

Uma nova geração  
de especialistas em  
**tecnologia**





**BIOLINGO**

**Grupo n° 14**

**Antonio Francisco**

**Eudesmagnus Miguel**

**José Vale**



# GERAL

- NOTA CONCEITUAL E PLANO DE IMPLEMENTAÇÃO :
  - CONTEXTO
  - OBJETIVOS
  - RELAÇÃO ODS
- DADOS
  - COLETA DE DADOS
  - ANÁLISE EXPLORATÓRIA DE DADOS (EDA) E ENGENHARIA DE RECURSOS
- SELEÇÃO E TREINAMENTO DE MODELOS
  - AVALIAÇÃO DE MODELOS E AJUSTE
  - REFINAMENTO E TESTE DE MODELOS
- RESULTADOS
- IMPLANTAÇÃO
- TRABALHO FUTURO

# Background

Angola possui uma das biodiversidades mais ricas do continente africano, com aproximadamente 276 espécies de mamíferos, 265 espécies de aves de reprodução e 5.185 espécies de plantas superiores. No entanto, esta riqueza natural enfrenta ameaças significativas, incluindo a caça furtiva, perda de habitat e os impactos residuais de três décadas de guerra civil. Segundo a IUCN, cerca de 75% dos animais e plantas listados na lista Vermelha em Angola estão classificados como vulneráveis, ameaçados ou criticamente ameaçados. Paralelamente, a pesquisa científica em bioacústica em África permanece limitada, com 74,3% dos autores sendo não-africanos, indicando uma lacuna crítica em capacidades locais de investigação. Angola está particularmente subrepresentada neste campo, apesar do seu potencial imenso.

Apresentamos o projecto BioLingo, uma plataforma digital inovadora que utiliza inteligência artificial para decifrar a comunicação animal, transformando sons da fauna em informações acessíveis e acionáveis. Com foco específico em Angola, o BioLingo visa contribuir para a conservação da biodiversidade, educação ambiental, investigação científica e ecoturismo, alinhando-se com os Objectivos de Desenvolvimento Sustentável (ODS) das Nações Unidas.

# OBJECTIVOS

## OBJETIVOSOS GERAIS:

- Criar uma base de dados bioacústica abrangente da fauna angolana.
- Desenvolver modelos de IA capazes de identificar espécies e interpretar vocalizações com alta precisão.
- Promover a educação ambiental e a ciência cidadã em Angola.
- Fornecer ferramentas de monitoramento para a conservação de espécies ameaçadas.

## OBJECTIVOS ESPECÍFICOS:

- Lançar um MVP focado em aves de Angola até março de 2026.
- Expandir para mamíferos emblemáticos (elefantes, primatas, palancas) até setembro de 2026.
- Implementar uma rede de sensores bioacústicos em parques nacionais até setembro de 2027.
- Estabelecer parcerias com o Ministério do Ambiente, universidades angolanas e ONGs

# RELAÇÃO COM AS ODS

## ODS 9 - INDÚSTRIA , INOVAÇÃO INFRA-ESTRUTURA

Uso de IA, big data e bioacústica para uma solução tecnológica inovadora, com API aberta e parcerias de pesquisa com universidades angolanas e centros de investigação.

## ODS 11 - CIDADES E COMUNIDADES SUSTENTÁVEIS

Monitoramento ambiental em áreas urbanas (parques, zonas verdes) e rurais, engajando as comunidades angolanas no registo de sons para proteção do ambiente.

## ODS 13 - AÇÃO CONTRA A MUDANÇA GLOBAL DO CLIMA

Ajuda a entender o impacto das mudanças climáticas na biodiversidade angolana através do monitoramento acústico, apoiando políticas públicas de adaptação.

