Universidade Estadual do Paraná
Campus Apucarana

FABRÍCIO PEREIRA DINIZ

CLASSIFICAÇÃO DE ESPÉCIES DE PÁSSAROS NO TERRITÓRIO BRASILEIRO COM BASE EM IMAGENS E METADADOS GEO-TEMPORAIS

APUCARANA–PR

2025

FABRÍCIO PEREIRA DINIZ

CLASSIFICAÇÃO DE ESPÉCIES DE PÁSSAROS NO TERRITÓRIO BRASILEIRO COM BASE EM IMAGENS E METADADOS GEO-TEMPORAIS

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Paraná para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. José Luis Seixas Junior
Coorientador: Prof. Dr. Rodrigo Clemente Thom de Souza

APUCARANA–PR

2025

Ficha catalográfica elaborada pelo Sistema de Bibliotecas da UNESPAR e
Núcleo de Tecnologia de Informação da UNESPAR, com Créditos para o ICMC/USP
e dados fornecidos pelo(a) autor(a).

Pereira Diniz, Fabrício
   Classificação de Espécies de Pássaros no Território Brasileiro com Base em Imagens e Metadados Geo-temporais / Fabrício Pereira Diniz. -- Apucarana-PR,2025.
   77 f.

   Orientador: José Luis Seixas Junior.
   Coorientador: Rodrigo Clemente Thom de Souza.
   Trabalho de Conclusão de Curso, Ciência da Computação - Universidade Estadual do Paraná, 2025.

   1. Classificação Multimodal. 2. Classificação Visual Fina. 3. Arquiteturas Transformer. 4. Classificação Downstream. 5. Ciência Cidadã. I - Luis Seixas Junior, José (orient). II - Clemente Thom de Souza, Rodrigo (coorient). III - Título.

FABRÍCIO PEREIRA DINIZ

CLASSIFICAÇÃO DE ESPÉCIES DE PÁSSAROS NO TERRITÓRIO BRASILEIRO COM BASE EM IMAGENS E METADADOS GEO-TEMPORAIS

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Paraná para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. José Luis Seixas Junior
Universidade Estadual do Paraná
Orientador

Prof. Dr. Lailla Milainny Siqueira Bine
Universidade Estadual do Paraná

Prof. Dr. Guilherme Corredato Guerino
Universidade Estadual do Paraná

Apucarana–PR, 17 de dezembro de 2025

Este trabalho é dedicado à minha família e às circunstâncias que fortalecem a importância da contemplação do ordinário.

AGRADECIMENTOS

Agradeço aos orientadores que auxiliaram no desenvolvimento deste trabalho, sendo fundamentais para a melhor compreensão dos assuntos aqui abordados, agradeço também aos professores responsáveis por ministrar a disciplina de Inteligência Artificial durante o período da graduação que, por sua vez, ampliaram minha visão sobre o desenvolvimento da ciência no Brasil e, com suas notórias habilidades acadêmicas, cultivaram meu apreço pela área.

Meu reconhecimento ao Manna Team e toda sua equipe, que prestaram total suporte e disponibilizaram seu servidor para a realização da pesquisa. O profissionalismo dos envolvidos não apenas facilitou os processos experimentais, mas também contribuiu para o aprendizado.

"E memo se tu não souber voar, agradeça suas asas
Porque o dia de sair do ninho chega
Mas não apenas desapareça, não enriqueça suas mágoas
Vivi da vida de pássaro que não sabia que podia voar"
— Gabriel "Big Rush" Almeida Mota, PÁSSARO

DINIZ, F. P.. **Classificação de Espécies de Pássaros no Território Brasileiro com Base em Imagens e Metadados Geo-Temporais**. 75 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual do Paraná, Apucarana–PR, 2025.

RESUMO

Este trabalho explora o uso de dados de ciência aberta para a classificação multimodal de aves no território brasileiro, destacando disparidades na distribuição dos dados entre os estados. Essas diferenças provavelmente estão relacionadas à ocorrência dos avistamentos reportados por cidadãos, indicando que, embora a ciência cidadã aumente a disponibilidade de amostras, os experimentos ainda dependem da distribuição geográfica dos colaboradores. A distribuição das espécies indica que a maior quantia de indivíduos está concentrada no primeiro e último quartil, especialmente no último, indicando que poucas espécies dominam as ocorrências enquanto muitas são incomuns, a delimitação de escopo é dependente do domínio do problema. Em relação ao desempenho dos modelos, a quantidade de exemplos não mostrou uma ligação forte com a acurácia. As mudanças na performance foram, em grande parte, por conta da arquitetura escolhida, principalmente quando só imagens foram usadas. Nos casos em que só dados em forma de tabela foram usados ou quando as duas modalidades foram combinadas, os resultados foram semelhantes, indicando que a escolha do modelo tem um efeito pequeno sobre a métrica utilizada. Os classificadores alcançaram médias de acurácia de 0,8550 para dados tabulares, 0,3458 para imagens e 0,8606 para a combinação das modalidades. A concatenação dos *embeddings* das duas modalidades demonstrou ganho de acurácia em alguns cenários, criando um espaço híbrido de características que combina a estabilidade dos dados tabulares com a expressividade das imagens. Essa abordagem reduz a sensibilidade à escolha do algoritmo e oferece boa capacidade de representação. De modo geral, este estudo estabelece uma baseline para a classificação multimodal de aves no Brasil e fornece subsídios para a aplicação de diferentes arquiteturas em problemas de Classificação Visual Fina.

Palavras-chave: Classificação Multimodal. Classificação Visual Fina. Arquiteturas Transformer. Classificação Downstream. Ciência Cidadã.

DINIZ, F. P.. **Classification of Bird Species in the Brazilian Territory Based on Images and Geo-Temporal Metadata**. 75 p. Final Project (Bachelor of Science in Computer Science) – State University of Paraná, Apucarana–PR, 2025.

