

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

Sérgio Henrique dos Santos Preiss

**Combate inteligente ao fogo: uma comparação entre diferentes modelos de visão
computacional para detecção de incêndios florestais**

CAÇADOR, 2025.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

Sérgio Henrique dos Santos Preiss

**Combate inteligente ao fogo: uma comparação entre diferentes modelos de visão
computacional para detecção de incêndios florestais**

Trabalho de Conclusão de Curso submetido
ao Instituto Federal de Educação, Ciência
e Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Bacharel em Sistemas de Informação.

Orientador:
Prof. Cristiano Mesquita Garcia, Me.

Coorientador:
Paulo Roberto Córdova, Dr.

CAÇADOR, 2025.

Preiss, Sérgio Henrique dos Santos
P924c Combate inteligente ao fogo: uma comparação entre diferentes
modelos de visão computacional para detecção de incêndios florestais /
Sérgio Henrique dos Santos Preiss ; orientador: Cristiano Mesquita
Garcia ; coorientador: Paulo Roberto Córdova. -- 2025.
54 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal
de Educação, Ciência e Tecnologia de Santa Catarina, Caçador, 2025.
Inclui bibliografias.

1. Visão computacional. 2. Inteligência artificial. 3. Incêndios florestais.
4. Monitoramento ambiental . I. Garcia, Cristiano Mesquita. II. Córdova,
Paulo Roberto. III. Instituto Federal de Educação, Ciência e Tecnologia
de Santa Catarina – Graduação em Sistemas de Informação. IV. Título.

CDD 621

Ficha catalográfica elaborada pela Bibliotecária
Janice Moser Corrêa – CRB-14/1865

**COMBATE INTELIGENTE AO FOGO: UMA COMPARAÇÃO ENTRE DIFERENTES
MODELOS DE VISÃO COMPUTACIONAL PARA DETECÇÃO DE INCÊNDIOS
FLORESTAIS**

SÉRGIO HENRIQUE DOS SANTOS PREISS

Este trabalho foi julgado adequado para obtenção do Título de Bacharel em Sistemas de Informação e aprovado pela banca examinadora do Curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

CAÇADOR, 07 de julho de 2025.

Banca Examinadora:

Cristiano Mesquita Garcia, Me.

Paulo Roberto Córdova, Dr.

Ademir Goulart, Dr.

Gabriel Neves Ferrari, Dr.

AGRADECIMENTOS

Agradeço, primeiramente, aos professores que guiaram meu aprendizado até aqui, por todos os conselhos, pela ajuda e paciência. Agradeço especialmente ao meu professor e orientador Cristiano Mesquita Garcia, e ao professor e coorientador Paulo Roberto Córdova, que desempenharam a função com excelência e dedicação. Agradeço também aos amigos e familiares por todo apoio e ajuda que deram para a realização deste trabalho.

RESUMO

Os incêndios florestais representam um dos maiores desafios ambientais no Brasil e no mundo, ameaçando a biodiversidade, contribuindo para a emissão de gases de efeito estufa e causando sérios prejuízos econômicos e sociais. Este trabalho compara diferentes modelos de inteligência artificial baseados em visão computacional com o objetivo de detectar incêndios com ou sem fumaça. A partir de imagens públicas anotadas, os modelos *YOLO*, *Faster R-CNN* e *Vision Transformer* passaram por *fine-tuning* e foram avaliados sob diferentes métricas, como tempo de inferência e acurácia, na identificação de incêndios e/ou fumaças em diferentes cenários. Através da análise de imagens aéreas, este trabalho busca responder à questão “Como a visão computacional pode auxiliar na identificação de áreas com focos ativos de incêndios?”. Os modelos avaliados são considerados estado da arte e tiveram seus desempenhos e viabilidade prática avaliados, auxiliando pesquisas futuras à tomada de decisão sobre o modelo mais adequado para utilização. Para alcançar essa meta, foram definidas etapas como a reunião e análise de imagens, o *fine-tuning* dos modelos, análises de métricas geradas e testes com imagens, visando ao final delas, que os modelos desenvolvidos sejam capazes de identificar incêndios com precisão comparável ao estado da arte. Adicionalmente, a pesquisa comparou o desempenho dos diferentes modelos implementados, além de demonstrar o potencial da visão computacional como ferramenta para o monitoramento ambiental e a gestão de recursos naturais. Como resultado, foi identificado que, apesar do modelo *Faster R-CNN* ter obtido os melhores valores para as métricas *recall* e acurácia, seu tempo de inferência pode inviabilizá-lo para cenários de tempo real. Ao contrário, o modelo *YOLO* obteve valores interessantes para as métricas *mAP*, *F1-score* e precisão, além de ser o mais leve entre os três, sugerindo seu potencial uso em aplicações que necessitam de detecções rápidas e em tempo real com recursos limitados.

Palavras-chave: visão computacional; inteligência artificial; incêndios florestais; monitoramento ambiental.

ABSTRACT

Fire outbreaks represent one of the greatest environmental challenges in Brazil and worldwide, threatening biodiversity, contributing to greenhouse gas emissions and causing serious economic and social losses. This article compares different artificial intelligence models based on computer vision, with the aim of identifying fires with or without smoke in forest regions. With analysis of public images, the models *YOLO*, *Faster R-CNN* and *Vision Transformer* were *fine-tuned* and tested using different metrics to evaluate, such as inference time and accuracy, in identifying fires and/or smoke in different situations. Through the analysis of aerial images, the model seeks to answer “How computer vision can help in the identification of areas with fire outbreaks?”. The models are considered state-of-the-art and their performance and usability were evaluated, helping future research to decide which model is most appropriate for use. To achieve this goal, steps were defined, such as the collection and analysis of images, model fine-tuning, analyzing generated metrics and testing with images. At the end of these steps, the developed model is expected to accurately identify fire hotspots with high precision. Additionally, the research compared the performance of the different models, demonstrates the potential of computer vision as a tool for environmental monitoring and natural resource management. The result showed identified that the *Faster R-CNN* model has the best values for the metrics *recall* and accuracy, but the inference time may make it unfeasible for real-time situations. However, the *YOLO* model has interesting values for the *mAP*, *F1-score* and *precision* metrics, and is also the lightest between the three, suggesting its potential use in applications that require fast and real-time detections with limited resources.

Keywords: computer vision; artificial intelligence; forest fires; environmental monitoring.

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolucional)
IA	Inteligência Artificial
INPE	Instituto Nacional de Pesquisas Espaciais
MCTI	Ministério da Ciência, Tecnologia e Inovação do Brasil
MIT	<i>Massachusetts Institute of Technology</i> (Instituto de Tecnologia de Massachusetts)
MNIST	<i>Modified National Institute of Standards and Technology</i> (Instituto Nacional de Padrões e Tecnologia modificado)
RPN	<i>Region Proposal Network</i> (Rede de Propostas Regionais)

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Justificativa	11
1.2	Definição do Problema	12
1.3	Objetivo Geral	12
1.4	Objetivos específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Incêndios florestais e seus impactos	14
2.2	Visão computacional e <i>Machine learning</i>	17
2.3	Modelos de detecção de objetos	23
3	PROCEDIMENTOS METODOLÓGICOS	29
3.1	Metodologia de pesquisa	29
3.1.1	Tipo de pesquisa	29
3.1.2	Etapas metodológicas	29
3.2	Procedimentos metodológicos	30
3.2.1	Coleta de bases de imagens	30
3.2.2	Desenvolvimento do algoritmo (treinamento e ajuste do modelo <i>YOLO</i>)	30
3.2.3	Desenvolvimento do algoritmo (treinamento e ajuste do modelo <i>Faster R-CNN</i>)	31
3.2.4	Desenvolvimento do algoritmo (treinamento e ajuste do modelo <i>ViT</i>)	31
3.2.5	Avaliação dos modelos	32
4	RESULTADOS	36
4.1	Resultados do modelo <i>YOLO</i>	36
4.2	Resultados do modelo <i>Faster R-CNN</i>	39
4.3	Resultados do modelo <i>Vision Transformer</i>	43
4.4	Comparação geral entre os modelos	46
5	CONSIDERAÇÕES FINAIS	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

Os incêndios florestais, que já são uma preocupação constante, têm se intensificado nos últimos anos. A cada evento alarmante, são gerados cada vez mais impactos severos das secas extremas, principalmente na Amazônia, como as de 2005, 2010 e 2015/2016, apresentadas por Souza *et al.* (2023) que resultaram na redução drástica dos níveis dos rios e na degradação da biodiversidade. Além disso, o desmatamento tem aumentado a vulnerabilidade da floresta no pós-incidente, dificultando sua recuperação e, conseqüentemente, tornando-a mais suscetível a eventos climáticos extremos (Souza *et al.*, 2023). Esses incêndios não apenas devastam imensas áreas de vegetação nativa, como também causam impactos relevantes na fauna, deslocando e matando animais, e ameaçam comunidades inteiras, que precisam lidar com a deterioração da qualidade do ar e com os riscos à saúde. Este trabalho de conclusão propõe entender e explorar melhor o uso da aplicação de visão computacional para identificar áreas que podem estar sofrendo queimadas ou incêndios.

Ao falarmos de regiões brasileiras, no Centro-Oeste o cenário não é menos alarmante. Em 2020, um aumento de 41% nos focos de incêndio foi registrado no Pantanal Norte, reflexo direto da seca prolongada e do uso do fogo para manejo de pastagens. Esse tipo de prática tem gerado danos irreparáveis à vegetação nativa, à fauna e aos recursos hídricos (Shimokawa; Mendes; Oliveira-Junior, 2024). Além dos impactos ecológicos, incêndios florestais geram conseqüências sociais significativas; cidades próximas às áreas afetadas têm registrado degradações significativas na qualidade do ar. Na Amazônia, por exemplo, a fumaça das queimadas contém micropartículas inaláveis em níveis muito acima do “seguro”, agravando doenças respiratórias como a COVID-19, especialmente em crianças e idosos (Fearnside, 2024a), contudo, a comunidade médica já está alerta na correlação entre o aumento das queimadas e o surgimento de problemas pulmonares (Maia, 2024).

Conforme o Ministério da Ciência, Tecnologia e Inovação do Brasil (MCTI) publicou em seu relatório ¹, as projeções climáticas para o Brasil indicam que secas prolongadas e ondas de calor podem se intensificar devido às mudanças climáticas. Estima-se que a temperatura em algumas regiões brasileiras suba em 4 °C até o final do século, aumentando ainda mais a frequência de queimadas. Além de previsões, o órgão nos informa que o Brasil é responsável por aproximadamente 2,91% das emissões de gases de efeito estufa no contexto global; sendo os incêndios florestais um dos grandes contribuintes e que esses eventos têm implicações diretas na economia, com perdas estimadas em R\$ 3,6 bilhões anuais devido a desastres climáticos.

¹ Mudança do clima no Brasil. 2024. 106 f. Ministério da Ciência, Tecnologia e Inovação do Brasil, Brasília, 2024. Disponível em: <https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/cgcl/arquivos/Relatorio_Mudanca_Clima_Brasil.pdf>. Acesso em: 04 dez. 2024.

Por meio do Portal Terrabrasilis ², plataforma de dados geográficos administrada pelo INPE, pode ser acessado o Portal de Queimadas, que fornece uma tabela com o índice de focos de incêndio registrado em 2024, comparando os países da América do Sul. Conforme a Figura 1 da seção “Fundamentação Teórica”, no resultado da pesquisa observa-se que o Brasil está em primeiro colocado nesse ranking, mais do que o dobro da Bolívia (recebendo a segunda colocação da lista).

A tecnologia tem desempenhado um papel cada vez mais relevante para reduzir a incidência de queimadas. Com o avanço de drones, satélites e modelos de visão computacional, é possível identificar e monitorar em tempo real, além de prever condições climáticas que favoreçam o surgimento de novos incêndios. Visando a prevenção, os recursos evoluíram em ambientes urbanos e industriais. A SMH Sistemas (empresa que atua há 30 anos no ramo de engenharia de combate a incêndios) apresenta em sua matéria ³, tecnologias utilizadas em *data centers*, onde sistemas inteligentes de detecção de fumaça e fogo, bem como soluções automáticas de supressão de incêndios, protegem infraestruturas críticas. Esses avanços tecnológicos oferecem uma atuação integrada e proativa na prevenção e controle de incêndios, complementando as iniciativas tradicionais e ampliando o alcance das medidas de combate. Em conjunto com políticas públicas e a conscientização da sociedade, a aplicação de soluções tecno-científicas é um passo importante na luta para reduzir a incidência de queimadas e minimizar seus efeitos devastadores sobre o meio ambiente e as populações afetadas.

O avanço de recursos tem desempenhado um papel fundamental no que tange ao meio ambiente e sustentabilidade, visando, principalmente, a prevenção e o combate a incêndios florestais. Analisando esse cenário e tendo grandes preocupações com o futuro, surge a seguinte questão de pesquisa que guia este trabalho: **“Como a visão computacional pode ser aplicada para identificar áreas com focos ativos de incêndios?”**. Este trabalho será realizado com base na necessidade de soluções tecnológicas que mitiguem os impactos do fogo na natureza, contribuindo para a preservação da fauna, flora e saúde pública, residindo na integração entre tecnologias emergentes e a conservação ambiental.

A proposta deste estudo está alinhada com esses avanços globais, com o objetivo de desenvolver e implementar uma solução tecnológica baseada em modelos de visão computacional que permita identificar incêndios com ou sem fumaça em regiões de mata, que posteriormente pode ser embarcada em drones.

² Programa Queimadas do INPE. Instituto Nacional de Pesquisas Espaciais, São Paulo. Disponível em: <<https://terrabrasilis.dpi.inpe.br/queimadas/portal/>>. Acesso em: 29 de nov. de 2024.

³ As Tecnologias para Combate a Incêndio mais utilizadas na Proteção de Data Centers e Ambientes de Missão Crítica. SMH Sistemas. Disponível em: <<https://www.smh.com.br/protecao-contra-incendio-para-data-center/>>. Acesso em: 10 de out. de 2024.

1.1 Justificativa

Os incêndios florestais representam um desafio crescente, especialmente no Brasil, onde eventos dessa natureza têm aumentado significativamente nos últimos anos (Felix, 2024). Esses desastres acarretam consequências graves para o meio ambiente, como a destruição de fauna e flora, além de impactos diretos na qualidade do ar, que afetam a saúde humana (Fidalgo; Fernandes, 2023). Embora existam soluções tecnológicas, como as apresentadas por Moura (2019), como drones equipados com câmeras térmicas e bolas extintoras, sua aplicação ainda é limitada e pouco difundida, o que evidencia a necessidade de explorar novas abordagens no setor ambiental (Moura, 2019).

Se o problema permanecer sem atenção, a frequência e a gravidade dos incêndios florestais tendem a se intensificar, resultando em secas prolongadas, perdas de biodiversidade e riscos à saúde das populações expostas (Fidalgo; Fernandes, 2023). Essas consequências, a curto e longo prazo, reforçam a urgência de desenvolver estratégias preventivas mais eficazes e acessíveis.

O presente trabalho propõe o uso de visão computacional como uma solução para monitorar áreas com risco e ocorrências de incêndios florestais. Visão computacional é uma ciência recente. Milano e Honorato (2010) definem esse recurso como a forma como um computador “enxerga” o meio à sua volta a partir de imagens ou vídeos capturados por câmeras, sensores e outros dispositivos (Milano; Honorato, 2010). O olho humano consegue perceber e interpretar objetos em uma imagem de forma muito rápida. Isso acontece dentro do córtex visual do cérebro, uma das partes mais complexas no sistema de processamento e, se baseando nisso, cientistas concentram seus estudos nessa parte para trazer ideias para a área de visão computacional, sendo o que pesquisadores do Massachusetts Institute of Technology (MIT) definem como “ensinar computadores a enxergarem como humano” (Milano; Honorato, 2010).

Projetadas especialmente para o uso de dados bidimensionais, o projeto fará uso de *Convolutional Neural Network* - Redes Neurais Convolucionais (CNN), redes neurais com camadas que processam e reconhecem dados estruturados em grade. Os dados são representados por diferentes “atributos” internos e, independentemente da dimensionalidade da entrada ou do tipo dos dados (imagem, som, vídeo ou até uma coleção desordenada de recursos). A representação pode sempre ser achatada em um vetor. No entanto, as características específicas que definem cada um dos atributos não são consideradas na transformação. Contudo, é possível explorar essas estruturas com o uso de Redes Neurais Convolucionais. A convolução discreta realizada é a transformação linear que preserva as noções de ordenação (Ferreira, 2021).