# RELAÇÃO COM AS ODS

## ODS 14 - VIDA NA ÁGUA

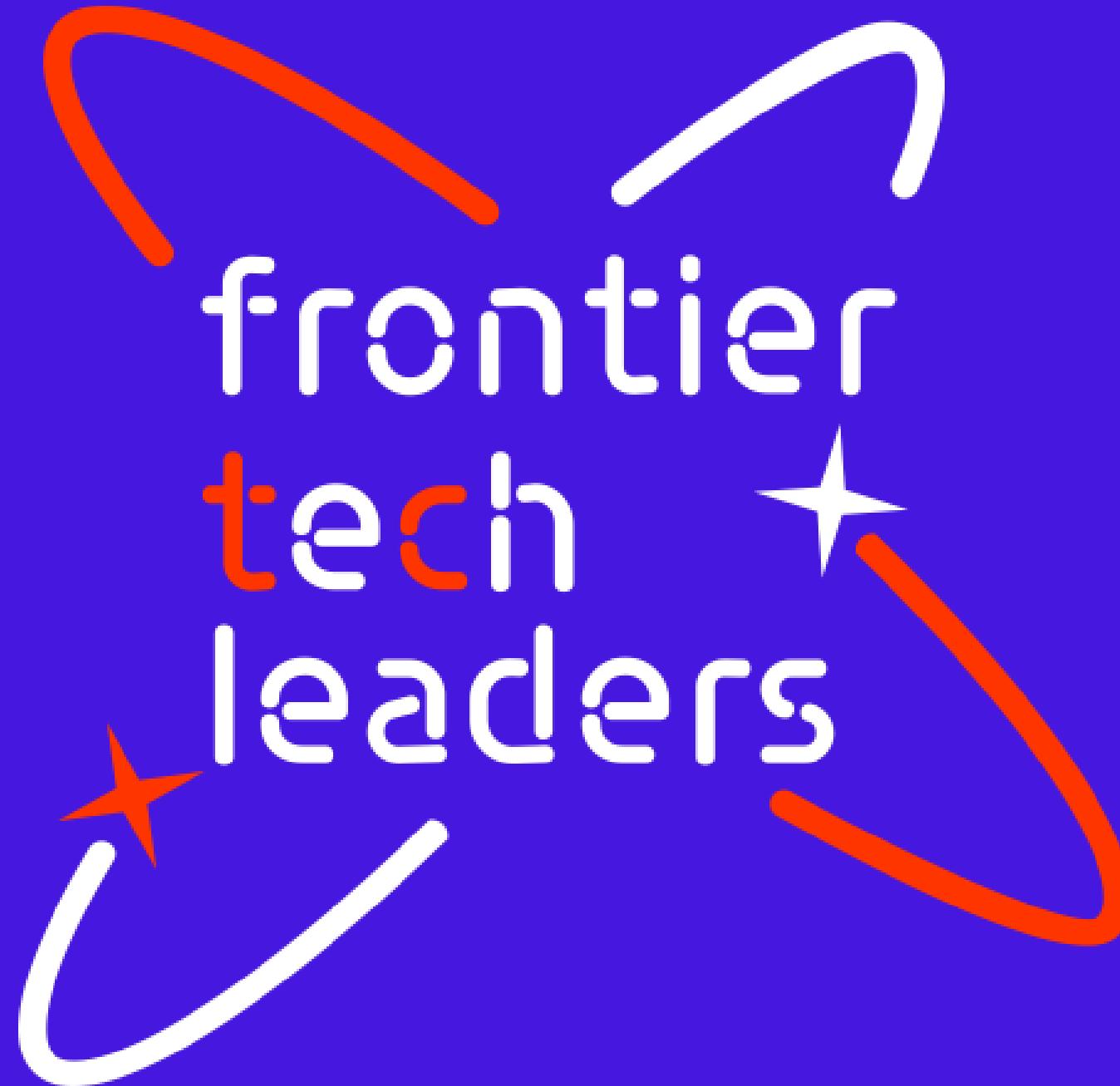
Potencial para monitoramento da vida marinha (bioacústica oceânica) na costa angolana, com parcerias de conservação marinha para proteger os oceanos.

## ODS 15 - VIDA TERRESTRE

Contribuição direta para a conservação da fauna e biodiversidade terrestre angolana, identificação de espécies em risco (como a Palanca Negra Gigante) e mapeamento colaborativo em parques nacionais como Quiçama e Maiombe.

## ODS 4 - EDUCAÇÃO DE QUALIDADE

Ferramenta educativa interativa sobre biodiversidade para escolas e universidades angolanas, promovendo o conhecimento sobre a nossa fauna.



DADOS



# DADOS

## RECOLHA DE DADOS

A recolha de dados para o treino dos modelos do BioLingo combina fontes globais e locais, garantindo diversidade e representatividade da fauna angolana.