ABSTRACT

This work explores the use of open science data for multimodal bird classification in Brazil, highlighting disparities in data distribution across states. These differences are likely related to the occurrence of sightings reported by citizen scientists, indicating that, although citizen science increases sample availability, experiments still depend on the geographic distribution of contributors. Species distribution indicates that the highest number of individuals is concentrated in the first and last quartiles, especially in the last one, suggesting that few species dominate occurrences while many are rare, and that the scope delimitation depends on the problem domain. Regarding model performance, the number of samples did not show a strong correlation with accuracy. Performance variations were largely due to the chosen architecture, especially when only images were used. In scenarios where only tabular data was used or when both modalities were combined, results were similar, indicating that model choice has limited effect on the selected metric. The classifiers achieved average accuracies of 0.8550 for tabular data, 0.3458 for images, and 0.8606 for the combination of modalities. The concatenation of embeddings from both modalities showed accuracy gains in some scenarios, creating a hybrid feature space that combines the stability of tabular data with the expressiveness of images. This method makes it less affected by the choice of algorithm while still offering powerful representation abilities. In conclusion, this research sets a standard for bird classification in Brazil using multiple data types and offers ideas for using various designs in detailed visual classification challenges.

Keywords: Multimodal Classification. Fine-Grained Visual Classification. Transformer Architectures. Downstream Classification. Citizen Science.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CBRO	Comitê Brasileiro de Registros Ornitológicos
RNNs	Redes Neurais Recorrentes
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
GRNNs	General Regression Neural Network
CNNs	Convolutional Neural Networks
ViTs	Vision Transformers
MLP	Multi-Layer Perceptron
ID3	Iterative Dichotomizer 3
GBDT	Árvores de Decisão com Boosting de Gradiente
t-SNE	t-Distributed Stochastic Neighbor Embedding
DBM	Máquina de Boltzmann Profunda
VLP	Vision-and-Language Pre-training
ViLT	Vision-and-Language Transformer
CLIP	Contrastive Language-Image Pre-training
k-NN	k-Nearest Neighbors
SVM	Support Vector Machine
SVN	Support-Vector Network
RBF	Função de Base Radial
XGBoost	eXtreme Gradient Boosting
CART	Classification and Regression Trees
BirdSAT	Cross-View Contrastive Masked Autoencoders for Bird Species Classification and Mapping
FGVC	Classificação Visual Fina de Espécies
CL	Contrastive Learning
MIM	Masked Image Modeling
CVE-MAE	Cross-View Embed MAE

CVM-MAE	Cross-View Metric MAE
DuSAFNet	Dual-path spectro–temporal Attention & Fusion Network
STA	Módulo de Atenção Espectro-Temporal
DPFM	Módulo de Extração de Recursos de Caminho Duplo
GFM	Mapeamento de Fusão Controlada

SUMÁRIO

1 INTRODUÇÃO

Estudos recentes demonstram que a conversão de habitats naturais em mosaicos de uso antrópico (e.g., agricultura, pastagens) reduz a riqueza de espécies, afetando principalmente as sensíveis às bordas e dependentes de fragmentos florestais extensos [1].

O aumento das atividades relacionadas à agricultura industrial vem impactando o habitat das espécies em solo brasileiro. Um exemplo é a Amazônia, onde espécies comuns demonstram contribuições únicas para o ecossistema [2].

A presença humana decorrente da atividade econômica pode ser observada tanto no surgimento de novos moradores em áreas de mata, semelhante ao ocorrido entre 1960 e 1980, quando a população de Manaus triplicou com a criação do Distrito Industrial [3], quanto nas rodovias que interligam a região ao resto da Amazônia. Estas servem para transportar produtos retirados das áreas exploradas, atuando como vetores ativos que impulsionam a interiorização da destruição florestal [4].

Esse avanço humano sobre as áreas naturais não afeta apenas a vegetação, mas também a fauna local, segundo o Comitê Brasileiro de Registros Ornitológicos[1](CBRO), o conhecimento ornitológico no Brasil vem crescendo graças a contribuições, especialmente por fotografias. Seu conteúdo serve como base taxonômica aviária para o maior portal de ciência cidadã sobre aves brasileiras na internet, o Wiki Aves[2] [5]. A invasão humana no habitat desses animais ao longo do tempo e o próprio avanço da tecnologia facilitam a coleta de dados por entusiastas e cientistas.

A identificação automatizada de espécies por meio do monitoramento é fundamental para a conservação ecológica, especialmente em cenários de perda de biodiversidade [6].

No entanto, embora a urbanização e a industrialização representem desafios para a preservação dessas espécies em seu habitat natural, a classificação das que apresentam maior incidência ao longo dos anos obtém viabilidade por meio do treinamento de modelos. Estas estão presentes na Lista de Aves do Brasil[3] e também no Wiki Aves, onde é disponibilizado um acervo de fotografias juntamente com metadados associados.

Tais informações, colhidas ao longo do tempo, podem ajudar a distinguir, catalogar e registrar a incidência das espécies tanto em localidades já conhecidas quanto em novas, podendo ainda revelar possíveis padrões de migração desses animais. Esse processo é essencial para compreender os efeitos do avanço da presença humana e das mudanças decorrentes sobre a fauna, permitindo identificar como ocorre a adaptação das espécies em ambientes antrópicos. Desse modo, a classificação automatizada contribui para mitigar a perda de biodiversidade e a fragmentação dos habitats naturais.

Este trabalho propõe a classificação multimodal de pássaros em território nacional utilizando suas imagens e seus respectivos metadados geo-temporais, escolha fundamentada na natureza do conjunto de dados disponível, oriundo de iniciativas de ciência cidadã. A fusão multimodal permite que informações contextuais, como localização, atributos taxonômicos e temporais façam parte da classificação juntamente com as imagens.

O Capítulo 2 apresenta a fundamentação teórico-metodológica e os trabalhos correlatos; O Capítulo 3

[1] https://www.cbro.org.br/
[2] https://www.wikiaves.com.br/
[3] https://www.cbro.org.br/listas/

contém o método de pesquisa do trabalho; O Capítulo 4 descreve os experimentos, e os resultados estão presentes no Capítulo 5; E por fim, a conclusão é descrita no Capítulo 6.

2 FUNDAMENTAÇÃO TEÓRICA

A arquitetura Transformer [7] foi criada para resolver os problemas de paralelização que existiam nas Redes Neurais Recorrentes [8] (RNNs) e nas convoluções usadas em tarefas de transdução de sequências. Arquiteturas como a Long Short-Term Memory [9] (LSTM) e General Regression Neural Network [10] (GRNNs) tinham limitações intrínsecas com relação a paralelização dentro dos exemplos de treinamento, um problema para sequências de maior comprimento.

