

ARARA: Monitoramento Acústico de Desmatamento Ilegal via Computação na Borda

Caio M. Carlos, Leonardo C. C. Bitencourt, Walter A. Gontijo, Eduardo L. O. Batista, Richard D. Souza

Resumo— Neste trabalho, é apresentado um sistema de monitoramento acústico florestal em tempo real que utiliza aprendizado de máquina no próprio dispositivo para detecção de eventos. A solução tem como base o uso da arquitetura do YAMNet como extrator de características para gerar um modelo refinado que classifica sons críticos como "motosserra", "motor" e "serrote". O sistema adquire segmentos de áudio e classifica os eventos acústicos diretamente em um Raspberry Pi 3. As classes identificadas são transmitidas via LoRaWAN. O sistema é alimentado por um painel solar e uma bateria para utilização em ambientes remotos. O modelo refinado apresentou acurácia global de 94% no conjunto de teste. Essa abordagem demonstra a viabilidade da implementação de classificação de áudio em plataformas embarcadas com recursos limitados (classificação na borda), possibilitando alertas em tempo real e aprimorando os esforços de preservação florestal em cenários reais.

Palavras-Chave— Aprendizado de Máquina, Computação de Borda, LoRaWAN, Monitoramento Acústico.

Abstract— In this work, a real-time forest acoustic monitoring system is presented that uses on-device machine learning for event detection. The solution is based on using the YAMNet architecture as a feature extractor to generate a refined model that classifies critical sounds, including 'chainsaw', 'engine', and 'hand saw'. The system acquires audio segments and classifies acoustic events directly on a Raspberry Pi 3. The identified classes are transmitted via LoRaWAN. The system is powered by a solar panel and battery for use in remote environments. The refined model presented overall accuracy of 94% on the test set. This approach demonstrates the feasibility of deploying audio classification on the edge using resource-constrained embedded platforms, enabling real-time alerts and enhancing forest preservation efforts in real-world scenarios.

Keywords— Machine Learning, Edge Computing, LoRaWAN, Acoustic Monitoring.

I. INTRODUÇÃO

O monitoramento acústico em tempo real é uma ferramenta essencial para a preservação florestal, permitindo a identificação de sons críticos como extração ilegal de madeira [1]-[3]. No entanto, abordagens tradicionais que adquirem o áudio e o transmitem para servidores centrais enfrentam severas limitações em áreas remotas, devido ao alto custo de transmissão de dados e restrições de energia [4].

Nesse contexto, este trabalho apresenta um sistema de monitoramento florestal em tempo real, denominado ARARA (*Acoustic Recognition and Alerting for Remote Areas*), que executa um modelo baseado em Redes Neurais Convolucionais (CNN) diretamente no dispositivo de borda. Para viabilizar

uma inferência local eficiente, o sistema utiliza a arquitetura YAMNet [5] como extratora de características. Por ser um modelo pré-treinado em um conjunto de dados de larga escala e amplamente validado em tarefas sobre reconhecimento de padrões acústicos, o YAMNet consolida-se como uma base robusta para a geração de um modelo customizado utilizando *Transfer Learning* [6].

Em [7], foi apresentada a arquitetura do sistema original. O presente trabalho estende aquele resultado refinando o modelo e integrando o subsistema de alimentação solar, o mecanismo de ativação por detecção de energia e a validação em campo com motosserras. Tal extensão demonstra a viabilidade do sistema completo em um cenário real.

II. TRABALHOS RELACIONADOS

O monitoramento acústico em dispositivos de borda exige estratégias para contornar restrições de hardware. Baucas e Spachos [4] propuseram uma arquitetura híbrida para classificação acústica que divide o esforço computacional entre a borda e a nuvem, otimizando o tempo de execução e o consumo energético, minimizando a transmissão de dados. Lamrini *et al.* [8] validaram que redes neurais convolucionais (CNNs) pré-treinadas apresentam forte desempenho sob restrições de memória, confirmando a viabilidade do aprendizado profundo embarcado.

No contexto da proteção florestal, abordagens clássicas também foram investigadas. Seabra *et al.* [3] exploraram a detecção de desmatamento ilegal na Amazônia utilizando Decomposição Wavelet e classificadores do tipo *Support Vector Data Description* (SVDD). Mais recentemente, avançando para o paradigma da computação na borda, Teixeira *et al.* [9] propuseram um sistema de monitoramento autônomo na floresta amazônica. Os autores desenvolveram um dispositivo que integra uma rede neural convolucional (CNN) 1D embarcada em microcontroladores e comunicação LoRaWAN para detectar sons de motosserras e tratores.

