

Detecção Automática de Pistas de Pouso Clandestinas na Amazônia Utilizando Técnicas de Aprendizado de Máquina

Gabriel R. Pardini¹, Gabriel Lessa de Araújo¹, Thiago José dos Santos Freitas¹, Elcio H. Shiguemori², Tahisa N. Kuck² e Marcos R. O. A. Máximo¹

¹Instituto Tecnológico de Aeronáutica - ITA, São José dos Campos/SP - Brasil

²Instituto de Estudos Avançados - IEAv, São José dos Campos/SP - Brasil

Resumo— A floresta amazônica é frequentemente alvo de atividades ilegais, como a mineração clandestina, que utiliza pistas de pouso para facilitar o transporte de materiais. Este estudo visa desenvolver uma metodologia automatizada para identificar essas pistas de pouso, utilizando técnicas de aprendizado de máquina. Foram aplicados modelos de *Transfer Learning* em combinação com diversos algoritmos de classificação, incluindo *Decision Tree*, *Random Forest*, *Naive Bayes* e *AdaBoost*. Os dados foram pré-processados e rotulados, e o desempenho dos algoritmos foi rigorosamente avaliado utilizando validação cruzada externa e *GridSearch* para otimizar os hiperparâmetros de cada modelo. Entre os modelos testados, o *XgBoost* apresentou a melhor acurácia média de 91%. No entanto, o modelo *AdaBoost* também obteve um resultado notável, alcançando uma acurácia de 89%, o que destaca sua eficácia e potencial como uma alternativa viável para a tarefa. A diferença de apenas 2% em comparação com o melhor modelo (*XgBoost*) demonstra que o *AdaBoost* é altamente competitivo, oferecendo uma combinação eficiente de simplicidade e precisão. Dessa forma, a metodologia proposta, visa considerar o desempenho de *AdaBoost*, que demonstra grande viabilidade para a automatização da detecção desses alvos, consolidando-se como uma abordagem robusta para identificar pistas de pouso ilegais na floresta amazônica.

Palavras-Chave— Inteligência artificial, Redes Neurais, Amazônia.

I. INTRODUÇÃO

A Amazônia é um dos biomas mais ricos e extensos da Terra, compreendendo mais da metade do território brasileiro [1]. A região é frequentemente alvo de atividades ilegais, sendo a mineração ilegal em terras indígenas uma das mais proeminentes [2], [3]. Por exemplo, em 2021, 91% da mineração brasileira foi realizada na Amazônia Legal [4]. Devido à alta densidade florestal, esses locais muitas vezes se encontram em regiões remotas, o que cria problemas logísticos.

Nesse sentido, as pistas de pouso não só facilitam a entrada e saída de equipamentos e pessoas, mas também o escoamento de minérios extraídos ilegalmente, contribuindo significativamente para a degradação ambiental e a violação dos direitos das comunidades indígenas [2].

Portanto, o mapeamento constante da localização das pistas de pouso neste bioma torna-se relevante, pois pode fornecer

informações cruciais sobre pontos críticos de atividades criminosas. Essa prática permite que as autoridades identifiquem e monitorem as áreas de maior risco, possibilitando ações mais eficazes de fiscalização e combate à mineração ilegal [4]. Entretanto, o fato dos alvos estarem em regiões remotas e com densa cobertura florestal, dificultam que o mapeamento seja feito de forma presencial.

No entanto, devido à vasta área da Amazônia, essa tarefa torna-se muito demorada e trabalhosa. O último mapeamento realizado por MapBiomas [5], usando imagens de 2021, foi feito por inspeção visual, ou seja, sem rotinas de automação, tornando a reprodução muito custosa. Nesse sentido, metodologias que permitem a automação da detecção tornam-se necessárias.

Após o trabalho seminal de [6] o uso de técnicas de inteligência artificial tornaram-se o estado da arte na automação de tarefas relacionadas à visão computacional. Entre essas, o *Transfer Learning* (TL) [7] se destaca, pois tem a capacidade de lidar com eficiência com uma pequena quantidade de dados. O TL envolve o uso de modelos pré-treinados em grandes conjuntos de dados e ajustá-los para tarefas específicas, o que é especialmente útil quando se tem acesso limitado a dados de treinamento. Esse método permite que modelos pré-treinados aprendam características gerais úteis e depois sejam adaptados para tarefas mais específicas, como o uso em classificação de imagens, acelerando o processo de treinamento e melhorando a eficiência geral do modelo. Normalmente, a técnica de TL é aplicada conjuntamente com redes neurais, que é um modelo computacional que se inspira no cérebro humano, organizada em camadas e que tem um processo de treinamento para entender diferentes padrões.