A ideia por trás do Transformer é que, ao contrário das outras estruturas, ele não precisa usar recorrências ou convoluções, utilizando mecanismos de Atenção em seu lugar [7]. O mecanismo de Atenção funciona associando três vetores: *queries*, *keys* e *values*, cada posição da entrada produz estes vetores, a partir disso a Atenção calcula a compatibilidade entre cada *query* e todas as *keys*, gerando pesos que determinam quanto cada value contribui para a saída.

A Atenção consiste em atribuir pesos diferentes às partes da entrada, de maneira que o modelo seja mais seletivo em elementos mais relevantes, o mecanismo mede a relação entre cada elemento da sequência com os demais, gerando uma combinação ponderada.

Através da Autoatenção o Transformer avalia a importância relativa de cada entrada em relação aos outros elementos da sequência, diferentemente de arquiteturas que utilizam a recorrência para processar os dados de forma sequencial e dependem de estados internos, a Atenção considera toda a sequência ao atribuir pesos para a informação de cada token com relação a sua representação final. Em arquiteturas baseadas em convolução as características locais são extraídas através de filtros deslizantes sobre a vizinhança espacial, diferente do Transformer que não assume relações locais.

O mecanismo de Autoatenção o Transformer reduz o número mínimo de operações sequenciais necessárias, tornando mais fácil a modelagem de dependências de longo alcance e facilitando a paralelização. O mecanismo de Atenção, mecanismo central dos Transformers funciona associando uma *query* (Q) e um conjunto de pares *chave-valor* (K, V) a uma saída.

A saída é calculada como uma soma ponderada dos valores (V), onde o peso atribuído a cada valor é determinado por uma função de compatibilidade entre a *query* e sua respectiva *key* (K). A implementação do Produto Escalar Escalonado no Mecanismo de Atenção aditiva otimiza a multiplicação de matrizes, o cálculo matricial para Atenção é expresso como:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

Q, K e V são matrizes onde cada linha representa um vetor da respectiva sequência, a partir disso a softmax transforma o vetor resultante do produto escalar em uma distribuição de probabilidades, onde cada elemento recebe um valor entre 0 e 1 de modo que a soma seja igual a 1, K^T representa a transposta de K que calcula a compatibilidade entre Q e K. Essa normalização enfatiza os elementos mais relevantes e atenua os menos importantes.

O termo $1/\sqrt{d_k}$ é um fator de escalonamento, onde d_k representa o tamanho dos vetores de *key*. Esse ajuste é muito importante, porque quando d_k é alto, os produtos escalares tendem a aumentar muito, fazendo com que a função softmax vá para áreas com gradientes muito baixos, dificultando o aprendizado.

Com essa abordagem, a estrutura consegue conectar diferentes posições, tornando mais fácil entender as relações globais.

A Atenção Multi-Cabeças é uma extensão do mecanismo de Atenção e permite que o modelo foque em diferentes relações da sequência de forma simultânea, cada cabeça aplica uma projeção linear independente aos vetores Q, KeV e realiza a atenção separadamente, esta extensão permite que o modelo capture padrões em diferentes subespaços. Isso permite que o modelo capte informações de diferentes partes de forma simultânea, melhorando ainda mais a habilidade de representar informações de forma global.

2.1 Classificações de imagem

A utilização de Convolutional Neural Networks [11] (CNNs) foi até a proposta dos Vision Transformers [12] (ViTs) o estado da arte relacionado a classificação de imagens [13]. Por mais que a arquitetura obtivesse resultados satisfatórios continuava sendo dependente de suas convoluções para capturar padrões nas imagens. Apesar dos avanços a sua limitação de captura a padrões locais, dependendo de configurações de filtros e camadas impulsionou o desenvolvimento dos Vision Transformers [14].

Entretanto ainda existem cenários onde as CNNs podem performar melhor que os ViTs como, por exemplo, quando há restrição de dados, hardware ou foco em padrões locais. Por sua aplicação ter menos limitações vemos que mesmo com alguns contrapontos ela continua sendo empregada em diversos cenários [15; 16].

2.1.1 Classificação com Vision Transformer

O Vision Transformer adapta a arquitetura Transformer, originalmente desenvolvida para processamento de linguagem natural, para tarefas de visão computacional, os ViTs utilizam o mecanismo de Atenção para modelar relações globais entre diferentes regiões da imagem [12; 14].

Inicialmente, uma imagem de dimensão $H \times W \times C$ é dividida em N blocos quadrados de tamanho $P \times P$, de modo que $N = \frac{HW}{P^2}$. Cada bloco $x_i \in \mathbb{R}^{P \times P \times C}$ é linearmente projetado em um vetor de dimensão fixa d por meio de uma camada de *embedding*:

$$z_i = E(x_i) \in \mathbb{R}^d, \tag{2.2}$$

onde $E(\cdot)$ representa a projeção linear aprendida. Para incorporar informações espaciais, cada bloco *embedding* recebe um vetor de posição p_i, gerando o *embedding* final do bloco:

$$\tilde{z}_i = z_i + p_i. \tag{2.3}$$

A sequência de *embeddings* $\{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_N\}$ é então processada por um *Transformer Encoder*, composto por camadas de Atenção Multi-Cabeças, normalização e redes feed-forward, cada camada feed-forward consiste em uma rede neural totalmente conectada aplicada a cada bloco de forma individual, refinando suas representações e extraindo características não-lineares. O mecanismo de Atenção permite que cada bloco interaja com todos os outros blocos da imagem, capturando dependências de longo alcance e relações globais:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.4}$$

onde Q, K e V são as matrizes de consulta, chave e valor derivadas dos *embeddings* dos blocos, e d_k é a dimensão das chaves.

A matriz de consulta Q contém representações que perguntam por informação relevante em outros blocos, na matriz chave K temos as representações que são comparadas com as consultas onde a matriz de valor V armazena as informações que serão combinadas de acordo com os pesos calculados.

Para realizar a classificação, a sequência de blocos recebe um token especial [CLS], cuja saída após o encoder representa um resumo da imagem inteira. Este vetor é passado por uma camada totalmente conectada para gerar as probabilidades das classes:

$$y = \text{softmax}(W_{\text{cls}}\, z_{\text{cls}} + b_{\text{cls}}), \tag{2.5}$$

onde z_{cls} é o *embedding* do token [CLS] após o encoder e $W_{\text{cls}}, b_{\text{cls}}$ são os parâmetros da camada de classificação.

