade ( $d(H1, H2)$ )	Saída da RNA ( $\Omega$ )	Argmax ( $g(s)$ )
4	A	0,2558	-0,0174	0
4	B	0,1945	0,0807	0
4	C	0,1634	0,0172	0
4	D	0,2219	0,2274	0
4	E	0,2553	0,1598	0
4	F	-0,0521	0,0086	0
4	G	-0,0125	0,0931	0
4	H	0,0424	0,0885	0
4	I	<b>0,4651</b>	<b>0,2991</b>	1
4	J	0,2374	0,0701	0

Fonte: Próprio Autor.

Adiante, será visto o comportamento da CNN diante da imagem de entrada exemplificada na Figura 18a. Por isso, apresentamos na Figura 22 os filtros e os mapas de características obtidos em cada camada convolucional após o treinamento da CNN (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023). Para esta análise, considere a imagem de entrada apresentada na Figura 18a.

Analisando os mapas de características na primeira camada convolucional, percebe-se que alguns mapas tornaram-se especialistas em identificar dois tipos de situação: (1) segmentar apenas a fumaça presente no ambiente; (2) visualizar o ambiente sem a presença de fumaça. Estas constatações são mostradas na Figura 23. Para facilitar a visualização, as imagens foram convertidas de tom de cinza

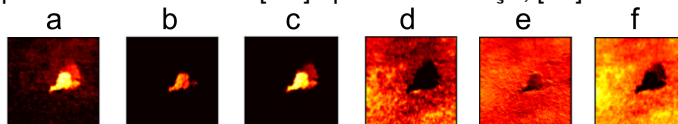
Figura 22 – Filtros e os Mapas de Características da EdgeFireSmoke (SILVA *et al.*, 2022) integrante do algoritmo proposto (ALMEIDA *et al.*, 2023).



Fonte: Próprio Autor.

para coloridas artificialmente em tons de vermelho.

Figura 23 – Mapas de Características da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023) com dois padrões observados: [a-c] apenas a fumaça; [d-f] ambiente sem a fumaça.



Fonte: Próprio Autor.

A visualização destes mapas nos leva a pensar sobre qual a importância dos filtros na rede neural convolucional e até a cogitar avaliar em trabalhos futuros qual seria o desempenho de uma nova CNN com apenas os filtros que produziram estes mapas mais significativos. No entanto, consideramos a

hipótese de todos os filtros terem sua importância, pois todos os filtros focam em características diferentes da imagem de entrada. De certo, durante o processo de treinamento as CNNs tem a capacidade de sintonizar automaticamente os seus filtros (pesos) dos seus neurônios convolucionais, sendo esta uma tarefa árdua para um programador de visão computacional, caso o fizesse manualmente ou por meio de experimentos computacionais. Nas Tabelas 18 à 23 são apresentadas as máscaras dos filtros [a, b, c, d, e, f], respectivamente, da Figura 23, no qual suas máscaras de tamanho  $3 \times 3$  estão separadas por canal da imagem RGB na primeira camada convolucional.

Tabela 18 – Máscaras do Filtro 17, Conv1.

R			G			B		
-0,19706145	-0,29275605	-0,013975053	-0,22577934	-0,15049036	0,10401013	-0,18742594	-0,28052023	-0,13265331
-0,18178317	-0,23017173	0,28017336	-0,2697188	-0,014552558	0,2112414	0,046835516	-0,28751525	0,13629441
-0,016065637	-0,25725985	0,18348473	-0,25263247	-0,06927824	0,28875777	-0,05199305	-0,33596817	0,18736503

Fonte: Próprio Autor.

Tabela 19 – Máscaras do Filtro 18, Conv1.

R			G			B		
0,3681603	-0,31606558	-0,42258766	0,3183993	-0,2211113	-0,11954478	0,15491824	-0,22545183	-0,08023499
0,18327135	-0,04338565	-0,43966267	0,16675232	-0,16996817	-0,25643232	0,30789658	-0,029909719	-0,105722055
0,20483045	0,005184866	-0,41448134	0,0029881033	0,041580044	-0,19268131	0,29223442	0,006386406	-0,027296418

Fonte: Próprio Autor.

Tabela 20 – Máscaras do Filtro 26, Conv1.

R			G			B		
0,052102007	-0,09728109	-0,18845363	-0,047694545	-0,14213324	-0,021861834	-0,009184317	0,05591397	-0,13219489
-0,113511115	-0,0077544027	-0,2003859	0,10053824	-0,12511651	-0,0026167517	-0,10785015	-0,16717319	-0,06304265
-0,03110659	-0,0040260497	0,06658879	-0,06282737	-0,059601255	-0,113837555	-0,14292964	0,062485464	0,05679968

Fonte: Próprio Autor.

Tabela 21 – Máscaras do Filtro 10, Conv1.

R			G			B		
-0,15930359	0,019446664	-0,29963154	-0,14985402	-0,09145833	-0,20954835	-0,03693815	0,08755877	-0,14840683
-0,23512295	-0,30327466	-0,23970254	-0,050986923	-0,2345303	-0,091962874	-0,118553564	-0,30574292	-0,23349172
0,03602149	0,089418456	0,19764528	0,11167918	0,30130506	0,08428877	0,15919934	0,33640176	0,27886915

Fonte: Próprio Autor.

Tabela 22 – Máscaras do Filtro 29, Conv1.

R			G			B		
0,112061106	-0,023221925	0,107743144	-0,2667928	-0,19859824	-0,28677347	-0,18927903	-0,052048165	-0,17860031
0,03275045	-0,14253859	0,11149771	-0,19455886	-0,11818829	-0,29409903	-0,06533899	-0,07952016	-0,080919296
0,14283308	0,11396691	-0,056608945	-0,2658596	-0,17088786	-0,17393234	-0,063261464	-0,2151378	0,047260985

Fonte: Próprio Autor.

Tabela 23 – Máscaras do Filtro 32, Conv1.

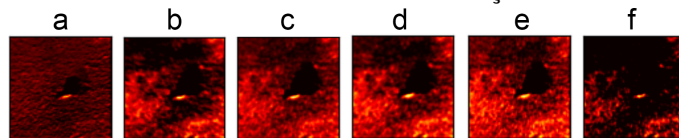
R			G			B		
-0,1360836	-0,24499288	-0,41610193	0,27227795	0,0069033224	0,040927917	0,017845625	-0,120379955	-0,13497214
-0,2778841	-0,2771752	-0,10574267	0,19979861	0,21430625	0,23907658	-0,13438776	-0,017738882	0,102680236
-0,23811087	-0,09473016	-0,17431968	0,33603072	0,32132456	0,1859462	-0,22555248	-0,105082005	-0,06461792

Fonte: Próprio Autor.

Porém, na segunda camada convolucional, pode-se observar um nível de abstração mais restritivo nos mapas de características. Nesta segunda camada, a CNN ajustou seus pesos para

identificar apenas a situação (2), descrita anteriormente, focando em ver o ambiente através de um recorte da região sem fumaça. Desta forma, pode-se observar que houve uma supressão da situação (1), condição esta ajustada pelos neurônios convolucionais nesta camada e mostrada na Figura 24. Nas Tabelas 24 à 29 são apresentadas as máscaras dos filtros [a, b, c, d, e, f], respectivamente, da Figura 24, no qual suas máscaras de tamanho  $3 \times 3$  estão separadas por canal da imagem RGB na segunda camada convolucional.

Figura 24 – Mapas de características da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023) na segunda camada convolucional. Os mapas [a-f] têm tamanho  $112 \times 112$  pixels e enxergam características do ambiente com ausência de fumaça.



Fonte: Próprio Autor.

Tabela 24 – Máscaras do Filtro 4, Conv2.

R			G			B		
-0,037274998	0,02607066	0,021273755	0,107412025	0,044788945	-0,12878387	-0,0164819	-0,11582065	0,047769073
-0,0144147	-0,08326288	-0,11819686	0,096847326	-0,02705968	-0,04474094	-0,044955604	-0,061164387	0,017928965
-0,082650386	-0,024333844	-0,01690306	0,032553643	0,021478578	0,06788591	0,06464106	-0,09676473	-0,030570826

Fonte: Próprio Autor.

Tabela 25 – Máscaras do Filtro 14, Conv2.

R			G			B		
-0,004799784	0,08423971	0,0637304	0,019336289	-0,05181738	-0,08369966	-0,44562382	-0,6298378	-0,47156617
0,04103046	-0,11501559	-0,13345048	-0,069433466	0,04041407	-0,03645189	-0,04272205	-0,16365473	-0,120181896
-0,0011330795	-0,10332922	-0,17302239	-0,18542834	-0,021520507	-0,029350901	-0,054878335	-0,20260043	-0,21687108

Fonte: Próprio Autor.

Tabela 26 – Máscaras do Filtro 20, Conv2.

R			G			B		
-0,05129799	0,030442964	0,09055483	0,15360528	0,21177307	0,137946	-0,21986881	-0,34464675	0,1312916
-0,008922467	0,10239477	0,17338772	0,20151563	0,19062991	0,12698765	-0,28586653	-0,44682828	-0,07309106
0,048745126	0,18032202	0,14827293	0,28787547	0,31372	0,18344125	-0,05652171	-0,32772177	-0,07979322

Fonte: Próprio Autor.

Tabela 27 – Máscaras do Filtro 21, Conv2.

R			G			B		
0,03469955	-0,20680514	-0,25151145	0,00052201445	0,16650069	0,1374679	0,6595647	0,29057747	-0,14245376
0,19991125	-0,079496786	-0,19380413	0,06633175	0,15306725	0,42092904	0,21707849	-0,2910533	-0,5214321
0,16852556	0,045528315	-0,0502483	0,0054058847	0,0767707	0,27013522	-0,32524467	-0,6135264	-0,8857493

Fonte: Próprio Autor.

Tabela 28 – Máscaras do Filtro 27, Conv2.

R			G			B		
0,13035917	0,1432867	0,06328632	0,12273763	-0,12922873	0,11651615	-0,1435501	-0,13861006	-0,04300414
0,022505691	0,06166115	0,007479702	0,030729156	-0,10921582	0,23047483	-0,1613766	-0,13463423	-0,14506698
-0,04667002	-0,15059629	-0,098718666	0,023129424	-0,016560702	0,147545	-0,032648616	0,03946992	0,0062560127

Fonte: Próprio Autor.

Tabela 29 – Máscaras do Filtro 28, Conv2.

R			G			B		
-0,14468165	-0,15244466	-0,10186379	0,025669325	0,1809738	0,1461692	-0,043747526	-0,038670324	0,037409175
-0,1558191	-0,18380213	-0,21315801	0,24084778	0,1113707	0,044787668	-0,0758196	-0,16038176	-0,12994722
-0,01996193	-0,064868495	-0,060474258	0,19714718	0,14986254	0,34944955	0,00747976	-0,05895226	-0,027043186

Fonte: Próprio Autor.

Uma outra importante análise consiste na visualização dos mapas de características gerados a partir das imagens de entrada. Para esta análise, considerou-se o algoritmo proposto e as quatro classes de imagens abordadas nesta pesquisa: (1) IAQ, (2) ICFF, (3) ICN e (4) IAV. Ao todo, avaliamos o algoritmo em 52 situações distintas de condições reais.

