

UNIVERSIDADE FEDERAL DE ITAJUBÁ
CAMPUS PROFESSOR JOSÉ RODRIGUES SEABRA

ENZO KOZONOE
HENRY MATHEUS PARENTE HAGEMANN
PAULO EMÍLIO MAIA CAUWILLA

**SISTEMA DE CLASSIFICAÇÃO DE CANTOS DE PÁSSAROS COM TINYML
PARA DISPOSITIVOS EMBARCADOS**

ITAJUBÁ, MG

2025

INSTITUTO DE ENGENHARIA DE SISTEMAS E TECNOLOGIA DE INFORMAÇÃO
TINYML - APRENDIZADO DE MÁQUINA APLICADO PARA DISPOSITIVOS IOT
EMBARCADOS (IESTI01)

ENZO KOZONOE

HENRY MATHEUS PARENTE HAGEMANN

PAULO EMÍLIO NAUA CAUWILLA

SISTEMA DE CLASSIFICAÇÃO DE CANTOS DE PÁSSAROS COM TINYML
PARA DISPOSITIVOS EMBARCADOS

Relatório apresentado ao curso de graduação
em Engenharia de Computação na
Universidade Federal de Itajubá, como
requisito final para a nota da disciplina
IESTI01.

Docentes: José Alberto Ferreira Filho e
Marcelo José Rovai.

ITAJUBÁ, MG

2025

SUMÁRIO

1. INTRODUÇÃO	5
2. OBJETIVOS	5
3. METODOLOGIA.....	5
3.1. Plataforma de Desenvolvimento	5
3.2. Aquisição e Preparação do <i>Dataset</i>	5
4. RESULTADOS	8
4.1. Performance do Treinamento e Validação	8
4.3. Performance no Dispositivo	9
5. DISCUSSÃO	10
6. CONCLUSÃO	11
7. QUESTÕES SEM SOLUÇÃO	11
8. REFERÊNCIAS.....	11

RESUMO

Este relatório detalha o processo de desenvolvimento, treinamento e validação de um modelo de *Machine Learning* para a classificação de cantos de pássaros da avifauna brasileira, utilizando técnicas de *TinyML*. O objetivo foi criar um sistema capaz de diferenciar entre as espécies *Turdus rufiventris* (Sabiá-laranjeira), *Zonotrichia capensis* (Tico-tico), e *Pitangus sulphuratus* (Bem-te-vi), além de identificar ruído ambiente. O *dataset* foi construído com amostras de áudio da plataforma Xeno-canto. O treinamento foi conduzido na plataforma *Edge Impulse*, empregando *Mel-Frequency Cepstral Coefficients* (MFCCs) para extração de características e uma Rede Neural Convolutiva para a classificação. O modelo final alcançou uma acurácia 81.93% em dados de teste, com um uso de memória estimado em 47.2KB de Flash e 14.4KB de RAM, demonstrando a viabilidade da sua implementação em microcontroladores de baixa potência como o *Arduino Nano 33 BLE Sense*.

1. INTRODUÇÃO

A monitorização bioacústica é uma ferramenta poderosa para estudos de biodiversidade, ecologia e conservação. A identificação automática de espécies através de seus cantos permite o monitoramento em larga escala e de forma não invasiva. No entanto, soluções tradicionais frequentemente dependem de equipamentos caros ou do envio de grandes volumes de dados para a nuvem. O *Tiny Machine Learning* emerge como uma solução promissora, permitindo que a inteligência artificial seja executada diretamente em dispositivos de *hardware* de baixo custo e baixo consumo, características da computação de borda, garantindo privacidade, baixa latência e autonomia energética.

2. OBJETIVOS

O principal objetivo deste trabalho é desenvolver e avaliar um modelo de classificação de áudio capaz de operar em um microcontrolador para identificar em tempo real os cantos de três espécies de pássaros comuns no Brasil, além de uma classe de ruído. Os objetivos específicos são:

- Construir um *dataset* balanceado a partir de fontes públicas (Xeno-canto);
- Treinar um modelo de classificação na plataforma *Edge Impulse*;
- Alcançar uma acurácia de validação superior a 80%;
- Analisar os requisitos de memória (Flash/RAM) e latência do modelo para garantir sua viabilidade no *hardware* alvo (*Arduino Nano 33 BLE Sense*).

3. METODOLOGIA

3.1. Plataforma de Desenvolvimento

- O treinamento e o modelo foram feitos na plataforma *Edge Impulse Studio* (versão de junho de 2025);
- O *Hardware* alvo placa de desenvolvimento *Arduino Nano 33 BLE Sense*.