Fonte	Tipo	Volume Estimado	Utilização
Xeno-canto	API pública	10.000+ gravações de aves africanas	Treino inicial e validação
Macaulay Library	API via eBird	5.000+ gravações de espécies angolanas	Treino e teste
Gravações Locais (Ciência Cidadã)	Aplicação móvel	2.000+ gravações (crescimento contínuo)	Fine-tuning e validação local
Sensores Bioacústicos	Parques Nacionais	1.000+ horas de gravação contínua	Dados de contexto e espécies raras
GBIF	Metadados de biodiversidade	Informações de ocorrência	Enriquecimento de metadados

## CRITÉRIOS DE QUALIDADE

- Cada gravação deve ter duração mínima de 3 segundos e máxima de 60 segundos
- Identificação de espécie confirmada por especialistas ou múltiplos observadores
- Qualidade de áudio deve ser classificada como boa ou excelente (relação sinal-ruído > 10 dB), e os metadados devem incluir localização geográfica, data, hora e tipo de habitat.
- Gravações de Angola e países vizinhos (Namíbia, Zâmbia, RDC, Congo) serão priorizadas para garantir relevância biogeográfica

## PREPARAÇÃO DE DADOS (LIMPEZA E VALIDAÇÃO)

- Remoção de gravações duplicadas através de hashing de áudio e validação da correspondência entre metadados e conteúdo de áudio.
- Identificação e correção de inconsistências taxonómicas, utilizando a nomenclatura científica padrão da IOC World Bird List.
- Filtro de gravações com ruído excessivo (tráfego, vento, interferência humana) através de análise espectral automatizada e revisão manual.

## AUMENTO DE DADOS (DATA AUGMENTATION)

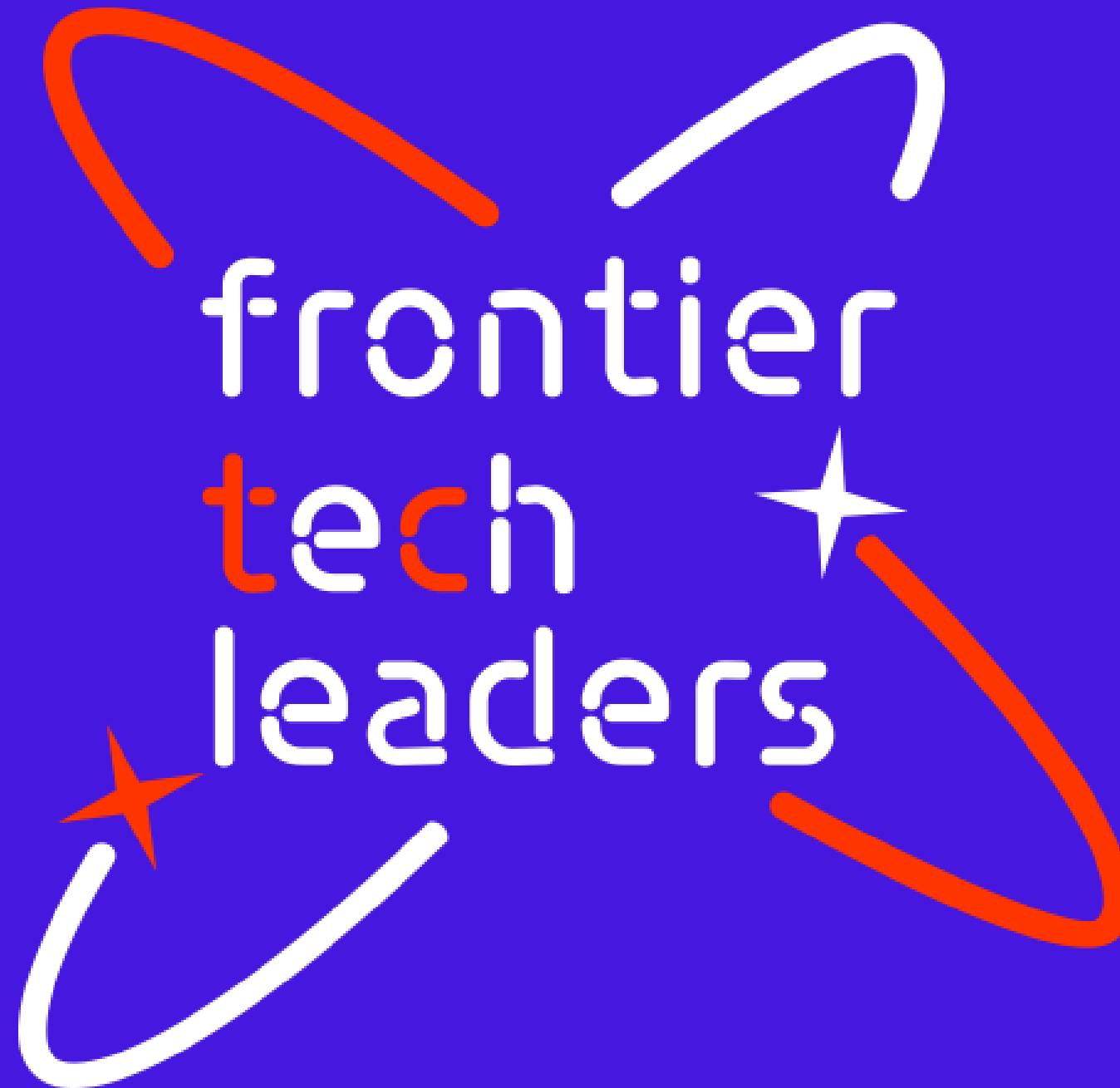
- Aplicação de técnicas de aumento de dados durante o treino para elevar a robustez dos modelos e compensar o desequilíbrio de classes, estas técnicas incluem :
  - **TIME STRETCHING**
  - **PITCH SHIFTING**
  - **TIME MASKING**
  - **FREQUENCY MASKING**