Iniciaremos apresentando as 13 amostras da classe ICFF na Figura 25, na qual contém imagens de incêndio florestal com presença de fogo e fumaça. Para facilitar a identificação, as imagens foram organizadas em formato matricial, em que as linhas são numeradas de 1 à 13 e as colunas nomeadas de a à h. Por exemplo, ao nos referir ao mapa da linha 1, coluna a, foi usada a notação (1a). As colunas a, b, c e d contém os mapas produzidos pela primeira camada convolucional, enquanto nas colunas e, f, g e h são mostrados os mapas gerados pela segunda camada convolucional.

Foram escolhidos alguns exemplos para aprofundar a análise. Observe a Figura 25(1a); em seguida perceba os mapas de características (a) à (h); veja que os quatro primeiros mapas nessa linha especializaram-se em identificar a região com fumaça, enquanto os quatro últimos mapas segmentam a região do solo e vegetação. Um exemplo muito claro é apresentado na linha 10, colunas a, b e c, na qual pode-se ver a segmentação da fumaça, enquanto nos outros mapas desta mesma linha pode-se ver a segmentação de outras características, como o solo e a vegetação. Observa-se este comportamento da rede neural convolucional em todas as imagens de entrada. Diante do exposto, foi possível visualizar como o algoritmo consegue ver o mundo a sua volta para que seja possível fazer a detecção de incêndios florestais por imagem.

Um segundo grupo de exemplos é apresentado na Figura 26. Neste caso, é possível observar como o algoritmo extrai as características do ambiente relacionadas a área queimada, rotulada como classe IAQ. Veja no mapa (1c) que a CNN faz a segmentação da região de solo escuro a partir da imagem de entrada, além de reconhecer o solo arenoso de cor clara no mapa 1a. Observe que em todas as situações apresentadas, a CNN tem a capacidade de identificar o solo escuro, típico de regiões que sofreram queimadas recentemente. No mapa 4a a rede identifica características do solo queimado e no mapa 4b consegue separar a região de céu azul claro. Eventualmente, consegue também identificar as bordas dos objetos no solo, como por exemplo no mapa 4g. Além de identificar a região queimada, os filtros permitem segmentar a vegetação baixa no solo, como no exemplo 9b.

A terceira análise dos mapas de características, apresentada na Figura 27, foca na classificação de imagens com neblina da classe ICN. Este tipo de imagem é conhecido no meio acadêmico como imagens *Hazy* e faz parte de um outro campo de estudo. Novamente a CNN apresentou dois comportamentos básicos. Nas colunas a, b, c e d, a rede identifica a região com neblina, e nas demais colunas segmenta apenas a região de solo e vegetação. Veja um exemplo muito interessante na linha 9, colunas a-h, em que a CNN produziu filtros capazes de identificar o céu, a região com neblina presente em toda a imagem, e por último conseguiu segmentar o solo e a vegetação. Situação semelhante



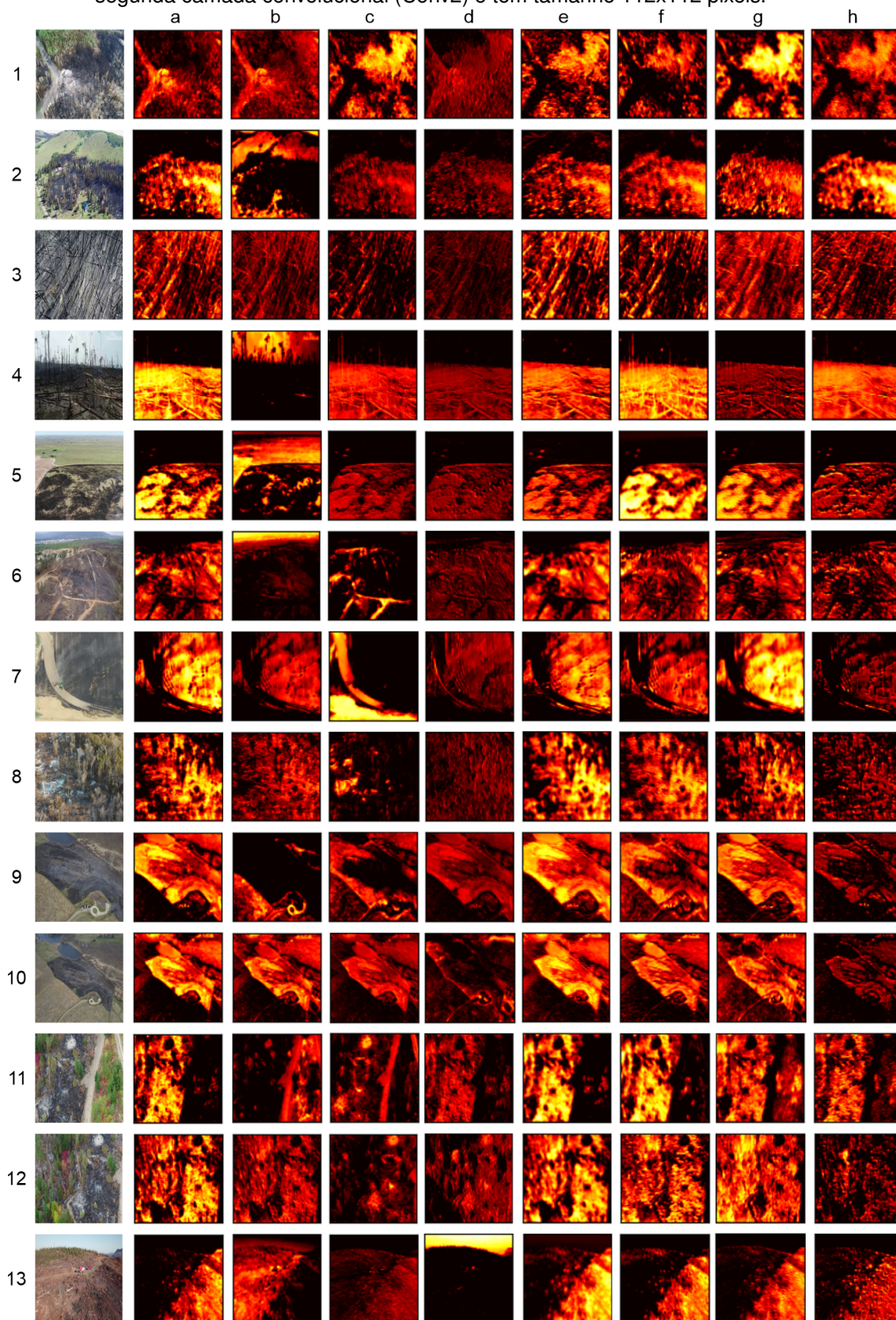
Figura 25 – Amostras da classe ICFF [1-13] e os oito principais mapas de características [a-h] da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023). Os mapas [a-d] são da primeira camada convolucional (Conv1) e têm tamanho 224x224 pixels. Os mapas [e-h] são da segunda camada convolucional (Conv2) e têm tamanho 112x112 pixels.



Fonte: Próprio Autor.



Figura 26 – Amostras da classe IAQ [1-13] e os oito principais mapas de características [a-h] da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023). Os mapas [a-d] são da primeira camada convolucional (Conv1) e têm tamanho 224x224 pixels. Os mapas [e-h] são da segunda camada convolucional (Conv2) e têm tamanho 112x112 pixels.



Fonte: Próprio Autor.

aconteceu nas linhas 3, 5 e 13.

Na última análise dos mapas de características, na Figura 28, destacamos a classe IAV. Foram gerados os mapas de 13 imagens distintas em área verde. O resultado são os principais mapas que nos mostram como a rede visualiza o solo preservado, com ausência de fumaça, área queimada ou neblina. As amostras contém áreas florestais verdes e com vegetação seca. Observe, por exemplo, na linha 9, colunas e-h, que a CNN consegue segmentar a copa das árvores, enquanto no mapa 9d, consegue ver o oposto. Situação semelhante ocorreu nas linhas 2, 4, 5 e 6. O conjunto de todas estas características contribui para a correta classificação dessas imagens. Um ponto que poderia ter sido avaliado seria a distinção entre área verde e área de vegetação seca, de forma a adicionar uma nova classe. No entanto, esta avaliação poderá ser feita em trabalhos futuros.

Por fim, a análise dos mapas de características foi muito relevante, pois nos permitiu compreender que os filtros da CNN foram sintonizados corretamente durante o treinamento para reconhecer as características peculiares do problema abordado. Além do mais, observou-se que dos 32 mapas em cada camada convolucional muitos se repetiram ou apresentaram muita semelhança entre si. Por este motivo, foram apresentados nesta análise apenas os oito principais mapas presentes nas duas camadas de convolução. Todas as características presentes nestes mapas são reunidas na camada de vetorização, chamada *Flatten*. Importante destacar que em todas as situações apresentadas a CNN obteve desempenho suficiente para garantir a capacidade de generalização do modelo, no entanto, além deste, outros aspectos serão discutidos mais adiante.

## 4.2 Análise Quantitativa

A partir de agora, serão comentados os dados numéricos e quantitativos. A Tabela 30 mostra o resultado obtido durante o treinamento da RNA. O treinamento da RNA foi realizado com 500 épocas de aprendizagem utilizando a configuração apresentada com 10 neurônios na camada de entrada, 128 neurônios ocultos e 10 neurônios na camada de saída. Após o treinamento registrou-se um RMSE de 0,28, no qual foi possível obter um modelo de pré-processamento, baseado em uma RNA, preciso e ajustado aos dados.