Nesse contexto, este trabalho aborda o uso de técnicas de TL aplicadas a vários algoritmos de classificação, com o objetivo de classificar automaticamente se a imagem analisada contém ou não uma pista de pouso. A abordagem utiliza um modelo pré-treinado em grandes conjuntos de dados, permitindo que a rede neural aproveite o conhecimento adquirido previamente para aprimorar a detecção de pistas de pouso em novas imagens.

O presente artigo é organizado da seguinte maneira: Introdução, Descrição do Problema e Trabalhos Relacionados, Materiais e Métodos, Resultados e Discussão, Conclusões.

II. DESCRIÇÃO DO PROBLEMA E TRABALHOS RELACIONADOS

A aplicação de técnicas de inteligência artificial (IA) e aprendizado de máquina (ML) em problemas de detecção e classificação de imagens tem crescido significativamente nos últimos anos [6], [8]. Um dos primeiros trabalhos relevantes foi realizado por LeCun et al. (1998), que demonstrou a eficácia das redes neurais em tarefas de visão computacional [9]. Desde então, diversas abordagens e melhorias foram propostas, incluindo o uso de técnicas de TL, que permitem a adaptação de modelos pré-treinados para novos domínios com conjuntos de dados limitados [10].

Na área específica de detecção de pistas de pouso e atividades ilegais na Amazônia, ainda há uma lacuna significativa na literatura. A maioria dos trabalhos focados em mapeamento da região utiliza métodos convencionais de processamento de imagens, sem incorporar técnicas avançadas de aprendizado de máquina. O estudo mais próximo foi realizado por MapBiomias (2021), que, apesar de fornecer um mapeamento detalhado, dependeu fortemente de inspeção visual manual, limitando sua escalabilidade e frequência de atualizações [5]. Tais estudos, apesar de serem eficazes, são ineficientes em relação ao aspecto temporal, devido a escala da Floresta Amazônica, dessa forma a utilização de técnicas convencionais tornariam o processo muito lento.

O uso de técnicas de classificação é um estudo bem fundamentado. As árvores de decisão (*Decision Tree*) [11] são um dos modelos mais antigos para tais tarefas, tendo resultados promissores em diversos problemas. Entretanto, essa modelagem sofre com a suscetibilidade ao *Overfitting*, ou seja, quando o modelo se ajusta demais aos dados de treinamento, perdendo a capacidade de generalização para novos dados [12].

Além disso, o *Naive Bayes* [13] adota uma abordagem distinta das árvores de decisão. Esse modelo utiliza uma abordagem probabilística, sendo robusto a ruídos e capaz de fornecer previsões rápidas e interpretáveis. No entanto, o classificador Bayesiano baseia-se na premissa de que os atributos são independentes dado a classe, o que pode não ser verdade para diversos domínios, limitando sua aplicabilidade em casos onde as características dos dados são fortemente correlacionadas.

Recentemente, técnicas de *ensemble learning*, como *Random Forest*, *AdaBoost* e *XgBoost*, têm demonstrado um desempenho superior em várias tarefas de classificação de imagens. *Random Forest* combina várias árvores de decisão para reduzir o risco de *Overfitting* e melhorar a robustez do modelo [14]. Esse método cria múltiplas árvores de decisão treinadas em diferentes subconjuntos dos dados. As previsões finais são obtidas através de uma votação ou média das previsões individuais das árvores, proporcionando maior estabilidade e precisão.

AdaBoost e *XGBoost*, por outro lado, são algoritmos de boosting que melhoram a precisão dos modelos de aprendizado de máquina combinando vários modelos simples e fracos em um único modelo robusto [15]. Eles atribuem pesos diferentes às instâncias de dados, permitindo que os modelos subsequentes deem mais atenção aos casos difíceis de classificar. Isso resulta em um modelo final capaz de generalizar melhor e lidar com dados mais complexos.

A fim de se verificar qual técnica selecionada tem o melhor resultado, o teste de Friedman [16] torna-se relevante. Tal teste é utilizado para detectar diferenças significativas entre algoritmos. A hipótese nula do teste postula que todos os algoritmos testados têm desempenhos equivalentes. Caso essa hipótese seja rejeitada, é possível afirmar que pelo menos um dos algoritmos apresenta um desempenho estatisticamente diferente dos demais. Para rejeitar a hipótese nula, utiliza-se um nível de confiança de 95%. Caso a hipótese nula não seja comprovada, para verificar as diferenças estatísticas entre os algoritmos utiliza-se o teste Pós-hoc Nemenyi [17], para verificar qual algoritmo tem diferenças estatísticas relevantes em relação ao outro.

