

Classificação de áreas de garimpo ilegal na Amazônia com algoritmos de visão computacional

Daniel Martins Monteiro da Silva¹, Dimas Irion Alves², Sarah Negreiros de Carvalho Leite³

Divisão de Engenharia Eletrônica - Instituto Tecnológico de Aeronáutica (ITA) - São José dos Campos/SP - Brasil

Abstract—A atividade de garimpo ilegal na Amazônia tem crescido exponencialmente nos últimos anos, aumentando proporcionalmente o número de áreas e a quantidade de dados que os órgãos de controle ambiental precisam analisar. Aliado a esses fatos, o monitoramento satelital sofre com longos intervalos de revisita e condições meteorológicas desfavoráveis, como a cobertura de nuvens persistente sobre a floresta. Para mitigar essas limitações, o emprego de veículos aéreos, como drones e aeronaves, equipados com sensores ópticos permite a aquisição de imagens de alta resolução. Diante do grande volume de dados, algoritmos de visão computacional despontam como instrumentos eficientes para classificar automaticamente essas áreas. Este estudo implementa e compara dois modelos de diferentes complexidades computacionais – *MobileNetV2* e *EfficientNetV2-S* – na classificação de cenas aéreas da Amazônia. Os resultados indicam que a *MobileNetV2* apresentou o melhor desempenho com acurácia de cerca de 98% na identificação de imagens com garimpo versus sem garimpo.

Index Terms—Visão Computacional, *deep learning*, Amazônia, garimpo.

I. INTRODUÇÃO

O Brasil, em particular a Amazônia, vivencia uma das mais graves crises ambientais de sua história, com o desmatamento e o garimpo ilegal ameaçando ecossistemas vitais. Em 2023, a área queimada na Amazônia atingiu 91.860 km², representando 2,2% do bioma, enquanto o garimpo ilegal causou significativa destruição, especialmente em terras indígenas [1]. A mineração ilegal expandiu-se mais de 1063% na Amazônia entre 1985 e 2023 e, em 2022, a área de garimpo ilegal superou a da mineração industrial no Brasil. Territórios indígenas como Yanomami, Kayapó e Mundurucu estão entre os mais afetados, com a presença de garimpeiros ilegais colocando em risco a saúde de comunidades isoladas e contribuindo para a contaminação por mercúrio [2].

A fiscalização e o combate a essas atividades são dificultados pelo difícil acesso às regiões de mata densa, onde o garimpo se concentra [3]. Nesse contexto, a tecnologia emerge como uma aliada poderosa. A inteligência artificial (IA) e o sensoriamento remoto – por meio de satélites e drones – oferecem novas soluções para detectar incêndios em tempo real, monitorar áreas remotas e mapear regiões afetadas pela exploração ilegal de recursos naturais [1].

O objetivo deste trabalho é desenvolver e comparar modelos de visão computacional para a identificação de atividades

de garimpo ilegal na Amazônia, utilizando imagens de alta resolução obtidas por drones e aeronaves, complementando os esforços de monitoramento ambiental existentes.

O artigo está organizado da seguinte forma: a Seção II apresenta os fundamentos teóricos relacionados ao sensoriamento remoto, com foco na Região Amazônica, e discute o uso de Redes Neurais Convolucionais (CNN) como ferramenta de apoio à classificação de imagens. A Seção III descreve a metodologia adotada para o processamento e preparação do banco de dados, bem como para o treinamento dos modelos. Na Seção IV são expostos os resultados e realizada uma análise comparativa entre os modelos quanto à acurácia e eficiência. Por fim, a Seção V sintetiza as conclusões do estudo e propõe direções para trabalhos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

O monitoramento da Amazônia por sensoriamento remoto é uma prática estabelecida há décadas. O Brasil possui programas consolidados, como o Projeto de Monitoramento do Desmatamento na Amazônia Legal por Satélite (PRODES), que desde 1998 calcula o desmatamento anual com dados de satélites *Landsat*, e o Sistema de Detecção de Desmatamento em Tempo Real (DETER), que, a partir de 2004, utiliza sensores de menor resolução para identificar áreas desmatadas em tempo quase real, enviando alertas às autoridades [4]. Contudo, a análise individual de vastas extensões é demorada e satélites podem ser limitados pela resolução espacial, pelo tempo de revisita e, crucialmente, pela cobertura de nuvens [4], [5].