III. ABORDAGEM PROPOSTA

O sistema ARARA baseia-se em uma arquitetura de processamento na borda projetada para operar de forma autônoma em ambientes florestais, fundamentando-se em escolhas estratégicas para solucionar as limitações de processamento local. Apesar dos avanços recentes validarem a combinação de redes LoRaWAN e inteligência na borda, o treinamento de arquiteturas convolucionais a partir do zero pode impor limitações na capacidade de generalização do modelo frente a cenários acústicos adversos. O ARARA diferencia-se ao adotar a arquitetura profunda YAMNet por meio de *transfer learning*. A principal vantagem dessa abordagem é a alta

capacidade de generalização herdada do modelo base, que já atua como um extrator de características robusto, facilitando consideravelmente a customização da rede para a aplicação proposta. A seguir, as três principais escolhas são descritas:

A. Arquitetura YAMNet

A seleção da arquitetura YAMNet como base justifica-se pelo seu balanço otimizado entre custo computacional e capacidade de representação acústica, tornando-a ideal para plataformas com recursos limitados. A arquitetura do YAMNet utiliza convoluções separáveis em profundidade (depthwise-separable convolutions) da MobileNet V1, reduzindo o número de parâmetros e operações de multiplicação e acúmulo (MACs) necessárias para a inferência, permitindo sua execução em um dispositivo de borda como o Raspberry Pi 3.

O YAMNet é amplamente validado por ser pré-treinado no conjunto de dados de larga escala AudioSet [6]. Essa base robusta permite que a rede atue como um extrator de características de alta confiabilidade. O YAMNet recebe o sinal de áudio amostrado em 16 kHz, converte em espectrogramas *log-mel* com 64 bandas mel e gera um vetor de *embedding* unidimensional de 1024 dimensões.

B. Transfer Learning e Conjunto de Dados

O emprego da técnica de *transfer learning* permitiu a customização do YAMNet para o contexto florestal, sem a necessidade de treinar uma rede profunda do zero. A partir dessa abordagem, os vetores de 1024 dimensões extraídos pela rede base alimentam uma camada de classificação, composta por uma camada densa oculta de 512 neurônios (ativação ReLU) e uma camada de saída *softmax* de quatro neurônios. Para o ajuste fino (*fine-tuning*) dessa nova estrutura, utilizou-se um subconjunto do *dataset* ESC-50 [10] (totalizando 320 áudios), selecionando-se amostras de áudios para compor as três classes alvo (*chainsaw*, *engine* e *hand_saw*), enquanto ruídos ambientais (como sons de tempestades, pássaros e insetos) foram agrupados na classe consolidada *others* para mitigar a ocorrência de falsos alarmes.

Para a avaliação, o conjunto de dados foi particionado empregando a estratégia *hold-out*, na qual 60% das amostras foram destinadas ao treinamento dos pesos da nova camada de classificação, 20% para validação durante o treinamento e 20% para testes.

C. Protocolo LoRaWAN

A escolha do protocolo LoRaWAN para transmissão dos dados responde diretamente às restrições de conectividade e energia encontradas em florestas ou propriedades rurais isoladas, cenários onde redes celulares (como 4G/5G) ou Wi-Fi são economicamente ou tecnicamente inviáveis. A viabilidade de utilizar uma rede de baixa taxa de transmissão de dados vincula-se intrinsecamente à presença da inteligência na borda. A transmissão contínua de sinais de áudio bruto impõe desafios significativos em áreas remotas. O aumento no tempo de transmissão ativa do rádio (*time-on-air*) resulta em um consumo maior de energia, dificultando a operação energeticamente sustentável e o monitoramento em tempo real.

TABELA I
DESEMPENHO DO CLASSIFICADOR NO CONJUNTO DE TESTE

Classe	Precisão	Recall	F1-Score	Suporte
chainsaw	0,89	0,93	0,91	80
engine	0,94	1,00	0,97	80
hand saw	0,81	0,96	0,88	80
others	0,98	0,93	0,95	400
Acurácia	94%			
Média Macro	0,91	0,94	0,94	640

Como o processamento e a inferência ocorrem de forma local no dispositivo de borda, o rádio LoRaWAN opera estritamente sob demanda. Identificada uma classe alvo, o resultado é encapsulado em um *payload* de apenas 1 *byte*, contendo o identificador da classe. Essa abordagem minimiza o uso de banda, reduz o consumo do rádio e possibilita que o arranjo atinja uma maior autonomia mesmo operando com baterias.

IV. IMPLEMENTAÇÃO E ARRANJO EXPERIMENTAL

A Fig. 1 mostra o diagrama do ARARA considerando seus principais blocos: Alimentação, Raspberry Pi 3 Model B [11], B-L072Z-LRWAN1 [12], LoRaWAN Gateway Khomp ITG200 [13] e servidor de rede The Things Network (TTN) [14]. A alimentação é composta por um painel solar de 10 W, um controlador de carga PWM, uma bateria de 12 V 9 Ah e um conversor DC-DC (12 V/5 V).