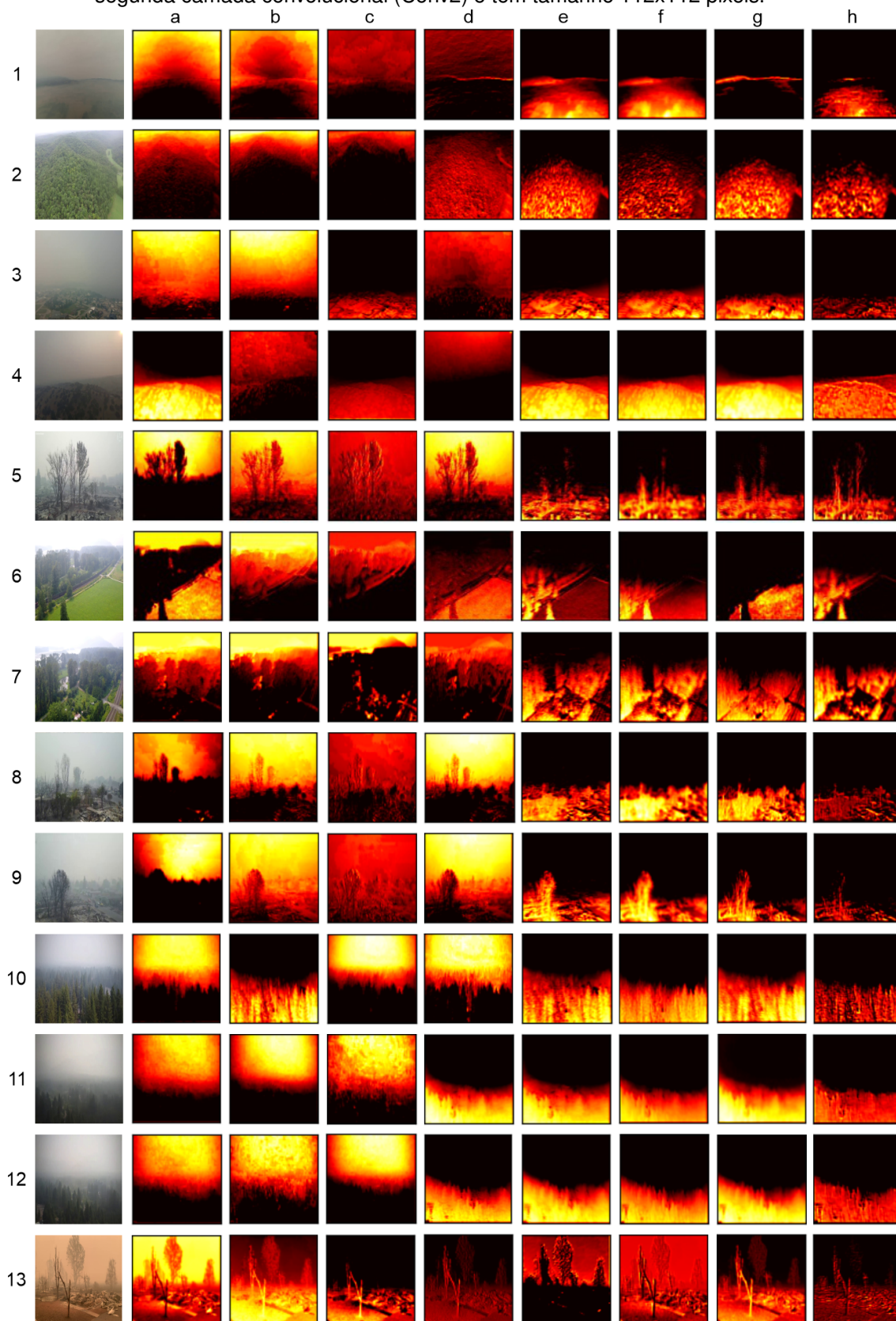
Tabela 30 – Parâmetros e resultados utilizados no treinamento da RNA.

Imagens	Modelo	Arquitetura	Épocas	Taxa de aprendizagem	RMSE
Dataset 1	RNA	Preprocessamento + RNA: 10/128/10	500	0,01	0,28

Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Na Tabela 31 temos o resultado obtido após o treinamento das CNNs. São apresentadas algumas informações relevantes dos modelos avaliados. Observa-se que o modelo EdgeFireSmoke++ tem a mais compacta arquitetura, com apenas 7MB de armazenamento e profundidade de apenas 10 camadas. Também é o que possui a menor quantidade de parâmetros, com um total de 1,62 milhões. Durante o treinamento o modelo EdgeFireSmoke++ registrou no treinamento uma acurácia de 98,25%, e no conjunto de validação uma acurácia de 95,45%. Este foi o melhor resultado obtido entre os outros

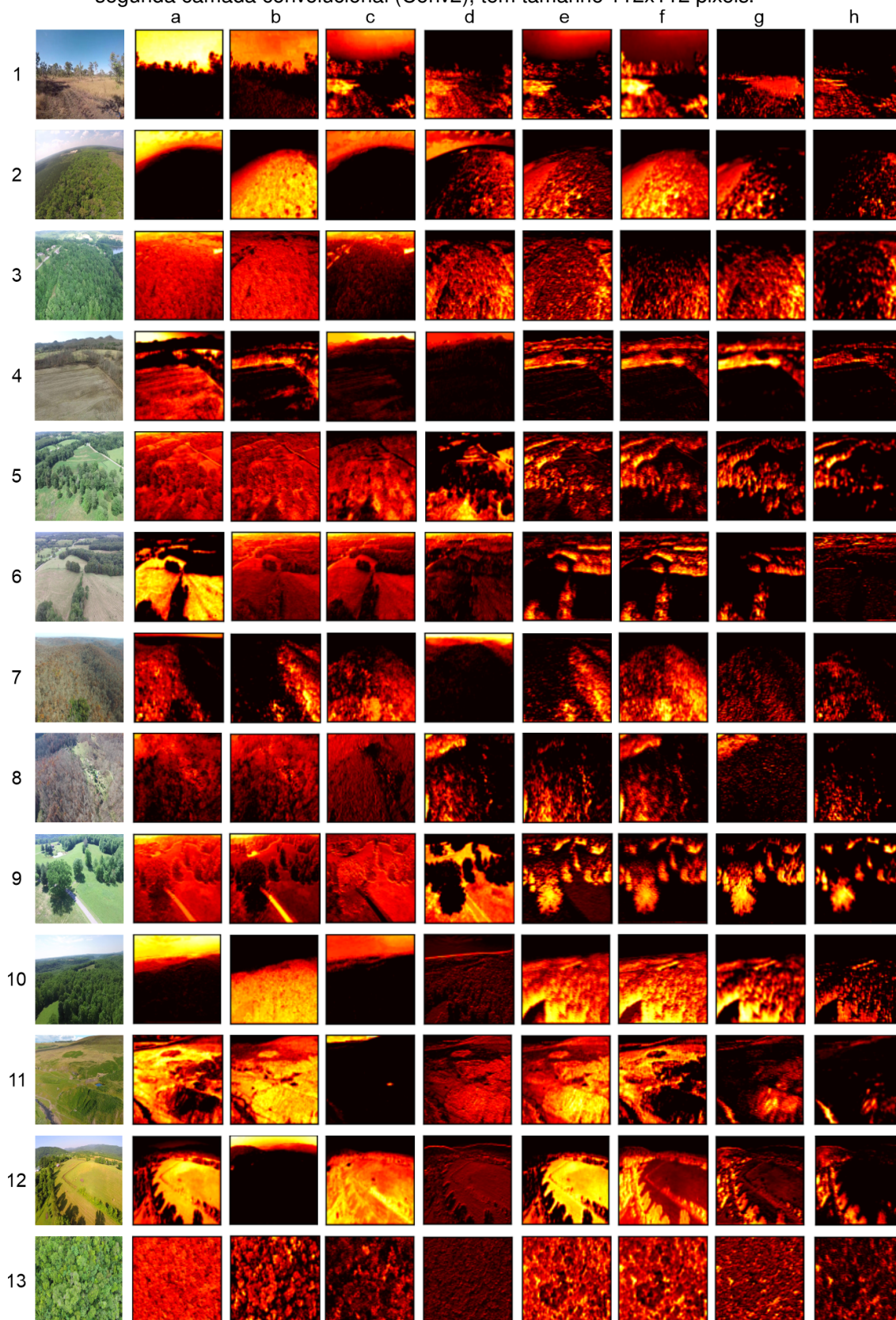
Figura 27 – Amostras da classe ICN [1-13] e os oito principais mapas de características [a-h] da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023). Os mapas [a-d] são da primeira camada convolucional (Conv1) e têm tamanho 224x224 pixels. Os mapas [e-h] são da segunda camada convolucional (Conv2) e têm tamanho 112x112 pixels.



Fonte: Próprio Autor.



Figura 28 – Amostras da classe IAV [1-13] e os oito principais mapas de características [a-h] da EdgeFireSmoke (SILVA *et al.*, 2022; ALMEIDA *et al.*, 2023). Os mapas [a-d] são da primeira camada convolucional (Conv1) e têm tamanho 224x224 pixels. Os mapas [e-h] são da segunda camada convolucional (Conv2), têm tamanho 112x112 pixels.



Fonte: Próprio Autor.

modelos avaliados.

Tabela 31 – Resultado do treinamento de modelos de aprendizagem profunda usando 25 épocas.

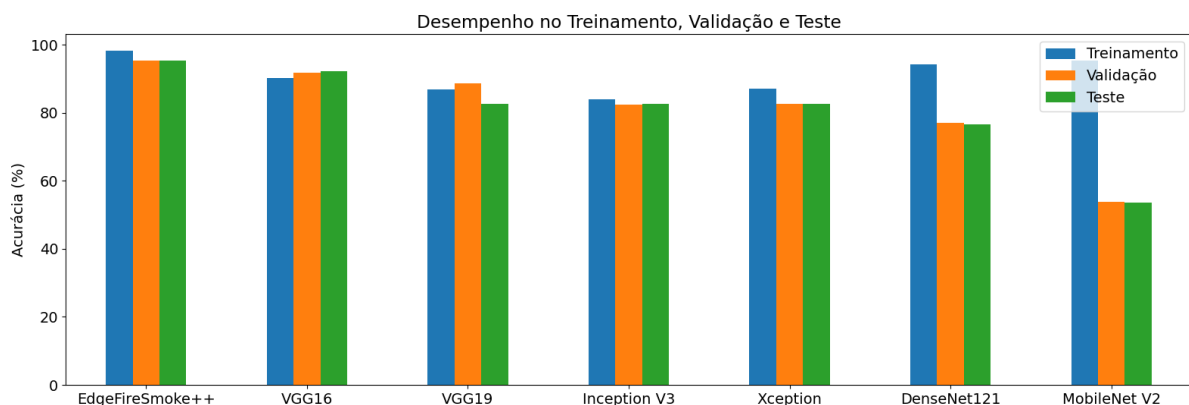
Imagens	Modelo CNN	Tam.(MB)	Camadas	Parâmetros			Acurácia (%)	
				Treinável	Não-treinável	Total	Treino	Validação
Dataset 2	EdgeFireSmoke++	7	10	1,62M	0,00M	1,62M	98,25	95,42
	VGG16	528	16	18,89M	14,71M	33,61M	90,29	91,88
	MobileNet V2	14	105	1,28M	2,25M	3,54M	95,25	53,84
	Inception V3	92	189	25,19M	21,80M	46,99M	84,03	82,36
	Xception	88	81	25,19M	20,86M	46,05M	87,13	82,53
	DenseNet121	33	242	20,99M	7,04M	28,03M	94,30	76,96
	VGG19	549	19	18,90M	20,02M	38,92M	86,87	88,73

Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

O segundo melhor resultado no treinamento foi obtido utilizando o modelo MobileNet V2, com Acurácia de 95,25%. Porém na validação registrou desempenho abaixo do esperado. Em termos de arquitetura, o MobileNet V2 é o segundo mais compacto, com 14MB de armazenamento e 3,54 milhões de parâmetros treináveis. O conjunto de validação serve como uma prévia do conjunto de teste, por isso, o desempenho obtido na validação refletirá no resultado obtido no teste.

Outros modelos de CNN também apresentaram resultados relativamente bons no treinamento e validação, ficando com o menor desempenho o modelo Inception V3, que obteve uma acurácia de 84,03%. Em todos os modelos CNN utilizamos 25 épocas de treinamento. Como esperado, modelos mais compactos conseguem aprender com algumas dezenas de épocas, enquanto modelos mais extensos necessitam de maior número de épocas. É importante destacar que o nosso principal objetivo foi avaliar qual modelo CNN permitirá um processamento em tempo real, juntamente com as outras etapas de processamento envolvidas. Por isso, independente da quantidade de épocas de treinamento, espera-se que o tempo de processamento seja muito próximo. Na Figura 29 temos um resumo que mostra o desempenho em acurácia obtido pelos modelos durante o treinamento, validação e teste.