## DIVISÃO DE DADOS

- 70% dos dados para treino, 15% para validação (ajuste de hiperparâmetros e early stopping) e 15% para teste (avaliação final do modelo). A divisão é estratificada por espécie e tipo de vocalização, garantindo que todas as classes estejam representadas em cada conjunto.
- **VALIDAÇÃO CRUZADA K-FOLD(K=5)** para avaliação mais robusta do desempenho.

## ENGENHARIA DE RECURSOS

- Transformação do áudio bruto em múltiplas representações numéricas que capturam diferentes aspectos do sinal acústico.
- ESPECTROGRAMA MEL: Representação visual do espectro de frequências ao longo do tempo, utilizando a escala Mel que aproxima a percepção auditiva humana. Configuração: 128 bandas Mel, janela de 2048 amostras, hop length de 512 amostras, resultando em imagens de 224x224 pixels (redimensionadas para compatibilidade com modelos pré-treinados).
- MFCCs (MEL-FREQUENCY CEPSTRAL COEFFICIENTS): Representação compacta das características espetrais do som, amplamente utilizada em reconhecimento de fala e sons.
- CONFIGURAÇÃO: 40 coeficientes MFCC, janela de 2048 amostras, hop length de 512 amostras, incluindo deltas e delta-deltas para capturar dinâmica temporal.
- CARACTERÍSTICAS TEMPORAIS : Pitch médio e desvio padrão (frequência fundamental), energia RMS (Root Mean Square) ao longo do tempo, taxa de cruzamento por zero (Zero Crossing Rate), duração total da vocalização e intervalos entre vocalizações.
- CARACTERÍSTICAS CONTEXTUAIS : Coordenadas GPS (latitude, longitude), hora do dia (manhã,tarde,noite), estação do ano (seca, chuvosa), tipo de habitat (floresta, savana, zona húmida, urbano) e presença de outras espécies (co-ocorrência).



Modelo



## MODELO 1 : CLASSIFICADOR DE ESPÉCIES

**ARQUITETURA :** ResNet-50 Com Transfer Learning (Camadas Pré-Treinadas No ImageNet ) Adequada Para Espectrogramas

### ESTRATÉGIA DE TREINO

#### TRANSFER LEARNING EM DUAS FASES

**Fase 1 :** Congelamos as camadas iniciais da ResNet-50 e treinamos as camadas finais com os nossos dados de espectrogramas de aves. Taxa de aprendizagem inicial de 0.001, otimizador Adam, função de perda Cross-Entropy Loss com pesos de classe para lidar com desequilíbrio, batch size de 32 e early stopping com paciência de 10 épocas.

**Fase 2 :** Descongelamos as últimas 20 camadas e realizamos fine-tuning com uma taxa de aprendizagem reduzida (0.0001), permitindo que o modelo adapte as características de nível superior aos padrões específicos das aves angolanas. LEARNING RATE scheduling (OnPlateau) para ajustar dinamicamente a taxa deprendizagem.

### MÉTRICAS DE DESEMPENHO ESPERADAS

Para o MVP com 50 espécies de aves angolanas, esperamos alcançar uma acurácia global superior a 85%, precisão média por classe superior a 82%, recall médio por classe superior a 80% e F1-score médio superior a 81%. O top-5 accuracy (espécie correta entre as 5 previsões mais prováveis) deve ser superior a 95%

## MODELO 2 : CLASSIFICADOR DE TIPO DE VOCALIZAÇÃO

### ARQUITETURA: CNN + LSTM (HÍBRIDA)

CNN Para Extração De Características Espaciais e LSTM Para Capturar Dependências Temporais

### ESTRATÉGIA DE TREINO

MFCCs como entrada, com sequências de comprimento fixo (200 frames, correspondendo aproximadamente 4–5 segundos de áudio).