No contexto do presente estudo, a aplicação de técnicas de TL e *ensemble learning* para a detecção automática de pistas de pouso na Amazônia visa preencher essa lacuna na literatura, oferecendo uma solução mais eficiente e escalável para monitorar e combater a mineração ilegal na região. Ao explorar a eficácia de diferentes algoritmos de aprendizado de máquina, buscamos identificar a melhor abordagem para este problema específico, contribuindo assim para a proteção do bioma amazônico e a sustentabilidade das comunidades locais.

III. MATERIAIS E MÉTODOS

Nessa seção, serão dispostos os materiais e métodos que foram utilizados no presente estudo.

A. Materiais

Para a realização deste estudo, foi utilizado um conjunto de imagens da Amazônia, focado na identificação de pistas de pouso clandestinas. As imagens foram obtidas do servidor de imagens de satélite *Image © 2023 Planet Labs PB* [18], que fornece imagens com quatro bandas ópticas: Vermelho, Verde, Azul e Infravermelho Próximo (NIR). Cada banda possui uma resolução radiométrica de 16 bits por canal, e a resolução espacial das imagens é de 4,7 metros, o que permite identificar e anotar características como pistas de pouso com precisão.

Para obter as imagens, utilizou-se o mapeamento do MapBiomias Brasil 2021 [5], que fornece a localização geográfica detalhada de pistas de pouso na Amazônia. As coordenadas fornecidas foram associadas às imagens de satélite, permitindo a criação de conjuntos de dados precisos para o treinamento das técnicas de aprendizado de máquina.

Além disso, realizou-se cortes quadrados de 2×2 quilômetros ao redor de todas as localizações mapeadas das pistas de pouso. Este processo envolveu a criação de imagens menores centradas nas coordenadas fornecidas pelo MapBiomias, garantindo que cada corte capturasse a área ao redor das pistas de pouso suspeitas.

Em seguida, foi realizada uma inspeção visual dessas imagens. Durante esta etapa, foram excluídas imagens que não correspondiam realmente a pistas de pouso, como, por exemplo, aquelas cobertas por nuvens, que impediam a visualização clara do alvo. Essa inspeção foi crucial para assegurar a qualidade e a precisão dos dados, garantindo que apenas imagens válidas fossem incluídas no conjunto de dados. Após esta filtragem, um total de 1.987 imagens de pistas de pouso foram selecionadas do servidor Planet.

Além disso, foi realizado um pré-processamento nas imagens, removendo-se a banda de infravermelho próximo (NIR), a fim de se obter apenas três canais de bandas ópticas na faixa do visível (Vermelho, Verde e Azul). Este procedimento teve como objetivo padronizar as imagens para análises subsequentes, garantindo a consistência dos dados ao focar exclusivamente nas informações disponíveis nas bandas RGB.

A remoção da banda NIR foi motivada pela necessidade de simplificação e uniformidade no conjunto de dados, considerando que muitas técnicas de aprendizado de máquina e modelos de classificação são inicialmente projetados para trabalhar com imagens RGB. Além disso, a exclusão da banda NIR ajuda a evitar potenciais complicações ou necessidades de ajustes específicos para dados multiespectrais, facilitando o uso de ferramentas e algoritmos comuns no processamento de imagens.

O conjunto de dados foi então dividido em duas categorias principais: imagens contendo pistas de pouso e imagens sem pistas de pouso. Para as imagens sem pistas de pouso, foi realizada uma amostragem aleatória de imagens da Amazônia, assegurando-se que a quantidade de imagens sem pistas de pouso fosse equivalente à quantidade de imagens contendo pistas de pouso. Todas as imagens amostradas são verificadas para garantir que não contenham pistas de pouso. Em seguida, as imagens selecionadas passaram pelo mesmo processo de pré-processamento descrito anteriormente, garantindo a consistência e a comparabilidade entre os conjuntos de dados. Na Figura 1 estão dispostas quatro amostras de imagens que compõem o *dataset*, duas contendo pistas e duas não contendo.