Figura 29 – Desempenho em acurácia obtido nos experimentos de treinamento, validação e teste.



Fonte: Próprio Autor.

Percebe-se que o algoritmo proposto apresentou o melhor resultado, no qual registrou uma acurácia de 98,25% no treinamento, uma leve redução de 2,83% em acurácia na validação, mantida com 95,41% de acurácia no teste. Podemos ver que os modelos DenseNet121 e MobileNetV2 tiveram perda significativa na capacidade de generalização, quando comparadas as acurácias no treinamento,

validação e teste. Nos demais modelos a acurácia no teste manteve-se próxima do valor obtido no treinamento, com variações para mais ou menos.

Na Tabela 32 foram organizados os resultados obtidos durante as etapas de treinamento, validação e teste dos modelos CNN. Para esta análise, criamos o coeficiente da capacidade de generalização do modelo (CCGM). Este coeficiente foi calculado a partir da razão entre a acurácia no teste e a acurácia no treinamento, em termos percentuais. Além disso, fizemos esta análise considerando três casos: (1) pouca capacidade de generalização; (2) intermediária capacidade de generalização; (3) Elevada capacidade de generalização. Desta forma, o coeficiente CCGM nos mostrou que os modelos *EdgeFireSmoke++*, *VGG16*, *InceptionV3*, *Xception*, e *VGG19*, apresentaram elevada capacidade de generalização, no qual mantiveram acurácia no teste muito semelhante a obtida no treinamento.

Tabela 32 – Resultado após análise do coeficiente CCGM.

Imagens	Modelo CNN	Acurácia (%)			CCGM	Análise do coeficiente CCGM
		Treinamento	Validação	Teste		
Dataset 2	EdgeFireSmoke++	98,25	95,42	95,41	0,97	Caso 3
	VGG16	90,29	91,88	92,15	1,02	Caso 3
	MobileNet V2	95,25	53,84	53,63	0,56	Caso 1
	Inception V3	84,03	82,36	82,65	0,98	Caso 3
	Xception	87,13	82,53	82,60	0,95	Caso 3
	DenseNet121	94,30	76,96	76,62	0,81	Caso 2
	VGG19	86,87	88,73	82,70	0,95	Caso 3

Fonte: Próprio Autor.

A principal vantagem da utilização do CCGM foi identificar, a partir de dados numéricos, quais os algoritmos que conseguiram aprender com as amostras apresentadas, seguindo uma mesma estratégia de treinamento. Nosso intuito não foi dizer que o algoritmo A é melhor que B, mas apenas mostrar que diferentes arquiteturas de CNN podem necessitar de estratégias mais elaboradas para o seu treinamento, como por exemplo, maior número de épocas, além do uso de outras técnicas, como o ajuste fino (*fine tuning*) e uso de outras funções de otimização. No entanto, todas as análises feitas nesta tese conduzem a busca de algoritmos leves, em termos computacionais, que tenham capacidade de processamento de vídeos em tempo real na tarefa de detecção de incêndios florestais.

Na Tabela 33 temos o resultado obtido com as imagens de teste no conjunto de dados 1. Os dados de teste foram particionados em 10 subconjuntos com igual quantidade. O modelo *EdgeFireSmoke++* obteve o melhor desempenho com as imagens de teste. Registrou uma Acurácia de 95,41%, e manteve equilíbrio entre Precisão, Recall e F1-Score. No teste, o segundo melhor modelo de CNN com ótimo desempenho no conjunto de dados 2 foi o *VGG16*, com 92,15% de acurácia e também equilibrado nas outras métricas. Apesar de compacto, característica interessante para processamento em tempo real, o modelo *MobileNet V2* registrou o pior desempenho no teste, com 53,63% de acurácia, 70,59% de Precisão, e um pouco acima dos 50% em Recall e F1-Score.

Na Figura 30 temos um resumo gráfico das principais métricas, no qual podemos destacar os três melhores desempenho obtidos pelos algoritmos: *EdgeFireSmoke++*, *VGG16* e *VGG19*; sendo o pior desempenho obtido pelo *MobileNetV2*.

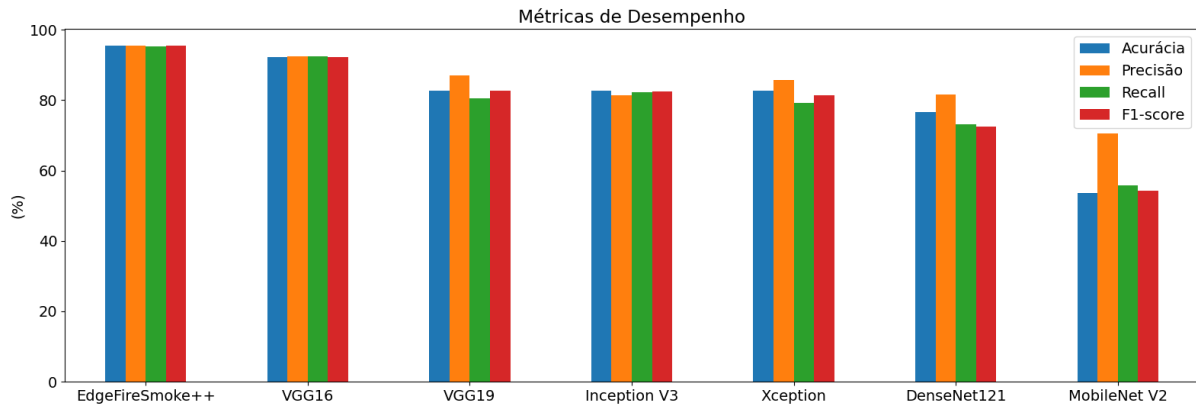
Um experimento relevante neste trabalho foi o teste de processamento em tempo real.

Tabela 33 – Resultado de desempenho no teste com 10 iterações.

Imagens	Modelo CNN	Teste			
		Acurácia (%)	Precisão (%)	Recall (%)	F1-Score (%)
Dataset 2	EdgeFireSmoke++	95,41+/-0,24	95,49+/-0,28	95,38+/-0,23	95,41+/-0,24
	VGG16	92,15+/-0,82	92,37+/-0,96	92,51+/-0,76	92,13+/-0,82
	MobileNet V2	53,63+/-1,65	70,59+/-1,67	55,79+/-1,76	54,20+/-1,83
	Inception V3	82,65+/-1,13	81,34+/-1,14	82,19+/-1,18	82,41+/-1,14
	Xception	82,60+/-1,04	85,76+/-1,16	79,13+/-1,27	81,49+/-1,13
	DenseNet121	76,62+/-0,82	81,53+/-1,16	73,03+/-0,95	72,51+/-1,26
	VGG19	82,70+/-1,46	87,13+/-1,13	80,52+/-1,75	82,70+/-1,52

Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Figura 30 – Desempenho obtido no teste com a CNN proposta e as variações de CNNs avaliadas.



Fonte: Próprio Autor.

Neste experimento, utilizamos as duas câmeras apresentadas na Tabela 3 e avaliamos o desempenho da RNA proposta juntamente com as CNNs. Os resultados alcançados são mostrados na Tabela 34.

Analisando a Tabela 34, percebe-se novamente que o melhor desempenho foi obtido com a RNA proposta, combinada com a CNN proposta originalmente por (SILVA *et al.*, 2022). Esta combinação, a qual nomeamos de EdgeFireSmoke++, por conter uma etapa extra de pré-processamento, obteve uma taxa de 33 frames por segundo (FPS) com a câmera IP de resolução HD, 1280x720 pixels. Processando as imagens em resolução VGA a partir da webcam, a EdgeFireSmoke++ obteve uma taxa de 40 FPS. O modelo proposto neste trabalho necessita de apenas 540MB de memória RAM e é capaz de funcionar com câmeras IP de até 30 FPS. É possível observar que com os outros modelos avaliados torna-se inviável o processamento em tempo real, pois não conseguem acompanhar o FPS da câmera.

Tabela 34 – Resultado do teste em tempo real, em milissegundos.

Imagens	Préprocessamento	Modelo CNN	FPS	RAM (GB)	Tempo (ms)			
					Média	Mediana	Máximo	Mínimo
Cam 1	RNA	EdgeFireSmoke++	33	0,54+/-0,01	29,86+/-2,59	29,86	54,13	22,11
		VGG16	4	0,89+/-0,03	241,08+/-9,81	241,70	289,25	221,70
		MobileNet V2	20	0,53+/-0,01	49,46+/-3,30	49,51	73,84	40,18
		Inception V3	12	0,77+/-0,01	77,26+/-3,54	76,77	89,81	68,64
		Xception	8	0,83+/-0,01	120,48+/-7,25	120,16	224,90	111,59
		DenseNet121	9	0,76+/-0,02	100,39+/-5,14	100,14	159,88	90,16
		VGG19	3	0,94+/-0,02	301,14+/-16,93	297,09	342,63	268,82
Cam 2	RNA	EdgeFireSmoke++	40	0,52+/-0,02	24,77+/-2,52	24,60	34,06	17,85
		VGG16	4	0,83+/-0,04	241,22+/-10,02	239,29	276,14	218,01
		MobileNet V2	25	0,48+/-0,01	38,99+/-2,94	38,91	48,36	30,29
		Inception V3	13	0,76+/-0,01	75,77+/-2,61	75,55	88,75	69,15
		Xception	8	0,79+/-0,01	113,67+/-3,08	113,50	130,89	105,79
		DenseNet121	10	0,68+/-0,01	95,78+/-4,74	95,00	111,34	84,31
		VGG19	3	0,88+/-0,04	292,07+/-6,33	292,07	319,90	274,89

Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

A métrica elaborada para mensurar a capacidade de processamento dos algoritmos foi o coeficiente CPTR, apresentado na Tabela 35. A partir de análises do CPTR, chegou-se a conclusão de que apenas o algoritmo proposto é capaz de fazer o processamento em tempo real, no qual obteve um CPTR de 1,1 e 1,34, respectivamente, com imagem das câmeras 1 e 2. Nesta tabela, o caso 1 engloba os algoritmos que não conseguem acompanhar o FPS da câmera, portanto, não funcionam em tempo real. No caso 2 temos os algoritmos que conseguem atender ao requisito de processamento em tempo real, e portanto, são capazes de realizar a tarefa.

Tabela 35 – Análise do coeficiente de processamento em tempo real, CPTR.

Imagens	Préprocessamento	Modelo CNN	FPS		CPTR	Análise do coeficiente CPTR
			Algoritmo	Câmera		
Cam 1	RNA	EdgeFireSmoke++	33	30	1,1	Caso 2
		VGG16	4	30	0,13	Caso 1
		MobileNet V2	20	30	0,66	Caso 1
		Inception V3	12	30	0,4	Caso 1
		Xception	8	30	0,27	Caso 1
		DenseNet121	9	30	0,3	Caso 1
		VGG19	3	30	0,1	Caso 1
		EdgeFireSmoke++	40	30	1,34	Caso 2
VGG16	4	30	0,13	Caso 1		
Cam 2	RNA	MobileNet V2	25	30	0,83	Caso 1
		Inception V3	13	30	0,43	Caso 1
		Xception	8	30	0,27	Caso 1
		DenseNet121	10	30	0,33	Caso 1
		VGG19	3	30	0,1	Caso 1

Fonte: Próprio Autor.