Taxa de aprendizagem inicial de 0.001, otimizador Adam, função de perda Cross-Entropy Loss, batch size de 64, gradient clipping (max\_norm=1.0) para estabilizar o treino de LSTMs e early stopping com paciência de 15 épocas.

### MÉTRICAS DE DESEMPENHO ESPERADAS

Para a classificação de 5 tipos de vocalização (alarme, acasalamento, territorial, social, contacto), esperamos alcançar uma acurácia global superior a 78%, precisão média por classe superior a 75%, recall médio por classe superior a 73% e F1-score médio superior a 74%.

```

import torch
import torch.nn as nn
from torchvision.models import resnet50, ResNet50_Weights
class BioLingoSpeciesClassifier(nn.Module):
    def __init__(self, num_species=50, pretrained=True):
        super(BioLingoSpeciesClassifier, self).__init__()

        # Carregar ResNet-50 pré-treinado no ImageNet
        if pretrained:
            self.backbone =
                resnet50(weights=ResNet50_Weights.IMAGENET1K_V2)
        else:
            self.backbone = resnet50(weights=None)

        # Congelar as primeiras camadas (transfer learning)
        for param in list(self.backbone.parameters())[:-20]:
            param.requires_grad = False

        # Substituir a camada final para o nosso número de espécies
        num_features = self.backbone.fc.in_features
        self.backbone.fc = nn.Sequential(
            nn.Dropout(0.5),
            nn.Linear(num_features, 512),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(512, num_species)
        )

    def forward(self, x):
        return self.backbone(x)

```

Configuração do modelo 1

```

import torch
import torch.nn as nn
class BioLingoVocalizationClassifier(nn.Module):
    def __init__(self, num_mfcc=40, num_classes=5):
        super(BioLingoVocalizationClassifier, self).__init__()

        # Camadas Convolucionais para extração de características
        self.conv1 = nn.Sequential(
            nn.Conv1d(num_mfcc, 128, kernel_size=5, padding=2),
            nn.BatchNorm1d(128),
            nn.ReLU(),
            nn.MaxPool1d(2)
        )
        self.conv2 = nn.Sequential(
            nn.Conv1d(128, 256, kernel_size=5, padding=2),
            nn.BatchNorm1d(256),
            nn.ReLU(),
            nn.MaxPool1d(2)
        )
        self.conv3 = nn.Sequential(
            nn.Conv1d(256, 512, kernel_size=3, padding=1),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.MaxPool1d(2)
        )

        # Camadas LSTM para capturar dependências temporais
        self.lstm = nn.LSTM(
            input_size=512,
            hidden_size=256,
            num_layers=2,
            batch_first=True,
            dropout=0.3,
            bidirectional=True
        )

```

Configuração do modelo 2 A

```

# Camadas totalmente conectadas para classificação
self.fc = nn.Sequential(
    nn.Dropout(0.5),
    nn.Linear(512, 128), # 512 devido ao LSTM bidirecional
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(128, num_classes)
)

def forward(self, x):
    # x shape: (batch, num_mfcc, time_steps)
    x = self.conv1(x)
    x = self.conv2(x)
    x = self.conv3(x)

    # Transpor para formato LSTM:(batch, time_steps, features)
    x = x.transpose(1, 2)

    # LSTM
    x, _ = self.lstm(x)

    # Usar apenas a última saída temporal
    x = x[:, -1, :]
    # Classificação
    x = self.fc(x)
    return x

```

Configuração do modelo 2 B

## ENSEMBLE E FUSÃO DE MODELOS

Para aumentar a robustez e precisão das previsões, implementamos uma estratégia de ensemble que combina as saídas de múltiplos modelos.

A fusão é realizada através de votação ponderada, onde cada modelo contribui com um peso proporcional à sua confiança e desempenho histórico.

```
class BioLingoEnsemble:  
    def __init__(self, species_model, vocalization_model):  
        self.species_model = species_model  
        self.vocalization_model = vocalization_model  
        self.species_weight = 0.6  
        self.vocalization_weight = 0.4  
  
    def predict(self, spectrogram, mfcc):  
        # Previsão de espécie  
        species_logits = self.species_model(spectrogram)  
        species_probs = torch.softmax(species_logits, dim=1)  
        species_pred = torch.argmax(species_probs, dim=1)  
        species_confidence = torch.max(species_probs, dim=1)[0]  
  
        # Previsão de tipo de vocalização  
        voc_logits = self.vocalization_model(mfcc)  
        voc_probs = torch.softmax(voc_logits, dim=1)  
        voc_pred = torch.argmax(voc_probs, dim=1)  
        voc_confidence = torch.max(voc_probs, dim=1)[0]  
  
        # Confiança combinada  
        combined_confidence = (  
            self.species_weight * species_confidence +  
            self.vocalization_weight * voc_confidence  
        )  
  
        return {  
            'species': species_pred,  
            'species_confidence': species_confidence,  
            'vocalization_type': voc_pred,  
            'vocalization_confidence': voc_confidence,  
            'combined_confidence': combined_confidence  
        }
```