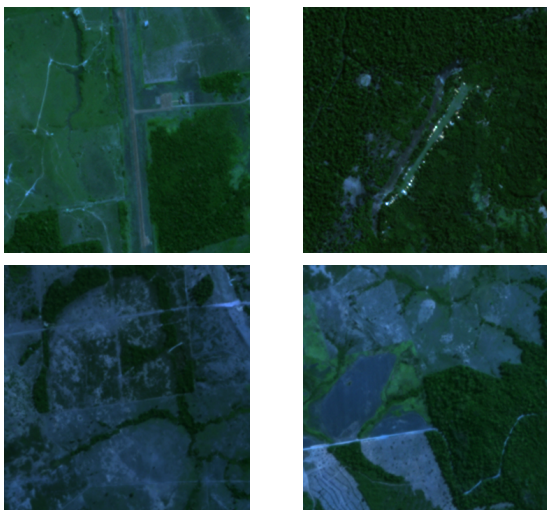


Fig. 1. Exemplos de imagens do *dataset* que contém pistas de pouso e duas imagens que não contém o alvo.

Os principais materiais e ferramentas utilizados incluem:

- **Ambiente de Programação:** O estudo foi desenvolvido utilizando o ambiente de programação Python, com as bibliotecas TensorFlow [19] e Keras [20] para a construção e treinamento dos modelos de aprendizado de máquina.
- **Hardware:** Todos os experimentos foram realizados em um computador com um processador AMD Ryzen 9 7900X, 256 GB de RAM e uma GPU Nvidia RTX 4090.
- **Software:** Ferramentas adicionais como OpenCV foram empregadas para a manipulação e pré-processamento das imagens. A biblioteca Pandas [21] foi utilizada para

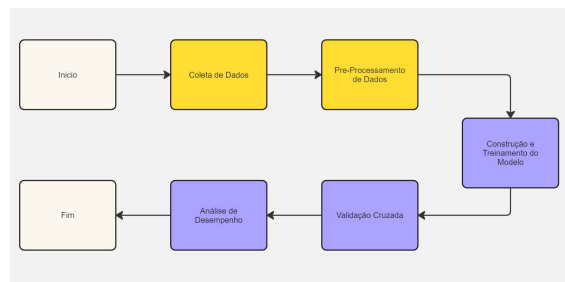


Fig. 2. Fluxograma do Projeto

a manipulação dos dados, e Matplotlib [22] para a visualização dos resultados.

B. Métodos

O métodos seguiram o fluxograma mostrado na Figura 2.

1) Pré-processamento dos Dados:

- **Leitura e Organização das Imagens:** Inicialmente, as imagens foram carregadas a partir de diretórios específicos, um contendo imagens de pistas de pouso e outro com imagens sem pistas de pouso. As listas de imagens foram embaralhadas para garantir a aleatoriedade dos dados durante o treinamento e a validação.
- **Rotulagem das Imagens:** As imagens foram rotuladas de acordo com sua categoria: 1 para imagens contendo pistas de pouso e 0 para imagens sem pistas de pouso. Esses rótulos foram armazenados em um dicionário para facilitar o acesso durante o treinamento do modelo.

2) Construção e Treinamento do Modelo:

- **Arquitetura do Modelo:** Utilizou-se a arquitetura VGG16 pré-treinada no ImageNet, uma rede neural convolucional (CNN) amplamente utilizada para tarefas de classificação de imagens. A VGG16 foi escolhida por sua eficácia comprovada em diversas competições de visão computacional [23].
- **Transfer Learning:** O TL aproveita o conhecimento prévio de uma rede neural, como a VGG16, treinada em um grande conjunto de dados. Para isso, remove-se as quatro camadas finais da VGG16, que são responsáveis pela classificação, mantendo apenas as camadas convolucionais, o que gera um vetor de características de tamanho de 25088, pois na última camada convolucional há uma matriz de $7 \times 7 \times 512$.
- **Treinamento:** O modelo foi treinado utilizando técnicas de augmentação de dados para aumentar a diversidade das amostras e prevenir *Overfitting*. Foram aplicados flips horizontais e verticais, além de ajustes de contraste nas imagens. Além disso, o treinamento foi realizado apenas no classificador final, a rede convolucional não foi treinada, sendo utilizada apenas para a extração de características.
- **Validação cruzada k-fold:** É uma técnica de validação que divide o conjunto de dados em k partes (*folds*) [24] de tamanho igual. O modelo é treinado k vezes, usando $k - 1$ partes como conjunto de treino e a parte restante como conjunto de validação. Isso permite uma avaliação robusta do modelo, utilizando diferentes combinações de dados de treino e validação. Será aplicado nesse modelo

80% dos dados e 20% será utilizado para um conjunto de testes separado.