Essa abordagem permite que os ViTs capturem relações globais de forma eficiente, superando limitações das CNNs em datasets grandes, ao modelar padrões complexos de maneira mais flexível.

2.1.2 Swin Transformer for Fine-Grained Recognition

A Figura 1 [17] ilustra a arquitetura SwinFG, na qual se efetua a integração de Mapas de Atenção locais dentro de uma estrutura global pensando em resolver problemas de Classificação Visual Fina de Espécies (FGVC) ou problemas similares, isto é, tarefas de distinção entre categorias próximas ou parecidas.

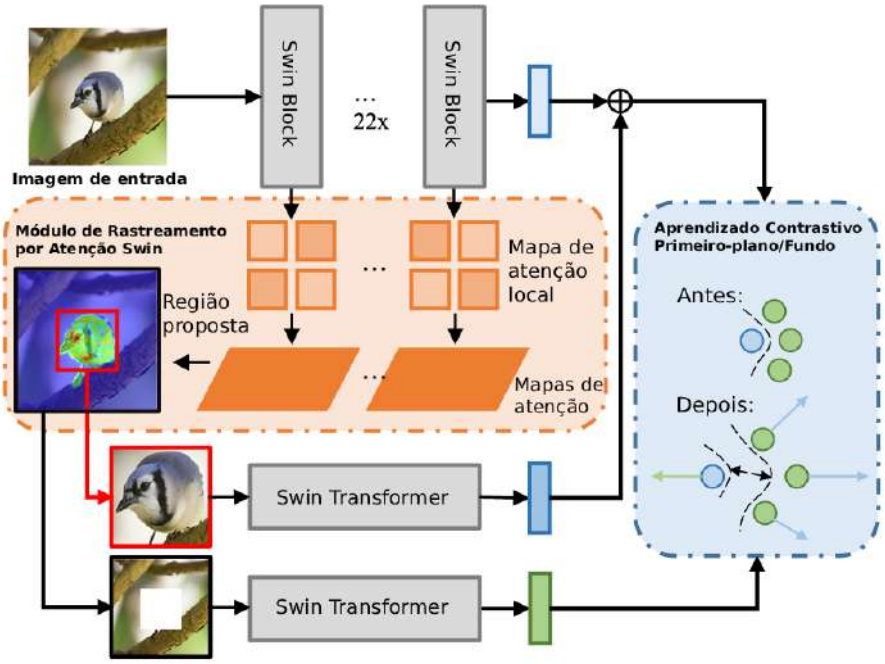


Figura 1 – Esquema da arquitetura do SwinFG.

Com fundamentação nos Mapas de Atenção gerados em distintas janelas do Swin Transformer, o modelo executa uma fusão iterativa, preservando a rastreabilidade da evolução dos pesos ao longo das camadas.

A metodologia é construída unindo Mapas de Atenção locais em uma matriz de afinidade global. Essa tabela é criada usando transformações repetitivas que aplicam multiplicação de matrizes de forma contínua, um aspecto importante que garante o valor de cada parte da imagem. Isso aumenta a diferença entre as áreas relevantes e o fundo, evitando que dados desnecessários possam prejudicar a acurácia do modelo.

O mecanismo de janelas deslocadas do Swin Transformer, também representado na Figura 1, restringe a operação de Autoatenção a regiões menores sem perder a capacidade de capturar relações de longo alcance, este mecanismo organiza a entrada em janelas de blocos, estas janelas são deslocadas entre camadas permitindo que as informações da imagem se misturem gradualmente.

2.2 Classificações tabulares

A aplicação de tarefas de classificação em dados tabulares é documentada em diversos contextos, um dos primeiros modelos que foi empregado nesse tipo de tarefa foi a Multi-Layer Perceptron (MLP), treinadas com retropropagação [8; 18]. Essas redes conseguem entender padrões complexos a partir de grupos de dados. A aplicação tem maior aplicabilidade quando existem relações entre diferentes variáveis que não seguem uma linha reta, permitindo aproximar funções complexas.

Por sua fácil interpretabilidade e adaptação uma alternativa às redes neurais são algoritmos com base nas árvores de decisão. O ID3 por exemplo, cria árvores de decisão utilizando medidas de entropia para escolher os atributos mais importantes de cada nó [19]. Com base nessa ideia, apareceram os métodos, como o Random Forest que junta várias árvores de decisão que funcionam sozinhas para diminuir a variância e melhorar a precisão do modelo [20].

Outro progresso significativo na classificação de dados tabulares é o boosting, que cria modelos ligeiramente melhores, denominados modelos fracos um após o outro, como árvores simples, para melhorar os erros dos modelos que vieram antes. Isso leva a um modelo mais forte e confiável [21]. Métodos de boosting, como o Gradient Boosting Machine, se tornaram muito populares porque conseguem trabalhar bem com dados de tabelas que são variados e complicados, muitas vezes se saindo melhor do que redes neurais e árvores isoladas em termos de robustez e acurácia.

Assim, a literatura [22; 23] mostra que, mesmo com o progresso das redes neurais, técnicas tradicionais que usam árvores e conjuntos ainda são consideradas importantes para classificar dados em tabelas, principalmente quando se precisa de interpretação, solidez e bom desempenho em conjuntos de dados variados.

2.2.1 TabTransformer

A modelagem de dados tabulares representa a forma de dados mais comum em aplicações reais [23], historicamente o estado da arte para dados tabulares tem sido dominado por métodos de conjunto baseados em árvores, como as Árvores de Decisão com Boosting de Gradiente (GBDT, do inglês Gradient Boosting Decision Trees - GBDT). Em contraste, modelos baseados em Aprendizado Profundo dominam as áreas de imagem e texto.

Modelos clássicos de Aprendizado Profundo, como o Multi-Layer Perceptron, utilizam *embeddings* paramétricos para representação de características categóricas aprendidas durante o treinamento, mas sofrem de limitações significativas, incluindo a falta de interpretabilidade e baixa robustez contra dados sem completude ou ruidosos. Mais importante, os MLPs geralmente não conseguem igualar a precisão de previsão dos modelos baseados em árvores, como o GBDT, na maioria dos conjuntos de dados.