Na Figura 31 temos um gráfico que apresenta o aprendizado da RNA durante as 500 épocas de treinamento. Enquanto que na Figura 32 podemos analisar e comparar as curvas de aprendizado das CNNs durante as 25 épocas de treinamento. Por fim, o gráfico da Figura 33 mostra de forma ordenada o desempenho no teste de tempo real obtido com a RNA proposta e suas variações de CNN. De forma clara, foi visto que o melhor desempenho em tempo de processamento foi alcançado pelo método EdgeFireSmoke++ proposto neste trabalho. O pior desempenho em tempo foi registrado pela combinação RNA+VGG19.

É de extrema importância ressaltar que os requisitos para o método proposto são: possuir fonte de iluminação natural ou artificial no ambiente; as câmeras precisam estar conectadas à internet; e as câmeras devem ter conexão via protocolo RTSP. Recomenda-se que o computador no qual o algoritmo está sendo processado seja equipado com pelo menos 16 GB de RAM, processador de 4 núcleos, com conexão ininterrupta à Internet.

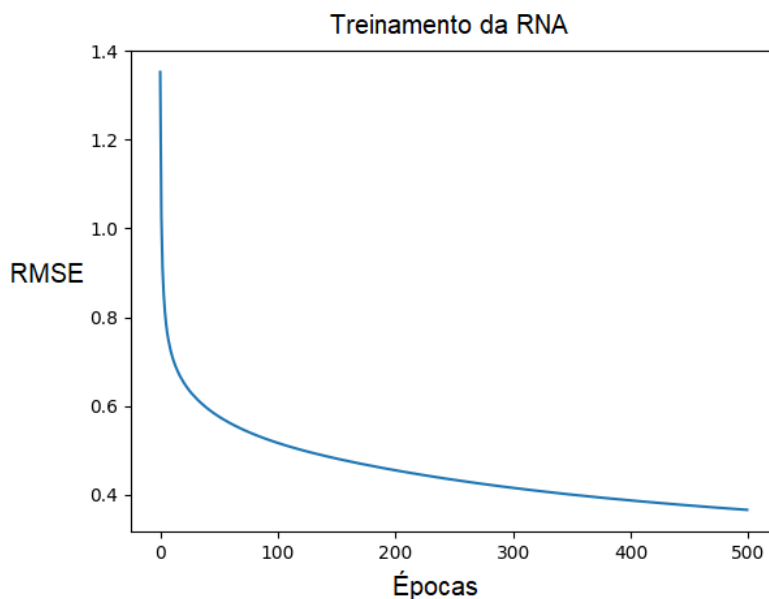
### 4.3 Análise do Índice de Risco de Incêndio Florestal

Neste momento, será comentado sobre o índice do risco de incêndio florestal, a qual chamamos IRIF. Esta análise e discussão será feita com base em dados hipotéticos de temperatura, umidade relativa do ar, e velocidade do vento, todos apresentados nas Tabelas 36 à 38.

Em todas as situações, foram simulados os valores de temperatura do ar (°C) partindo de 5°C até 50°C; mantendo a velocidade do vento (km/h) e a umidade relativa do ar (%) constantes, variando apenas esta última entre as tabelas. Com base nessas simulações, foi possível ver que os

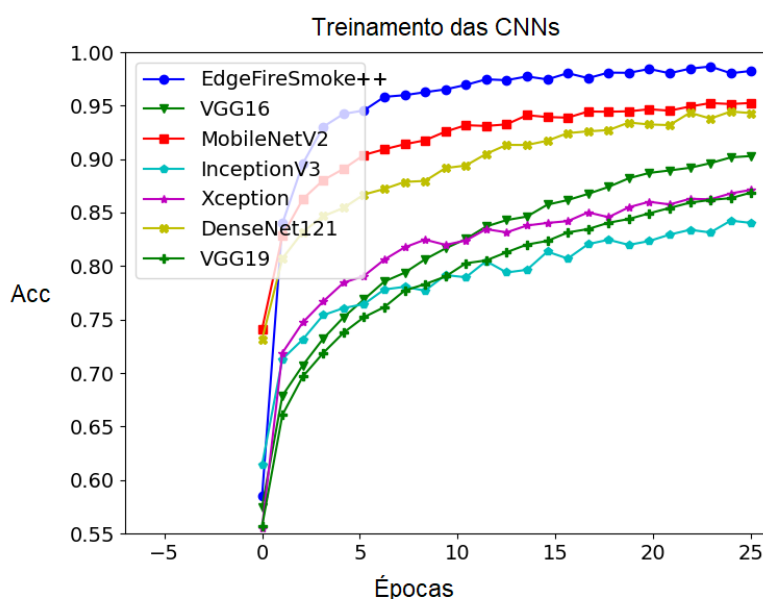


Figura 31 – Curva de aprendizagem da RNA durante 500 épocas.



Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Figura 32 – Curvas de aprendizagem das CNNs durante 25 épocas.

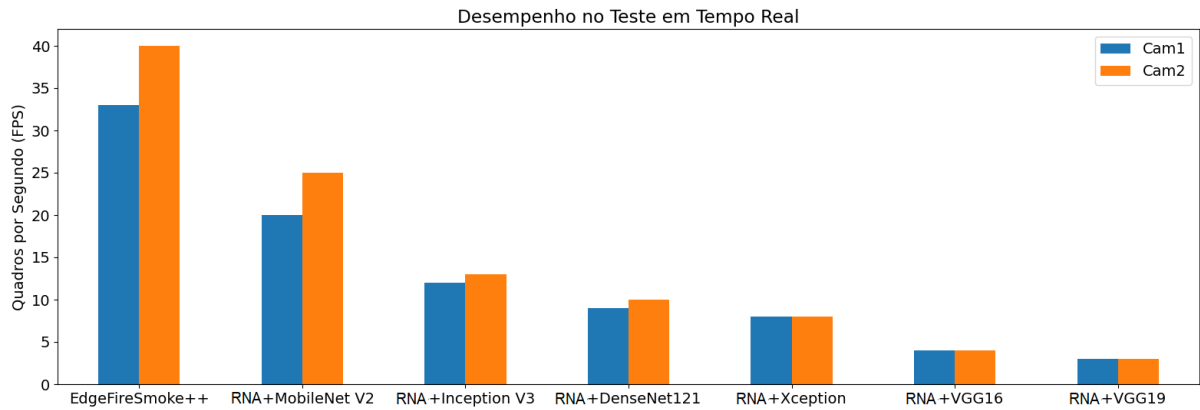


Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

prontos críticos que podem contribuir para o surgimento de incêndios florestais são a temperatura do ar acima dos 25°C e a umidade relativa do ar abaixo dos 30%. Observe, por exemplo, que na Tabela 36 para a temperatura de 30°C, 35% de umidade relativa e um vento de 25km/h, tem-se a estimativa de risco elevado para incêndios.

Na Tabela 37 esse risco é adiantado para 25°C devido a redução da umidade relativa do ar para 30%. Na Tabela 38 o risco é ainda maior, sendo antecipado para 20°C, quando considerada uma umidade relativa do ar de 25% e mesma velocidade do vento. Logo, percebe-se que quanto maior a temperatura e menor a umidade relativa do ar, maior será o risco de ocorrer um incêndio florestal, de forma que a velocidade do vento pode contribuir para a rápida propagação do fogo. Consequentemente,

Figura 33 – Desempenho obtido com a RNA proposta e as variações de CNNs avaliadas no teste.



Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Tabela 36 – Análise 1 do índice IRIF utilizando simulações de dados meteorológicos nas seguintes condições climáticas: (1) Temperatura [T] variável; (2) Umidade relativa do ar [U] constante em 35%; (3) Velocidade do vento [V] constante de 25 km/h.

Simulação	Temperatura do ar (°C)	Umidade relativa do ar (%)	Velocidade do vento (km/h)	Índice IRIF	Grau de Risco	Escala de cores
1	5	35	25	0,39	Reduzido	Azul
2	10	35	25	0,53	Moderado	Verde
3	15	35	25	0,67	Moderado	Verde
4	20	35	25	0,82	Moderado	Verde
5	25	35	25	0,96	Moderado	Verde
6	30	35	25	1,11	Elevado	Amarelo
7	35	35	25	1,25	Elevado	Amarelo
8	40	35	25	1,39	Elevado	Amarelo
9	45	35	25	1,53	Muito Elevado	Laranja
10	50	35	25	1,67	Muito Elevado	Laranja
<b>Média</b>	27,5	35	25	1,03	Elevado	Amarelo
<b>Máximo</b>	50	35	25	1,67	Muito Elevado	Laranja
<b>Mínimo</b>	5	35	25	0,39	Reduzido	Azul

Fonte: Próprio Autor.

Tabela 37 – Análise 2 do índice IRIF utilizando simulações de dados meteorológicos nas seguintes condições climáticas: (1) Temperatura [T] variável; (2) Umidade relativa do ar [U] constante em 30%; (3) Velocidade do vento [V] constante de 25 km/h.

Simulação	Temperatura do ar (°C)	Umidade relativa do ar (%)	Velocidade do vento (km/h)	Índice IRIF	Grau de Risco	Escala de cores
1	5	30	25	0,41	Reduzido	Azul
2	10	30	25	0,58	Moderado	Verde
3	15	30	25	0,75	Moderado	Verde
4	20	30	25	0,91	Moderado	Verde
5	25	30	25	1,08	Elevado	Amarelo
6	30	30	25	1,25	Elevado	Amarelo
7	35	30	25	1,41	Elevado	Amarelo
8	40	30	25	1,58	Muito Elevado	Laranja
9	45	30	25	1,75	Muito Elevado	Laranja
10	50	30	25	1,91	Muito Elevado	Laranja
<b>Média</b>	27,5	30	25	1,16	Elevado	Amarelo
<b>Máximo</b>	50	30	25	1,91	Muito Elevado	Laranja
<b>Mínimo</b>	5	30	25	0,41	Reduzido	Azul

Fonte: Próprio Autor.

Tabela 38 – Análise 3 do índice IRIF utilizando simulações de dados meteorológicos nas seguintes condições climáticas: (1) Temperatura [T] variável; (2) Umidade relativa do ar [U] constante em 25%; (3) Velocidade do vento [V] constante de 25 km/h.

Simulação	Temperatura do ar (°C)	Umidade relativa do ar (%)	Velocidade do vento (km/h)	Índice IRIF	Grau de Risco	Escala de cores
1	5	25	25	0,45	Reduzido	Azul
2	10	25	25	0,65	Moderado	Verde
3	15	25	25	0,85	Moderado	Verde
4	20	25	25	1,05	Elevado	Amarelo
5	25	25	25	1,25	Elevado	Amarelo
6	30	25	25	1,45	Elevado	Amarelo
7	35	25	25	1,65	Muito Elevado	Laranja
8	40	25	25	1,85	Muito Elevado	Laranja
9	45	25	25	2,05	Extremo	Vermelho
10	50	25	25	2,25	Extremo	Vermelho
<b>Média</b>	27,5	25	25	1,35	Elevado	Amarelo
<b>Máximo</b>	50	25	25	2,25	Extremo	Vermelho
<b>Mínimo</b>	5	25	25	0,45	Reduzido	Azul

Fonte: Próprio Autor.

o grau de risco é maior nessas situações. Adicionalmente, temos a escala de cores, que inicia na cor azul e conduz ao nível crítico identificado pela cor vermelho.