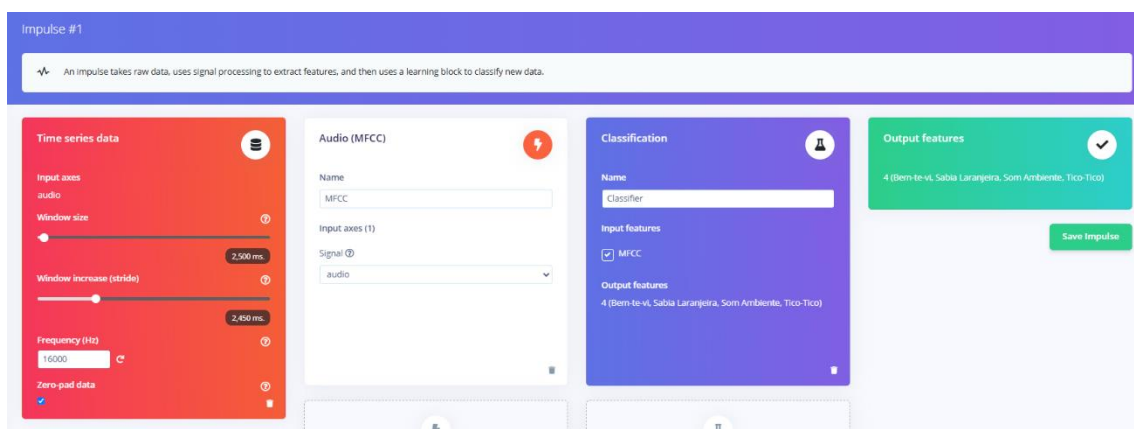
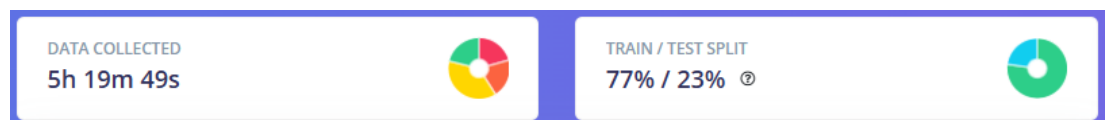
3.2. Aquisição e Preparação do *Dataset*

- Os áudios dos cantos pássaros foram obtidos da plataforma colaborativa [xeno-canto](#);
- O projeto foi estruturado com quatro classes distintas:

- *Sabiá-laranjeira (Turdus rufiventris)*;
- *Tico-tico (Zonotrichia capensis)*;
- *Bem-te-vi (Pitangus sulphuratus)*;
- *Som ambiente* (sons ambientes diversos, como silêncio, vozes, tráfego, vento).
- Foram coletados múltiplos arquivos de áudio para cada classe. Primeiramente foi baixado arquivos de qualidade “A” da plataforma xeno-canto, em seguida os áudios foram pré-processados, para o padrão abaixo, utilizando-se o fre:ac:
 - Áudio Mono;
 - 16 KHz de taxa de amostragem;
 - Formato de Signed 16 bit PCM.
 - Codificador Windows Wave File Output (.wav)

Após esse procedimento, elaborou-se código em *python* para dividir os áudios longos em áudios de 10 segundos com uma pequena sobreposição de 0,1s.

- Para cada classe de som de pássaro, utilizou-se em média 50 minutos de dados. Para dados de Sons Ambiente, utilizou-se tanto *datasets* de ruídos e de fala humana, gerando um total de 1h e 33 minutos de dados.



[illegible]

Neural Network settings

Training settings

Number of training cycles

100

Use learned optimizer

☐

Learning rate

0.005

Training processor

CPU

Advanced training settings

Audio training options

Data augmentation

☐

Neural network architecture

Architecture presets

1D Convolutional (Default)

2D Convolutional

Input layer (1,612 features)

Reshape layer (13 columns)

1D conv / pool layer (8 filters, 3 kernel size, 1 layer)

Dropout (rate 0.25)

1D conv / pool layer (16 filters, 3 kernel size, 1 layer)

Dropout (rate 0.25)

Flatten layer

Add an extra layer

Output layer (4 classes)