O sistema utiliza dois microfones com funções distintas. O microfone analógico MAX4466, conectado ao módulo B-L072Z, realiza a detecção de energia do áudio ambiente. Quando essa ultrapassa um limiar pré-definido, o B-L072Z envia um sinal para despertar o Raspberry Pi 3, que permanece em modo *sleep* para reduzir o consumo energético. Uma vez desperto, o Raspberry Pi 3 adquire o áudio por meio do microfone digital INMP441, que oferece maior fidelidade para a etapa de classificação. O modelo customizado do YAMNet então classifica o segmento de áudio. O resultado é enviado de volta ao B-L072Z, que transmite¹ o código da classe identificada para o Gateway LoRaWAN, que por sua vez encaminha os dados para o servidor de rede da TTN, sendo finalmente enviados para um servidor de aplicação.

V. RESULTADOS E AVALIAÇÃO EXPERIMENTAL

Conforme a Tabela I, o modelo atingiu 94% de acurácia global. A matriz de confusão (Fig. 2) mostra o desempenho do sistema na detecção das classes alvo, com acertos de 93% para *chainsaw*, 100% para *engine*, 96% para *hand saw* e 93% para *others*. Tais resultados evidenciam que as classes alvo não estão comprometidas por falsos negativos. O tempo médio de classificação do áudio é de 1,25 s. A latência fim-a-fim do sistema, incluindo o despertar via MAX4466, aquisição de áudio (10s), inferência e transmissão via LoRaWAN, resulta em um tempo médio de 30 segundos.

Para validar o funcionamento do ARARA em um ambiente real, foram realizados testes utilizando dois modelos de motosserras (STIHL MS 170 e HT 75). O sistema classificou corretamente sons de motosserra a uma distância de até 150 metros.

¹O B-L072Z foi programado para operar como um dispositivo LoRaWAN do tipo "Classe A" no modo "adaptive data rate" habilitado.

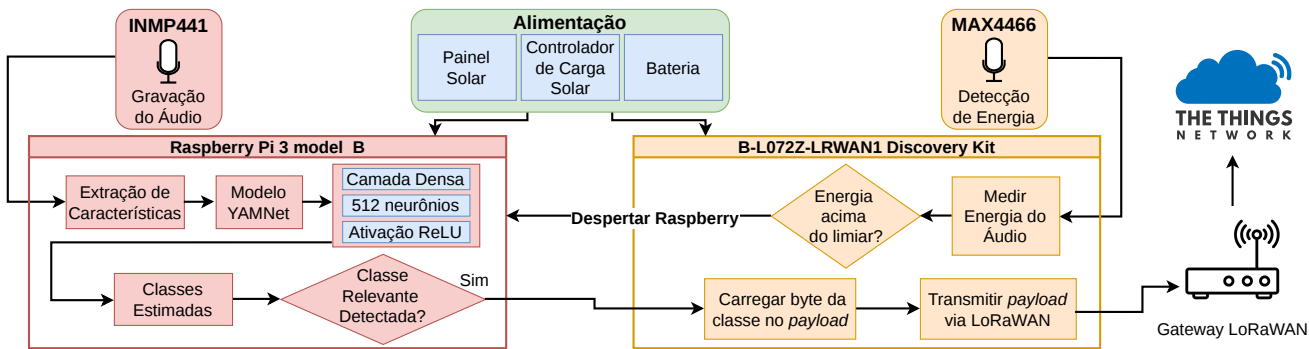


Fig. 1. Diagrama de blocos do ARARA

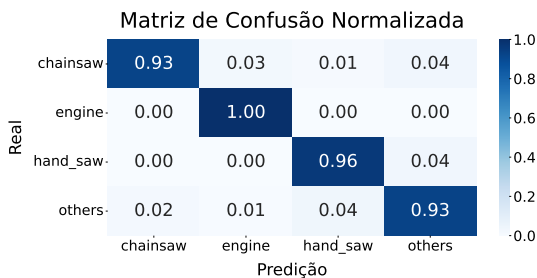


Fig. 2. Matriz de Confusão Normalizada do modelo refinado.

Os experimentos mostraram um consumo médio de 2,01 W durante os ciclos de aquisição e classificação. Já em modo de espera (sleep), o consumo reduziu para 1,81 W, devido à limitação de *hardware* do Raspberry Pi 3. A autonomia obtida considerando apenas a bateria (12 V, 9 Ah), sem contribuição do painel solar, foi da ordem de 40 horas.