O uso de CNN visa não apenas a detecção precoce, mas também a prevenção desses eventos, contribuindo diretamente para a preservação ambiental. Além

disso, a aplicação prática do projeto tem potencial para ser integrada em sistemas de monitoramento em tempo real, beneficiando tanto organizações ambientais quanto a sociedade como um todo. Trazendo a ideia prática e viável de prevenção, este trabalho mostra o impacto positivo que a tecnologia pode ter no combate aos incêndios florestais, promovendo melhorias indiretas na qualidade do ar, preservação da biodiversidade e a redução dos danos causados ao meio ambiente e às populações humanas.

1.2 Definição do Problema

Conforme Fidalgo e Fernandes (2023), o crescente aumento das ocorrências de incêndios no Brasil tem gerado sérias consequências ambientais, sociais e econômicas, colocando em risco a biodiversidade, a qualidade do ar e a vida de inúmeras espécies, incluindo a humana. Em 2019, por exemplo, o Pantanal registrou o maior número de focos de incêndio da década, e, em 2020, observou-se um aumento de 41% dos focos concentrados na parte norte, quando comparado ao ano anterior (Shimokawa; Mendes; Oliveira-Junior, 2024). Esses números evidenciam a urgência de soluções tecnológicas eficazes para enfrentar o problema.

A questão principal que norteia este estudo é: como a visão computacional pode ser aplicada na identificação de áreas com ocorrência de queimadas? Para isso, a pesquisa propõe uma solução baseada em *machine learning*, que, ao treinar diferentes modelos de visão computacional e compará-los entre si, possibilitará a identificação de áreas em que incêndios estejam ocorrendo. Assim, espera-se fornecer uma ferramenta eficiente que auxilie profissionais especializados a tomarem decisões rápidas e assertivas, reduzindo o impacto ambiental e preservando o ecossistema e a biodiversidade brasileira.

Com isso, surge a dúvida: “Quais são as técnicas e abordagens necessárias para desenvolver um modelo de *machine learning* baseado em visão computacional que permita identificar e monitorar, em tempo real, padrões em regiões com diferentes climas, vegetações e relevos, com o objetivo de detectar focos ativos de incêndio com ou sem fumaça em regiões florestais?”

1.3 Objetivo Geral

O objetivo geral deste trabalho é avaliar diferentes modelos de *machine learning* baseados em visão computacional para identificar padrões em regiões com diferentes climas, vegetações e relevos, visando detectar incêndios com ou sem fumaça em regiões florestais.

1.4 Objetivos específicos

Os objetivos específicos deste trabalho são:

- a) Coletar *datasets* públicos com imagens de áreas com focos ativos de incêndios com e sem fumaça, em diferentes estágios e vegetações;
- b) Treinar e ajustar modelos de visão computacional para realizar a detecção de fogo e fumaça;
- c) Analisar e comparar métricas resultantes do treinamento entre os diferentes modelos;
- d) Testar os modelos com imagens.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os principais conceitos, teorias e debates acadêmicos relevantes ao tema. O estudo consiste na literatura especializada e será dividido entre temas, permitindo uma maior compreensão dos eventos e estabelecer as conexões necessárias para a condução da pesquisa. Este capítulo apresentará conceitos sobre incêndios e seus impactos, visão computacional e *machine learning* e o uso de tecnologias em aeronaves remotamente pilotadas.

2.1 Incêndios florestais e seus impactos

Ao pensarmos em incêndio, pensamos em fogo, em ardência, em calor e até mesmo em combustível. Para compreender de forma mais clara esta seção, Silva *et al.* (2018) nos apresentam suas definições: “combustão” é uma ação que libera calor como consequência da mistura de uma substância combustível com um oxidante (substância que se mistura com oxigênio). “Fogo” é um exemplo de processo de combustão acompanhado por fumaça, chama ou ambos. Já “incêndio” é a combustão rápida do fogo que se alastra de forma descontrolada, ou seja, basicamente é o fogo fora do controle (Silva *et al.*, 2018).

De acordo com Valencia-Zuleta *et al.* (2024), os incêndios de grandes proporções representam uma ameaça crescente à biodiversidade, afetando espécies aquáticas, semiaquáticas e fossoriais, como anuros, crocodilianos e quelônios. Esses eventos de combinação de seca e fogo destroem habitats importantes, aumentando a vulnerabilidade de diversas espécies e dificultando sua detecção durante situações de emergência (Valencia-Zuleta *et al.*, 2024). Incêndios florestais representam um desafio ambiental e social significativo, com consequências para a biodiversidade, o clima e as comunidades locais. Fearnside (2024b) aborda os impactos das queimadas na Amazônia, ressaltando a conexão entre desmatamento e a propagação da queimada. Ele destaca que os incêndios resultam em perdas significativas de carbono do solo, além de liberarem grandes quantidades de gás carbônico na atmosfera, contribuindo para o aquecimento global. Esses eventos são principalmente de origem antropogênica, tendo seus pontos de ignição gerados pela expansão de atividades humanas, como a pecuária e a exploração madeireira (Fearnside, 2024b).

Fearnside (2024b) alega também que os incêndios geram níveis elevados de micropartículas PM2.5 (inaláveis), que ultrapassam os limites recomendados pela Organização Mundial da Saúde (OMS), prejudicando a saúde respiratória das populações locais (Fearnside, 2024b). Considerando que os eventos tenham, principalmente origem humana, Shimokawa, Mendes e Oliveira-Junior (2024) nos traz que o fogo é utilizado como uma ferramenta de manejo, contudo, causa sérios riscos ambientais;

mais de 50% dos focos de incêndio ocorreram em áreas de pastagem, devido à frequência de queimadas propositalis para facilitar o manejo do gado (Shimokawa; Mendes; Oliveira-Junior, 2024). Além da Amazônia, outra região afetada é o Pantanal, com um aumento de 41% dos focos de incêndio em 2020 em comparação ao ano anterior, sendo um reflexo da seca prolongada e do uso do fogo proposital para limpeza de pastos (Shimokawa; Mendes; Oliveira-Junior, 2024).

Alegando também que as queimadas no Brasil são majoritariamente causadas por atividades humanas, como o manejo inadequado do solo, Rodrigues (2024) nos argumenta que a educação ambiental e o treinamento de comunidades para práticas sustentáveis podem reduzir significativamente esses eventos (Rodrigues, 2024). Já Pimentel, Costa e Cunha (2024), esclarecem que as secas afetam 81,3% das áreas rurais e 50% das áreas urbanas, referindo-se às atividades de serviço como: agricultura, ecossistemas e abastecimento de água (Pimentel; Costa; Cunha, 2024).

Explorando a plataforma Terrabrasilis, site que nos informa dados geográficos (administrado pelo MCTI), podemos ter acesso ao Programa de Queimadas, portal que apresenta informações atualizadas sobre incêndios e queimadas, de diferentes formas em diferentes regiões. Citado anteriormente (na seção Introdução), conforme a Figura 1 o resultado de uma das pesquisas nos mostra que o Brasil está em primeiro lugar na lista dos países da América do sul com o maior número de focos de incêndio registrados no ano de 2024 (com 268.362 focos); mais que o dobro do segundo colocado no ranking (Bolívia, com 89.729 focos registrados).

Figura 1 – Focos de incêndio entre os países da América do Sul em 2024.

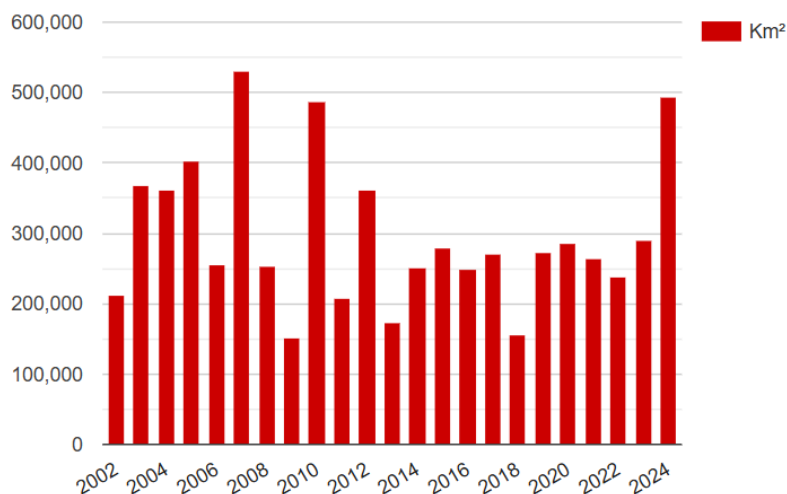
Brasil	268362
Bolivia	89729
Venezuela	42071
Argentina	31158
Paraguai	22581
Colombia	16521
Perú	16329
Equador	3350
Guyana	3142
Chile	2707
Suriname	1063
Guyana Francesa	272
Uruguai	271

Fonte: Ministério da Ciência, Tecnologia e Inovação (2024).

Analisando agora a Figura 2, podemos ressaltar que em 2024, o Brasil teve quase 500.000 km² de sua área queimada e que por pouco, situa-se atrás apenas do ano de 2007.

E por fim, as Figuras 3 e 4, mostram que estados com biomas do tipo Amazônia, Caatinga e Cerrado se tornam mais vulneráveis a terem focos de incêndio,

Figura 2 – Gráfico sobre áreas queimadas no Brasil, com o passar dos anos.



Fonte: Ministério da Ciência, Tecnologia e Inovação (2024).

devido à grande atividade agropecuária e a fatores climáticos, como umidade e clima das localizações.

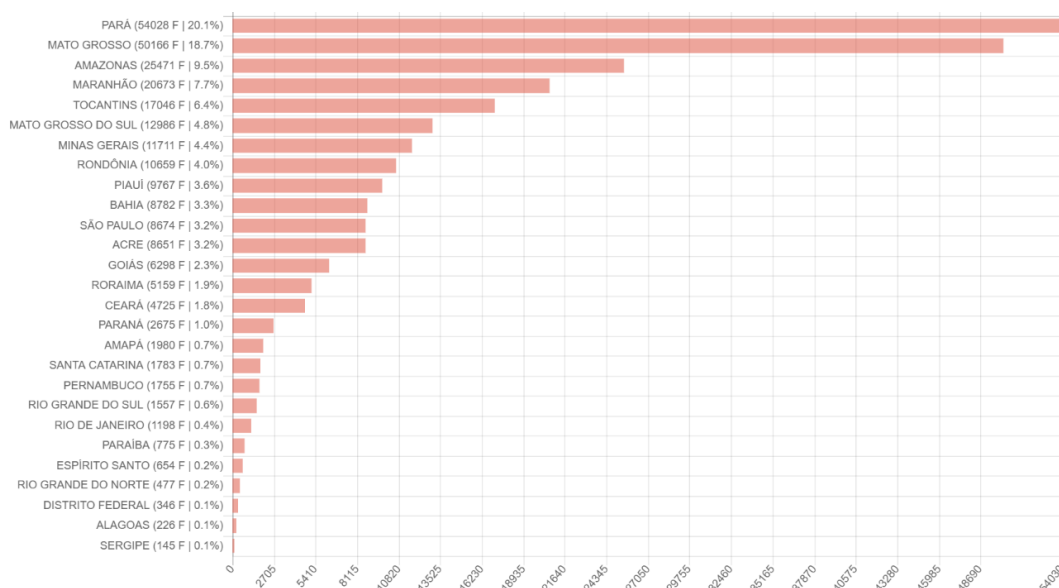
Figura 3 – Área brasileira queimada conforme o bioma.

Ano	Amazônia	Caatinga	Cerrado	Mata Atlântica	Pampa	Pantanal	Total anual
2002	51.562	33.660	104.359	7.212	27	14.398	211.218
2003	94.791	51.233	171.439	40.613	2.477	7.410	367.963
2004	123.530	14.333	190.070	15.191	1.569	17.547	362.240
2005	142.157	37.665	174.898	19.867	1.128	27.229	402.944
2006	84.136	19.433	114.725	29.413	1.853	6.874	256.434
2007	138.766	31.857	316.310	25.393	1.024	18.288	531.638
2008	61.640	27.020	130.430	23.157	1.425	9.556	253.228
2009	37.205	14.631	70.391	13.274	2.169	12.970	150.640
2010	106.960	39.864	301.429	22.495	781	15.730	487.259
2011	32.366	17.396	133.603	19.274	816	4.318	207.773
2012	52.445	37.013	243.642	14.829	1.954	12.844	362.727
2013	30.220	18.421	108.293	10.426	743	5.229	173.332
2014	47.659	19.000	166.786	14.089	741	1.992	250.267
2015	62.021	28.101	171.521	10.890	1.349	6.604	280.486
2016	55.092	19.547	143.684	18.229	1.509	10.835	248.896
2017	69.469	18.758	154.601	16.010	1.571	9.635	270.044
2018	36.929	18.164	83.782	13.002	602	2.862	155.341
2019	64.993	31.271	140.662	17.331	1.369	18.137	273.763
2020	73.364	20.843	135.474	14.970	3.077	37.779	285.507
2021	45.245	36.013	142.529	21.149	1.184	18.799	264.919
2022	71.860	30.165	122.887	9.290	955	2.360	237.517
2023	68.122	63.968	137.597	13.581	1.174	4.859	289.301
2024	128.112	65.057	228.474	45.115	271	26.855	493.884

Fonte: Ministério da Ciência, Tecnologia e Inovação (2024).

A fim de mitigar esses efeitos, Souza, Vale e Chaves (2024) nos trazem ideias de proteção contra tais eventos, como o uso de plantas de baixa inflamabilidade para criação de uma “barreira verde”, dificultando a propagação do fogo (Souza; Vale; Chaves, 2024). Existe também a proposta de Moraes (2004) que aborda o uso de

Figura 4 – Gráfico sobre focos de incêndio no Brasil, dividido por estados.



Fonte: Ministério da Ciência, Tecnologia e Inovação (2024).

tecnologias para combater incêndios florestais, destacando avanços em monitoramento, prevenção e resposta rápida. Ele cita o uso de *Big Data* como integração de dados para análise e tomada de decisões estratégicas e *machine learning* para aplicação de algoritmos para prever e classificar regiões vulneráveis a incêndios.

2.2 Visão computacional e *Machine learning*

Tratando-se do ramo de Inteligência Artificial (IA), Braga *et al.* (2019) apresentam como uma ciência da computação que imita os processos de pensamento humano, a capacidade de aprendizagem e o armazenamento de conhecimento. Essa tecnologia, quando comparada a um humano, foi capaz de demonstrar uma gama maior de informações diante da combinação de previsões feitas por seres humanos com as previsões de um algoritmo de aprendizado de máquina (*machine learning*) (Braga *et al.*, 2019). Existem diversas abordagens de aprendizado e cada categoria possui seus próprios algoritmos de treinamento e modelos resultantes. Para treinamento de modelos, são utilizados os algoritmos de *machine learning*.