Save & train

4. RESULTADOS

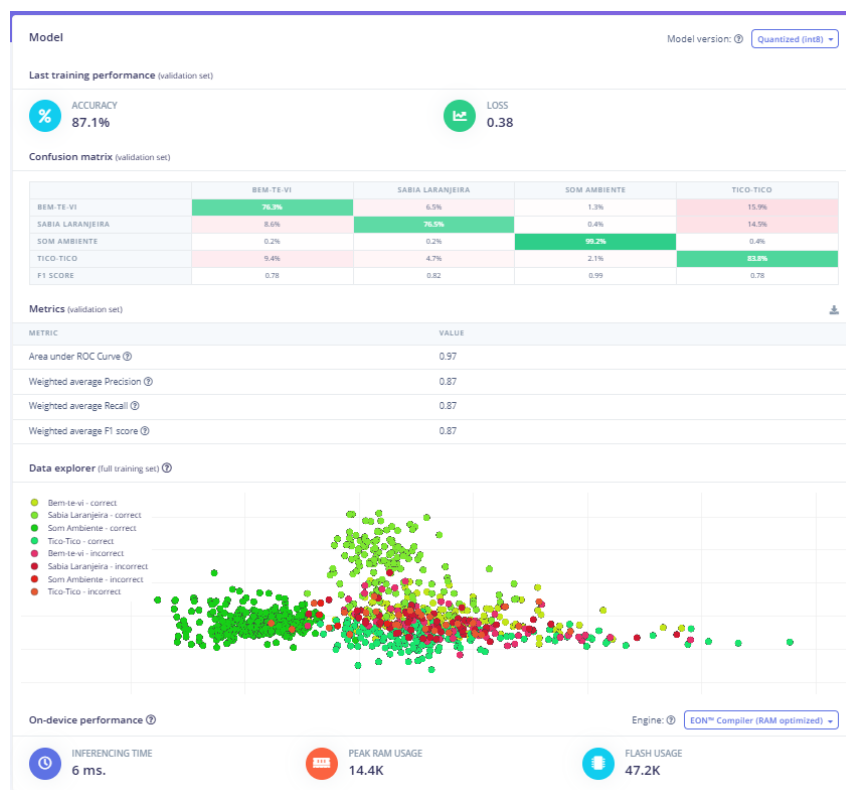
4.1. Performance do Treinamento e Validação

Ao final do treinamento, o modelo atingiu os seguintes níveis de performance:

- Acurácia: 87,1%
- Perda: 0,38.

4.2. Matriz de Confusão

A matriz de confusão abaixo, gerada com o conjunto de teste, detalha a performance do modelo para cada classe. A diagonal principal representa as classificações corretas.



A Matriz de Confusão detalha o desempenho por classe:

- A classe "Som Ambiente" é classificada com alta precisão, atingindo 96.2% de acurácia. Este resultado é reforçado por um F1 Score de 0.99, sugerindo que o modelo é robusto na identificação de ruídos de fundo e na sua distinção de vocalizações de pássaros.
- Entre as espécies de pássaros, o "Tico-Tico" apresenta a melhor performance individual, com 83.8% de acurácia.

- As classes "Bem-te-vi" e "Sabiá Laranjeira" mostram acurácias de 76.3% e 76.5%, respectivamente. A análise da matriz revela confusões notáveis: o "Bem-te-vi" é frequentemente confundido com "Tico-Tico" (15.9%) e "Sabiá Laranjeira" (6.5%), enquanto o "Sabiá Laranjeira" é confundido com "Tico-Tico" (14.5%) e "Bem-te-vi" (8.6%). O "Tico-Tico" também é confundido, principalmente com "Bem-te-vi" (9.8%) e "Sabiá Laranjeira" (4.7%). Essas confusões indicam sobreposição nas características vocais entre as três espécies de aves.

As métricas complementares corroboram o desempenho geral, com uma Área sob a Curva ROC de 0.97 e uma média ponderada (*Weighted average*) de Precisão, Recall e F1 Score de 0.87, alinhados com a acurácia global.

O Data Explorer visualiza a separação das classes no espaço de características. As amostras classificadas corretamente (bolinhas verdes) formam clusters distintos, enquanto as classificações incorretas (bolinhas vermelhas) aparecem nas regiões de intersecção entre as classes de pássaros, confirmando visualmente as confusões observadas na matriz.

Em termos de desempenho para implantação em dispositivos embarcados, o modelo é notavelmente eficiente: o tempo de inferência é de apenas 6 ms, com um uso máximo de RAM de 14.4 KB e de Flash de 47.2 KB. Esses números demonstram que o modelo é extremamente leve e rápido, ideal para ambientes com recursos computacionais limitados.

4.3. Performance no Dispositivo

As estimativas de performance no hardware alvo, fornecidas pelo Edge Impulse, são cruciais para um projeto de TinyML.