- **Grid Search:** É uma abordagem sistemática para encontrar os melhores hiperparâmetros para um modelo. Todas as combinações possíveis de hiperparâmetros especificadas em um grid são testadas, e cada combinação é avaliada usando validação cruzada (neste caso, com 3 folds). Isso ajuda a determinar os hiperparâmetros que otimizam o desempenho do modelo sem a necessidade de ajuste manual [25].

3) Avaliação do Modelo:

- **Teste do Modelo:** As métricas de desempenho, incluindo a acurácia média e o desvio padrão, foram calculadas para cada algoritmo testado: *Decision Tree*, *Random Forest*, *Naive Bayes*, *AdaBoost* e *XgBoost*.
- **Análise de Desempenho:** Os resultados foram analisados e comparados, identificando o algoritmo com melhor desempenho na classificação utilizando o teste de Friedman [16] e após utilizando o teste Pós-hoc Nemenyi [17]. Após isso, uma análise a respeito dos valores de acurácia é feito.

IV. RESULTADOS E DISCUSSÃO

Todos os códigos relacionados podem ser acessados nesse link: <https://drive.google.com/file/d/1NiWVzn-tVF4ryF5MN7wNISKbKfIgeU1j/view?usp>.

A. Análise estatística

Para a execução do problema, foram utilizados quatro algoritmos de aprendizado de máquina, para verificar a coerência dos resultados: *Decision Tree*, *Random Forest*, *Naive Bayes*, e *AdaBoost*. Cada algoritmo foi ajustado usando Grid Search e validação cruzada (5-fold). O motivo para se utilizar 5 execuções advém ao tempo demorado para o treinamento das técnicas, cada fold demora, em média, mais de uma hora para ser executado, dessa forma, um número maior geraria um aumento significativo no tempo da validação cruzada. Foram utilizados os parâmetros mostrados na Tabela I. Na Tabela II estão os resultados de acurácia para as 5 execuções da validação cruzada e na Figura 3 está representado tais execuções em um boxplot. Na Tabela III está a média e desvio padrão para as execuções.

TABELA I

DESEMPENHO DOS ALGORITMOS COM DIFERENTES HIPERPARÂMETROS, SENDO OS EM NEGRITO OS PARÂMETROS QUE TIVERAM MELHOR ACURÁCIA.

Algoritmo	Hiperparâmetros
<i>Decision Tree</i>	Profundidade máxima: 5, 10 , 20
<i>Random Forest</i>	Número de estimadores: 50, 100, 200
<i>Naive Bayes</i>	Número de características (k): 3, 5, 10
<i>AdaBoost</i>	Número de estimadores: 50, 100, 200
<i>XgBoost</i>	Número de estimadores: 50, 100, 200

Com base nos dados da Tabela II, foram calculados a estatística de Friedman e o valor de p (p -value), resultando em 14,093 e 0,003, respectivamente. Como o valor de p é menor que o nível de significância de 0,05, rejeitamos a hipótese nula. Isso nos permite concluir que há evidência estatística para afirmar que pelo menos um dos algoritmos é significativamente diferente dos outros.

TABELA II
RESULTADOS DAS EXECUÇÕES DOS ALGORITMOS DE APRENDIZADO DE MÁQUINA.

Execução	<i>Decision Tree</i>	<i>Random Forest</i>	<i>Naive Bayes</i>	<i>AdaBoost</i>	<i>XgBoost</i>
0	0,799	0,872	0,815	0,887	0,908
1	0,832	0,884	0,809	0,895	0,915
2	0,791	0,848	0,777	0,862	0,910
3	0,819	0,885	0,767	0,904	0,907
4	0,808	0,848	0,782	0,880	0,901

TABELA III
DESEMPENHO MÉDIO DOS ALGORITMOS.