Para superar essas limitações, tecnologias complementares são exploradas. O programa Brasil MAIS, do Ministério da Justiça e Segurança Pública, já concede acesso a imagens diárias de alta resolução da *Planet* [6]. O sensoriamento por radar (SAR), como o do *Sentinel-1*, pode penetrar nuvens e operar à noite, sendo valioso para detectar embarcações, como balsas de mineração [7]. Paralelamente, drones inteligentes oferecem vantagens como maior detalhamento, menor tempo de revisita e menor sensibilidade à cobertura de nuvens, sendo ideais para missões de fiscalização em áreas remotas [1], [8].

A. Redes Neurais Convolucionais

No campo do aprendizado profundo (*Deep Learning*), as CNNs permitem analisar imagens, sendo eficazes na extração de padrões e características para tarefas como a identificação de mudanças no uso da terra [9]. A Figura 1 ilustra a estrutura clássica de uma CNN. Essa arquitetura é composta por

D. M. M. Silva, monteirodms@ita.br; D. I. Alves, dimasirion@ita.br; S. N. C. Leite, sarahnc@ita.br.

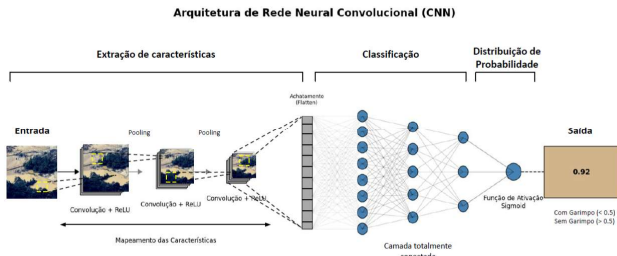


Fig. 1. Estrutura de uma CNN para classificação binária

camadas convolucionais intercaladas por camadas de *pooling*, seguidas por uma camada de *flatten* que converte os mapas de ativação em um vetor unidimensional. Em seguida, ocorre a fase de classificação, realizada por camadas totalmente conectadas (*fully connected*).

A camada convolucional realiza a operação de correlação em cada posição (u, v) da imagem I , utilizando um *kernel* $K \in \mathbb{R}^{m \times n}$

$$S(u, v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} K_{ij} \cdot I_{(u+i)(v+j)} + b, \quad (1)$$

em que K_{ij} representa o peso do *kernel* na posição (i, j) , $I_{(u+i)(v+j)}$ é o valor do pixel da imagem na posição deslocada, b é o viés (*bias*) adicionado ao resultado da operação, e $m \times n$ define as dimensões do *kernel*. O escalar $S(u, v)$ resultante é então passado por uma função de ativação não linear, como a ReLU

$$f(x) = \max(0, x). \quad (2)$$

Em seguida, aplica-se opcionalmente uma operação de *pooling*, responsável por reduzir a resolução espacial dos mapas de ativação. Após sucessivas combinações de camadas convolucionais e de *pooling*, obtém-se um vetor de características h na saída da camada de *flatten*, que reorganiza os dados em uma representação unidimensional.

Esse vetor h é então passado para o classificador, geralmente composto por uma ou mais camadas totalmente conectadas, conforme representado a seguir

$$z = Wh + b, \quad (3)$$

em que W e b são os pesos e o viés da camada totalmente conectada. A saída z é então transformada em probabilidades de classe por meio da função *softmax*

$$\hat{y}_k = \frac{\exp(z_k)}{\sum_j \exp(z_j)}. \quad (4)$$

Como o problema em questão envolve apenas duas classes, a camada *softmax* pode ser substituída por um único neurônio de saída com função de ativação sigmoide

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \quad (5)$$

Valores de $\sigma(z)$ próximos de 1 indicam a classe positiva, enquanto valores próximos de 0 correspondem à classe negativa. Durante o treinamento, todos os parâmetros da rede são ajustados por meio da minimização da função de perda de entropia cruzada

$$\mathcal{L} = - \sum_k y_k \log \hat{y}_k, \quad (6)$$

utilizando o algoritmo de retropropagação do gradiente em conjunto com a descida do gradiente estocástica (SGD) [10]. Esse processo permite que a CNN aprenda representações hierárquicas progressivamente mais abstratas dos padrões visuais presentes nos dados.