- Uso de Flash (ROM): 47.2KB
- Uso de RAM: 14.4KB
- Latência de Inferência: 6 ms

Configure your deployment

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

Arduino library

SELECTED DEPLOYMENT



Arduino library
An Arduino library with examples that runs on most Arm-based Arduino development boards.

MODEL OPTIMIZATIONS

Model optimizations can increase on-device performance but may reduce accuracy.



EON™ Compiler
Same accuracy, 34% less RAM, 45% less ROM.

Quantized (int8)

Selected

	MFCC	CLASSIFIER	TOTAL
LATENCY	630 ms.	6 ms.	636 ms.
RAM	29.5K	5.7K	29.5K
FLASH	-	31.9K	-
ACCURACY			-

Unoptimized (float32)

Select

	MFCC	CLASSIFIER	TOTAL
LATENCY	630 ms.	73 ms.	703 ms.
RAM	29.5K	14.5K	29.5K
FLASH	-	32.4K	-
ACCURACY			81.93%

To compare model accuracy, run model testing for all available optimizations.

Run model testing

Estimate for Cortex-M4F 80MHz [Change target](#)

Build

5. DISCUSSÃO

Os resultados obtidos confirmam a eficácia da abordagem *TinyML* para a classificação de cantos de pássaros. A acurácia superior a 80% indica que o modelo aprendeu características distintivas para cada espécie. O baixo consumo de memória *Flash* e RAM (< 50 KB e < 20 KB, respectivamente) está bem dentro dos limites do *Arduino Nano 33 BLE Sense* (1MB de *Flash*, 256 KB de RAM), validando a viabilidade técnica do *deploy*.

A principal limitação encontrada foi a qualidade variável dos áudios do Xenocanto, onde ruídos de fundo por vezes se sobrepõem ao canto principal, podendo ter causado as confusões observadas na matriz.

Como trabalhos futuros, sugere-se:

1. **Aumento do *Dataset*:** Adicionar mais amostras, especialmente de cantos com variações regionais.
2. ***Data Augmentation*:** Aplicar técnicas de aumento de dados, como adicionar ruído sintético ou variar o *pitch* do áudio, para tornar o modelo mais robusto.
3. **Otimização de Hiperparâmetros:** Realizar um estudo mais aprofundado para otimizar a taxa de aprendizagem e a arquitetura da rede.

10

6. CONCLUSÃO

Este projeto demonstrou com sucesso a criação de um classificador de cantos de pássaros utilizando a plataforma *Edge Impulse*, resultando em um modelo leve, rápido e preciso, adequado para execução em microcontroladores. A tecnologia *TinyML* provou ser uma ferramenta acessível e poderosa para o desenvolvimento de soluções de monitoramento ambiental de baixo custo, abrindo portas para novas pesquisas e aplicações práticas em conservação.

7. QUESTÕES SEM SOLUÇÃO

- Como a diversidade do dataset impacta a generalização? Visto que a principal limitação foi a qualidade e variabilidade dos áudios, uma questão em aberto é: qual seria o ganho de acurácia e robustez ao adicionar amostras com diferentes variações regionais dos cantos e com maior controle sobre o ruído de fundo?
- Quais técnicas de *Data Augmentation* são mais eficazes? O relatório sugere aplicar *Data Augmentation* para tornar o modelo mais robusto. A questão pendente é: qual técnica específica (adição de ruído sintético, variação de *pitch*, etc.) traria o melhor custo-benefício para reduzir as confusões observadas entre as espécies, especialmente entre "Tico-Tico", "Bem-te-vi" e "Sabiá Laranjeira"?
- É possível otimizar o modelo ainda mais? O projeto propõe um estudo mais aprofundado para otimizar hiperparâmetros e a arquitetura da rede. A questão que permanece é: existe uma configuração de hiperparâmetros ou uma arquitetura de rede neural alternativa que poderia diminuir a taxa de perda (atualmente em 0.38) e aumentar a acurácia (atualmente em 87.1%), sem exceder significativamente os limites de memória e latência já alcançados (14.4 KB de RAM, 47.2 KB de Flash, 6 ms de inferência)?

8. REFERÊNCIAS

- Edge Impulse. (2025). *Edge Impulse Studio*. <https://www.edgeimpulse.com>
- Xeno-canto Foundation. (2025). *Xeno-canto: Community database of shared bird sounds from around the world*. <https://xeno-canto.org>.
- Vídeo da implementação do modelo no *hardware* alvo: [iesti01-final-video](#).