Segundo Calanca, Matheus e Raphael (2023), *machine learning* permite que os computadores aprendam e melhorem seu desempenho através de um conjunto de dados. Para esse ensinamento são utilizadas técnicas como:

- a) **Aprendizado supervisionado:** computadores são treinados utilizando exemplos de entrada e saída correspondente. Esse tipo de aprendizado é utilizado principalmente como modelo de classificação, onde com ele, é possível identificar padrões associados a diferentes categorias;
- b) **Aprendizado não supervisionado:** neste aprendizado os computadores

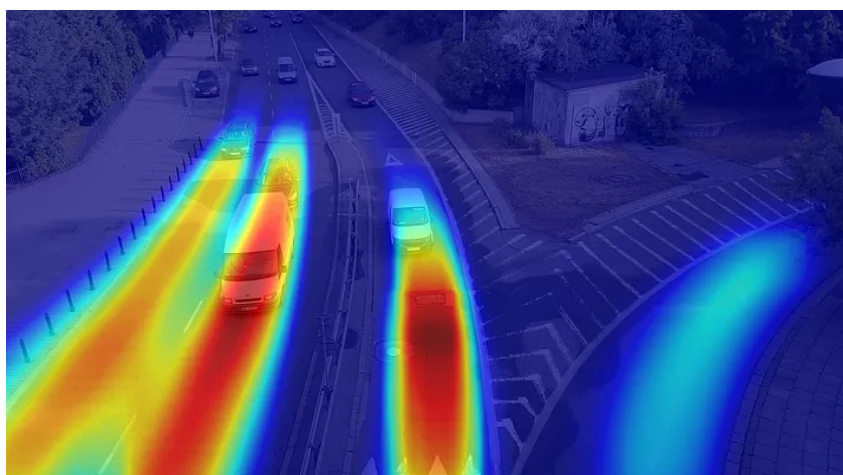
exploram dados não rotulados para descobrir padrões e estruturas ocultas. Uma técnica comum para este tipo de aprendizado é o agrupamento, processo de juntar e categorizar grupos de dados;

- c) Aprendizado semi-supervisionado: aqui temos um exemplo onde parte dos dados é rotulada e outra parte é não rotulada. Esses modelos analisam dados rotulados, identificando padrões e relacionamentos e, em seguida, aplicam esses padrões aos dados não rotulados, tornando possível assim, o sistema fazer previsões ou recomendações, por exemplo;
- d) Aprendizado por reforço: neste modelo, um sistema é treinado para tomar sequência de decisões, recebendo recompensas ou penalidades em troca, visando que a aplicação aprenda a tomar as melhores ações para maximizar as recompensas ao longo do tempo;

Esses são exemplos de treinamento de modelos de *machine learning* em que há interferência humana envolvida, porém uma IA vai além disso, sendo capaz de agir e aprender mais com base no que foi aprendido por ela mesma anteriormente (Calanca; Matheus; Raphaell, 2023). Como exemplos de tarefas de mineração de dados que são treinadas por meio de aprendizado supervisionado, destacam-se a classificação e a regressão. A classificação tem como objetivo prever categorias discretas, rotulando os dados com base em exemplos previamente identificados, ou seja, é usada para identificar e separar os dados em grupos definidos. Já a regressão busca modelar a relação entre variáveis independentes e uma variável dependente contínua, ou seja, é aplicada para prever um valor numérico, analisando como diferentes fatores influenciam esse resultado, sendo amplamente aplicada em contextos como a previsão de receitas de vendas (Monard; Baranauskas, 2003).

Neste estudo, o foco é em uma específica área de IA que aplica técnicas de *machine learning*, a visão computacional utilizando o modelo de aprendizado supervisionado para classificação. Desde seu surgimento, a visão computacional tinha o objetivo de criar “olhos” para as máquinas, possibilitando que sistemas automáticos interpretem imagens e padrões visuais (Milano; Honorato, 2010). Nagao (2024) nos mostra que as aplicações já utilizadas deste instrumento incluem:

- a) Mapas de calor: identificar movimentos e alocar cores relacionadas ao volume de tráfego para diferentes áreas, ajudando a definir e testar novas estratégias de *layouts* (Figura 5);
- b) Espelhos virtuais e mecanismos de recomendação: para experiências de clientes, espelhos virtuais (equipados com câmera de visão computacional e tecnologia de realidade aumentada) podem se tornar o foco central da personalização e melhoria da experiência, sendo possível exibir uma

Figura 5 – Rastreamento de tráfego com visão computacional.

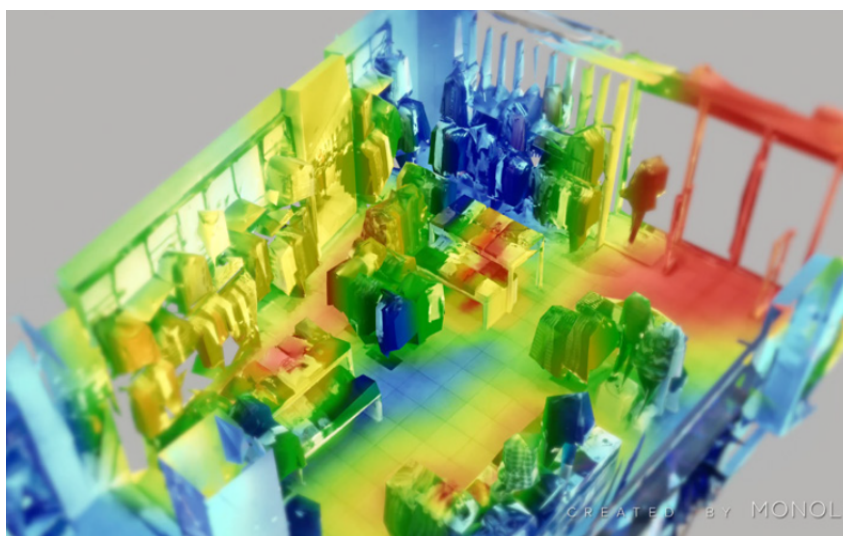
Fonte: Dijinga (2024).

ampla gama de informações contextuais, melhorando a conexão com os consumidores (Figura 6);

Figura 6 – Espelho virtual utilizando visão computacional.

Fonte: Voutik (2018).

- c) Análise de dados: em lojas, os sistemas detectam padrões em tráfegos para identificar setores e itens mais acessados, fazendo assim, com que seja possível impulsionar de forma mais rápida campanhas de *marketing* e mensagens personalizadas (Figura 7);
- d) Gestão de estoque: em estabelecimentos que necessitam de um controle de seus itens, é possível automatizar a contagem de ciclo de inventário utilizando visão computacional (Figura 8);
- e) Prevenção de perdas: observando comportamento de pessoas para identificar atitudes suspeitas relacionadas à fraude ou roubo (Figura 9);

Figura 7 – Rastreamento de produtos utilizando visão computacional.

Fonte: Koptelov (2022).

Figura 8 – Gestão de estoque com visão computacional.

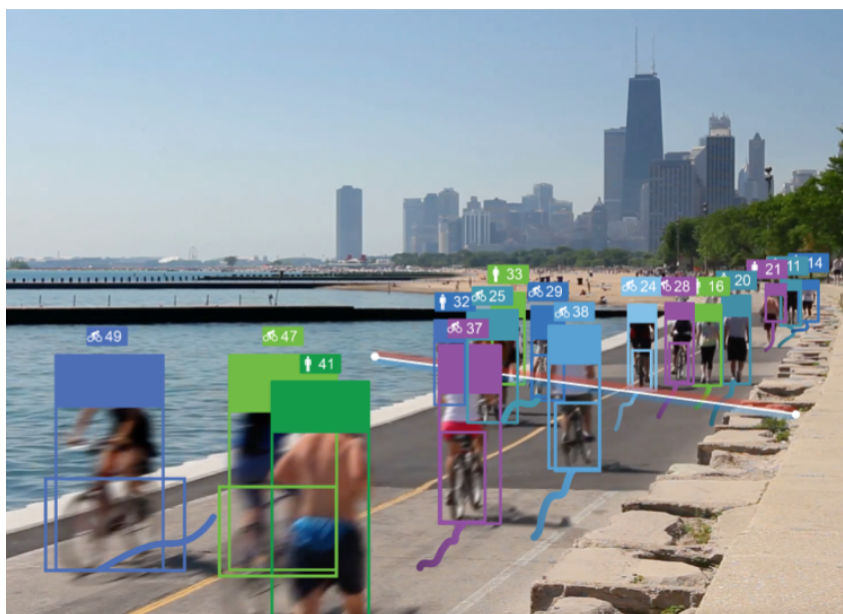
Fonte: Toshniwal (2023).

- f) Análises de multidão: rastreando jornadas de pessoas por um determinado local, além de poder calcular o tempo gasto para determinada ação e garantir que os estabelecimentos ou pessoas sigam protocolos padronizados (Figura 10);
- g) Planogramas: analisando determinado objeto e verificar seu posicionamento, determinando se estão ou não fora de ordem (Figura 11).

Fora as apresentadas, existem mais tarefas que utilizam essa tecnologia (como reconhecimento facial de celulares no dia a dia). A cada novo uso que surge para esse recurso, o departamento responsável pela implementação se revoluciona, permitindo a reunião de *insights* valiosos e a otimização de operações (Nagao, 2024). A aplicação de visão computacional em drones permite a detecção rápida e precisa

Figura 9 – Prevenção de roubos com visão computacional.

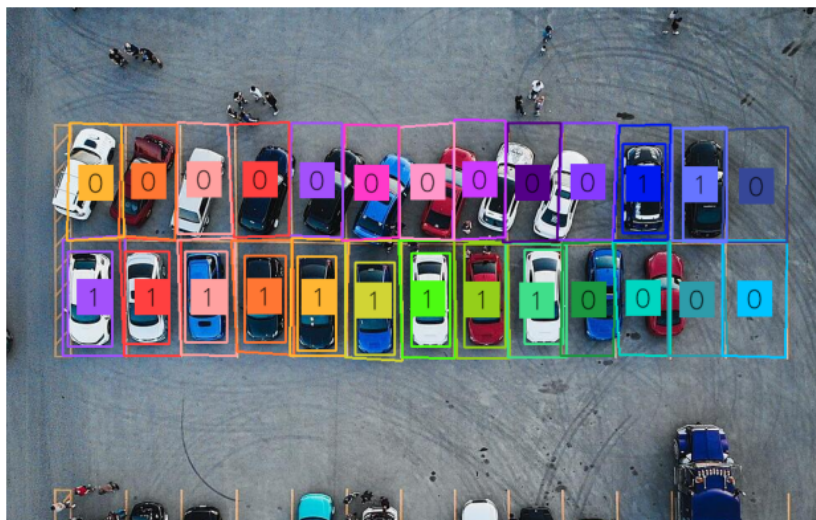
Fonte: Vina (2024).

Figura 10 – Contador de pessoas utilizando visão computacional.

Fonte: VisualCortex (2025).

de incêndios, mesmo em condições adversas. Combinados a algoritmos de *machine learning*, esses sistemas podem aprender a identificar padrões característicos de fumaça e fogo, otimizando as operações de monitoramento e combate (Machado, 2019).

Conforme O'shea e Nash (2015), as Redes Neurais Convolucionais (CNNs) são um tipo de rede neural projetada para processar dados estruturados como imagens, sendo amplamente utilizadas em visão computacional. Elas se destacam por sua capacidade de capturar padrões espaciais e relacionamentos entre pixels, tornando-se ideais para tarefas como reconhecimento facial, diagnóstico médico, sistemas de

Figura 11 – Alinhamento de estacionamento utilizando visão computacional.

Fonte: Lynn (2025).

segurança e classificação de imagens. Voulodimos *et al.* (2018) defendem a ideia de que CNNs superam métodos tradicionais (que dependem da extração manual de características) em tarefas como detecção de objetos e reconhecimento de padrões. Eles nos apontam exemplos de *datasets* utilizados para treinamento e validação de modelos, como o *MNIST* (conjunto de imagens em tons de cinza de dígitos manuscritos), o *CIFAR* (imagens de objetos naturais em cores), o *COIL* (conjunto de imagens 3D de objetos rotacionados) e o *Youtube-8M* (banco massivo de vídeos rotulados com categorias) (Voulodimos *et al.*, 2018).

Como todo treinamento, existem desafios no caminho, problemas como *overfitting* (quando o modelo “memoriza” o conjunto usado no treinamento, prejudicando sua capacidade de generalizar). Contudo, técnicas como regularização por *dropout* (técnica que desativa aleatoriamente conexões entre neurônios durante o treinamento, reduzindo dependências específicas) podem ser utilizadas para contornar o empecilho (Voulodimos *et al.*, 2018).

Conforme Moumgiakmas, Samatas e Papakostas (2021), alguns dos modelos de algoritmos populares de visão computacional incluem o *YOLO* (*You Only Look Once*), um sistema de detecção de objeto capaz de prever classes e classificar objetos em apenas uma única passagem da imagem) (Redmon *et al.*, 2016); o *Faster R-CNN*, um modelo que alinha uma rede *Region Proposal Network* - Rede de Propostas Regionais (RPN), tecnologia que prevê simultaneamente limites de objetos e pontuações de objetividade em cada posição), com uma rede neural profunda utilizada para processamento de dado em forma de grade (Ren *et al.*, 2015); e o *VGG-16*, um algoritmo baseado em estruturas de redes neurais profundas, contando com 16 camadas (Simonyan; Zisserman, 2014). Moumgiakmas, Samatas e Papakostas (2021) também

citam que classes de redes neurais como a CNN possuem precisões com valores acima de 90% de acerto em cenários perfeitos (*dataset* bem balanceado, escolha de parâmetros e tipo do modelo escolhido) (Moumgiakmas; Samatas; Papakostas, 2021).

2.3 Modelos de detecção de objetos

Jiang *et al.* (2022) apresentam o *YOLO* como um modelo de detecção de objetos de única etapa que utiliza regressão para prever “caixas” delimitadoras, fazendo com que ele seja altamente eficiente para grandes conjuntos de dados. Sua popularidade tem crescido devido à sua capacidade de oferecer alta velocidade e precisão, sendo uma alternativa viável para aplicações em tempo real. Além disso, segundo Jiang *et al.* (2022), o *YOLO* é amplamente utilizado em monitoramento ambiental, reconhecimento de placas de veículos para segurança pública e na agricultura para identificação de pragas e colheitas. No futuro, espera-se que o modelo continue sendo aprimorado para alcançar maior precisão e que continue sendo integrado a dispositivos embarcados, permitindo análises ainda mais rápidas e eficientes em tempo real.

No site oficial da *Ultralytics*¹, empresa responsável pelo modelo, é possível encontrar a documentação que define hiperparâmetros e configurações que afetam o desempenho, a velocidade e a precisão do treinamento do modelo. Alguns dos parâmetros incluem:

- a) *epochs*: número total de passagem completa por todo conjunto de dados;
- b) *batch*: quantidade de imagens que a rede processa de uma vez antes de atualizar os pesos durante o treinamento;
- c) *patience*: número de épocas sem melhoria nas métricas de validação antes de parar o treino antecipadamente;
- d) *imgsz*: tamanho da imagem alvo para treino;
- e) *optimizer*: escolha do otimizador para ajustar os pesos da formação;
- f) *lr0*: taxa de aprendizagem inicial;
- g) *weight_decay*: termo de regularização L2, que penaliza os pesos grandes para evitar o sobreajuste.

Conforme evidenciado no estudo de Neto *et al.* (2024), não são somente os parâmetros que influenciam no resultado do modelo. O número de imagens utilizadas no treinamento exerce a maior influência sobre a performance final do *YOLO*, seguido pela quantidade de épocas e, por último, pela taxa de aprendizado. O autor revelou que quanto maior o número de imagens, melhor a rede aprende os padrões e variações das classes; e que mais épocas permitem o refinamento dos pesos. Porém, sem dados

¹ <https://www.ultralytics.com/pt/yolo>

suficientes o modelo tende a memorizar o treino (*overfitting*). Número de imagens e número de épocas estão diretamente relacionados: mais imagens permitem mais épocas sem *overfitting*. Já a taxa de aprendizado precisa ser ajustada de acordo com o volume de dados e épocas: poucos dados funcionam com taxa de aprendizado menor e menos épocas e muitos dados funcionam com taxa de aprendizado mais alta e mais épocas. A interação entre esses parâmetros mostra que a ampliação do *dataset* possibilita a utilização de taxas de aprendizado mais elevadas e um maior número de épocas sem comprometer a capacidade de generalização do modelo. Assim, o planejamento adequado desses parâmetros é essencial para garantir a estabilidade do treinamento e evitar problemas como *overfitting*.

A Tabela 1 e a Tabela 2 resumem as principais recomendações mostradas no estudo de Neto *et al.* (2024), citando como os parâmetros interagem entre si e impactam a performance dos modelos.

Tabela 1 – Influência dos parâmetros no desempenho do modelo.

Parâmetro	Recomendação	Observações
Nº de imagens	Quanto mais, melhor	Aprendizado mais completo dos padrões e variações; permite mais épocas sem <i>overfitting</i> .
Nº de épocas	Aumentar com mais dados	Refinamento dos pesos; com poucos dados pode causar <i>overfitting</i> .
Taxa de aprendizado	Ajustar conforme dados e épocas	Poucos dados exigem taxa menor e menos épocas; muitos dados permitem taxa maior e mais épocas.

Fonte: O autor.

Tabela 2 – Relações entre os parâmetros no desempenho do modelo.

Parâmetros	Tipo de Relação	Observações
Imagens e épocas	Direta	Mais imagens permitem mais épocas sem <i>overfitting</i> .
Taxa e dados	Proporcional ao volume de dados	Volume maior de dados possibilita taxas de aprendizado mais altas.
Ajuste conjunto	Planejamento integrado dos parâmetros	Garante estabilidade no treinamento e reduz risco de <i>overfitting</i> .