Algoritmo	Acurácia Média \pm Desvio Padrão
<i>Decision Tree</i>	0,81 \pm 0,02
<i>Random Forest</i>	0,87 \pm 0,02
<i>Naive Bayes</i>	0,79 \pm 0,02
<i>AdaBoost</i>	0,89 \pm 0,02
<i>XgBoost</i>	0,91 \pm 0,01

Para verificar possíveis diferenças estatísticas entre os algoritmos, utilizou-se o teste Pós-hoc de Nemenyi. Os resultados estão resumidos na Tabela IV, onde os valores de p indicam as comparações estatísticas entre cada par de algoritmos. Um valor de p menor que 0,05 foi considerado como indicativo de diferença estatisticamente significativa, com um nível de confiança de 95%. Da análise da tabela, observa-se que há diferença estatística entre os algoritmos *Decision Tree* e *XgBoost*, assim como entre *Naive Bayes* e *XgBoost*. Portanto, com base nos resultados obtidos, pode-se afirmar que o *XgBoost* apresenta desempenho estatisticamente superior ao *Decision Tree* e ao *Naive Bayes*.

TABELA IV
VALORES P PARA O TESTE DE NEMENYI PARA OS ALGORITMOS TESTADOS.

	<i>Decision Tree</i>	<i>Random Forest</i>	<i>Naive Bayes</i>	<i>AdaBoost</i>	<i>XgBoost</i>
<i>Decision Tree</i>	1.0000	0.7243	0.9000	0.1794	0.0119
<i>Random Forest</i>	0.7243	1.0000	0.3744	0.8378	0.2659
<i>Naive Bayes</i>	0.9000	0.3744	1.0000	0.0409	0.0013
<i>AdaBoost</i>	0.1794	0.8378	0.0409	1.0000	0.8378
<i>XgBoost</i>	0.0119	0.2659	0.0013	0.8378	1.0000

B. Análise qualitativa

Primeiramente, realizando uma comparação temporal, na Tabela V estão demonstrados os tempos de treinamento e de inferência para os algoritmos testados. Percebe-se que, apesar dos tempos de treinamento terem escala de horas, tal resultado é promissor, pois os tempos de inferência para cada imagem é relativamente pequeno. Além disso, percebe-se que o *XgBoost*, apesar de ter a melhor performance, tem um tempo de inferência pior, o que é um trade-off que deve ser considerado ao se aplicar tal técnica. Para a Tabela VI os modelos foram aplicados no conjunto de teste separado, cada modelo foi treinado em toda a base que foi utilizada para a validação cruzada.

A comparação entre *Random Forest*, *AdaBoost* e *XGBoost* indica que não há uma diferença expressiva significativa entre esses algoritmos. No entanto, ao analisar os resultados

TABELA V

TEMPO DE TREINAMENTO DOS ALGORITMO (1 FOLD) E TEMPO DE INFERÊNCIA MÉDIO APLICADO AO CONJUNTO DE TESTES.

Algoritmo	Tempo: Treinamento (h)	Tempo: Inferência (ms)
<i>Decision Tree</i>	1:03	23
<i>Random Forest</i>	1:25	29
<i>Naive Bayes</i>	0:47	15
<i>AdaBoost</i>	2:17	33
<i>XgBoost</i>	2:23	35

TABELA VI

DESEMPENHO DO MELHOR ALGORITMO NUM CONJUNTO DE TESTE SEPARADO.

Algoritmo	Acurácia	Precisão	Sensibilidade
<i>Decision Tree</i>	0,821	0,811	0,844
<i>Random Forest</i>	0,887	0,865	0,919
<i>Naive Bayes</i>	0,791	0,782	0,813
<i>AdaBoost</i>	0,903	0,909	0,899
<i>XgBoost</i>	0,905	0,915	0,902

das Tabelas III e VI, observa-se que o *XGBoost* apresenta acurácia superior em todas as execuções. Portanto, apesar de não haver alta confiança estatística, há indícios de que o *XGBoost* demonstra um desempenho geral melhor. Para determinar o melhor algoritmo, todo o processo foi conduzido utilizando 80% da base de dados, revelando que o número ideal de estimadores para o *XGBoost* foi 200. Esses modelos de *ensemble* mostraram um desempenho superior aos modelos individuais, como *Decision Tree* e *Naive Bayes*.

Como informado o *Naive Bayes* apresentou a pior performance. Isso pode ser atribuído à premissa de independência condicional entre características, mostrando-se, aqui, não verdadeira para imagens. Além disso, o fato de imagens não serem dados discretizados atrapalha a performance final, haja visto a dificuldade deste algoritmo em atributos contínuos. Um outro ponto importante é que os atributos são extraídos de uma CNN e operações de convolução são aplicadas em mais de uma célula da matriz, então os atributos são inerentemente correlacionados.