1) *Arquiteturas de Referência para Aplicações Embarcadas*: Arquiteturas leves e eficientes são relevantes para a implementação em plataformas com recursos computacionais limitados, como drones. A MobileNetV2 utiliza convoluções separáveis em profundidade (*depthwise separable convolutions*) para reduzir o custo computacional, mantendo uma acurácia competitiva [11]. Sua eficácia em aplicações de monitoramento ambiental com drones foi demonstrada por Suleiman et al. [12], que a empregaram na classificação de imagens de incêndios florestais.

A EfficientNetV2-S é um modelo mais robusto que combina blocos Fused-MBConv com escalonamento composto para otimizar acurácia e eficiência [13]. Ela realiza um escalonamento composto de profundidade, largura e resolução [13], embora apresente uma demanda computacional superior. Um levantamento recente sobre técnicas de *deep learning* aplicadas ao monitoramento ambiental reforça a proeminência dessas arquiteturas [14].

Neste trabalho, ambas as arquiteturas foram implementadas e comparadas no âmbito da detecção de garimpo ilegal na Amazônia a partir de imagens de drones, buscando uma solução que alie alto desempenho e viabilidade operacional.

III. METODOLOGIA

O protocolo experimental proposto neste trabalho é composto por quatro etapas principais: construção da base de dados, pré-processamento, treinamento dos modelos e avaliação do desempenho.

A. Construção da Base de Dados

O conjunto inicial de dados foi composto por 499 imagens rotuladas manualmente, sendo 379 referentes à classe ‘*área com garimpo*’ e 120 à classe ‘*área sem garimpo*’, refletindo um cenário severamente desbalanceado. Todas as imagens foram adquiridas por sensores ópticos embarcados em plataformas aéreas, com resolução nativa de 1920×1080 pixels. As bordas contendo metadados foram removidas, resultando em imagens de 1020×600 pixels.

Para mitigar o viés gerado pelo desbalanceamento, adotou-se uma estratégia baseada em classificação não supervisionada assistida. Um primeiro modelo MobileNetV2 foi treinado no dataset desbalanceado. Este modelo foi então utilizado para classificar 1 319 imagens não rotuladas. Após inspeção

manual das predições, 331 amostras, sendo 132 corretamente classificadas como *área sem garimpo* e 199 falsos positivos, foram incorporadas ao dataset, ampliando significativamente a representatividade da classe minoritária. O dataset balanceado resultante passou a conter 379 imagens da classe *área com garimpo* e 451 da classe *área sem garimpo*.

Com essa base de dados foram treinados dois modelos de redes convolucionais para comparação de desempenho e custo computacional: uma outra MobileNetV2 e uma EfficientNetV2-S.

Vale ressaltar que, em todas as aplicações, a base foi particionada em conjuntos de treinamento (70%), validação (15%) e teste (15%), mantendo a proporção das classes. Para compensar eventuais desequilíbrios residuais, aplicou-se a ponderação de classes via

$$w_k = \frac{N}{C \cdot n_k}, \quad (7)$$

em que N representa o total de amostras, $C = 2$ é o número de classes e n_k a quantidade de amostras na classe k .

B. Pré-processamento

As imagens foram redimensionadas para o tamanho requerido por cada arquitetura: 224×224 para MobileNetV2 e 384×384 para EfficientNetV2S. Aplicou-se as etapas de pré-processamento indicadas especificamente para cada modelo, por meio das funções: `mobilenet_v2.preprocess_input` e `efficientnet_v2.preprocess_input`, respectivamente.

C. Treinamento dos modelos

1) *MobileNetV2*: O modelo adotado neste trabalho utiliza a arquitetura MobileNetV2 [11] pré-treinada na base de dados do ImageNet, com as camadas convolucionais congeladas (`trainable = False`) para atuar como extrator de características. Sobre essa base, foi adicionada uma cabeça de classificação customizada, cuja estrutura é detalhada na Tabela I.

TABLE I
ARQUITETURA DE CLASSIFICAÇÃO PARA O MODELO MOBILENETV2.