Fonte: O autor.

Como base de comparação, Patki (2024) traz a base para o entendimento de outros modelos. O autor apresenta o *Faster R-CNN* como uma solução precisa e poderosa para a detecção de objetos. No artigo, o modelo é exposto como uma arquitetura de duas etapas, onde inicialmente uma RPN gera propostas de regiões de interesse a partir de um mapa de características extraído por uma rede *backbone* (rede responsável por extrair as características da imagem), e posteriormente, essas regiões são refinadas e classificadas por meio do *Fast R-CNN detection head*, a “cabeça” por trás do modelo. Um dos destaques do modelo é a utilização do *RoI Pooling*, técnica responsável por selecionar as regiões de interesse e transformar seus mapas de características em um tamanho fixo, independentemente do tamanho da região original. Assim, é permitido que as áreas mapeadas sejam processadas por camadas totalmente conectadas para a previsão da classe e a regressão das coordenadas dos objetos detectados. Conforme Patki (2024) o ponto forte do *Faster R-CNN* ocorre principalmente em cenários com maior rigor de precisão.

Sitaula e Hossain (2021) apresentam o modelo *VGG-16*, uma rede convolucional profunda amplamente utilizada para extração de características em tarefas de classificação e detecção de imagens. Conforme comentado, o *VGG-16* é conhecido por sua arquitetura composta por blocos de convolução com filtros pequenos (3×3) e camadas de *pooling* (operação feita para diminuir o tamanho dos mapas de características e evitar sobreajuste), organizados de forma hierárquica para capturar desde informações mais simples até padrões mais complexos.

No estudo de Ali *et al.* (2024), os modelos *YOLO* e *VGG-16* são combinados entre si, cada um atuando em uma etapa distinta do processo: enquanto o *YOLO* é responsável pela detecção de objeto nas imagens, localizando e delimitando as regiões, o *VGG-16* atua posteriormente na classificação das detecções anteriormente realizadas. Ali *et al.* (2024) mostram que essa divisão de funções é essencial porque o *YOLO* não realiza identificação individual, apenas mostra a localização do objeto, enquanto o *VGG-16*, por sua estrutura de rede convolucional voltada para classificação de imagens, é capaz de diferenciar e reconhecer as características.

Já em estudos como os de Tan *et al.* (2021) e Olorunshola, Jemitola e Ademuwagun (2023), a comparação é feita entre os modelos de detecção. Os principais resultados de comparação foram obtidos analisando métricas como mAP (*Mean Average Precision*) e IoU (*Intersection over Union*). Conforme Wang (2022), mAP é uma medição utilizada para avaliar a precisão de um modelo de detecção de objetos. Ela calcula a média das precisões para todas as classes presentes no conjunto de dados e mede o quão bem o modelo detecta objetos corretamente, ou seja, quanto maior o mAP, melhor o modelo. Um mAP de 80% significa que, em média, o modelo tem 80% de precisão na detecção das classes avaliadas. Já Rezatofighi *et al.* (2019)

cita que IoU mede a sobreposição entre a caixa predita pelo modelo e a caixa real, ou seja, mede a precisão da localização do objeto. Quanto maior o IoU, melhor a detecção do modelo. Um IoU de 0.5 (50%) é um padrão mínimo para considerar uma detecção válida, enquanto 0.75 ou mais indica alta precisão.

A comparação realizada por Tan *et al.* (2021) destaca um equilíbrio entre precisão e velocidade, fatores essenciais para aplicações práticas. O estudo revelou que o *Faster R-CNN* alcançou a maior precisão (mAP de 87,69%), enquanto o *YOLO* foi significativamente mais rápido (51 FPS) e ainda manteve uma precisão considerável (mAP de 80,17%). Para aplicações em tempo real, o *YOLO* se mostra mais vantajoso devido à sua alta velocidade, sendo também mais eficiente para execução em *hardware* menos potente, como dispositivos móveis e drones. Por outro lado, Tan *et al.* (2021) cita que o *Faster R-CNN*, apesar de oferecer maior precisão, tem um alto custo computacional e baixa velocidade (7 FPS), o que o torna inviável para cenários que exigem resposta rápida. Em amostras mais difíceis, o *YOLO* manteve sua vantagem em velocidade (69 FPS), enquanto o *Faster R-CNN* apresentou um pequeno ganho de precisão (79,63%), mas com uma velocidade extremamente baixa (3 FPS). Dessa forma, para aplicações que exigem rapidez e eficiência computacional, o *YOLO* se destaca como a melhor escolha.

O estudo de Olorunshola, Jemitola e Ademuwagun (2023) destaca uma diferença fundamental entre velocidade e precisão. O *YOLO*, sendo um modelo de uma etapa, é mais rápido e eficiente para tarefas em tempo real, alcançando até 54 FPS em aplicações de monitoramento por drones, com um mAP de 74,36% a um IoU de 50. Por essa razão, é amplamente utilizado para vigilância aérea e aplicações de tráfego. Já o *Faster R-CNN*, por ser um modelo de duas etapas, é mais preciso, atingindo um mAP de 82% em aplicações como reconhecimento facial e diagnósticos médicos, onde a alta precisão é essencial. Olorunshola, Jemitola e Ademuwagun (2023) revisa mais de 30 estudos e conclui que o *YOLO* é a opção mais popular, tornando-se uma escolha interessante para aplicações que exigem velocidade e eficiência computacional. Por outro lado, o *Faster R-CNN*, apesar de mais preciso, pode não ser adequado para aplicações em tempo real devido ao seu maior custo computacional. Assim, a escolha entre os dois depende da necessidade específica: o *YOLO* é mais vantajoso para cenários que exigem alta velocidade, enquanto o *Faster R-CNN* é melhor quando a precisão é prioridade.

Já Fu (2022) cita outro tipo de modelo utilizado na detecção de objetos. O autor escreve que o *Vision Transformer (ViT)* surgiu como uma proposta inovadora para tarefas de visão computacional, adaptando o mecanismo de autoatenção (mecanismo que permite o modelo avaliar a importância de diferentes partes de uma mesma entrada), que foi desenvolvido primeiramente para o Processamento de Linguagem

Natural (NLP). Diferente das CNNs, que utilizam filtros locais para extrair características espaciais de uma imagem, o *ViT* trabalha com relações globais entre as regiões da imagem, isto é, considera a influência de cada parte da imagem sobre todas as demais simultaneamente, sem se limitar apenas a áreas vizinhas ou próximas. Isso permite que um pixel no canto superior esquerdo influencie a interpretação de um objeto no canto inferior direito, se houver relação entre eles, algo que as CNNs não fazem, pois estas olham apenas regiões locais. Isso é possível porque o *ViT* divide a imagem em pequenos *patches* ou fragmentos, transforma cada um em um vetor de *embeddings* (representações numéricas de dados) e aplica um mecanismo de *self-attention*, mecanismo que calcula a importância relativa entre todos os dados, no caso *patches*, simultaneamente, considerando suas interações em toda a imagem. Segundo Fu (2022), essa abordagem permite que os *ViTs* aprendam padrões de longo alcance sem depender de vieses estruturais locais, como as CNNs dependem de convoluções.

Conforme Han *et al.* (2022), ao eliminar a necessidade de operações convolucionais, os *ViTs* passaram a se destacar em diversas aplicações, como classificação de imagens, detecção de objetos, segmentação semântica e reconhecimento em vídeo. Os autores ressaltam que, embora os *ViTs* inicialmente enfrentassem dificuldades para generalizar bem com conjuntos de dados pequenos, essa limitação foi contornada com o uso de pré-treinamento em *datasets* numerosos, como o JFT-300M² e o JFT-3B³. Além disso, os autores destacam que arquiteturas híbridas, que combinam CNNs e *transformers*, também vêm sendo exploradas para unir as vantagens de ambos os paradigmas.

Fu (2022) expõe que a eficiência e a versatilidade dos *ViTs* são evidenciadas por seus resultados em *benchmarks* famosos, como o *ImageNet*. De acordo com Khan *et al.* (2022), o *ViT-L* treinado no gigantesco *dataset* JFT (contendo 300 milhões de imagens) apresentou uma melhora de 13% na acurácia do *ImageNet* em comparação com sua versão treinada apenas no *ImageNet*. Além disso, o estudo aponta soluções para os desafios computacionais dos *transformers*, como o uso de *deformable attention*, *axial attention* e *window attention*, técnicas que viabilizam a aplicação eficiente do *self-attention* em imagens de alta resolução visando a redução do custo quadrático (custo que cresce proporcionalmente ao quadrado do número de elementos de entrada) tradicional dessa operação. Khan *et al.* (2022) cita que outros modelos derivados, como o *Deformable DETR* ou *Swin Transformer*, também confirmam esse avanço, otimizando custo operacional, proporcionando tempos de treinamento até 10× menores e melhor desempenho em detecção de objetos em imagens pesadas sem sacrificar a precisão.

Por fim, segundo Fu (2022), a arquitetura do *Vision Transformer* evoluiu para diferentes especializações e variações. Existem versões como o *PVT* (*Pyramid Vision*

² <https://paperswithcode.com/dataset/jft-300m>

³ <https://paperswithcode.com/dataset/jft-3b>

Transformer), que reduz a complexidade da autoatenção incorporando hierarquias multi-escala, criando representações da imagem em diferentes níveis de resolução, semelhantes às CNNs, além disso, modelos como o *TimeSformer* foram adaptados para vídeo, aplicando autoatenção temporal (relações entre diferentes momentos) e espacial (relações entre diferentes regiões de um *frame*) de forma sequencial. O autor conclui que a diversidade de variantes demonstra que o potencial dos *Vision Transformers* não está apenas em superar as CNNs em tarefas tradicionais como a classificação de imagens, mas também em abrir novas possibilidades para aplicações complexas e multimodais em visão computacional, como o poder de reconhecimento de emoções em vídeos ou a geração de imagens a partir de descrições.

3 PROCEDIMENTOS METODOLÓGICOS

Esta seção descreve os procedimentos metodológicos adotados para desenvolver o trabalho, especificando as etapas, ferramentas e técnicas utilizadas para alcançar os objetivos propostos anteriormente.

3.1 Metodologia de pesquisa

Esta pesquisa procura explorar os fatores associados aos incêndios florestais no Brasil, com foco nas causas, nos impactos ambientais e sociais e nas possíveis soluções tecnológicas para sua detecção e combate. O estudo foi conduzido por meio de uma abordagem exploratória, dividida em etapas metodológicas que garantem a profundidade e a validade científica dos resultados. Para uma abordagem de categorização dos dados, utilizou-se a proposta de Gil (2002), que nos apresenta como principal característica a organização das informações com base em características comuns e com o intuito de identificar padrões.

3.1.1 Tipo de pesquisa

A pesquisa é exploratória, com ênfase qualitativa e quantitativa, buscando identificar padrões. Conforme Gil (2002), a pesquisa exploratória se caracteriza por ter o propósito de permitir ao pesquisador conseguir entender determinado evento, identificar variáveis relevantes e conduzir estudos posteriores mais conclusivos, aprimorando as ideias.

3.1.2 Etapas metodológicas

- a) Etapa 1: revisão bibliográfica;
- b) Etapa 2: coleta de imagens;
- c) Etapa 3: experimentos;
- d) Etapa 4: análise dos resultados;

Na primeira etapa um levantamento bibliográfico foi realizado, abrangendo fontes como artigos científicos em bases de dados reconhecidas, relatórios institucionais, livros e teses relacionados a incêndios florestais, inteligência artificial e impacto ambiental. Os tópicos investigados incluíram temas como Métodos de detecção e monitoramentos de queimadas, métodos de previsão de áreas com riscos de incêndios, tecnologias emergentes contra tais eventos e impactos sociais, econômicos e ambientais.

A segunda etapa teve foco na coleta das imagens. As imagens foram obtidas a partir de fontes secundárias, como bases públicas disponibilizadas na comunidade

técnico-científica. O *dataset* escolhido deveria conter imagens com e sem fogo, com e sem fumaça, visando um treinamento baseado em diferentes eventos, além de imagens térreas e aéreas.

Já na terceira etapa, foram conduzidos experimentos utilizando três arquiteturas de modelos de detecção de objetos: *YOLO*, *Faster R-CNN* e *Vision Transformer*. Foi realizado o *fine-tuning* dos modelos pré-treinados com base no conjunto de dados selecionado. Os treinamentos incluíram o pré-processamento das imagens, organização das anotações e divisão entre dados de treino, validação e teste. Após o treinamento, cada modelo foi avaliado, além da análise da inferência visual dos modelos sobre novas imagens.

Por fim, na quarta etapa, realizou-se a análise dos resultados, por meio de uma abordagem qualitativa e quantitativa. A abordagem qualitativa ocorreu na interpretação das classificações feitas pelos modelos de visão computacional, avaliando a relevância das áreas apontadas pelo modelo. Já a abordagem quantitativa envolveu a mensuração de métricas dos modelos utilizados.

3.2 Procedimentos metodológicos

3.2.1 Coleta de bases de imagens

Visando o treinamento, o modelo precisou de imagens para conseguir cada vez mais identificar padrões nas fotos/vídeos. Para isso, sites como *Kaggle*¹, *Roboflow*² e *UCI Machine Learning Repository*³ disponibilizam *datasets* gratuitos para treinamento de modelos. A base escolhida foi retirada do *Roboflow*, contendo 1.847 imagens com pequenos focos, grandes incêndios, fumaças claras, escuras e em diferentes ângulos (aéreo e térreo). O *dataset* foi dividido entre imagens de treinamento, validação e testes, respectivamente 75% - 10% - 15%, como o trabalho apresentado por Neto *et al.* (2024).

3.2.2 Desenvolvimento do algoritmo (treinamento e ajuste do modelo *YOLO*)

Após a coleta e organização de imagens, o desenvolvimento foi iniciado. A fim de realizar o primeiro treinamento, o modelo escolhido foi o *YOLO*. Nesta etapa, o desenvolvimento foi realizado dentro de um *notebook* da plataforma *Kaggle*. No ambiente, realizou-se o *upload* do *dataset*, as importações dos módulos, treinamento do modelo, projeção das métricas e, em primeira instância, testes com algumas imagens para termos resultados visíveis e correções de erro no código.

Para definir os parâmetros finais do *fine-tuning* (processo que adapta um modelo pré-treinado para uma tarefa específica) do *YOLO*, foram escolhidos valores

¹ <https://www.kaggle.com/datasets>

² <https://universe.roboflow.com/>

³ <https://archive.ics.uci.edu/>

similares aos utilizados por Dave *et al.* (2023). Neste caso, os autores utilizaram um número próximo de imagens (1.619 imagens) próximo ao que foi utilizado neste trabalho de conclusão (1.847), 120 épocas (*epochs*) de treinamento e a versão número 8 do modelo. Dessa forma, espera-se que a demanda computacional e em termos de parâmetros também estejam próximos.

Após a definição dos parâmetros, foi realizado o carregamento da biblioteca *Ultralytics*, a importação do modelo pré-treinado *YOLOv8n*, a configuração do treinamento com parâmetros como *epochs*, *batch*, *imgsz* e *patience*, e em seguida o treinamento do modelo sobre o conjunto de dados. Ao final do processo, o modelo foi testado com imagens específicas para avaliar visualmente a eficácia das previsões e ajustar, se necessário, os parâmetros.

3.2.3 Desenvolvimento do algoritmo (treinamento e ajuste do modelo *Faster R-CNN*)

Para uma abordagem comparativa com o *YOLO*, foi utilizado também o modelo *Faster R-CNN*. O treinamento foi desenvolvido na mesma plataforma, aproveitando as bibliotecas do *PyTorch*⁴ e *Torchvision*⁵, bibliotecas mesmas que disponibilizam implementações pré-treinadas do *Faster R-CNN*. A escolha dos parâmetros de *fine-tuning* desse modelo se baseou no trabalho de Hung *et al.* (2020), que treinaram o *Faster R-CNN* utilizando aproximadamente 1.500 imagens e 200 épocas de treinamento.

Inicialmente, as imagens e os rótulos foram convertidos para o formato compatível com a estrutura do *PyTorch*, e foram definidos os parâmetros de *data augmentation*. Em seguida, o modelo foi carregado com pesos pré-treinados, realizando o *fine-tuning* para as classes do projeto.

O processo de treinamento incluiu a definição do número de épocas, taxa de aprendizado (*learning rate*), otimizador (SGD) e tamanho de *batch*. Por fim, as previsões foram testadas sobre imagens para verificação visual da qualidade das detecções.