## PIPELINE DE TREINO

**CARREGAMENTO E PRÉ-PROCESSAMENTO DE DADOS:** Engenharia de recursos (espectrogramas e MFCCs), aumento de dados (data augmentation), treino com validação contínua, avaliação em conjunto de teste, ajuste de hiperparâmetros e registo de métricas e artefatos.

## HIPERPARÂMETROS PRINCIPAIS

Selecionados através de uma combinação de valores padrão da literatura e ajuste manual baseado em validação cruzada. Para o classificador de espécies (ResNet-50), utilizamos taxa de aprendizagem inicial de 0.001 (fase 1) e 0.0001 (fase 2), batch size de 32, épocas máximas de 100 com early stopping, dropout de 0.5 e 0.3 nas camadas finais e weight decay de 0.0001.

Para o classificador de vocalização (CNN+LSTM), utilizamos taxa de aprendizagem inicial de 0.001, batch size de 64, épocas máximas de 80 com early stopping, dropout de 0.5 e 0.3 e gradient clipping com max\_norm de 1.0.

## ESTRATÉGIAS DE OTIMIZAÇÃO

**REGULARIZAÇÃO:** Para prevenir overfitting, aplicamos múltiplas técnicas de regularização, incluindo dropout nas camadas totalmente conectadas, weight decay (L2 regularization) no otimizador, early stopping baseado na perda de validação, aumento de dados durante o treino e batch normalization nas camadas convolucionais.

**BALANCEAMENTO DE CLASSES:** O desequilíbrio de classes é uma realidade nos dados bioacústicos, com algumas espécies tendo muito mais gravações do que outras. Implementamos pesos de classe na função de perda (inversamente proporcionais à frequência), oversampling de classes minoritárias durante o treino e focal loss para focar o treino em exemplos difíceis.

**AJUSTE DE HIPERPARÂMETROS:** Utilizamos Optuna, uma biblioteca de otimização de hiperparâmetros, para realizar busca bayesiana dos melhores valores. Os hiperparâmetros otimizados incluem taxa de aprendizagem, tamanho de batch, número de camadas e neurônios, taxas de dropout e weight decay.

## AVALIAÇÃO DE DESEMPENHO

**MÉTRICAS PRIMÁRIAS:** Para avaliar o desempenho dos modelos, utilizamos um conjunto abrangente de métricas que capturam diferentes aspectos da qualidade das previsões.

**ACURÁCIA GLOBAL:** Percentagem de previsões corretas sobre o total de previsões. Métrica principal para avaliação geral do modelo.

**PRECISÃO POR CLASSE:** Para cada espécie ou tipo de vocalização, calculamos a precisão (proporção de previsões positivas que são verdadeiramente positivas). Importante para avaliar a confiabilidade das identificações.

**RECALL POR CLASSE:** Para cada classe, calculamos o recall (proporção de exemplos positivos que foram corretamente identificados). Crucial para garantir que não estamos a perder espécies raras ou ameaçadas.

**MATRIZ DE CONFUSÃO:** Visualização das previsões corretas e incorretas para cada par de classes, permitindo identificar confusões sistemáticas entre espécies similares.

**F1-Score:** Média harmónica entre precisão e recall, fornecendo uma métrica equilibrada. Utilizamos tanto o F1-score macro (média não ponderada) quanto o F1-score ponderado (ponderado pela frequência de cada classe).

**Top-K Accuracy:** Percentagem de casos em que a classe correta está entre as K previsões mais prováveis. Utilizamos K=3 e K=5 para avaliar a utilidade do modelo em cenários onde o utilizador pode escolher entre múltiplas opções.

**Métricas Secundárias:** Além das métricas primárias, monitorizamos métricas secundárias que fornecem insights adicionais sobre o desempenho do modelo. Curvas ROC e AUC: Para cada classe, calculamos a curva ROC (Receiver Operating Characteristic) e a área sob a curva (AUC), avaliando a capacidade do modelo de discriminar entre classes.