É importante deixar claro que as informações apresentadas nestas tabelas constituem apenas uma simulação com dados hipotéticos para que tenhamos uma noção da abrangência e condições de aplicação do IRIF. Portanto, as tabelas não substituem um estudo em campo mais aprofundado que poderá ser realizado com apoio de profissionais especializados no assunto.

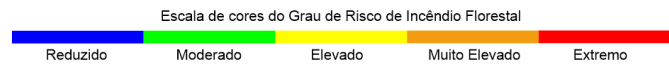
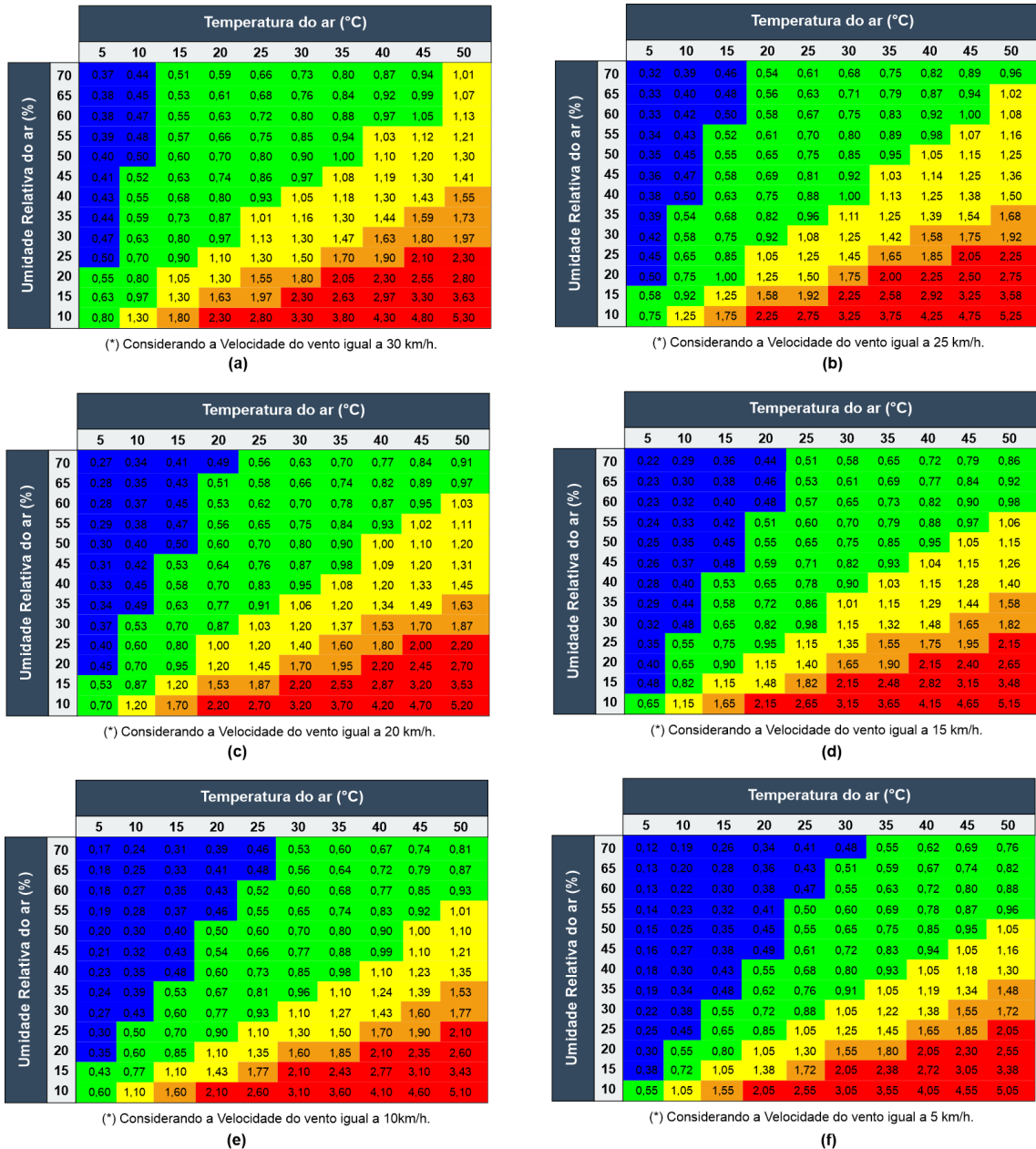
A última análise do IRIF é apresentada na Figura 34, na qual foram consideradas seis condições de velocidade do vento, variando de 30km/h à 5km/h, além de considerar as condições adversas de temperatura e umidade relativa do ar. Com base nestas tabelas, pode-se identificar o grau de risco através da escala de cores. No entanto, este cálculo está embarcado no algoritmo proposto e realiza toda esta estimativa do grau de risco automaticamente. Pode-se com estas informações tentar prever condições futuras de clima, capazes de influenciar na ocorrência dos incêndios florestais. Porém, como já mencionado, este estudo tem foco no algoritmo proposto, e por este motivo, este pode ser complementado por estudos mais aprofundados que relacionem as condições meteorológicas ao risco de ocorrência de incêndios florestais.

#### 4.4 Interface de Visualização de Dados (IoT-IHM)

A Internet das Coisas (IoT) tem revolucionado a maneira como lidamos com a detecção e o alerta de fogo, fumaça e incêndios florestais, permitindo uma abordagem mais eficaz e abrangente. Esta tecnologia, quando combinada com métodos clássicos e modernos, tem se mostrado fundamental para prevenir e responder a esses desastres naturais.

Foi utilizada neste trabalho uma tecnologia clássica conhecida por Interface Humano-Máquina (IHM). A IHM representa um elemento crucial na interação entre seres humanos e sistemas computacionais, desempenhando um papel fundamental na área de visão computacional. Uma IHM eficaz em visão computacional deve cumprir vários requisitos técnicos e científicos. Neste trabalho, ela

Figura 34 – Tabelas de obtenção do IRIF calculadas a partir de dados hipotéticos de Temperatura do ar (°C) e da Umidade Relativa do ar (%), considerando as condições [a-f] com diferentes velocidades do vento, variando de 30 km/h à 5 km/h.



Fonte: Próprio Autor.

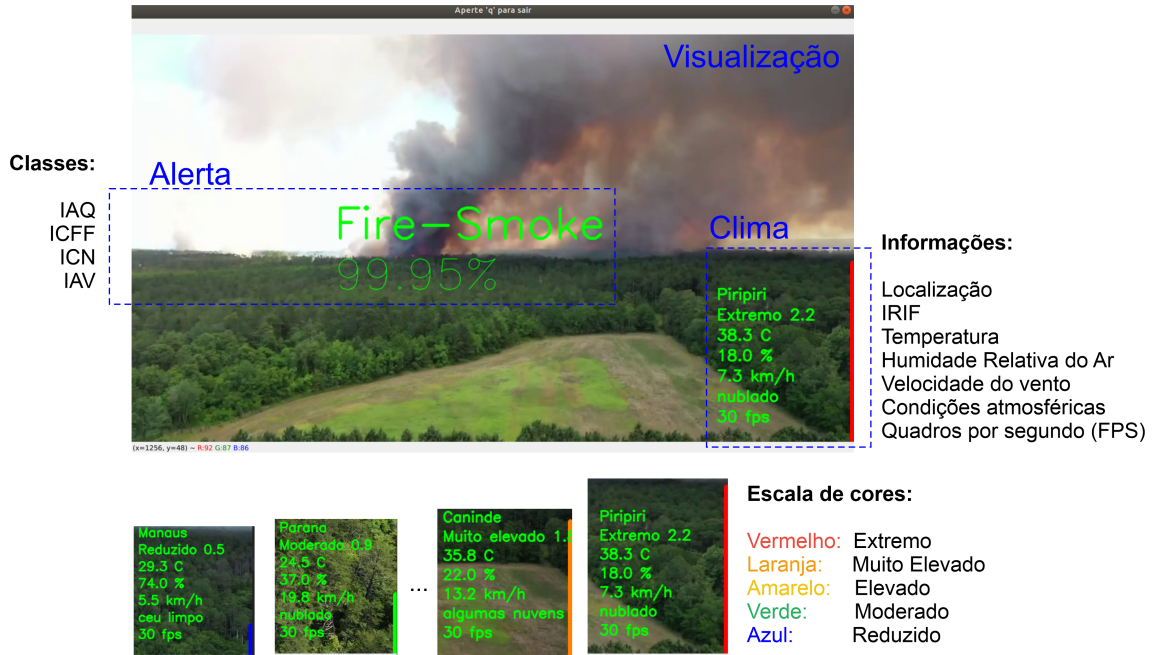
permite que o usuário interaja de forma intuitiva com os resultados do processamento de vídeo. Isso inclui a capacidade de visualizar dados brutos, os resultados do algoritmo EdgeFireSmoke e as análises meteorológicas de maneira clara e compreensiva.

A interface IoT-IHM desenvolvida nesta tese mostra as previsões do algoritmo em tempo real, as informações meteorológicas e o risco de incêndios florestais usando a escala de cores. Esta interface interativa de visualização de dados, conectada à internet, denominamos IoT-IHM. Esta interface permite ao sistema desenvolvido obter informações meteorológicas em tempo real do local monitorado.

As informações são obtidas a partir da internet usando o algoritmo de detecção de incêndios florestais, que utiliza uma API de acesso a informações do clima.

Os resultados experimentais do algoritmo proposto podem ser visualizados na Figura 35.

Figura 35 – Interface IoT-IHM que mostra as previsões do algoritmo em tempo real, as informações meteorológicas da API meteorológica e o risco de incêndios florestais usando a escala de cores.



Fonte: Próprio Autor, originalmente publicada em Almeida *et al.* (2023).

Podemos ver no centro da interface a classificação feita pelo algoritmo da classe prevista, que pode ser de quatro tipos (IAQ, ICFF, ICN e IAV); no canto inferior direito vemos as informações obtidas a partir da API OpenWeather, as quais são utilizadas para o cálculo do IRIF e usadas para a representação da escala de cores. A interface mostra as detecções em tempo real obtidas com o algoritmo EdgeFireSmoke.

#### 4.5 Conclusão deste Capítulo

Neste capítulo, foram explorados de forma abrangente os resultados obtidos, além de realizar uma análise e discussão aprofundada do modelo proposto e suas características construtivas.

A partir dos resultados obtidos, foi possível observar os filtros em cada camada da rede, reforçando a importância do modelo de aprendizagem profunda. Além disso, identificamos o efeito produzido pelos filtros nas situações abordadas, demonstrando o impacto de um modelo CNN otimizado para um problema de apenas quatro classes. Adicionalmente, foi discutido sobre a estimativa do risco de incêndio, ressaltando a necessidade de considerar as variáveis meteorológicas.