3.2.4 Desenvolvimento do algoritmo (treinamento e ajuste do modelo *ViT*)

Além dos modelos baseados em CNN, foi implementado também o modelo utilizando *transformers*, o *ViT*. O modelo foi implementado com apoio das bibliotecas *Timm*⁶ e *Transformers*⁷, utilizando uma versão pré-treinada do *ViT* e adaptada para tarefas de detecção com *fine-tuning*. O processo de *fine-tuning* e escolha de parâmetros foram inspirados no trabalho de (Smith; Du; Kurien, 2023), que utilizaram *Vision Transformers* com 200 épocas de treinamento para detecção de objetos utilizando imagens

⁴ <https://pytorch.org/>

⁵ <https://pytorch.org/vision/stable/index.html>

⁶ <https://timm.fast.ai/>

⁷ <https://huggingface.co/docs/transformers/en/index>

de baixa resolução e um conjunto de imagens contendo 1600 imagens, característica semelhante ao cenário deste trabalho.

O código iniciou com o pré-processamento das imagens (redimensionamento, normalização e aplicação de *data augmentation*). O modelo ViT foi carregado com modelos pré-treinados, e adaptado para a tarefa de detecção com o acréscimo de uma cabeça de saída customizada para as classes do projeto. Os parâmetros que se basearam no trabalho de Smith, Du e Kurien (2023) foram número de épocas, taxa de aprendizado e *batch*, valores estes que mostraram ser adequados para conjuntos deste tamanho.

3.2.5 Avaliação dos modelos

Em sucessão ao treinamento do modelo, sua eficácia foi testada utilizando imagens estáticas e vídeos previamente salvos no computador. O modelo será executado para analisar imagens, com resultados já concluídos, selecionadas anteriormente. Essas simulações permitiram avaliar a precisão do modelo na detecção de focos ativos de incêndio. Estudos como o de Khanam, Asghar e Hussain (2025), apresentam métricas que permitem avaliar e mensurar a qualidade das detecções de um modelo:

- a) Verdadeiros Positivos (TP) - ocorre quando o modelo detecta corretamente um objeto na imagem;
- b) Verdadeiros Negativos (TN): ocorre quando o modelo não detecta nenhum tipo de objeto em regiões onde realmente não há nenhum objeto presente;
- c) Falsos Positivos (FP) — ocorre quando o modelo identifica um objeto onde não há nenhum, ou a detecção está mal posicionada;
- d) Falsos Negativos (FN) - ocorre quando há um objeto real na imagem, mas o modelo não consegue detectar;
- e) Acurácia - avalia o percentual de acertos, sendo obtida pela razão entre a quantidade de acertos e o total de entradas. Quanto maior seu valor, melhor será a capacidade do modelo de classificar corretamente as detecções:

$$\text{Acurácia} = \frac{TP+TN}{TP+TN+FP+FN}$$

- f) Precisão - avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos. Quanto maior seu valor, menor será a quantidade de falsos alarmes, o que indica que o modelo é confiável:

$$\text{Precisão} = \frac{TP}{TP+FP}$$

- g) *Recall*: mede a proporção de objetos relevantes detectados corretamente. Quanto maior o valor, menor a chance de o modelo deixar de detectar objetos reais, ou seja, é fundamental para garantir cobertura completa da imagem:

$$Recall = \frac{TP}{TP+FN}$$

- h) *F1-Score*: avalia o equilíbrio entre detectar todos os objetos (*recall*) e errar pouco (precisão), trazendo o valor da média harmônica entre as duas métricas. Quanto maior seu valor, melhor será o desempenho do modelo ao combinar baixa taxa de falsos positivos e alta taxa de detecção correta:

$$F1-Score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall}$$

- i) *mAP (Mean Average Precision)*: é a média da precisão do modelo em diferentes situações de detecção. Avalia o quão bem o modelo identifica corretamente os objetos e suas posições. Quanto maior seu valor, melhor será a capacidade do modelo em detectar corretamente a posição e o tipo dos objetos, com alta confiança:

$$mAP = \frac{1}{N} \sum_i^N AP_i$$

Onde N é o número de classes e AP_i é a precisão média para a classe i

- j) *IoU (Intersection over Union)*: avalia a sobreposição entre a caixa predita pelo modelo e a caixa delimitadora real do objeto. Quanto maior o valor de IoU, mais precisa é a localização do objeto detectado.

$$IoU = \frac{\text{Área da interseção}}{\text{Área da união}}$$

- k) Tempo de inferência: tempo médio que o modelo leva para fazer a predição em uma imagem. Quanto menor o tempo, melhor será a aplicabilidade do modelo em cenários de tempo real.

Além das métricas apresentadas por Khanam, Asghar e Hussain (2025), este projeto também monitorou métricas relacionadas à eficiência computacional e à performance operacional do modelo:

- a) Tempo total de treinamento - mede a duração total para que o modelo finalize o processo de aprendizado. Quanto menor for o tempo, mais eficiente será o processo de desenvolvimento do modelo;

- b) Tamanho do modelo final - espaço em disco ocupado pelo modelo treinado gerado. Quanto menor, melhor para fins de armazenamento, portabilidade e implantação em dispositivos com recursos limitados;
- c) Utilização de GPU durante o treinamento - quantidade de memória GPU utilizada, indicando a demanda pelos recursos gráficos disponíveis. Valores menores demonstram o uso mais eficiente dos recursos computacionais disponíveis;
- d) Utilização de RAM durante o treinamento - quantidade de memória usada no treinamento do modelo. Valores menores demonstram uma execução mais leve e eficiente;
- e) Latência na inferência - tempo médio que o modelo leva para processar e gerar uma predição. Quanto menor for seu valor, mais adequado será o modelo para aplicações que exigem resposta imediata;
- f) Tempo médio de treinamento por época - tempo gasto em cada ciclo completo de treinamento. Quanto menor o valor, maior será a velocidade de aprendizado por ciclo, favorecendo experimentos e ajustes mais rápidos:

$$\text{Tempo médio por época} = \frac{\text{Tempo total}}{\text{Número de épocas}}$$

A Tabela 3 resume as métricas utilizadas neste trabalho, acompanhadas de suas respectivas descrições.

Tabela 3 – Métricas utilizadas na avaliação dos modelos.

Métrica	Descrição	Melhor Valor
Acurácia (%)	Percentual de predições corretas.	Maior
mAP@50 (%)	Média da precisão para IoU \geq 0,5.	Maior
mAP@[50:95] (%)	Média da precisão em múltiplos IoUs.	Maior
<i>F1-score</i>	Média harmônica entre precisão e <i>recall</i> .	Maior
Precisão	Proporção de acertos entre predições positivas.	Maior
<i>Recall</i>	Proporção de acertos entre casos positivos reais.	Maior
Inferência (ms)	Tempo por predição.	Menor
Trein. Total (h)	Duração total do treinamento.	Menor
Temp. Época (min)	Tempo médio por época.	Menor
Tam. Modelo (MB)	Tamanho do arquivo final do modelo.	Menor
GPU (MB)	Uso de memória da GPU.	Menor
RAM (MB)	Uso de RAM durante execução.	Menor
Latência (ms)	Tempo entre entrada e saída da predição.	Menor

Fonte: O autor.

4 RESULTADOS

A presente seção descreve os resultados obtidos com o desenvolvimento deste projeto, evidenciando as metas a serem alcançadas e os potenciais impactos que terá perante diferentes frentes. Os resultados representam a materialização dos objetivos traçados.

4.1 Resultados do modelo YOLO

Inicialmente, o treinamento do modelo *YOLO* foi realizado com base no artigo de Dave *et al.* (2023), que indicava o uso de 120 épocas. No entanto, os resultados obtidos com esse treinamento apresentaram métricas insatisfatórias, com indícios de *overfitting* e baixo desempenho. Diante disso, optou-se por realizar uma nova abordagem: treinar o modelo em diferentes estágios de 25 em 25 épocas, até atingir novamente as 120 épocas. O objetivo foi observar o comportamento do modelo ao longo do tempo e identificar em qual ponto ele atinge o melhor resultado. A Tabela 4 apresenta os principais resultados obtidos para os diferentes estágios de treinamento.

Tabela 4 – Resultados do modelo YOLO em diferentes épocas de treinamento. Valores em negrito são os melhores por métrica.

Métrica	25 ép.	50 ép.	75 ép.	100 ép.	120 ép.
Acurácia (%)	63.39	64.71	62.82	63.23	62.66
<i>mAP@50</i> (%)	65.82	66.75	65.90	65.27	64.09
<i>mAP@[50:95]</i> (%)	41.93	42.84	42.39	41.76	40.13
<i>F1-score</i> (%)	63.18	64.18	62.80	62.98	62.55
Precisão (%)	67.01	70.56	64.06	67.15	65.29
<i>Recall</i> (%)	62.57	58.86	61.59	59.30	60.03
Inferência (ms)	7.11	7.55	7.74	6.75	6.83
Latência (ms)	17.65	18.27	18.45	35.21	32.88
Trein. Total	6min	13min	20min	25min	30min
Temp. Época	15.19s	15.59s	16.43s	15.36s	15.19s
Tam. Modelo (MB)	5.96	5.96	5.96	5.96	5.96
<i>Overfitting</i>	Não	Não	Leve	Sim	Sim
Uso GPU	347 MB	329 MB	317 MB	341 MB	341 MB
Uso RAM	3892 MB	4089 MB	4127 MB	4125 MB	4159 MB

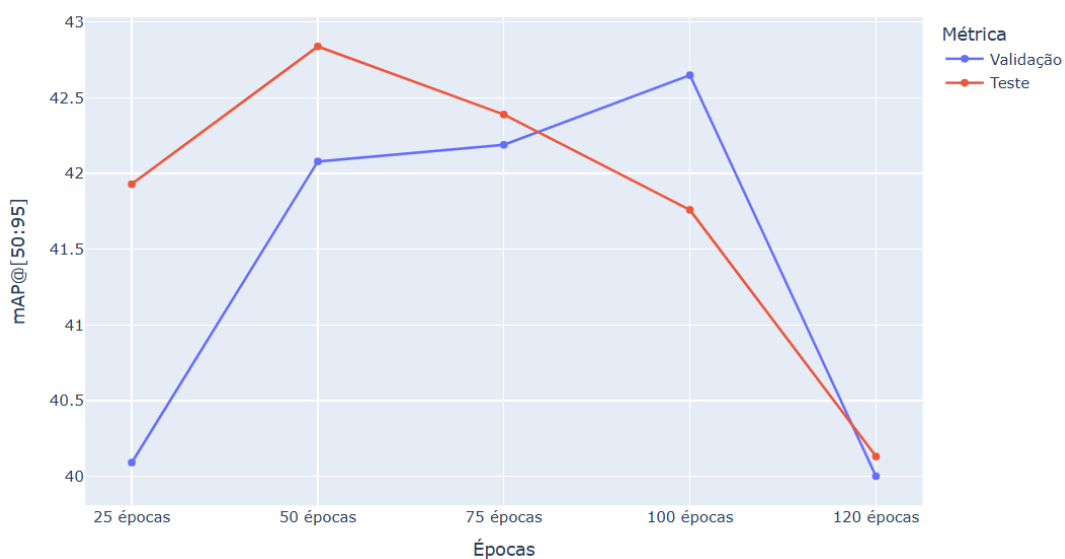
Fonte: O autor.

O estudo de Padilla, Netto e Silva (2020) apresenta $mAP@[50:95]$, $mAP@50$, $F1$ -score, precisão e *recall* como métricas de extrema importância, respectivamente nessa ordem, para a avaliação de modelos de detecção de objetos. Com base nos dados gerados, o melhor desempenho foi alcançado com 50 épocas de treinamento. Esse treinamento apresentou os melhores valores em $mAP@50$, $mAP@[50:95]$, $F1$ -score e precisão. Segundo Venturini (2023), *overfitting* ocorre quando os valores das métricas de avaliação para o conjunto de validação continuam melhorando (ou se mantêm), mas as métricas para o conjunto de teste pioram; ou então, há divergência crescente entre validação e teste. Com 50 épocas de treinamento, o modelo não sinais de *overfitting*. Em contrapartida, o modelo treinado com 120 épocas demonstrou uma queda em quase todas as métricas e apresenta indícios claros de *overfitting*, o que prejudica sua capacidade de generalização.

A Figura 12 apresenta um gráfico de linhas mostrando valores da métrica $mAP@[50:95]$ (que conforme Padilla, Netto e Silva (2020) é a métrica mais importante de avaliação) para o conjunto de validação e de teste do treinamento em todas as épocas. Nela, é possível observar que, até o treinamento de 75 épocas, o valor para o conjunto de teste estava maior, sinais de que não houve *overfitting*, porém, de 100 épocas em diante, o valor de $mAP@[50:95]$ para o conjunto de validação já ultrapassa o valor do conjunto de teste, indicando positividade para *overfitting*. Conforme Venturini (2023), não é apenas uma métrica isolada que determina a presença de *overfitting*. É fundamental analisar o conjunto completo de métricas de desempenho (como acurácia, $F1$ -score, precisão e *recall*) em ambos os conjuntos (validação e teste) antes de concluir. A métrica $mAP@[50:95]$, embora muito relevante, representa apenas uma das possíveis evidências ao comparar o desempenho entre os conjuntos avaliados.

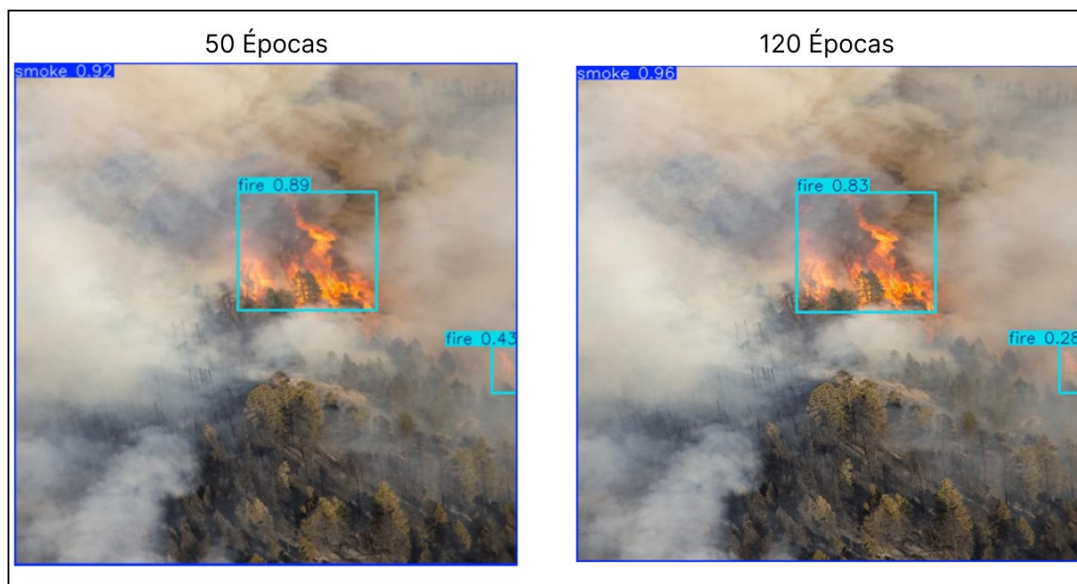
Ainda na Figura 12, ao analisar a variação de valores de $mAP@[50:95]$ para o conjunto de teste, nota-se que até 50 épocas de treinamento, o valor da métrica estava em crescimento, e que após atingir o ápice (em 50 épocas), começou uma queda até 120 épocas do treinamento, mostrando uma tendência de degradação do desempenho do modelo. Essa queda pode estar relacionada à ocorrência de *overfitting*, uma vez que o modelo passa a se ajustar aos dados de treinamento, perdendo capacidade de generalização.

A Figura 13 e a Figura 14 são comparações de duas inferências geradas a partir dos modelos treinados com 50 e 120 épocas, respectivamente. Pode-se observar que, na Figura 13, na imagem com 50 épocas, o fogo foi detectado com confiança de 89% e 43%, e a fumaça com 92%, enquanto na imagem com 120 épocas, os mesmos focos foram detectados com confiança de 83% e 28%, e a fumaça com 96%. Já na Figura 14, nota-se uma vantagem na capacidade de detecção, uma vez que o treinamento com 120 épocas conseguiu capturar 2 focos a mais de incêndio do que o

Figura 12 – Comparação de $mAP@[50:95]$ entre os conjuntos de validação e teste do modelo YOLO.