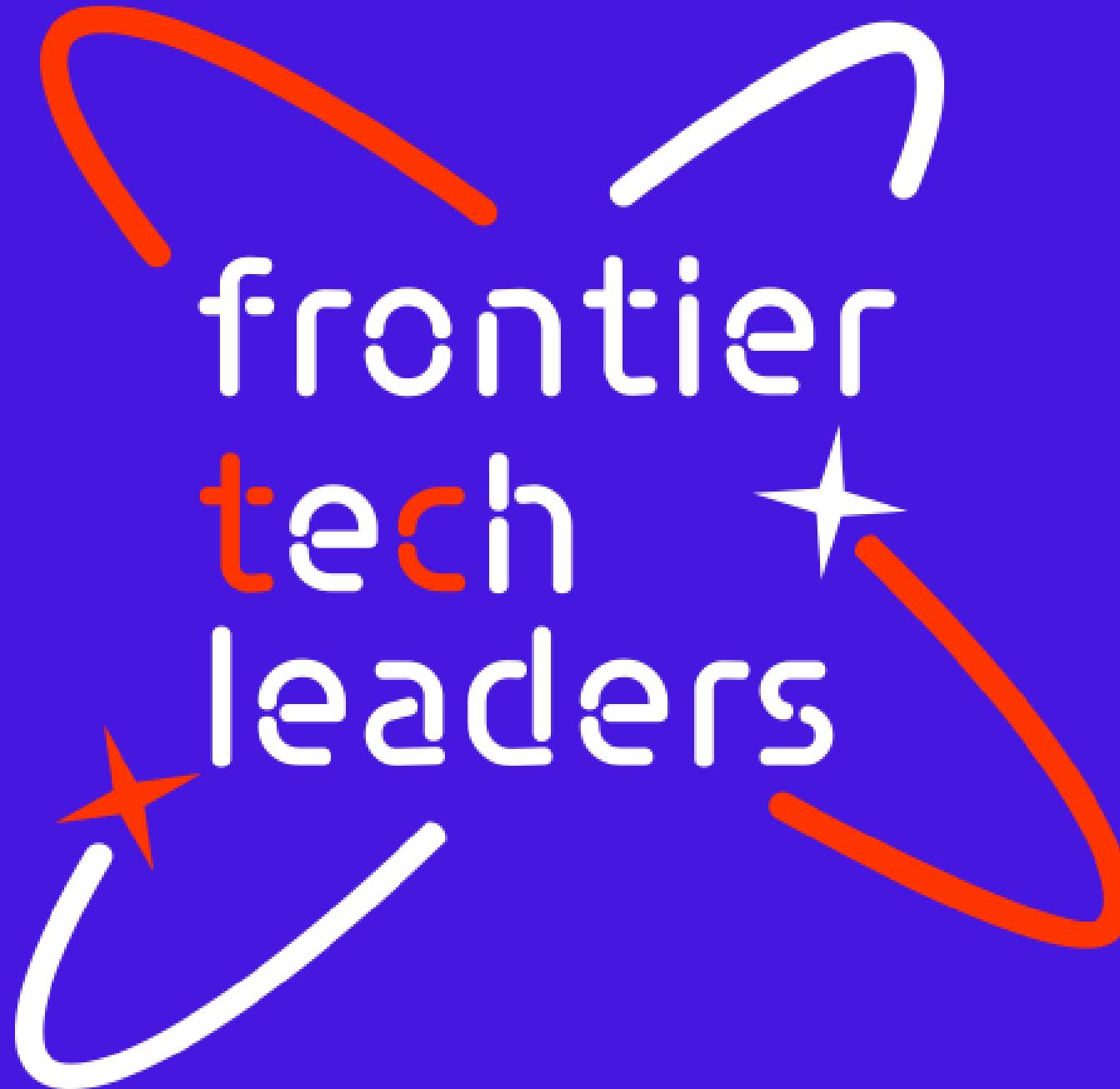
**Calibração de Confiança:** Avaliamos se as probabilidades previstas pelo modelo refletem a verdadeira probabilidade de acerto, utilizando diagramas de calibração e Expected Calibration Error (ECE).

**Tempo de Inferência:** Medimos o tempo necessário para processar uma gravação de áudio, desde o pré-processamento até a previsão final. Objetivo: < 2 segundos em hardware de servidor, < 5 segundos em dispositivos móveis.

## INFRAESTRUTURA DE TREINO

**HARDWARE :** O treino dos modelos é realizado em GPUs NVIDIA (Tesla T4 ou superior) disponíveis em plataformas de nuvem como Google Colab Pro, AWS EC2 (instâncias p3) ou Azure ML. Para o MVP, estimamos 40-60 horas de treino total para ambos os modelos.