Essas descobertas fortalecem a fundamentação teórica e contribuem para a compreensão mais aprofundada do abordado. Por fim, vale ressaltar que este capítulo, juntamente com os demais, é fundamental para a consolidação do trabalho como um todo, fornecendo bases sólidas para a conclusão.

## 5 CAPÍTULO V: CONCLUSÃO

Nesta pesquisa, foi apresentado um novo algoritmo capaz de detectar e alertar a ocorrência de incêndios florestais a partir das imagens digitais capturadas pelos sistemas de CFTV IP.

O algoritmo apresentou melhorias em relação ao trabalho proposto em sua etapa inicial (SILVA *et al.*, 2022). Dentre elas, apresentamos a inclusão de uma etapa de pré-processamento baseada em Rede Neural Artificial (ANN). A ANN faz uma comparação de similaridade entre a imagem de entrada e padrões para reconhecer a presença ou ausência de ambiente florestal. Desta forma, a detecção de incêndios é realizada em duas etapas (ANN-CNN), aumentando assim a confiabilidade do algoritmo.

Adicionalmente, foi criada uma interface de visualização de dados (IoT-IHM) capaz de calcular o índice do risco de ocorrer incêndio florestal. O índice mencionado é calculado a partir de variáveis meteorológicas. Estas melhorias no algoritmo possibilitaram a publicação de um segundo trabalho (ALMEIDA *et al.*, 2023).

O EdgeFireSmoke++ proposto nesta tese foi superior aos outros métodos CNN da literatura avaliados. Apresentou vantagens em relação ao custo computacional. Na abordagem do primeiro artigo (SILVA *et al.*, 2022), foi o método mais rápido entre os avaliados, com tempo de detecção por imagem de 30 milissegundos na GPU Tesla T4, 60 milissegundos na GTX 1650 e 210 milissegundos na Jetson Nano. Na abordagem do segundo artigo (ALMEIDA *et al.*, 2023) realizada no Dataset 2, considerando o conjunto de teste, foi obtido 95,41% de acurácia, 95,49% de precisão, 95,38% de Recall e 95,41% de F1-score.

O algoritmo proposto registrou a melhor taxa de *frames* por segundo (FPS) utilizando a câmera IP avaliada, no qual alcançou 33 FPS. Para seu funcionamento, o algoritmo proposto mostrou-se bastante leve em termos computacionais, podendo trabalhar em uma CPU de 4 núcleos, 2.1GHz, com consumo médio de 540MB de memória RAM. Logo, conseguiu cumprir os requisitos necessários para ser uma *lightweight CNN*.

São vantagens significativas do algoritmo: i) possibilidade de monitorar pequenas áreas florestais de  $30m^2$  ou mais, de forma a suprir ao menos uma das dificuldades enfrentada por satélites, e ii) possui uma interface de visualização de dados em tempo real com uso de inteligência artificial para alerta e estimativa do risco de incêndio florestal.

Além do mais, foram identificadas duas limitações do algoritmo: i) a necessidade de usar o algoritmo durante o dia, pois o treinamento do modelo foi feito a partir de câmeras convencionais diurnas, e ii) a necessidade de usar câmeras conectadas à internet.

Importante destacar que o algoritmo proposto, EdgeFireSmoke, e sua versão expandida EdgeFireSmoke++, mostraram-se promissores para a detecção de incêndios florestais por videomonitoramento em tempo real (CFTV IP). Sua interface de visualização de informações (IoT-IHM) constitui uma importante vantagem de uso. O algoritmo pode ser muito útil no monitoramento de florestas, reservas ambientais e na conservação dos biomas brasileiros.

Por fim, como trabalhos futuros, sugere-se adicionar uma etapa de segmentação de fumaça e calcular a densidade da fumaça na imagem. Além do mais, sugere-se a criação de uma API ou

biblioteca, a criação de uma central de monitoramento baseada em *dashboard*, a integração com outros dados meteorológicos, implementar o envio de alertas por SMS e E-mail, analisar novos analíticos de vídeo, e fazer o mapeamento de regiões que são mais suscetíveis a ocorrência de incêndios florestais, fazendo a integração com robôs móveis e aéreos não tripulados.

## REFERÊNCIAS

- AGARWAL, P.; JHA, G. Forest fire detection using classifiers and transfer learning. In: IEEE. **2021 IEEE International Conference on Robotics, Automation and Artificial Intelligence (RAAI)**. [S.l.], 2021. p. 29–33.
- AGIRMAN, A. K.; TASDEMIR, K. Blstm based night-time wildfire detection from video. **PLoS one**, Public Library of Science San Francisco, CA USA, v. 17, n. 6, p. e0269161, 2022.
- ALMEIDA, J. S.; JAGATHEESAPERUMAL, S. K.; NOGUEIRA, F. G.; ALBUQUERQUE, V. H. C. de. Edgefiresmoke++: A novel lightweight algorithm for real-time forest fire detection and visualization using internet of things-human machine interface. **Expert Systems with Applications**, Elsevier, v. 221, p. 119747, 2023.
- AMAZONIAREAL.COM.BR. **Incêndio atinge Parna Jaú e ICMBio pede apoio para combater o fogo**. 2021. Disponível em: <<https://amazoniareal.com.br/incendio-atinge-parna-jau-e-icmbio-pede-apoio-para-combater-o-fogo/>>.
- AZAMI, M. H. bin; ORGER, N. C.; SCHULZ, V. H.; CHO, M. *et al.* Demonstration of wildfire detection using image classification onboard cubesat. In: IEEE. **2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS**. [S.l.], 2021. p. 5413–5416.
- BBC, N. B. **Forest fires around the world are the biggest in scale and CO2 emissions in 18 years**. 2020. Disponível em: <<https://www.bbc.com/portuguese/geral-54202546>>.
- BEBIS, G.; GEORGIOPOULOS, M. Feed-forward neural networks. **Ieee Potentials**, IEEE, v. 13, n. 4, p. 27–31, 1994.
- BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006. v. 4.
- CHATRAGADDA, A.; CHALASANI, S. H. V.; CHALLA, N.; GUPTA, N. V. R.; OLETI, P. P.; AMARENDRA, K. Convolutional neural networks based enhanced forest monitoring system for early fire detection. In: IEEE. **2022 7th International Conference on Communication and Electronics Systems (ICCES)**. [S.l.], 2022. p. 425–432.
- CHAUHAN, R.; VAIDYA, H.; SINGHAL, M.; BARDDHAN, A.; AWASTHI, S.; NARUKA, M. S.; CHAUHAN, S. S. An iot-based novel framework for early prediction of forest fire. In: IEEE. **2023 International Conference on Disruptive Technologies (ICDT)**. [S.l.], 2023. p. 727–732.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1251–1258.
- DIARIODONORDESTE.COM.BR. **Incêndio no Parque do Cocó teve causa involuntária, diz perícia**. 2022. Disponível em: <<https://diariodonordeste.verdesmares.com.br/ceara/incendio-no-parque-do-coco-teve-causa-involuntaria-diz-pericia-1.3194044>>.
- FAN, R.; PEI, M. Lightweight forest fire detection based on deep learning. In: IEEE. **2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)**. [S.l.], 2021. p. 1–6.
- FILHO, J. E. F.; ARAÚJO, A. C. Análise de ocorrências de incêndios florestais na área do parque estadual do cocó, região metropolitana de fortaleza, ce. **Revista Brasileira de Meteorologia**, SciELO Brasil, v. 36, p. 563–569, 2021.
- G1.GLOBO.COM. **Incêndio em Chapada dos Guimarães (MT) já dura 5 dias e destruiu 4,6 mil hectares**. 2021. Disponível em: <<https://g1.globo.com/mt/mato-grosso/noticia/2021/08/23/incendio-em-chapada-dos-guimaraes-mt-ja-dura-5-dias-e-destruiu-46-mil-hectares.ghtml>>.
- G1.GLOBO.COM. **Bombeiros combatem incêndio no Parque Estadual da Serra da Boa Esperança**. 2022. Disponível em: <<https://g1.globo.com/mg/sul-de-minas/noticia/2022/10/25/bombeiros-combatem-incendio-no-parque-estadual-da-serra-da-boja-esperanca.ghtml>>.
- G1.GLOBO.COM. **Incêndio atinge 500 hectares da Estação Ecológica do Taim, no Sul do RS**. 2022. Disponível em: <<https://g1.globo.com/rs/rio-grande-do-sul/noticia/2022/12/13/incendio-atinge-500-hectares-da-estacao-ecologica-do-taim-no-sul-do-rs.ghtml>>.



- GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998.
- GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. [S.l.]: Alta Books, 2019.
- GOMES, D. T.; SAKAMOTO, M. S.; SILVA, A. E. F. d. Avaliação da distribuição de focos de calor às margens de rodovias federais no estado do ceará/brasil. **Revista Brasileira de Meteorologia**, SciELO Brasil, v. 35, p. 467–475, 2020.
- GONZÁLEZ-MOLES, M. Á.; RAMOS-GARCÍA, P.; WARNAKULASURIYA, S. An appraisal of highest quality studies reporting malignant transformation of oral lichen planus based on a systematic review. **Oral diseases**, Wiley Online Library, v. 27, n. 8, p. 1908–1918, 2021.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer, 2009. v. 2.
- HAYKIN, S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice Hall PTR, 1994.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.
- HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.
- HUBEL, D. H. Single unit activity in striate cortex of unrestrained cats. **The Journal of physiology**, Wiley Online Library, v. 147, n. 2, p. 226–238, 1959.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields of single neurones in the cat's striate cortex. **The Journal of physiology**, Wiley Online Library, v. 148, n. 3, p. 574–591, 1959.
- HUSSAIN, T.; MUHAMMAD, K.; SER, J. D.; BAIK, S. W.; ALBUQUERQUE, V. H. C. de. Intelligent embedded vision for summarization of multiview videos in iiot. **IEEE Transactions on Industrial Informatics**, IEEE, v. 16, n. 4, p. 2592–2602, 2019.
- IBAMA. **Burn Monitoring in Satellite Images**. 2022. Disponível em: <<https://www.gov.br/ibama/pt-br/servicos/consultas/incendios-florestais/monitoramento-de-focos-de-queimadas-em-imagens-de-satelites>>.
- ILSVRC2014. **ImageNet Large Scale Visual Recognition Challenge 2014**. 2014. Disponível em: <<https://www.image-net.org/challenges/LSVRC/2014/>>.
- INPE. **Monitoramento dos Focos de Fogo Ativo por Países**. 2021. Disponível em: <[https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas\\_paises/](https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas_paises/)>.
- INPE. **Monitoramento dos Focos de Fogo Ativo por Países**. 2023. Disponível em: <[https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas\\_paises/](https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas_paises/)>.
- JAGATHEESAPERUMAL, S. K.; MUHAMMAD, K.; SAUDAGAR, A. K. J.; RODRIGUES, J. J. Automated fire extinguishing system using a deep learning based framework. **Mathematics**, Multidisciplinary Digital Publishing Institute, v. 11, n. 3, p. 608, 2023.
- KHALIFE, S.; CHENG, H.; BASU, A. Neural networks with linear threshold activations: structure and algorithms. **Mathematical Programming**, Springer, p. 1–24, 2023.
- KHAN, S.; MUHAMMAD, K.; HUSSAIN, T.; SER, J. D.; CUZZOLIN, F.; BHATTACHARYYA, S.; AKHTAR, Z.; ALBUQUERQUE, V. H. C. de. Deepsmoke: Deep learning model for smoke detection and segmentation in outdoor environments. **Expert Systems with Applications**, Elsevier, v. 182, p. 115125, 2021.
- KHAN, S.; MUHAMMAD, K.; MUMTAZ, S.; BAIK, S. W.; ALBUQUERQUE, V. H. C. de. Energy-efficient deep cnn for smoke detection in foggy iot environment. **IEEE Internet of Things Journal**, v. 6, n. 6, p. 9237–9245, 2019.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGESS, C.; BOTTOU, L.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Disponível em: <[https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM**, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LI, W.; XIAOBO, S.; JUN, C.; YING, L. Research on forest fire image recognition algorithm based on color feature statistics. In: IEEE. **2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)**. [S.l.], 2021. p. 346–349.