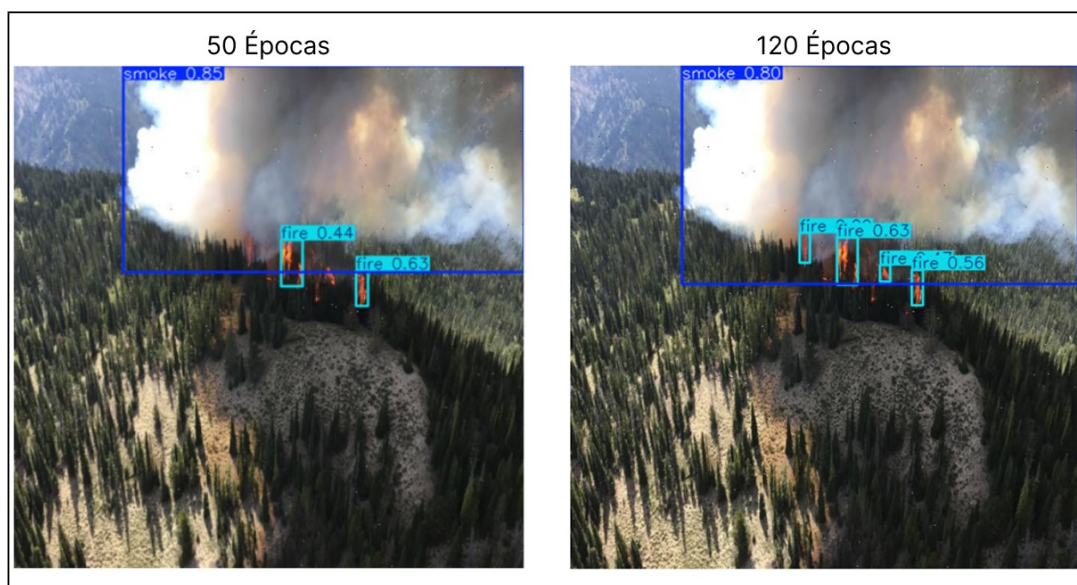
Fonte: O autor.

treinamento com 50 épocas. Dentro da Figura 15, a imagem com 50 épocas os focos de fogo foram detectados com 44% e 63% e a fumaça com 85%, enquanto na imagem com 120 épocas os focos foram detectados com 63% e 56% (outros dois valores não identificados) e a fumaça com 80%.

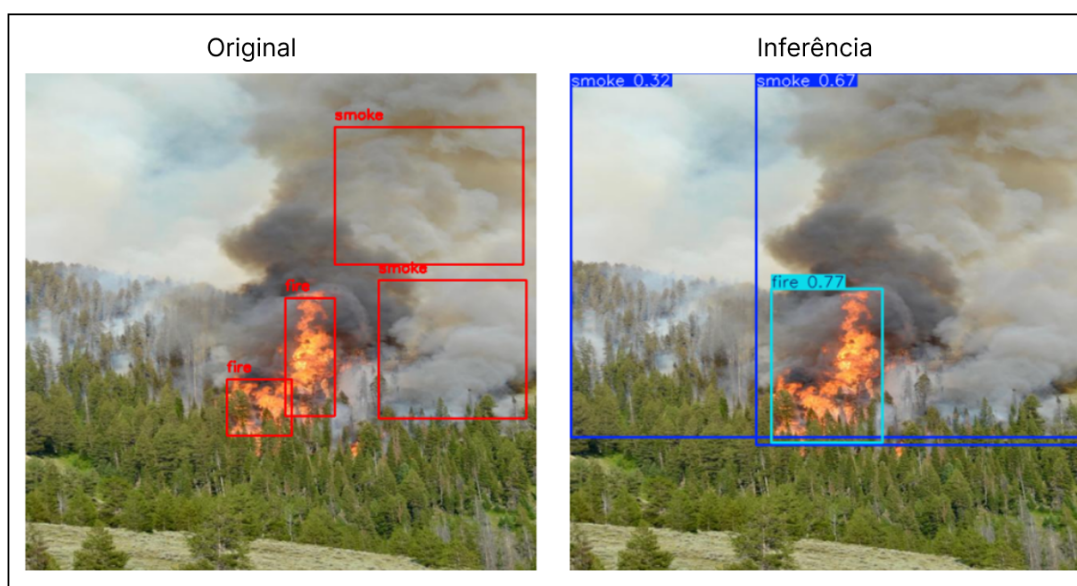
Figura 13 – Primeira comparação entre inferências do YOLO com 50 e 120 épocas.

Fonte: O autor.

Comparando a inferência com a anotação real da imagem, observa-se que, no modelo de 50 épocas, o fogo foi detectado com boa sobreposição em relação à anotação real, enquanto a fumaça foi identificada de forma clara, visualmente até melhor em relação à anotação, como é possível observar na Figura 15.

Figura 14 – Segunda comparação entre inferências do YOLO com 50 e 120 épocas.

Fonte: O autor.

Figura 15 – Comparação entre anotação e inferência com 50 épocas de treinamento do YOLO.

Fonte: O autor.

4.2 Resultados do modelo *Faster R-CNN*

O treinamento inicial do modelo *Faster R-CNN* foi realizado com base no artigo de Hung *et al.* (2020), que treinou o modelo por 200 épocas. No entanto, assim como observado no modelo *YOLO*, os resultados obtidos com esse treinamento mostraram métricas insatisfatórias, indicando desempenho inferior em comparação a outras épocas possíveis de aprendizado. Diante disso, foi adotada a mesma estratégia: realizar treinamentos em estágios de 25 em 25 épocas, variando de 25 até 200 épocas, com o objetivo de identificar em qual ponto o modelo apresenta seu melhor desempenho. A

Tabela 5 apresenta os principais resultados obtidos ao longo dos diferentes treinamentos, considerando sempre os melhores valores entre os conjuntos de validação e teste para cada métrica.

Tabela 5 – Resultados do modelo *Faster R-CNN* em diferentes épocas de treinamento. Valores em negrito são os melhores por métrica.

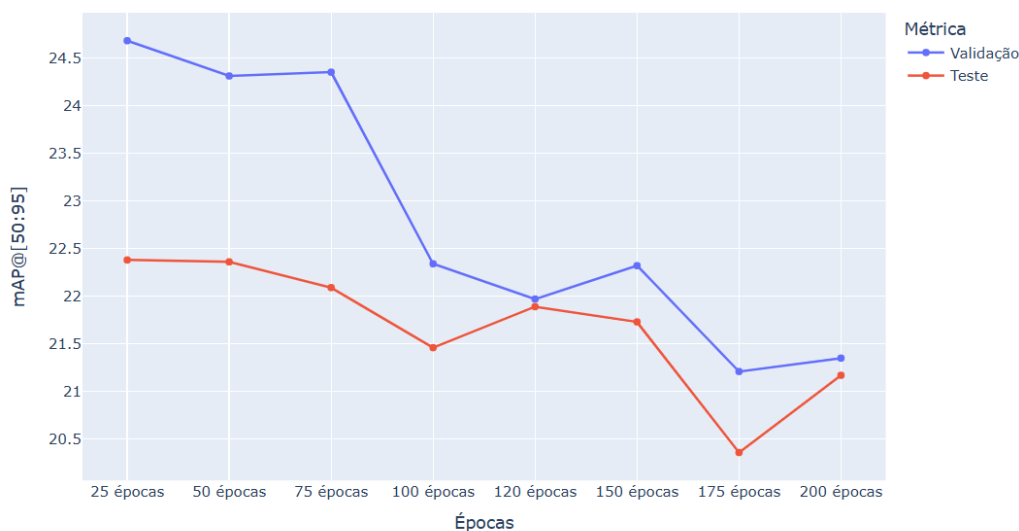
Métrica	25	50	75	100	120	150	175	200
Acurácia (%)	56.34	59.14	54.61	64.10	65.13	64.55	67.50	66.83
<i>mAP@50</i> (%)	48.60	48.60	47.15	44.72	44.02	44.08	42.18	42.71
<i>mAP@[50:95]</i> (%)	22.38	22.36	22.09	21.46	21.89	21.73	20.36	21.17
<i>F1-score</i> (%)	52.38	53.85	50.92	60.92	60.84	59.61	62.45	61.63
Precisão (%)	52.46	53.88	51.11	60.95	60.83	59.56	62.18	61.45
<i>Recall</i> (%)	52.62	54.24	51.20	62.54	62.94	61.36	64.71	63.98
Inferência (ms)	121.19	120.93	121.15	121.93	120.70	121.06	120.56	121.23
Latência (ms)	449.97	455.36	456.50	501.22	449.87	471.92	444.13	454.15
Trein. Total	107min	218min	286min	384min	479min	565min	674min	766min
Temp. Época	256.4s	262.6s	229.2s	234.4s	257.8s	223s	261.1s	241.8s
Tam. Modelo (MB)	158	158	158	158	158	158	158	158
<i>Overfitting</i>	Não	Não	Não	Não	Não	Não	Não	Não
Uso GPU (MB)	494	517	496	496	500	498	500	497
Uso RAM (MB)	1631	1675	1633	1631	1638	1628	1625	1631

Fonte: O autor.

Com base no estudo de Padilla, Netto e Silva (2020) e nos resultados obtidos, o modelo treinado por 175 épocas demonstrou o melhor desempenho geral em relação às métricas principais, superando o treinamento completo de 200 épocas. Apesar do leve ganho de *mAP* no modelo de 200 épocas, o treinamento com 175 épocas apresentou melhores resultados em métricas relevantes como *F1-score*, precisão e *recall*, além de exigir menos tempo de treinamento.

A Figura 16 apresenta um gráfico de linhas semelhante ao gerado para o modelo anterior, mostrando valores da métrica *mAP@[50:95]* para os conjuntos de validação e teste ao longo de todas as épocas de treinamento. Observa-se que, em nenhum momento, o valor obtido para o conjunto de teste ultrapassou o valor do conjunto de validação, o que indica de sinais evidentes de inexistência de *overfitting* durante o processo de treinamento.

Figura 16 – Comparação de $mAP@[50:95]$ entre os conjuntos de validação e teste do modelo *Faster R-CNN*.



Fonte: O autor.

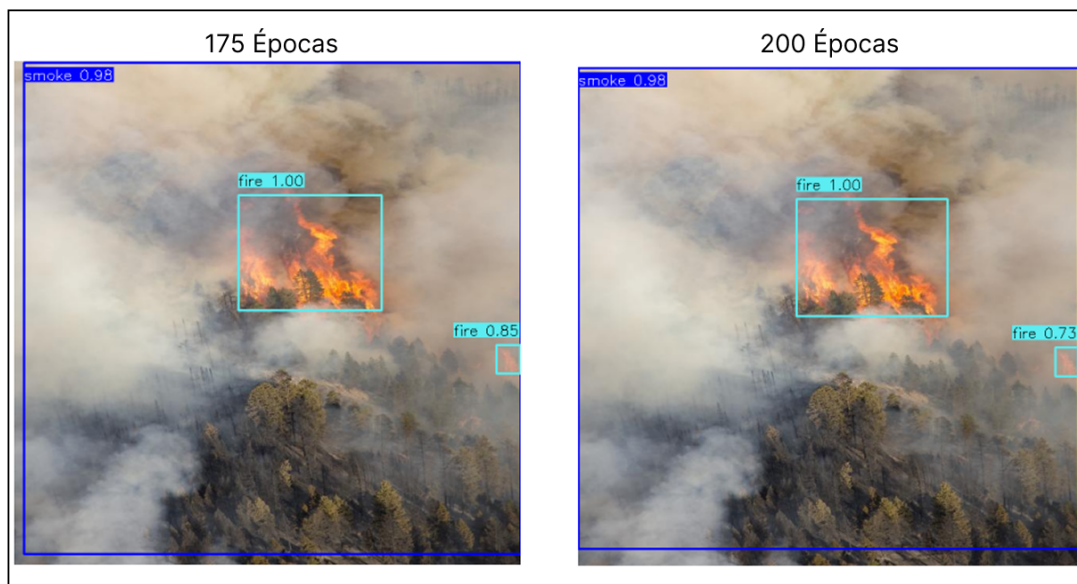
Também na Figura 16, observando a variação de valores da métrica $mAP@[50:95]$ para o conjunto de teste, é possível notar uma tendência de declínio contínuo desde o primeiro treinamento (25 épocas) até o último (200 épocas), sugerindo uma possível limitação do modelo quanto à sua capacidade de generalização com o avanço do treinamento. Coincidentemente, o menor valor da métrica apresenta-se no treinamento com 175 épocas; treinamento esse que obteve melhores resultados em outras importantes métricas (acurácia, $F1$ -score, precisão e *recall*), como é possível analisar na Tabela 5.

Esse cenário reforça a importância de não se basear em uma única métrica para a escolha final de um modelo, como destacado por Venturini (2023). A decisão deve considerar um conjunto amplo de métricas complementares, analisando o equilíbrio entre desempenho em detecção, capacidade de generalização, tempo de inferência e robustez. Portanto, mesmo que o valor de $mAP@[50:95]$ tenha decrescido, o modelo treinado com 175 épocas demonstrou, de forma geral, mais eficiente e estável.

A Figura 17 e Figura 18 apresentam comparações de inferência entre os modelos gerados pelos dois treinamentos (175 épocas e 200 épocas). Na Figura 17 observa-se que os focos de fogo foram detectados com 100% e 85% de confiança no modelo de 175 épocas, enquanto no modelo de 200 épocas as confianças caíram para 100% e 73%. Já a fumaça foi detectada com 98% de confiança no modelo de 175 épocas e se manteve com mesmo valor no modelo de 200 épocas. Já na Figura 18, observa-se novamente a presença de múltiplas caixas delimitadoras indicando “fogo” e “fumaça”, com níveis de confiança semelhantes entre os dois modelos. A detecção de fumaça permaneceu alta (98% para 175 épocas e 97% para 200 épocas), enquanto o fogo apresentou leves variações, com uma das regiões passando de 68%

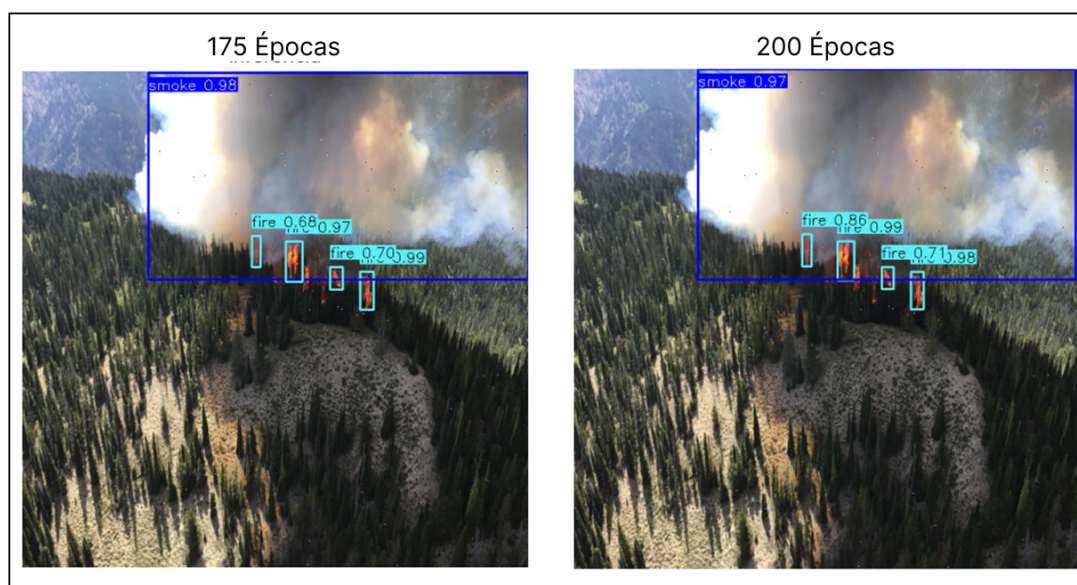
com 175 épocas para 86% de confiança com 200 épocas do treinamento. Apesar da diferença sutil, o modelo de 175 épocas apresentou melhor clareza e precisão nas regiões identificadas nas imagens.

Figura 17 – Primeira comparação entre inferências do *Faster R-CNN* com 175 e 200 épocas.



Fonte: O autor.