A proposta da arquitetura do TabTransformer [24] é abordar as limitações do MLP e fechar a lacuna de desempenho em relação ao GBDT através da Autoatenção que antes era aplicada ao domínio de Processamento de Linguagem Natural. Camadas Transformer baseadas em Autoatenção transformam os *embeddings* paramétricos de características categóricas em *embeddings* contextuais, sendo composto por três componentes principais: i) Uma camada de *Embedding* de Coluna; ii) Uma pilha de N camadas de Transformer; iii) Uma MLP posicionada no topo da arquitetura, responsável pela etapa final de predição.

Embedding de Coluna e Identificador Único

Cada característica categórica x_i é embutida em um *embedding* paramétrico de dimensão d, denotado por $e_{\varphi_i}(x_i) \in \mathbb{R}^d$. Para dados tabulares, o método de *embedding* de coluna é único e inclui um identificador único (c_{φ_i}) para distinguir as classes em uma coluna daquelas em outras colunas. O *embedding* para um valor $x_i = j$ é definido pela concatenação:

$$e_{\varphi_i}(j) = [c_{\varphi_i}, w_{\varphi_{ij}}] \tag{2.6}$$

onde $c_{\varphi_i} \in \mathbb{R}^l$ é o identificador único da coluna, e $w_{\varphi_{ij}} \in \mathbb{R}^{d-l}$ é o *embedding* específico do valor da característica. Diferentemente do Transformer original, o TabTransformer não utiliza codificações posicionais, pois os dados tabulares não possuem ordenação intrínseca das características.

Camadas de Transformer e Contextualização

Os *embeddings* paramétricos $E_\varphi(x_{cat}) = \{e_{\varphi_1}(x_1), \ldots, e_{\varphi_m}(x_m)\}$ são alimentadas nas camadas de Transformer, representadas por uma função f_θ. Através da agregação sucessiva de contexto de outros *embeddings*, cada *embedding* paramétrico é transformado em um *embedding* contextual h_i, de modo que:

$$\{h_1, \ldots, h_m\} = f_\theta(\{e_{\varphi_1}(x_1), \ldots, e_{\varphi_m}(x_m)\}) \tag{2.7}$$

Uma camada de Transformer consiste em uma camada de Atenção Multi-Cabeças seguida por uma camada *position-wise feed-forward*, com adição por elemento e normalização de camada após cada subcamada.

O mecanismo de Autoatenção calcula o quanto cada *embedding* de entrada atende aos outros *embeddings*, transformando-o em uma representação contextual. A Atenção é calculada pela matriz $A \in \mathbb{R}^{m \times m}$, definida como:

$$\text{Attention}(K, Q, V) = A \cdot V, \quad \text{onde } A = \text{softmax}\left(\frac{QK^T}{\sqrt{k}}\right) \tag{2.8}$$

Onde m é o número de *embeddings* de entrada (características categóricas) e k é a dimensão dos vetores chave e consulta.

Os *embeddings* contextuais $\{h_1, \ldots, h_m\}$ são concatenados com as características contínuas x_{cont} para formar um vetor de dimensão $(d \times m + c)$, que é então introduzido em um MLP superior g_ψ para prever o alvo y.

O treinamento do TabTransformer é realizado de forma *end-to-end*, minimizando a função de perda $L(x, y)$ aprendendo simultaneamente os parâmetros φ (para *embedding* de coluna), θ (para camadas Transformer) e ψ (para o MLP superior):

$$L(x, y) \equiv H(g_\psi(f_\theta(E_\varphi(x_{cat})), x_{cont}), y) \tag{2.9}$$

Os *embeddings* contextuais aprendidos pelo TabTransformer são altamente robustos contra dados ruidosos e ausentes, superando o MLP basal nesses cenários. Essa robustez é atribuída à propriedade contextual dos *embeddings*, onde uma característica ruidosa pode extrair informações das características contextuais corretas, permitindo um grau de correção.

O TabTransformer demonstrou superar significativamente o estado da arte em Aprendizado Supervisionado, método que aprende a partir de dados já rotulados corretamente, além disso, se destaca especialmente quando a quantidade de dados não rotulados é grande, igualando seu desempenho com modelos GBDT [24].

2.3 Classificações multimodais

A informação no mundo real é inerentemente multimodal, provindo de múltiplos canais de entrada, como imagens associadas a legendas, ou sinais visuais e auditivos em vídeos. A integração de múltiplas mídias, suas características ou decisões intermediárias para a realização de uma tarefa de análise é referida como fusão multimodal.

A fusão de múltiplas modalidades pode fornecer informações complementares e aumentar a precisão do processo de tomada de decisão geral. Contudo, as diferentes modalidades possuem características distintas, incluindo formatos e taxas de captura variadas, o que impõe desafios à sincronização e à modelagem das correlações [25].

Níveis e Estratégias de Fusão Clássicas

As estratégias de fusão são tradicionalmente classificadas em dois níveis: fusão em nível de característica, ou fusão precoce e fusão em nível de decisão, ou fusão tardia [26].

Na fusão em nível de característica, as características extraídas das modalidades de entrada são combinadas em uma única representação antes de serem enviadas a uma única unidade de análise, ou seja é *single-branch*. Este método produz uma verdadeira representação multimídia, pois as características são integradas desde o início, e exige apenas uma fase de aprendizado. A dificuldade está em combinar características em uma representação comum pois a sincronização temporal entre as características multimodais é complexa de representar.

A fusão em nível de decisão junta escolhas locais feitas com base em características específicas. Essas escolhas, por estarem em um nível de significado, normalmente têm a mesma forma, o que torna a fusão mais fácil e permite a combinação mais flexível de múltiplas decisões. Além disso, a fusão tardia dá a liberdade de utilizar métodos mais apropriados para analisar cada modalidade de forma separada, aplicação *multi-branch*. O principal desafio é que isso exige um esforço maior de aprendizado, já que cada modalidade precisa de uma fase de Aprendizado Supervisionado diferente, além de uma etapa final para combinar tudo.

A fusão tardia tende a fornecer um desempenho ligeiramente melhor para a maioria dos conceitos analisados, quando a fusão precoce é mais eficaz, as melhorias de desempenho são notavelmente mais significativas [26].