A técnica de *Decision Tree* apresentou uma acurácia moderada para o problema em questão. Observou-se que essa técnica não mostrou grande instabilidade nos resultados, conforme ilustrado na Tabela II. Uma hipótese para essa estabilidade é a limitação imposta no tamanho da árvore durante a execução, o que restringiu sua capacidade de representação, mas reduziu o *Overfitting* que poderia ter sido gerado. Por outro lado, o *Random Forest* demonstrou ser mais robusto que a *Decision Tree*. Esse método combina diversas árvores para diminuir o *Overfitting*, o que é especialmente eficaz dada a extensão da nossa base de dados.

O melhor desempenho foi obtido com os modelos *AdaBoost* e *XgBoost*, cuja performance pode ser explicada pela técnica de *boosting*, tal técnica ajusta-se iterativamente, penalizando os classificadores incorretos e permitindo que o novo classificador se concentre nos exemplos incorretamente classificados, dessa forma, o comitê de classificadores se ajusta melhor ao tipo de dado, mantendo a generalização do modelo e, conseqüentemente, tendo uma acurácia superior. Além disso, toda a base de dados foi inspecionada visualmente, o que reduziu consideravelmente os ruídos nos dados e facilitou o bom desempenho das técnicas baseadas em *boosting*, pois o principal problema dessa técnica é a grande quantidade de

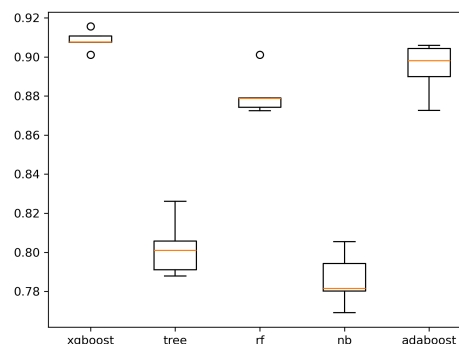


Fig. 3. Boxplot com a performance dos algoritmos nas 5 execuções da validação cruzada.

ruídos que possam ter na base de dados. Numa comparação direta entre os modelos, percebe-se que o *XgBoost* tem um resultado superior, devido a ser um modelo mais novo que apresenta a técnica mencionada de forma mais eficiente.

Em todo modelo, há saídas erradas. No caso de um modelo que visa reconhecer locais de crimes ambientais, é imprescindível que seja entendido quais são os falsos positivos e falsos negativos e como eles são reconhecidos pelo operador. Na Figura 4 está ilustrada uma amostra de 2 erros que foram gerados pelo *AdaBoost*, em uma das execuções da validação cruzada.

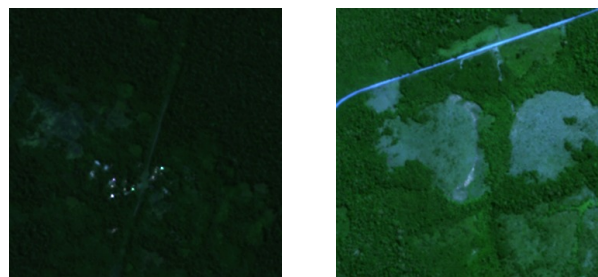


Fig. 4. Classificações erradas geradas pelo *AdaBoost* no conjunto de testes.

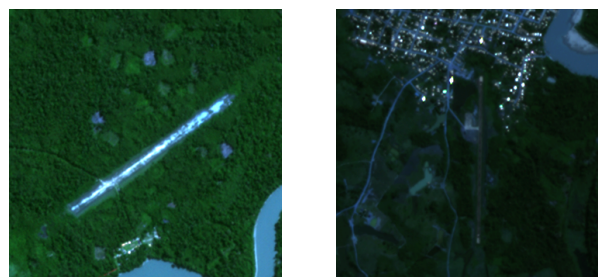


Fig. 5. Classificações corretas geradas pelo *AdaBoost* no conjunto de testes.

É importante destacar que existem características específicas que podem dificultar a predição dos modelos. Um exemplo relevante é o horário em que as fotos foram tiradas. Fotos capturadas à noite, como ilustrado na Figura 4, são mais difíceis de classificar. Perceba pela Figura 5 na qual ilustra 2 acertos que foram gerados, que o brilho das imagens é relativamente maior que as classificações erradas, demonstrando que tal efeito tem influência relevante no problema. Nesse sentido, devido à falta de padronização no período do

dia em que as imagens são capturadas, as bases de dados de treinamento e teste acabam contendo uma mistura de imagens diurnas e noturnas. Essa variabilidade temporal pode introduzir desafios adicionais para a acurácia dos modelos de classificação.