Figura 18 – Segunda comparação entre inferências do *Faster R-CNN* com 175 e 200 épocas.

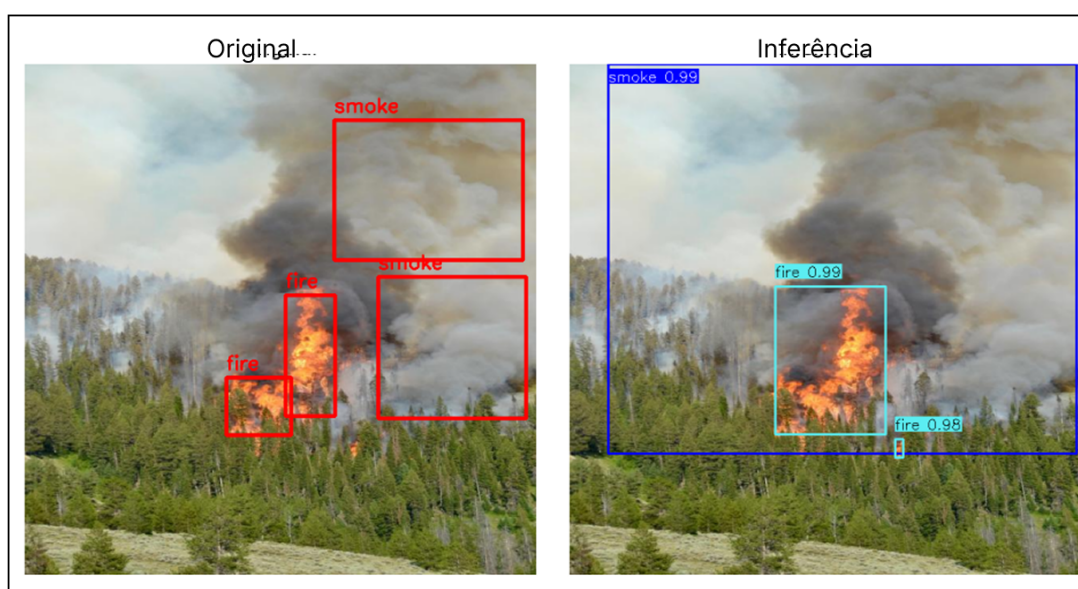


Fonte: O autor.

Além da comparação entre os dois modelos, é fundamental avaliar o quão próximo o modelo chega das anotações reais. A Figura 19 apresenta a comparação entre a inferência gerada e a marcação do *dataset*. Comparando a anotação com a inferência da imagem, podemos ver que o fogo foi corretamente localizado, com 99% de confiança, além de conseguir identificar um pequeno foco de incêndio não mapeado,

indicando bom entendimento do modelo, cobrindo praticamente toda a área anotada. No caso da fumaça, o modelo foi capaz de detectar a presença com boa precisão e 99% de confiança na inferência. Assim, a sobreposição entre predição e anotação é significativa, comprovando a efetividade do modelo.

Figura 19 – Comparação entre anotação e inferência com 175 épocas de treinamento do *Faster R-CNN*.



Fonte: O autor.

4.3 Resultados do modelo *Vision Transformer*

O modelo Vision Transformer (*ViT*), quando treinado com os parâmetros e estrutura indicados no artigo de Smith, Du e Kurien (2023), apresentou métricas insatisfatórias, indicando baixo desempenho na tarefa de detecção de incêndios florestais. Diante disso, optou-se também por realizar a mesma abordagem utilizada nos modelos anteriores, onde o modelo foi treinado progressivamente de 25 em 25 épocas, até alcançar as 200 épocas propostas no artigo original. A Tabela 6 apresenta os principais resultados obtidos nas diferentes etapas de treinamento.

Com base no estudo de Padilla, Netto e Silva (2020) e nos resultados obtidos, o modelo treinado por 75 épocas apresentou o melhor desempenho geral entre todos os experimentos realizados. Esse treinamento se destacou por atingir o maior valor de $mAP@50$ (10.66%) e $mAP@[50:95]$ (2.61%), além de manter um bom equilíbrio entre $F1-score$ (41.63%), precisão (38.32%) e $recall$ (45.55%). Em comparação com o modelo treinado por 200 épocas, os resultados de 75 épocas indicam menor tendência a *overfitting*, menor tempo de treinamento e desempenho superior nas métricas de avaliação mais relevantes para tarefas de detecção de objetos.

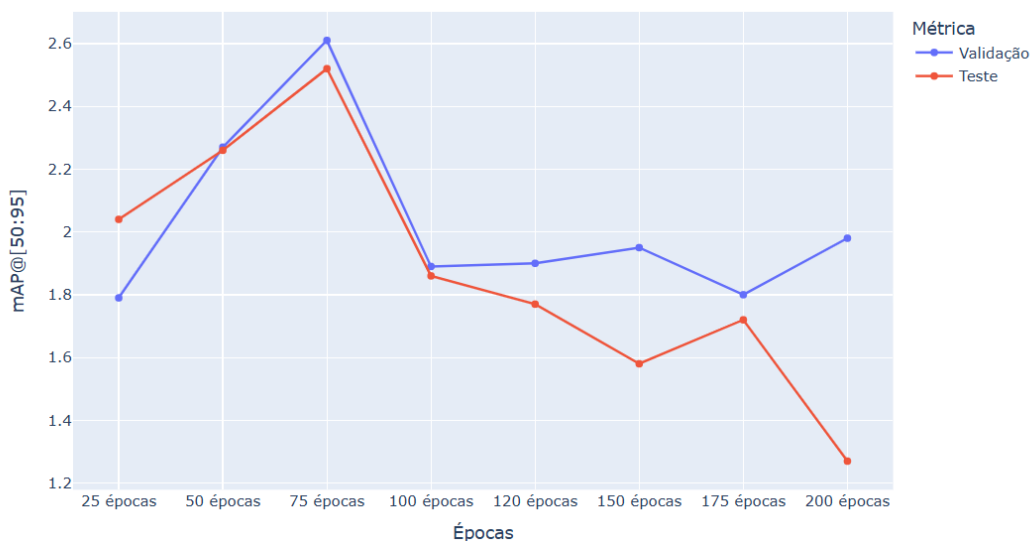
Tabela 6 – Resultados do modelo ViT em diferentes épocas de treinamento. Valores em negrito são os melhores por métrica.

Métrica	25	50	75	100	120	150	175	200
Acurácia (%)	22.64	26.57	26.28	26.28	27.70	26.62	26.49	24.38
<i>mAP@50</i> (%)	8.20	9.81	10.66	7.80	7.34	6.60	7.07	5.35
<i>mAP@[50:95]</i> (%)	2.04	2.26	2.52	1.86	1.77	1.58	1.72	1.27
<i>F1-score</i> (%)	36.91	41.99	41.63	41.62	43.38	42.05	41.89	39.2
Precisão (%)	32.68	39.69	38.32	47.18	48.05	46.38	51.25	57.48
<i>Recall</i> (%)	42.42	44.56	45.55	37.23	39.54	38.47	35.42	29.74
Inferência (ms)	111.01	112.96	113.36	111.31	110.12	108.59	117.02	113.08
Latência (ms)	686.83	716.88	770.88	684.38	680.25	685.26	826.58	728.51
Trein. Total	89min	175min	249min	330min	403min	511min	600min	689min
Temp. Época	213s	210.4s	200s	194.2s	218.2s	216.4s	213s	214.6s
Tam. Modelo (MB)	158	158	158	158	158	158	158	158
<i>Overfitting</i>	Não	Não	Não	Leve	Sim	Sim	Sim	Sim
Uso GPU (MB)	2954	2954	2954	3123	3123	3120	3120	3123
Uso RAM (MB)	5066	5044	5080	3434	2971	2971	2966	2967

Fonte: O autor.

A Figura 20 apresenta um gráfico de linhas que ilustra a variação da métrica *mAP@[50:95]* para os conjuntos de validação e teste ao longo de todas as épocas de treinamento do modelo ViT, com o objetivo de identificar sinais de *overfitting*. Nota-se que em 50 épocas os valores da métrica para ambos os conjuntos estavam relativamente próximos. A partir desse ponto, o valor do conjunto de validação ultrapassou o valor do conjunto de teste, indicando um possível início de *overfitting*. No entanto, como reforçado por Venturini (2023), a conclusão de perda de generalização não pode ser feita com base em uma única métrica, sendo necessário considerar o conjunto completo de métricas.

Ao analisar somente o valor de *mAP@[50:95]* para o conjunto de teste na Figura 20, observa-se uma tendência de crescimento da métrica entre as épocas 25 e 75, indicando melhorias progressivas no desempenho do modelo durante esse intervalo. A partir desse ponto, os valores passam por uma queda contínua, com um pequeno acrise no treinamento de 175 épocas, seguida por nova queda no último ciclo (200 épocas). Esse comportamento reforça a importância da escolha criteriosa do

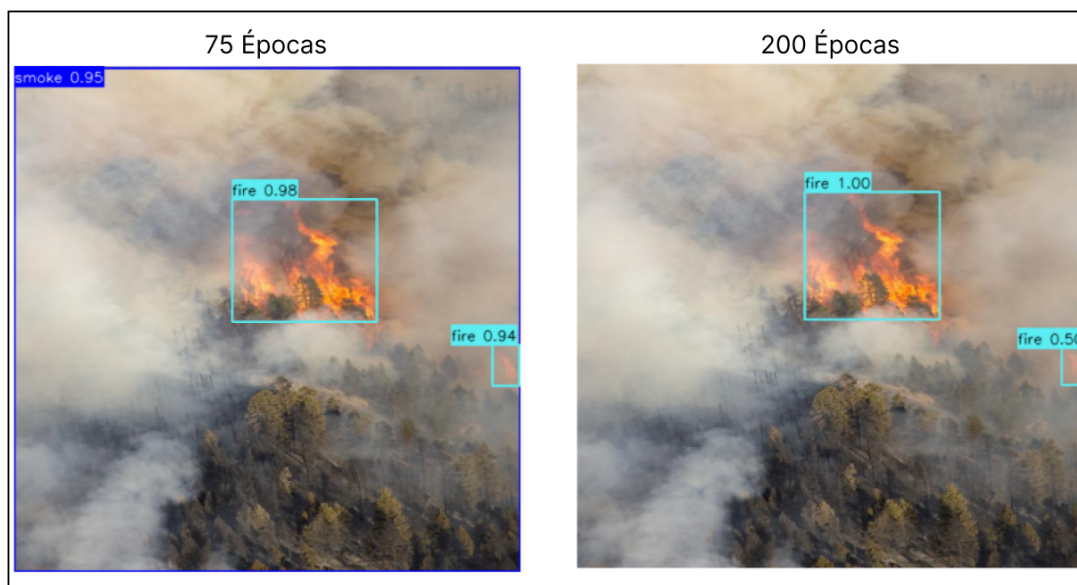
Figura 20 – Comparação de $mAP@[50:95]$ entre os conjuntos de validação e teste do modelo ViT.

Fonte: O autor.

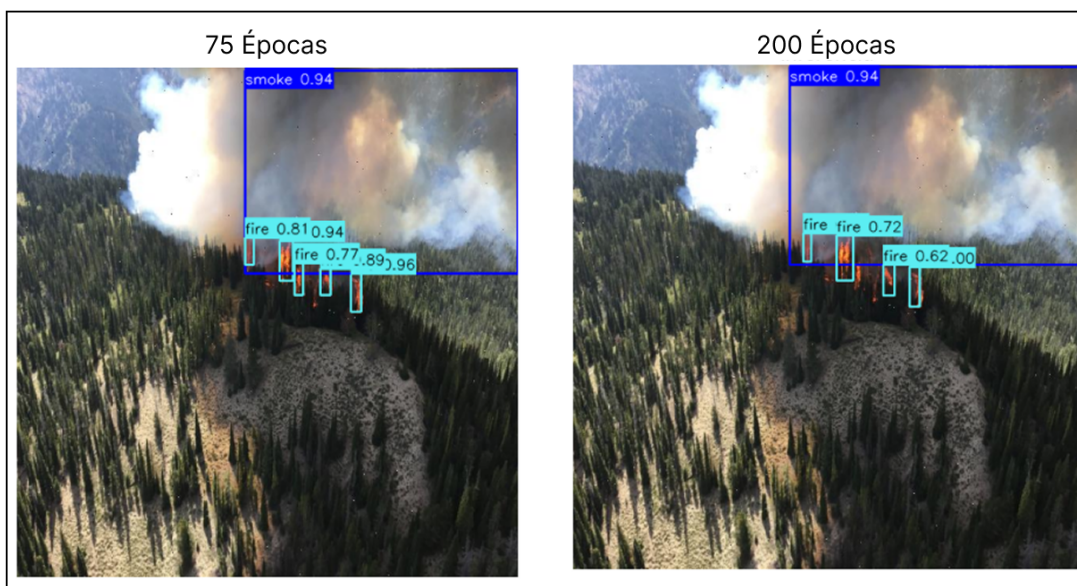
número de épocas, já que treinamentos excessivamente prolongados podem levar à degradação do desempenho, mesmo em métricas fundamentais como o $mAP@[50:95]$.

A seguir, a Figura 21 e a Figura 22 apresentam a comparação visual entre as inferências feitas pelos modelos treinados com 75 e 200 épocas. Na Figura 21, o modelo de 75 épocas, o fogo foi detectado com confiança de 98% e 94%, enquanto a fumaça foi detectada com 95%. Já o modelo com 200 épocas detectou o fogo com 100% e 50%, porém, sem a detecção de fumaça na imagem. Nota-se que o modelo de 75 épocas demonstrou maior eficiência nas detecções e com menos oscilações de confiança. Já na Figura 22, o treinamento com 75 épocas detectou 5 focos de incêndio com confiança superior a 77%, chegando em confianças de até 96%, enquanto o treinamento com 200 épocas encontrou 4 focos, com confiança mínima de 62% e máxima de 72%. A fumaça foi encontrada nas duas inferências com pouco deslocamento da caixa delimitadora, porém, a confiança se manteve em 94%.

Comparando a anotação com a inferência, apresentadas na Figura 23, observa-se que o modelo de 75 épocas conseguiu identificar corretamente as principais regiões com presença de fogo e fumaça. As caixas de “fire” coincidem com os focos reais de incêndio, enquanto a grande área de fumaça foi bem delimitada. Embora ainda existam pequenas divergências de localização e tamanho das caixas (principalmente na detecção de fumaça), o modelo demonstrou bom alinhamento com os dados anotados, e boa localização de focos de incêndio, reforçando sua capacidade de detecção.

Figura 21 – Primeira comparação entre inferências do ViT com 75 e 200 épocas.

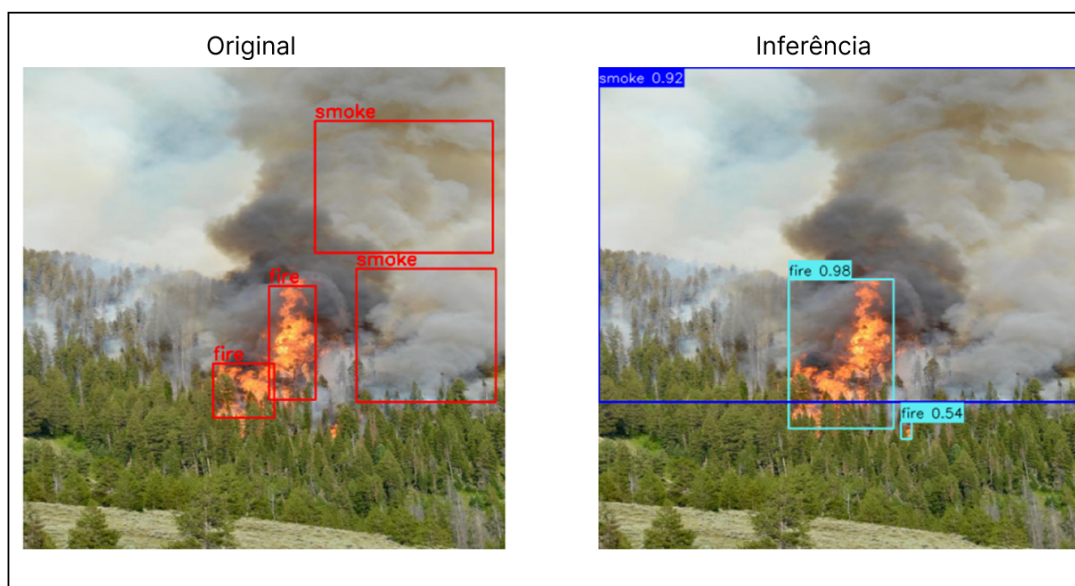
Fonte: O autor.