Camada	Configuração
<i>Base Model</i>	MobileNetV2 (congelado)
<i>Global Average Pooling 2D</i>	–
<i>Dropout</i>	Taxa = 0,5
<i>Dense (ReLU)</i>	128 neurônios
<i>Dense (Sigmoid)</i>	1 neurônio (saída)

O modelo foi compilado utilizando o otimizador Adam, com taxa de aprendizado de 10^{-3} , função de perda *binary cross-entropy* e acurácia como métrica de avaliação. Durante o treinamento, foram empregados os seguintes callbacks:

- *EarlyStopping*: paciência de 10 épocas, monitorando a perda na validação;
- *ReduceLROnPlateau*: redução da taxa de aprendizado em fator 0,2 após 5 épocas sem melhora na validação;

- *ModelCheckpoint*: salvamento do melhor modelo com base na acurácia de validação;
- *Callback* customizado: encerramento automático do treinamento quando a acurácia de validação atinge 1,0.

Para aumentar a robustez do modelo e mitigar *overfitting*, foi aplicado *data augmentation* sobre o conjunto de treinamento, utilizando as seguintes transformações:

- Rotações aleatórias de até $\pm 30^\circ$;
- Deslocamentos horizontais e verticais de até 0,2;
- Cisalhamento de até 0,2;
- *Zoom* aleatório de até 0,2;
- Espelhamento horizontal;
- Preenchimento das regiões vazias utilizando o método *nearest*.

Após o treinamento inicial, o modelo MobileNetV2 foi salvo e empregado para realizar a etapa de classificação não supervisionada sobre um conjunto de 1.319 imagens não rotuladas. A partir desse processo, e após análise manual das predições, foram incorporadas ao dataset inicial 331 imagens pertencentes à classe minoritária, com o objetivo de promover o balanceamento das classes. Em seguida, o modelo foi re-treinado utilizando o dataset balanceado, mantendo exatamente os mesmos hiperparâmetros e configurações adotados no treinamento anterior da MobileNetV2.

2) *EfficientNetV2S*: O modelo adotado utiliza a arquitetura EfficientNetV2S como base, pré-treinada no ImageNet [13], com as camadas convolucionais congeladas (`trainable = False`) durante o treinamento, atuando como extrator de características. Sobre essa base, foi implementada uma cabeça de classificação, detalhada na Tabela II.

TABLE II
ARQUITETURA DE CLASSIFICAÇÃO PARA O MODELO EFFICIENTNETV2S.

Camada	Configuração
<i>Base Model</i>	EfficientNetV2-S (congelado)
<i>Global Average Pooling 2D</i>	–
<i>Dropout</i>	Taxa = 0,3
<i>Dense (ReLU)</i>	128 neurônios
<i>Dense (Sigmoid)</i>	1 neurônio (saída)

O modelo foi compilado utilizando o otimizador Adam, com taxa de aprendizado de 10^{-3} , função de perda *binary cross-entropy* e acurácia como métrica de avaliação. Durante o treinamento foram empregados os seguintes *callbacks*:

- *EarlyStopping*: paciência de 8 épocas, monitorando a perda na validação;
- *ReduceLROnPlateau*: redução da taxa de aprendizado em fator 0.2 após 4 épocas sem melhora na validação;
- *ModelCheckpoint*: salvamento do melhor modelo com base na acurácia de validação.

Além disso, foi aplicado *data augmentation* durante o treinamento, com as mesmas configurações descritas para a MobileNetV2.

TABLE III
COMPARAÇÃO DAS MÉTRICAS POR CLASSE E ACURÁCIA GLOBAL DOS MODELOS AVALIADOS.

Modelo	Acurácia	Acc. Ponderada	Com Garimpo			Sem Garimpo		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
MobileNetV2 (Desbalanceado)	0,96	0,96	0,95	1,00	0,97	1,00	0,83	0,91
Classificação Não Supervisionada	0,84	0,84	0,83	0,99	0,90	0,92	0,40	0,56
MobileNetV2 (Balanceado)	0,98	0,98	0,95	1,00	0,97	1,00	0,96	0,98
EfficientNetV2S (Balanceado)	0,97	0,97	0,93	1,00	0,97	1,00	0,94	0,97

IV. RESULTADOS E DISCUSSÃO

A Tabela III resume as métricas de desempenho dos quatro cenários avaliados. Nota-se que os modelos treinados com o dataset balanceado (MobileNetV2 e EfficientNetV2S) apresentaram desempenhos superiores e equilibrados entre as classes.