Foi estruturado um repositório no GitHub [15] contendo tanto os áudios originais capturados nestes testes de campo quanto versões desses misturados a ruídos de ambientes florestais com SNR de 10dB. Todos os arquivos deste conjunto foram classificados corretamente, permitindo verificar a eficácia do sistema mesmo em cenários de interferência acústica.

VI. CONCLUSÕES

Este artigo apresentou um sistema de monitoramento acústico florestal em tempo real capaz de detectar sons de “motoserra”, “motor” e “serrote” diretamente em um dispositivo de borda. O sistema, alimentado por painel solar e bateria, utiliza uma versão refinada do YAMNet que obteve 94% de acurácia no conjunto de teste. Testes de campo com motosserras validaram a eficácia da classificação em cenários reais. Como trabalhos futuros, pretende-se integrar o ARARA em uma única plataforma de hardware (e.g., microcontrolador com suporte a TensorFlow Lite Micro) para reduzir o consumo energético, e incluir interface com comunicação via satélite.

AGRADECIMENTOS

Ao Eng. Rafael Luiz Ferreira e ao casal Sr. Jeronimo Werkhaizer/Sra. Carla Werkhaizer pelo auxílio nos experimentos de campo utilizando motosserras.

REFERÊNCIAS

[1] I. Mporas, I. Perikos, V. Kelefouras, and M. Paraskevas, “Illegal logging detection based on acoustic surveillance of forest,” *Applied Sciences*, vol. 10, no. 20, p. 7379, Oct. 2020, doi: 10.3390/app10207379.

[2] A. Srisuphab *et al.*, “Illegal logging listeners using IoT networks,” in *Proc. 2020 IEEE Region 10 Conf. (TENCON)*, Nov. 2020, pp. 1277–1282, doi: 10.1109/TENCON50793.2020.9293935.

[3] W. Seabra, J. Colonna, and E. Nakamura, “Detecção de desmatamento ilegal na floresta amazônica baseada em processamento de áudio,” em *Anais do XXXVI Concurso de Trabalhos de Iniciação Científica da SBC*, 2017, pp. 2512–2521.

[4] M. J. Baucas and P. Spachos, “Using cloud and fog computing for large scale IoT-based urban sound classification,” *Simulation Modelling Practice and Theory*, vol. 101, p. 102013, Mar. 2020, doi: 10.1016/j.simpat.2019.102013.

[5] M. Plakal and D. Ellis, “YAMNet,” <https://github.com/tensorflow/models/tree/master/research/audioset/yamnet>, 2020.

[6] N. H. Valliappan, S. D. Pande, and S. Reddy Vinta, “Enhancing gun detection with transfer learning and YAMNet audio classification,” *IEEE Access*, vol. 12, pp. 58940–58949, Apr. 2024, doi: 10.1109/ACCESS.2024.3392529.

[7] M. A. d. Oliveira, C. M. Carlos, W. A. Gontijo, and R. D. Souza, “Edge machine learning for forest acoustic monitoring implementation,” in *Proc. 2025 Symposium on Internet of Things (SIoT)*, Campinas, Brazil, 2025, pp. 1–4, doi: 10.1109/SIoT68426.2025.11368807.

[8] M. Lamrini, M. Y. Chkouri, and A. Touhafi, “Evaluating the performance of pre-trained convolutional neural network for audio classification on embedded systems,” *Sensors*, vol. 23, no. 13, p. 6227, Jul. 2023, doi: 10.3390/s23136227.

[9] T. A. Teixeira, N. L. Vilaça, A. L. Printes, R. C. S. Gomes, I. G. Torné, T.-Y. A. Araújo, and A. G. D. e. Dias, “Development of a Monitoring System against Illegal Deforestation in the Amazon Rainforest Using Artificial Intelligence Algorithms,” *Engineering Proceedings*, vol. 58, no. 1, p. 21, Nov. 2023.

[10] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” em *Proceedings of the 23rd Annual ACM Conference on Multimedia*. Brisbane, Australia: ACM Press, Oct. 2015, pp. 1015–1018. [Online].

[11] Raspberry Pi Foundation, “Raspberry Pi 3 Model B,” <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>, 2016.

[12] STMicroelectronics, “B-L072Z-LRWAN1 LoRa/Sigfox Discovery kit,” <https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html>.

[13] Khomp, “Gateway Indoor ITG 200,” <https://app.khomp.com/iot/pt/produto/gateway-itg-200-indoor/>, Acessado em: 6 maio 2026.

[14] The Things Network, “The Things Network,” <https://www.thethingsnetwork.org/>. Acessado em: 6 maio 2026.

[15] C. M. Carlos, “Deforestation-detection,” <https://github.com/CaioBinhal/Deforestation-Detection>, Sep. 2025, GitHub repository.