Figura 22 – Segunda comparação entre inferências do ViT com 75 e 200 épocas.

Fonte: O autor.

4.4 Comparação geral entre os modelos

A Tabela 7 apresenta a comparação geral entre os três modelos avaliados no projeto: *YOLO*, *Faster R-CNN* e *ViT*. Os modelos foram analisados e avaliados com base em diversas métricas, incluindo acurácia, $mAP@50$, $mAP@[50:95]$, $F1$ -score, precisão e *recall*. Também foram considerados fatores práticos como tempo de inferência, tempo de treinamento, uso de recursos computacionais (GPU e RAM) e tamanho do modelo. Essa análise permitiu identificar as vantagens e limitações de cada abordagem, considerando tanto desempenho quanto viabilidade prática de aplicação

Figura 23 – Comparação entre anotação e inferência com 75 épocas de treinamento do ViT.

Fonte: O autor.

em contextos reais.

Tabela 7 – Comparação geral entre os modelos treinados. Valores em negrito são os melhores por métrica.

Métrica	YOLO	Faster R-CNN	ViT
Acurácia (%)	64.71	67.50	26.28
<i>mAP@50</i> (%)	66.75	42.18	10.66
<i>mAP@[50:95]</i> (%)	42.84	21.21	2.61
<i>F1-score</i> (%)	64.18	62.45	41.63
Precisão (%)	70.56	62.18	38.32
<i>Recall</i> (%)	58.86	64.71	45.55
Inferência (ms)	7.55	120.56	113.36
Núm. Épocas	50	175	75
Trein. Total	13min	674min	249min
Temp. Época	15.59s	261.1s	200s
Tam. Modelo (MB)	5.96	158	158

Fonte: O autor.

Com base nos resultados obtidos e no estudo de Padilla, Netto e Silva (2020), que destaca a importância de métricas como *mAP@50*, *mAP@[50:95]*, *F1-score*, precisão e *recall* na avaliação de desempenho de modelos de detecção de objetos, observa-se que o modelo YOLO apresentou o melhor desempenho geral. Ele

obteve os maiores valores nas métricas principais: $mAP@50$ (66.75%), $mAP@[50:95]$ (42.84%), $F1$ -score (64.18%) e precisão (70.56%), baixo tempo de inferência (7.55 ms) e sendo o mais leve (5.96 MB) entre os avaliados.

O modelo *Faster R-CNN*, por sua vez, destacou-se pelo melhor *recall* (64.71%) e boa acurácia (67.50%), mostrando-se também eficaz na identificação, embora com menor desempenho em mAP e com maior tempo de inferência. Trata-se de um modelo robusto, mas mais pesado e demorado para treinar e executar.

Já o modelo baseado em *ViT*, apesar de utilizar uma arquitetura mais moderna, apresentou resultados consideravelmente inferiores nas métricas principais ($mAP@50$ de apenas 10.66%, e $F1$ -score de 41.63%), indicando que ainda necessita de otimizações para problemas como detecção de incêndios florestais com dataset limitado.

Para aplicações em que há necessidade de detecção rápida, com recursos computacionais limitados e cenários de uso em tempo real (como em drones, sistemas de vigilância ou sensores móveis), o modelo *YOLO* se mostra como a escolha mais apropriada. Seu tamanho, velocidade de inferência e bom desempenho geral fazem dele uma excelente alternativa para operações em campo, onde agilidade e eficiência são fundamentais.

Em contrapartida, em contextos com maior disponibilidade de recursos computacionais e menor preocupação com tempo de resposta (como centros de monitoramento, estações fixas de análise de dados ou servidores dedicados), o *Faster R-CNN* pode ser uma boa alternativa, oferecendo alta acurácia e *recall*. Em trabalhos futuros, recomenda-se a ampliação do conjunto de dados, principalmente com inclusão de imagens com variações climáticas. Com isso, pode existir melhoras significativas no desempenho dos três modelos.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresenta uma robusta comparação entre métodos inteligentes que utilizam visão computacional e IA para a detecção e monitoramento de focos de incêndio. O objetivo geral deste trabalho foi avaliar diferentes modelos de *machine learning* baseado em visão computacional para identificar padrões em regiões com diferentes climas, vegetações e relevos visando detectar incêndios com ou sem fumaça em regiões florestais. Conforme apresentado nos resultados, o objetivo geral foi atingido com sucesso. A escolha do tema se justifica devido ao crescente número de queimadas e a grande preocupação com os incêndios florestais no país, os quais causam danos ambientais, econômicos e sociais. Posteriormente, a utilização de drones equipados com câmeras e a aplicação de algoritmos de visão computacional representam uma abordagem promissora para a prevenção e combate a esses eventos.

A revisão da literatura permitiu identificar um crescente interesse na aplicação de técnicas de *machine learning* para a análise de imagens de satélite e aéreas, com o objetivo de detectar padrões relacionados a incêndios florestais. No entanto, a aplicação dessas técnicas em tempo real e em grandes áreas ainda é um desafio. Casos de sucesso foram registrados com o uso de câmeras térmicas, que detectam a emissão de calor em regiões específicas.

A metodologia utilizada neste trabalho envolve a coleta e o pré-processamento de um conjunto de imagens aéreas, o *fine-tuning* de dois modelos de *machine learning* baseados em redes neurais convolucionais (*YOLO* e *Faster R-CNN*) e um modelo baseado em *Transformers* (*ViT*), além da avaliação do desempenho dos modelos, utilizando métricas de classificação e inferências em imagens estáticas.

Com base nos resultados obtidos, o modelo *YOLO* se destacou por apresentar um ótimo equilíbrio entre desempenho, tempo de inferência e tamanho do modelo. Ele obteve a melhor pontuação geral em métricas essenciais, além de ser o modelo mais leve e rápido em tempo de execução. Dessa forma, mostra-se o modelo mais equilibrado. Essa característica o torna especialmente promissor para aplicações embarcadas, por exemplo, o uso em drones equipados com câmeras, possibilitando a detecção de incêndios em tempo real, mesmo em áreas remotas ou de difícil acesso. Futuramente, espera-se que a implementação desse tipo de sistema em plataformas móveis permita ampliar a cobertura de monitoramento, reduzir o tempo de resposta a focos de incêndio e, conseqüentemente, minimizar seus impactos.

Como recomendações para aplicações dos modelos, em cenários com alta disponibilidade de recursos, o *Faster R-CNN* seria o mais adequado dentre os modelos avaliados. Ainda, o *YOLO* é considerado mais adequado em cenários de pouca disponibilidade de recursos, dado seu curto tempo de inferência e pequeno

volume em MB. O *ViT* não obteve métricas interessantes no contexto deste trabalho, sendo necessário um estudo mais aprofundado sobre seu comportamento prático como trabalho futuro.

REFERÊNCIAS

- ALI, N. S. *et al.* An effective face detection and recognition model based on improved yolo v3 and vgg 16 networks. *J. homepage <http://iieta.org/journals/ijcmem>*, v. 12, n. 2, p. 107–119, 2024.
- BRAGA, A. V. *et al.* Machine learning: o uso da inteligência artificial na medicina. *Brazilian Journal of Development*, v. 5, n. 9, p. 16407–16413, 2019.
- CALANCA, P.; MATHEUS, Y.; RAPHAELL, B. *Quais são os 4 tipos de aprendizagem na IA, algoritmos e usos no dia a dia*. 2023. <https://www.alura.com.br/artigos/quais-sao-tipos-aprendizagem-ia-inteligencia-artificial?srsltid=AfmBOooA6qmls0NuW7totK6zZjTKfu068LA9igJMKuKVmU83LCwR_ke7>.
- DAVE, B. *et al.* Wild animal detection using yolov8. *Procedia Computer Science*, Elsevier, v. 230, p. 100–111, 2023.
- DIJINGA, F. J. *Python: Creating a Traffic Heatmap using Computer Vision and Artificial Neural Networks (YOLOv8)*. 2024. <https://medium.com/@fernando.dijkinga/python-creating-a-traffic-heatmap-using-computer-vision-and-artificial-neural-networks-yolov8-0ce>. Publicado em: 10 out. 2024. Acesso em: 22 abr. 2025.
- FEARNSIDE, P. M. Fogo na amazônia: impactos ambientais e sociais: 1–tipos de fogo. 2024a.
- FEARNSIDE, P. M. Fogo na amazônia: impactos ambientais e sociais: 2–fogo e desmatamento. 2024b.
- FELIX, T. *Brasil bate recorde de queimadas no primeiro semestre de 2024*. 2024. <https://www.cnnbrasil.com.br/nacional/brasil-bate-recorde-de-queimadas-no-primeiro-semester-de-2024/>.
- FERREIRA, M. H. S. Reconhecimento facial para detecção de emoções utilizando redes neurais convolucionais com tensorflow. Pontifícia Universidade Católica de Goiás, 2021.
- FIDALGO, A.; FERNANDES, A. S. Efeito dos incêndios florestais no ecossistema: revisão integrativa. *Territorium*, n. 30 (II), p. 5–13, 2023.
- FU, Z. Vision transformer: Vit and its derivatives. *arXiv preprint arXiv:2205.11239*, 2022.
- GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: Editora Atlas SA, 2002.
- HAN, K. *et al.* A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 45, n. 1, p. 87–110, 2022.
- HUNG, G. L. *et al.* Faster r-cnn deep learning model for pedestrian detection from drone images. *SN Computer Science*, Springer, v. 1, p. 1–9, 2020.
- JIANG, P. *et al.* A review of yolo algorithm developments. *Procedia computer science*, Elsevier, v. 199, p. 1066–1073, 2022.
- KHAN, S. *et al.* Transformers in vision: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, v. 54, n. 10s, p. 1–41, 2022.

- KHANAM, R.; ASGHAR, T.; HUSSAIN, M. Comparative performance evaluation of yolov5, yolov8, and yolov11 for solar panel defect detection. In: MDPI. *Solar*. [S.l.], 2025. v. 5, n. 1, p. 6.
- KOPTELOV, A. *Computer Vision in Retail: Applications, Benefits and Examples*. 2022. <https://www.itransition.com/blog/computer-vision-retail>. Publicado em: 20 jun. 2022. Acesso em: 22 abr. 2025.
- LYNN, T. *20+ Real-World Computer Vision Examples and Projects*. 2025. <https://blog.roboflow.com/computer-vision-examples/>. Publicado em: 13 fev. 2025. Acesso em: 22 abr. 2025.
- MACHADO, U. T. Inspeção automática de frascos utilizando visão computacional em sistema embarcado. 2019.
- MAIA, E. *Comunidade medica estuda relação das queimadas com doenças pulmonares*. 2024. <https://www.cnnbrasil.com.br/nacional/comunidade-medica-estuda-relacao-das-queimadas-com-doencas-pulmonares/>.
- MILANO, D. de; HONORATO, L. B. Visão computacional. *Faculdade de Tecnologia, Universidade Estadual de Campinas*, 2010.
- Ministério da Ciência, Tecnologia e Inovação. *Portal de Monitoramento de Queimadas - TerraBrasilis*. 2024. Acesso em: 12 set. 2024. Disponível em: <https://terrabrasilis.dpi.inpe.br/queimadas/portal/>.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, v. 1, n. 1, p. 32, 2003.
- MORAIS, J. C. M. de. Tecnologia de combate aos incêndios florestais. 2004.
- MOUMGIAKMAS, S. S.; SAMATAS, G. G.; PPAKOSTAS, G. A. Computer vision for fire detection on uavs—from software to hardware. *Future Internet*, MDPI, v. 13, n. 8, p. 200, 2021.
- MOURA, L. S. B. d. Drone de combate aos incêndios florestais. 2019.
- NAGAO, A. C. *Conheça 8 aplicações da visão computacional no varejo*. 2024. <https://panoramafarmacutico.com.br/visao-computacional-no-varejo/>.
- NETO, J. F. da S. *et al.* Conem2024-0826 influência dos parâmetros de treino na performance da rede neural yolo para classificação de defeitos em pré-formas de aisi 316 l-si obtidas por mada. 2024.
- OLORUNSHOLA, O.; JEMITOLA, P.; ADEMUWAGUN, A. Comparative study of some deep learning object detection algorithms: R-cnn, fast r-cnn, faster r-cnn, ssd, and yolo. *Nile Journal of Engineering and Applied Science*, Nile University of Nigeria Abuja, v. 1, n. 1, p. 0–0, 2023.
- O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

- PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A survey on performance metrics for object-detection algorithms. In: IEEE. *2020 international conference on systems, signals and image processing (IWSSIP)*. [S.l.], 2020. p. 237–242.
- PATKI, Y. R. Object detection using faster r-cnn. 2024.
- PIMENTEL, A. d. S.; COSTA, L. C.; CUNHA, A. P. M. do A. Avaliação dos impactos das secas: Proposta piloto de formulário colaborativo. 2024.
- REDMON, J. *et al.* You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 779–788.
- REN, S. *et al.* Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, v. 28, 2015.
- REZATOFIGHI, H. *et al.* Generalized intersection over union: A metric and a loss for bounding box regression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 658–666.
- RODRIGUES, N. A. d. S. A educação ambiental para reduzir potenciais riscos de queimadas e incêndios florestais no Brasil. Universidade Federal do Tocantins, 2024.
- SHIMOKAWA, M. C. d. S.; MENDES, E. F.; OLIVEIRA-JUNIOR, E. S. Pantanal sob pressão do desmatamento: pecuária e riscos de incêndio florestal. *Cadernos de Agroecologia*, v. 19, n. 1, 2024.
- SILVA, T. C. B. M. A. d. *et al.* Tópicos introdutórios: ciências do fogo. *Corpo de Bombeiro Militar de Santa Catarina*, 2018.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- SITAUULA, C.; HOSSAIN, M. B. Attention-based vgg-16 model for covid-19 chest x-ray image classification. *Applied Intelligence*, Springer, v. 51, n. 5, p. 2850–2863, 2021.
- SMITH, A. D.; DU, S.; KURIEN, A. Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset. *Applied sciences*, MDPI, v. 13, n. 15, p. 8716, 2023.
- SOUZA, J. R. d. *et al.* O panorama dos eventos de seca na Amazônia: Uma análise bibliométrica. 2023.
- SOUZA, M. A. d.; VALE, A. T. do; CHAVES, B. S. Análises térmicas das plantas de baixa inflamabilidade para barreiras verdes de proteção contra incêndios florestais. *Promptus: Revista Técnico-Científica do CBMDF*, v. 1, p. 80–94, 2024.
- TAN, L. *et al.* Comparison of yolo v3, faster r-cnn, and ssd for real-time pill identification. 2021.
- TOSHNIWAL, A. *Retail Revolution: A Comprehensive Learning Guide for the Purchase of Tomorrow's Goods*. 2023. <https://www.labellerr.com/blog/retail-revolution-a-comprehensive-learning-guide-for-the-purchase-of-tomorrows-goods/>. Publicado em: 21 dez. 2023. Acesso em: 22 abr. 2025.

VALENCIA-ZULETA, A. *et al.* O fogo e a herpetofauna no pantanal: observações durante e após os incêndios. *Biodiversidade Brasileira*, v. 14, n. 4, 2024.

VENTURINI, G. C. *Técnicas de aprendizagem de máquina para classificação do nível de satisfação do estudante com base em seu perfil e percepção sobre uma universidade*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2023.

VINA, A. *Computer Vision for Theft Prevention: Enhancing Security*. 2024. <https://www.ultralytics.com/blog/computer-vision-for-theft-prevention-enhancing-security>. Publicado em: 15 nov. 2024. Acesso em: 22 abr. 2025.

VisualCortex. *People Detection – VisualCortex Use Cases*. 2025. <https://visualcortex.com/usecases/people-detection>. Acesso em: 22 abr. 2025.

VOULODIMOS, A. *et al.* Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, Wiley Online Library, v. 2018, n. 1, p. 7068349, 2018.

VOUTIK, L. *How Virtual Mirror Technology Will Change the Way You Shop*. 2018. <https://www.quytech.com/blog/how-virtual-mirror-technology-will-change-the-way-you-shop/>. Publicado em: 6 jul. 2018. Acesso em: 22 abr. 2025.

WANG, B. A parallel implementation of computing mean average precision. *arXiv preprint arXiv:2206.09504*, 2022.