O modelo MobileNetV2, quando treinado no dataset desbalanceado, apresentou acurácia de 0,84, com forte viés para a classe majoritária (*área com garimpo*). O F1-score para a classe *sem garimpo* foi de apenas 0,40, evidenciando uma baixa capacidade de identificar corretamente regiões sem mineração. A matriz de confusão (Fig. 2) da Classificação Não Supervisionada revelou 199 falsos negativos para essa classe, ou seja, áreas sem atividade de garimpo identificadas incorretamente como áreas com garimpo pelo modelo.

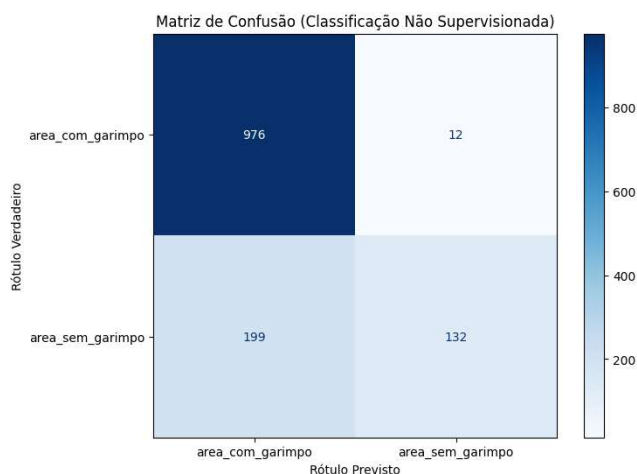


Fig. 2. Matriz de confusão da classificação não supervisionada da MobileNetV2 com dados desbalanceados. Elevado número de falsos negativos na classe *área sem garimpo*.

Após a etapa de classificação não supervisionada, seguida de curadoria manual, a inclusão de 331 amostras da classe minoritária produziu um dataset balanceado. O *retraining* da MobileNetV2 neste conjunto elevou a acurácia para 0,978 e os F1-scores de ambas as classes ultrapassaram 0,97. A matriz de confusão indicou uma drástica redução na taxa de falsos negativos, de 0,60 para apenas 0,04.

A EfficientNetV2S, embora apresente uma ligeira redução na acurácia final (0,970), manteve o mesmo equilíbrio nas métricas por classe (F1-score de 0,97 para ambas).

A análise de custo computacional, baseada no cálculo de FLOPs, revelou que a MobileNetV2 executa 0.6 GFLOPs por imagem, enquanto a EfficientNetV2S demanda aproximadamente 6,5 GFLOPs. Isso representa um aumento de mais de 10 vezes no custo computacional da EfficientNetV2S, sem ganhos substanciais de acurácia.

A. Erros de classificação

A Fig. 3 ilustra cenas de falsos-negativos, que correspondem a cenas da classe 1 (*área sem garimpo*) que o modelo classificou como “com garimpo”. Observa-se solo exposto, clareiras recentes e pequenas estradas de acesso — indícios visuais típicos de frentes de mineração —, embora ali não haja atividade garimpeira. A presença de áreas desmatadas, tanques de água esverdeada e telhados metálicos leva a rede a ativar filtros aprendidos para a classe 0. A falta de pistas específicas, como maquinário pesado ou balsas dragas, sugere que o classificador depende fortemente da textura de solo nu, gerando erro quando essa característica aparece fora do contexto de mineração.

Já os falsos-positivos referem-se a imagens da classe 0 (*área com garimpo*) rotuladas como “sem garimpo”. Na Fig. 4, os pontos de garimpo situam-se em pleno leito de rio, onde balsas dragas e pontões metálicos ficam parcialmente submersos ou camuflados pelo reflexo da lâmina d’água. A homogeneidade cromática do rio, aliada ao pequeno porte das embarcações na escala 224×224 px, reduz a saliência dos traços estruturais que a rede aprendeu para detectar mineração, fazendo-a subestimar a classe positiva. Estratégias, como uso de janelas maiores, incorporação de bandas NIR/SWIR para realçar plumas de sedimento e *data augmentation* específico para rios podem mitigar esses erros.

No que tange as condições operacionais, os resultados indicam que a MobileNetV2 balanceada atende aos requisitos de missões a bordo de *small UAVs*: acurácia elevada, baixa latência (14 ms) e $\approx 2,5$ M parâmetros. Já a EfficientNet-V2S é recomendada para estações em solo ou veículos aéreos com maior orçamento energético, obtendo leve ganho em robustez ao custo de maior consumo computacional.