Aprendizado Multimodal com Modelos Generativos Profundos

Apesar da utilidade das estratégias de fusão nos níveis de característica e decisão, a modelagem de dados multimodais, onde as modalidades possuem propriedades estatísticas muito distintas, apresenta

desafios significativos para modelos rasos.

Modelos convencionais discriminativos não conseguem lidar muito bem com a falta de certos tipos de entrada ou usar enormes quantidades de dados não rotulados. Uma boa representação que envolve múltiplas modalidades deve servir tanto para tarefas que diferenciam quanto para aquelas que buscam informação, mas também precisa ser simples de conseguir mesmo quando alguns tipos estão faltando, permitindo que dados que estão ausentes sejam preenchidos.

A Máquina de Boltzmann Profunda (DBM do termo em inglês para Deep Boltzmann Machine) [27] é um modelo gráfico não direcionado que aprende uma densidade de probabilidade conjunta sobre o espaço de entradas multimodais. A DBM alcança a fusão aprendendo uma representação unificada a partir dos estados de variáveis latentes.

O modelo é formado juntando DBMs que são feitas para diferentes tipos, colocando uma camada oculta binária extra em cima para unir essas modalidades. Os trajetos de cada tipo podem ser treinados antes, sem supervisão, aproveitando muitos dados que não têm rótulos.

A representação que combina diferentes entradas é obtida a partir da camada oculta central da rede. Essa combinação é considerada a mais útil, pois elimina as dependências específicas de cada tipo de dado quando avança na rede [27].

Uma característica importante do DBM Multimodal é que ele consegue gerar novos dados. Isso significa que ele pode criar informações para modalidades que estão faltando, como produzir texto a partir de uma imagem ou encontrar imagens usando uma descrição em palavras.

2.3.1 Classificação com Transformer Multimodal

O esquema de pré-treinamento e ajuste fino (*pre-train-and-fine-tune*) foi expandido para o domínio conjunto de visão e linguagem, dando origem à categoria de modelos *Vision-and-Language Pre-training* (VLP). Esses modelos são pré-treinados em pares de imagem e texto alinhados, utilizando objetivos como casamento de imagem e texto (*image text matching*) e modelagem de linguagem mascarada (*masked language modeling*), e são subsequentemente ajustados para tarefas multimodais *downstream*.

Historicamente, as abordagens VLP dependiam fortemente de processos de extração de características visuais, que geralmente envolviam a supervisão de região e arquiteturas convolucionais profundas. Tais métodos criavam gargalos de eficiência, pois a extração de características visuais exigia muito mais computação do que as etapas subsequentes de interação multimodal.

A nova geração de arquiteturas multimodais baseadas em Transformer busca superar essas limitações, introduzindo caminhos de processamento visual mais leves e unificados, para as quais se identificam técnicas *single-branch* e *dual-branch* aplicadas a diferentes domínios.

Vision-and-Language Transformer

O Vision-and-Language Transformer [28] (ViLT) é uma arquitetura VLP mínima e monolítica, ao implementar um esquema simples de *visual embedding*, utilizando projeção linear em blocos de imagem, método introduzido pelo Vision Transformer, simplifica as entradas tratando-as de maneira unificada e sem convolução, em vez de extrair características visuais por meio de convoluções como em modelos VLP baseados em regiões, o ViLT transforma diretamente blocos da imagem em embeddings através de uma projeção linear.

Essa abordagem faz com que esta arquitetura seja mais rápida que VLPs baseados em características de região.

Sendo uma arquitetura de fluxo único onde a maior parte da computação é concentrada na modelagem das interações modais e embora tenha sido originalmente projetado e pré-treinado para tarefas como *Visual Question Answering* e recuperação, sua arquitetura permite a adaptação direta para classificação supervisionada de pares imagem-texto. O modelo utiliza a representação agrupada, da sequência multimodal final para alimentar uma cabeça *downstream* para predição de classes.

Contrastive Language-Image Pre-training

O Contrastive Language-Image Pre-training [29] (CLIP) representa uma abordagem distinta, focada no aprendizado de representações visuais transferíveis a partir da supervisão em linguagem natural. Ele foi pré-treinado em um vasto conjunto de 400 milhões de pares de imagem e texto com um objetivo contrastivo simples de prever qual legenda corresponde a qual imagem.

O modelo consiste em dois codificadores separados e igualmente dispendiosos que mapeiam as entradas para um espaço de *embedding* multimodal, onde a similaridade de cosseno entre pares correspondentes é maximizada. Diferentemente do ViLT, a interação entre as modalidades no CLIP é rasa, limitada a um produto escalar entre os vetores de *embedding* extraídos.

Esta característica é crucial para a classificação supervisionada, o CLIP não foi treinado para classificação direta. No entanto, seus *embeddings* podem ser aproveitados em modelos *downstream* para tarefas supervisionadas. A principal aplicação para classificação é a transferência *zero-shot*, o codificador de texto é reutilizado para sintetizar um classificador linear (*zero-shot classifier*) ao codificar os nomes ou descrições das classes, e a predição é feita pelo cálculo da similaridade de cosseno entre o *embedding* da imagem e os *embeddings* das classes textuais.

Perceiver

O Perceiver [30] é um *framework* de *representation learning* projetado para a percepção geral, capaz de processar simultaneamente entradas de alta dimensão de múltiplas modalidades sem depender de pressupostos arquitetônicos específicos de domínio, ou seja, não depende de convoluções, recorrências ou estruturas específicas de dados, ao utilizar atenção e projeções lineares ele processa qualquer tipo de entrada.

Para enfrentar a grande quantidade de informações de entrada, o Perceiver usa um sistema de Atenção assimétrica que ajuda a transformar essas informações em um espaço menor e fixo. Essa transformação é feita por meio de um módulo de Atenção cruzada que converte o conjunto de dados de entrada em um conjunto menor.

A arquitetura então processa o *latent array* através de uma pilha profunda de blocos Transformer Autoatenção. Essa estrutura permite a fusão de informação em todos os níveis, já que o modelo pode iterativamente extrair informações relevantes da entrada através de múltiplos módulos de Atenção cruzada.