V. CONCLUSÕES

A mineração clandestina se mostra difícil de ser identificada pela estrutura visual da floresta. Técnicas anteriores de mapeamento deste tipo de atividade ilegal se mostraram custosas em relação ao tempo de classificação, uma vez que a identificação de imagens é manual. Uma forma de contornar essa dificuldade seria a implementação de metodologias que pudessem tratar as imagens de forma automática.

No presente estudo, aplicou-se o aprendizado de máquina para classificar pistas de pouso ilegais utilizando imagens de satélite. Foram utilizados modelos *ensemble* e modelos individuais, como *Decision Trees* e *Naïve Bayes* (individuais), além de *Random Forest*, *AdaBoost* e *XgBoost* (*ensemble*). Os modelos *ensemble* apresentaram performance superior aos modelos individuais, com destaque para o *XgBoost*, que obteve uma acurácia de 91%. Essa conclusão foi confirmada por análise estatística. Essa eficácia pode ser explicada pelas características dos modelos *ensemble*, que utilizam vários classificadores e ajustam-se iterativamente aos erros dos classificadores anteriores, melhorando continuamente a precisão.

Dessa forma, pode-se observar que a utilização de modelos *ensemble*, para o problema analisado, pode ser uma boa ferramenta para identificação de pistas de pousos ilegais, uma vez que pode reduzir o tempo de identificação das áreas com práticas criminosas. Além disso, este tipo de ferramenta pode ser expandido para outros processos de fiscalização da manutenção do meio ambiente e, conseqüentemente, preservação de áreas vulneráveis como, por exemplo, monitoramento de desmatamento e extração de madeira ilegais, monitoramento de queimadas e incêndios florestais, fiscalização de áreas de conservação e proteção de fauna, monitoramento de mudanças em corpos d'água. Por fim, reforça-se que, após a utilização do algoritmo, ainda é necessária a avaliação do operador para verificar se é um falso positivo.

Como trabalhos futuros, pode-se utilizar a metodologia proposta na aplicação de uma busca iterativa no bioma a fim de classificar toda a floresta em relação ao alvo proposto. Além disso, pode-se verificar o desempenho comparativo das técnicas mencionadas com redes neurais convolucionais completas e modelos avançados de deep learning, como: YOLO e Transformers.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

[1] Instituto Brasileiro de Geografia e Estatística. (2020, jun) IBGE atualiza Mapa da Amazônia Legal. [Online]. Available: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/28089-ibge-atualiza-mapa-da-amazonia-legal>

[2] C. F. A. da Silva, M. O. de Andrade, A. M. dos Santos, V. A. Falcão, and S. F. S. Martins, "The drivers of illegal mining on Indigenous Lands in the Brazilian Amazon," *The Extractive Industries and Society*, vol. 16, p. 101354, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214790X23001442>

[3] R. R. Soares, L. Pereira, and R. Pucci, "Ilegalidade e Violência na Amazônia," *Report Amazônia 2030*, vol. 202112, no. ed26, Dec 2021.

[4] MapBiomas, "Mapeamento das pistas de pouso e garimpo na Amazônia," 2023. [Online]. Available: https://brasil.mapbiomas.org/wp-content/uploads/sites/4/2023/08/MapBiomas-Pistas_de_Pouso_06.02.2023_-1.pdf

[5] MapBiomas, "Projeto mapbiomas – mapa de pistas de pouso da amazônia 2021 (v1)," Accessed on 05/04/2024, available at <https://plataforma.brasil.mapbiomas.org/pistas-de-pouso>.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[7] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[11] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed. Chapman and Hall/CRC, 1984.

[12] K. Faceli, A. C. Lorena, J. Gama, and A. C. P. d. L. F. d. Carvalho, *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2011.

[13] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.

[14] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[16] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522>

[17] P. Nemenyi, *Distribution-free Multiple Comparisons*. Princeton University, 1963. [Online]. Available: <https://books.google.com.br/books?id=nhDMtgAACAAJ>

[18] P. L. PBC, "Planet Application Program Interface: In Space for Life on Earth," Planet, 2018–. [Online]. Available: <https://api.planet.com>

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," <https://www.tensorflow.org/>, 2015, software available from tensorflow.org.

[20] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.

[21] W. McKinney, "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, pp. 56–61, 2010.

[22] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *International Joint Conference on Artificial Intelligence*, vol. 14, pp. 1137–1145, 1995.

[25] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.