V. CONCLUSÃO

Este trabalho desenvolveu e avaliou um sistema de classificação de imagens aéreas da Amazônia para detecção de áreas com garimpo ilegal, utilizando arquiteturas de redes neurais convolucionais. Inicialmente, foi treinado um modelo



Fig. 3. Exemplos de falsos negativos da classificação não supervisionada: Áreas sem atividade de garimpo identificadas incorretamente como áreas com garimpo.



Fig. 4. Exemplos de falsos positivos da classificação não supervisionada: Áreas com atividade de garimpo identificadas incorretamente como áreas sem garimpo.

MobileNetV2 sobre um dataset desbalanceado, cuja baixa capacidade de generalização para a classe minoritária evidenciou a necessidade de um balanceamento.

A estratégia de classificação não supervisionada, seguida de curadoria manual, demonstrou-se eficaz na expansão do dataset e na melhoria das métricas de desempenho, elevando o F1-score da classe minoritária de 0,46 para 0,97. A arquitetura MobileNetV2, após balanceamento, apresentou acurácia de 0,978, desempenho comparável ao da EfficientNetV2S, mas com uma fração do custo computacional (0,6 GFLOPs versus 6,5 GFLOPs).

Futuras pesquisas pretendem incluir a detecção de objetos, como dragas e estruturas associadas ao garimpo ilegal.

REFERENCES

- [1] Inatel. (2024, Oct.) A tecnologia como resposta à crise ambiental no brasil. Blog do Inatel. [Online]. Available: <https://www.inatel.br/blog/a-tecnologia-como-resposta-a-crise-ambiental-no-brasil>
- [2] M. Lang, J. Silva, and F. Rocha, "Monitoramento da amazônia: desafios e estratégias," *Revista Brasileira de Geoinformação*, vol. 17, no. 1, pp. 12–28, 2025.
- [3] J. Fernandes and R. Filho, "Desafios logísticos no combate ao garimpo ilegal na amazônia," Relatório técnico, 2016, disponível mediante solicitação à instituição responsável.
- [4] N. E. Observatory, "Amazon deforestation," Online, 2019, disponível em: <https://earthobservatory.nasa.gov/features/Deforestation>.
- [5] R. Pratiwi, A. Nugroho, and H. Permana, "Application of cnns in illegal activity detection from aerial images," *International Journal of Remote Sensing*, vol. 42, no. 12, pp. 4635–4654, 2021.
- [6] P. Torres, R. Almeida, and M. Costa, "Impacto do mercúrio em comunidades indígenas da amazônia," *Saúde e Meio Ambiente*, vol. 11, no. 2, pp. 55–67, 2022.
- [7] J. Lee, E. Lin, M. Wang, and S. Maity, "Computer vision for detection of illegal mining barges in the rio madeira," Harvard College project report, 2021, disponível em: https://projects.iq.harvard.edu/files/illegal-mining/files/computer_vision_for_detection_of_illegal_mining_barges_in_the_rio_madeira_1.pdf.
- [8] Aeroscan. (2025, May) Drones inteligentes: o que são, para que servem e como estão transformando o mercado. Aeroscan Blog. [Online]. Available: <https://www.aeroscan.com.br/blog/drones-inteligentes-o-que-sao-para-que-servem-e-como-estao-transformando-o-mercado/>
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: <https://www.deeplearningbook.org>
- [10] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th world symposium on applied machine intelligence and informatics (SAMI)*. IEEE, 2020, pp. 281–286.
- [11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," arXiv preprint arXiv:1801.04381, 2018, disponível em: <https://arxiv.org/abs/1801.04381>.
- [12] D. Suleiman, A. Al-Azzawi *et al.*, "Employing cnn mobilenetv2 and ensemble models in classifying drones forest fire detection images," *International Journal of Data and Network Science*, vol. 9, pp. 297–316, 2025.
- [13] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," *arXiv preprint arXiv:2104.00298*, 2021.
- [14] A. Faustine, A. Micheli, and P. Gamba, "A review of deep learning techniques for environmental monitoring with remote sensing images," *Remote Sensing*, vol. 14, no. 5, p. 1150, 2022.