LI, Y.; SHEN, Z.; LI, J.; XU, Z. A deep learning method based on srn-yolo for forest fire detection. In: IEEE. **2022 5th International Symposium on Autonomous Systems (ISAS)**. [S.l.], 2022. p. 1–6.

LI, Y.; ZHANG, W.; LIU, Y.; SHAO, X. A lightweight network for real-time smoke semantic segmentation based on dual paths. **Neurocomputing**, Elsevier, v. 501, p. 258–269, 2022.

LIMA, M. B.; COSTA, G. B.; PEREIRA, W. D. S.; SILVA, D. T. C. da; MATIAS, L. V. D. S.; MACAMBIRA, A. C. D. S.; HEIDEMANN, M. D. A.; ALMADA, N. B. Aspectos sazonais de um índice de risco de incêndios florestais na região norte do Brasil por meio de dados meteorológicos. **Biodiversidade Brasileira-BioBrasil**, v. 11, n. 4, p. 46–59, 2021.

LOURENÇO, L. **Uma fórmula expedita para determinar o índice meteorológico de risco de eclosão de fogos florestais em Portugal Continental**. [S.l.: s.n.], 1991.

LOURENÇO, L.; GONÇALVES, A. B.; LOUREIRO, J. Sistema de informação de risco de incêndio florestal. **ENB, Revista Técnica e Formativa da Escola Nacional de Bombeiros**, p. 16–25, 1997.

LUO, Y.; WONG, Y.; KANKANHALLI, M.; ZHAO, Q.  $\mathcal{G}$ -softmax: Improving intraclass compactness and interclass separability of features. **IEEE Transactions on Neural Networks and Learning Systems**, v. 31, n. 2, p. 685–699, 2020.

MAUGER, J. S.; CARVALHO, F. S. H.; TAVARES, M.; CARVALHO, I.; SILVA, V. P.; MARQUES, D. S.; ARMANDO, M. S. Cartilha de incêndios florestais: causas, consequências e como evitar. **Disponível em: <https://www.ibram.df.gov.br/wp-content/uploads/2018/02/Cartilha-Inc%C3%AAndios-Florestais-Causas-Consequ%C3%AAncias-e-Como-Evitar.pdf>**. Acesso em: 23/02/2022, v. 1, 2009.

MITCHELL, T. M. *et al.* **Machine learning**. [S.l.]: McGraw-hill New York, 2007. v. 1.

MOHNISH, S.; P, A. K.; S, G. R.; A, S. V.; P, P.; S, E. Deep learning based forest fire detection and alert system. In: **2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)**. [S.l.: s.n.], 2022. p. 1–5.

MUHAMMAD, K.; HUSSAIN, T.; SER, J. D.; PALADE, V.; ALBUQUERQUE, V. H. C. D. Deepres: A deep learning-based video summarization strategy for resource-constrained industrial surveillance scenarios. **IEEE Transactions on Industrial Informatics**, IEEE, v. 16, n. 9, p. 5938–5947, 2019.

MUHAMMAD, K.; HUSSAIN, T.; TANVEER, M.; SANNINO, G.; ALBUQUERQUE, V. H. C. de. Cost-effective video summarization using deep cnn with hierarchical weighted fusion for iot surveillance networks. **IEEE Internet of Things Journal**, IEEE, v. 7, n. 5, p. 4455–4463, 2019.

- MUHAMMAD, K.; KHAN, S.; PALADE, V.; MEHMOOD, I.; ALBUQUERQUE, V. H. C. D. Edge intelligence-assisted smoke detection in foggy surveillance environments. **IEEE Transactions on Industrial Informatics**, IEEE, v. 16, n. 2, p. 1067–1075, 2019.
- MUHAMMAD, K.; KHAN, S.; PALADE, V.; MEHMOOD, I.; ALBUQUERQUE, V. H. C. de. Edge intelligence-assisted smoke detection in foggy surveillance environments. **IEEE Transactions on Industrial Informatics**, v. 16, n. 2, p. 1067–1075, 2020.
- MUHAMMAD, K.; RODRIGUES, J. J.; KOZLOV, S.; PICCIALLI, F.; ALBUQUERQUE, V. H. C. de. Energy-efficient monitoring of fire scenes for intelligent networks. **IEEE Network**, IEEE, v. 34, n. 3, p. 108–115, 2020.
- MURTAGH, F. Multilayer perceptrons for classification and regression. **Neurocomputing**, Elsevier, v. 2, n. 5-6, p. 183–197, 1991.
- NGUYEN, A.; NGUYEN, H.; TRAN, V.; PHAM, H. X.; PESTANA, J. A visual real-time fire detection using single shot multibox detector for uav-based fire surveillance. In: IEEE. **2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)**. [S.l.], 2021. p. 338–343.
- OLIVER, A. S.; ASHWANTHIKA, U.; ASWITHA, R. Detection of forest fire using convolutional neural networks. In: IEEE. **2020 7th International Conference on Smart Structures and Systems (ICSSS)**. [S.l.], 2020. p. 1–6.
- OMAR, N.; AL-ZEBARI, A.; SENGUR, A. Deep learning approach to predict forest fires using meteorological measurements. In: IEEE. **2021 2nd International Informatics and Software Engineering Conference (IISEC)**. [S.l.], 2021. p. 1–4.
- OMM. **Earth had hottest three-month period on record, with unprecedented sea surface temperatures and much extreme weather**. 2023. Disponível em: <<https://public.wmo.int/en/media/press-release/earth-had-hottest-three-month-period-record-unprecedented-sea-surface>>.
- ONU. **Planeta Terra teve o trimestre mais quente já registrado**. 2023. Disponível em: <<https://news.un.org/pt/story/2023/09/1820042>>.
- PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 10, p. 1345–1359, 2010.
- PARK, M.; JEON, Y.; BAK, J.; PARK, S. *et al.* Forest-fire response system using deep-learning-based approaches with cctv images and weather data. **IEEE Access**, IEEE, v. 10, p. 66061–66071, 2022.
- PRİYADARSHINI, I.; COTTON, C. A novel lstm–cnn–grid search-based deep neural network for sentiment analysis. **The Journal of Supercomputing**, Springer, v. 77, n. 12, p. 13911–13932, 2021.
- RACHANA, P.; RAJALAKSHMI, B.; BHAT, T.; KAUR, S.; BIMALI, S. Comparative study of different methods for fire detection using convolutional neural network (cnn). In: IEEE. **2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)**. [S.l.], 2022. p. 1759–1765.
- RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation function for artificial neural network. In: IEEE. **2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)**. [S.l.], 2020. p. 281–286.
- REDDY, P. R.; KALYANASUNDARAM, P. Novel detection of forest fire using temperature and carbon dioxide sensors with improved accuracy in comparison between two different zones. In: IEEE. **2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)**. [S.l.], 2022. p. 524–527.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.
- SADI, M.; ZHANG, Y.; XIE, W.-F.; HOSSAIN, F. A. Forest fire detection and localization using thermal and visual cameras. In: IEEE. **2021 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.], 2021. p. 744–749.

- SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520.
- SANTOS, D. C.; SILVA, F. A. da; PEREIRA, D. R.; ALMEIDA, L. L. de; ARTERO, A. O.; PITERI, M. A.; ALBUQUERQUE, V. H. Real-time traffic sign detection and recognition using cnn. **IEEE Latin America Transactions**, IEEE, v. 18, n. 03, p. 522–529, 2020.
- SILVA, J.; HUANG, C.; NOGUEIRA, F.; BHATIA, S.; ALBUQUERQUE, V. Edgefiresmoke: a novel lightweight cnn model for real-time video fire-smoke detection. **IEEE Transactions on Industrial Informatics**, IEEE, 2022.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: SPRINGER. **AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings 19**. [S.l.], 2006. p. 1015–1021.
- SOUSA, M. J.; MOUTINHO, A.; ALMEIDA, M. Thermal infrared sensing for near real-time data-driven fire detection and monitoring systems. **Sensors**, MDPI, v. 20, n. 23, p. 6803, 2020.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions (2014). **arXiv preprint arXiv:1409.4842**, arXiv, v. 10, 2014.
- SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 2818–2826.
- TAHIR, H. U. A.; WAQAR, A.; KHALID, S.; USMAN, S. M. Wildfire detection in aerial images using deep learning. In: IEEE. **2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)**. [S.l.], 2022. p. 1–7.
- TLIG, M.; BOUCHOUICHA, M.; SAYADI, M.; MOREAU, E. Infrared-visible images' fusion techniques for forest fire monitoring. In: IEEE. **2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)**. [S.l.], 2022. p. 1–6.
- UNEP. **Incêndios florestais, poluição sonora e descompasso dos ciclos de vida: Relatório da ONU identifica ameaças ambientais iminentes**. 2022. Disponível em: <<https://www.unep.org/pt-br/noticias-e-reportagens/comunicado-de-imprensa/incendios-florestais-poluicao-sonora-e-descompasso>>.
- UNEP. **Número de incêndios florestais aumentará em 50 governos não estão preparados, alertam especialistas**. 2022. Disponível em: <<https://www.unep.org/pt-br/noticias-e-reportagens/comunicado-de-imprensa/numero-de-incendios-florestais-aumentara-em-50-ate>>.
- WANG, J.; BELATRECHE, A.; MAGUIRE, L.; MCGINNITY, M. Online versus offline learning for spiking neural networks: A review and new strategies. In: IEEE. **2010 IEEE 9th international conference on cyberntic intelligent systems**. [S.l.], 2010. p. 1–6.
- WANG, S.; CHEN, T.; LV, X.; ZHAO, J.; ZOU, X.; ZHAO, X.; XIAO, M.; WEI, H. Forest fire detection based on lightweight yolo. In: IEEE. **2021 33rd Chinese Control and Decision Conference (CCDC)**. [S.l.], 2021. p. 1560–1565.
- YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? **Advances in neural information processing systems**, v. 27, 2014.
- YUAN, F.; LI, K.; WANG, C.; FANG, Z. A lightweight network for smoke semantic segmentation. **Pattern Recognition**, Elsevier, v. 137, p. 109289, 2023.