Embora o Perceiver tenha sido desenvolvido como um *framework* genérico, e não seja intrinsecamente um classificador direto, ele pode ser aplicado a tarefas *downstream* de classificação supervisionada. A saída do Perceiver é tipicamente obtida pela média do módulo final de Autoatenção latente sobre a dimensão de índice, produzindo um vetor de resumo global que é então projetado para o número de classes alvo por uma camada linear.

2.4 Cabeças de classificação

Cabeças de classificação são camadas finais de um modelo pré-treinado que são adicionadas para tarefas específicas conhecidas como *downstream tasks*. O modelo base aprende as representações a partir do pré-treinamento e a Cabeça de classificação converte as representações em previsões ou categorias específicas, as Cabeças permitem que o mesmo backbone seja reutilizado em múltiplos cenários.

2.4.1 k-Nearest Neighbors

O classificador k-Nearest Neighbors [31] (k-NN) é reconhecido como um procedimento de decisão não paramétrico, o que significa que ele opera independentemente de quaisquer pressupostos sobre as estatísticas subjacentes da distribuição conjunta. Este método atribui a um ponto de amostra que se deseja classificar, denominado x, a mesma classificação do ponto mais próximo encontrado em um conjunto de amostras previamente classificadas.

Para que o procedimento seja executado, é fornecido um conjunto de n pares corretamente classificados: $(\mathbf{x}_1, \theta_1), (\mathbf{x}_2, \theta_2), \ldots, (\mathbf{x}_n, \theta_n)$. As observações $\mathbf{x}_i$ representam as medições de um indivíduo e tomam valores em um espaço métrico X, no qual está definida uma métrica d. As variáveis θ_i representam a categoria à qual o i-ésimo indivíduo pertence, tomando valores no conjunto $\{1, 2, \ldots, M\}$.

O objetivo é estimar a categoria θ de uma nova observação $\mathbf{x}$ utilizando a informação contida no conjunto de pontos classificados. O núcleo do algoritmo reside na identificação do vizinho mais próximo, $\mathbf{x}_i^*$, em relação a $\mathbf{x}$, o vizinho mais próximo é definido como o ponto $\mathbf{x}_i^*$ que minimiza a distância $d(\mathbf{x}_i, \mathbf{x})$ para todos os i no conjunto de n amostras.

Uma vez identificado o vizinho mais próximo $\mathbf{x}_i^*$, a regra do Vizinho Mais Próximo toma a decisão de classificar $\mathbf{x}$ na categoria θ_i^* correspondente a esse vizinho. Este é o procedimento de decisão não paramétrico mais simples desta forma, pois a classificação de $\mathbf{x}$ depende exclusivamente da classificação de seu vizinho mais próximo, ignorando as classificações dos $n - 1$ pontos restantes.

A abordagem simples reside na suposição heurística de que observações que estão próximas terão a mesma classificação, ou pelo menos terão distribuições de probabilidade posteriores quase idênticas em relação às suas respectivas classificações.

A eficácia desta regra é notável, pois, mesmo na análise de grandes amostras, a probabilidade de erro R da regra NN é limitada superiormente por duas vezes a probabilidade de erro de Bayes R^*. A probabilidade de erro de Bayes (R^*) é o mínimo possível sobre todas as regras de decisão, servindo como uma referência para a excelência que não pode ser superada. No sentido de que R é no máximo o dobro de R^*, metade da informação de classificação em um conjunto infinito de amostras está contida no vizinho mais próximo.

Uma extensão deste conceito é a regra do k-Vizinho Mais Próximo (k-NN), que atribui ao ponto não classificado a classe mais representada entre seus k vizinhos mais próximos. No entanto, o procedimento de Vizinho Mais Próximo Único (1-NN) foi mostrado ser admissível entre a classe das regras k-NN para o problema de n amostras, sugerindo que, em certas distribuições, ele é estritamente melhor, pois possui uma probabilidade de erro mais baixa.

No exemplo ilustrado na Figura 2 [32], o k-NN classifica um novo ponto, representado pelo quadrado preto, com base nas classes predominantes entre seus k vizinhos mais próximos.

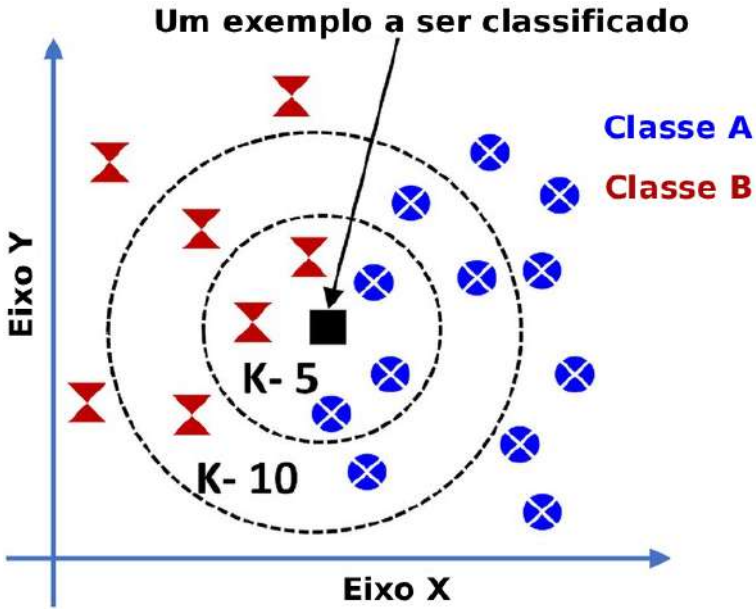


Figura 2 – Demonstração da classificação de um novo ponto com diferentes valores de K.

2.4.2 Random Forest

O Random Forest [20] é um classificador que consiste numa coleção de preditores estruturados em árvores, conforme ilustrado na Figura 3 [33]. Para que este classificador funcione, cada árvore que compõe a floresta depende dos valores de um vetor aleatório, $\mathbf{\Theta}_k$, amostrado de forma independente e com a mesma distribuição para todas as árvores.

O elemento comum em procedimentos de ensemble, método que combina múltiplos modelos para melhorar o desempenho utilizando aleatoriedade, utiliza para a k-ésima árvore, um vetor aleatório $\mathbf{\Theta}_k$ é gerado, independentemente dos vetores aleatórios passados, mas com a mesma distribuição. Uma árvore é então desenvolvida utilizando o conjunto de treinamento e $\mathbf{\Theta}_k$, resultando em um classificador $h(\mathbf{x}, \mathbf{\Theta}_k)$, onde $\mathbf{x}$ é o vetor de entrada. Após um grande número de árvores ser gerado, elas votam para a classe mais popular.