**SOFTWARE :** Utilizamos PyTorch 2.0+ como framework principal de deep learning, Librosa para processamento de áudio, Torchvision para transformações de imagens, Scikit-learn para métricas e validação, Weights & Biases (wandb) para rastreamento de experimentos e MLflow para gestão de modelos.

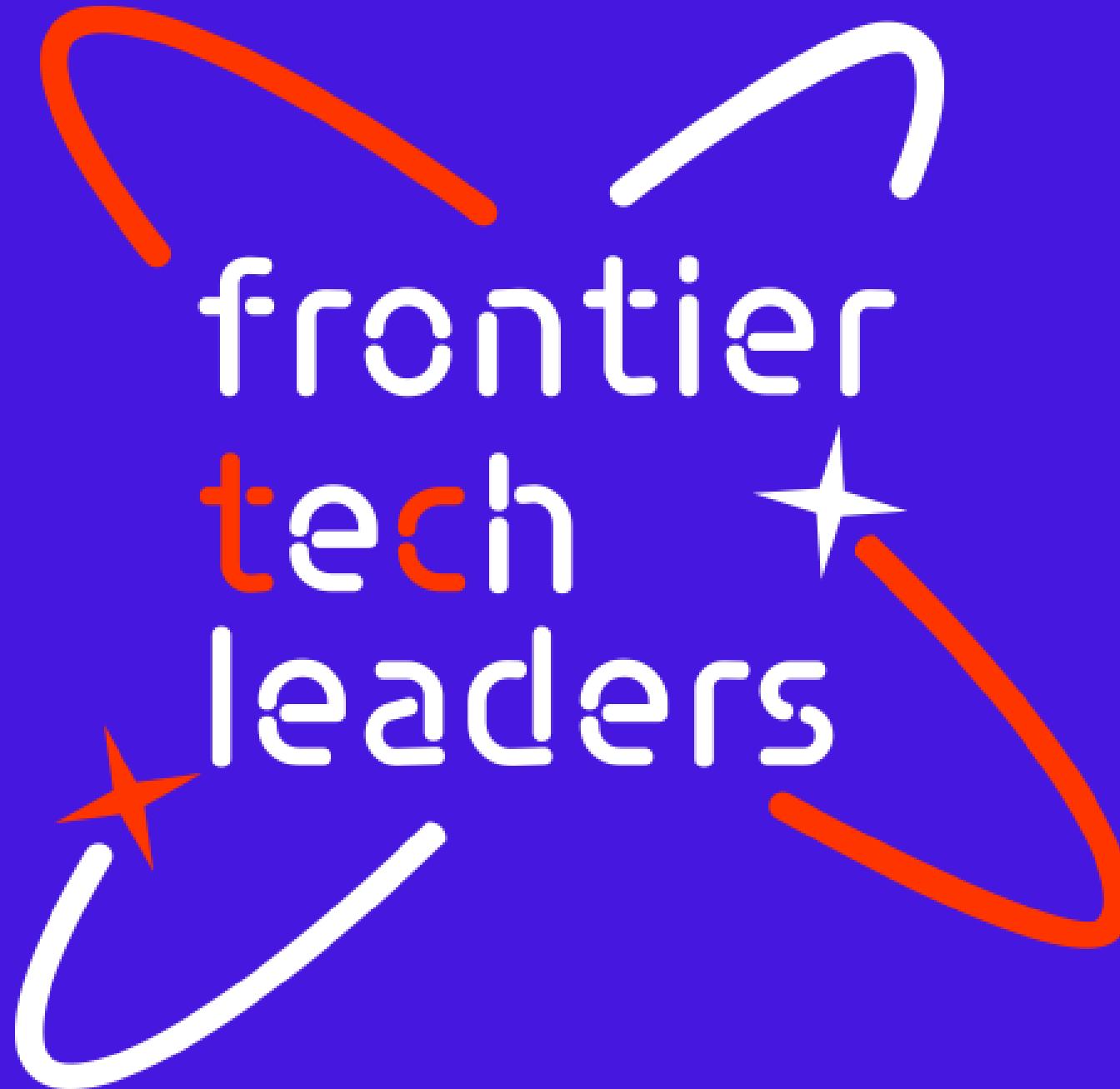


# Resultado



# Resultados

- Apresente os resultados do seu projeto aqui.



Implantação



# Implantação

- Visão geral da fase de implantação
- Serialização de modelos, veiculação e integração de API

# Conclusão e Trabalho futuro



Obrigado!

frontier  
tech  
leaders