Formalmente, uma Random Forest é definida como um classificador que consiste numa coleção de classificadores estruturados em árvores $\{h(\mathbf{x}, \mathbf{\Theta}_k), k = 1, \dots\}$, onde os $\{\mathbf{\Theta}_k\}$ são vetores aleatórios independentes e identicamente distribuídos, e cada árvore lança um voto unitário para a classe mais popular na entrada $\mathbf{x}$. Uma vez que o número de árvores no forest aumenta, o erro de generalização converge quase certamente para um limite, o que significa que o *overfitting* não é um problema.

A precisão do Random Forest depende fundamentalmente de dois fatores: a força dos classificadores de árvores individuais na floresta e a correlação entre eles. A força (s) é medida pelo valor esperado da função de margem $m_r(\mathbf{X}, Y)$, que quantifica a extensão em que o número médio de votos para a classe correta (Y) excede o voto médio para qualquer outra classe. A correlação média ($\bar{\rho}$) é a correlação entre as funções de margem bruta de pares de árvores.

Um limite superior para o erro de generalização (PE^*) de um Random Forest pode ser expresso em termos destas duas variáveis: $PE^* \leq \bar{\rho}(1 - s^2)/s^2$. Isto implica que, para um bom desempenho, o algoritmo deve injetar aleatoriedade para minimizar a correlação ($\bar{\rho}$), ao mesmo tempo em que mantém a força (s) dos classificadores individuais.

O método proposto no trabalho original [20] alcança isso usando a seleção aleatória de características para determinar a divisão em cada nó da árvore. A forma mais simples é selecionar aleatoriamente, em cada nó, um pequeno grupo de variáveis de entrada para realizar a divisão, as árvores são desenvolvidas até o tamanho máximo e não são podadas.

A utilização da técnica de bagging que consiste em gerar múltiplos conjuntos de treinamento por amostragem com reposição e combinar os resultados dos modelos treinados, em conjunto com a seleção aleatória de características, é frequentemente empregada, onde novos conjuntos de treinamento (bootstraps) são gerados com reposição a partir do conjunto original. Os resultados empíricos mostram que a injeção de aleatoriedade tem como objetivo uma baixa correlação $\bar{\rho}$ enquanto mantém uma força razoável.

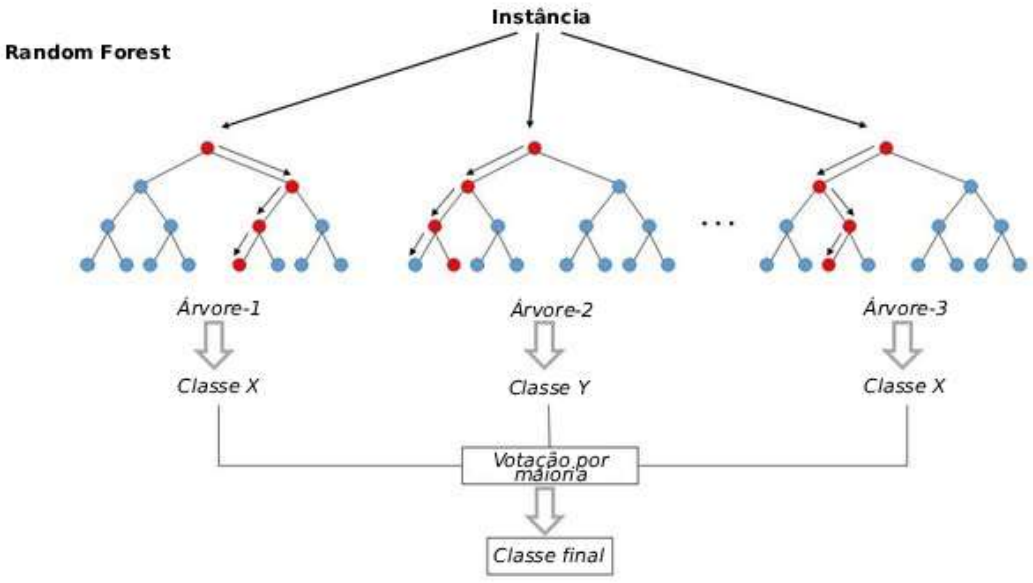


Figura 3 – Classificação de uma instância através de várias árvores de decisão independentes.

2.4.3 Support Vector Machine

A Support Vector Machine [34] (SVM), anteriormente denominada Support-Vector Network (SVN) é uma máquina de aprendizado desenvolvida inicialmente para problemas de classificação binária. O conceito fundamental que a SVN implementa é que os vetores de entrada são transformados de maneira não linear para um espaço de características de dimensão muito alta, Z, onde é construída uma superfície de decisão linear. As propriedades dessa superfície de decisão garantem uma alta capacidade de generalização da máquina de aprendizado.

O principal desafio teórico é encontrar um hiperplano que consiga separar bem os dados, mesmo quando estão em espaços com muitas características. A resposta para esse problema é a ideia de um hiperplano ótimo. O hiperplano ótimo é definido como a função de decisão linear que possui a margem máxima entre os vetores de treinamento das duas classes. A margem é a distância entre a superfície de decisão e os pontos de dados mais próximos de cada classe, que são chamados de vetores de suporte.

A grande capacidade de generalização da SVN decorre de uma limitação teórica: se os vetores de

treinamento forem separados sem erros por um hiperplano ótimo como visto na Figura 4 [35], a probabilidade esperada de erro em um exemplo de teste é limitada superiormente pela razão entre o valor esperado do número de vetores de suporte e o número de vetores de treinamento.

A limitação não depende da dimensionalidade do espaço de separação, permitindo que o algoritmo generalize bem mesmo em espaços de características que podem atingir bilhões de dimensões. Para o caso separável sem erros, o hiperplano ótimo (definido por $\mathbf{w}_0 \cdot \mathbf{z} + b_0 = 0$) é aquele que minimiza $\mathbf{w} \cdot \mathbf{w}$ sujeito às restrições de separação. O vetor de pesos $\mathbf{w}_0$ é uma combinação linear dos vetores de suporte $\mathbf{z}_i$.

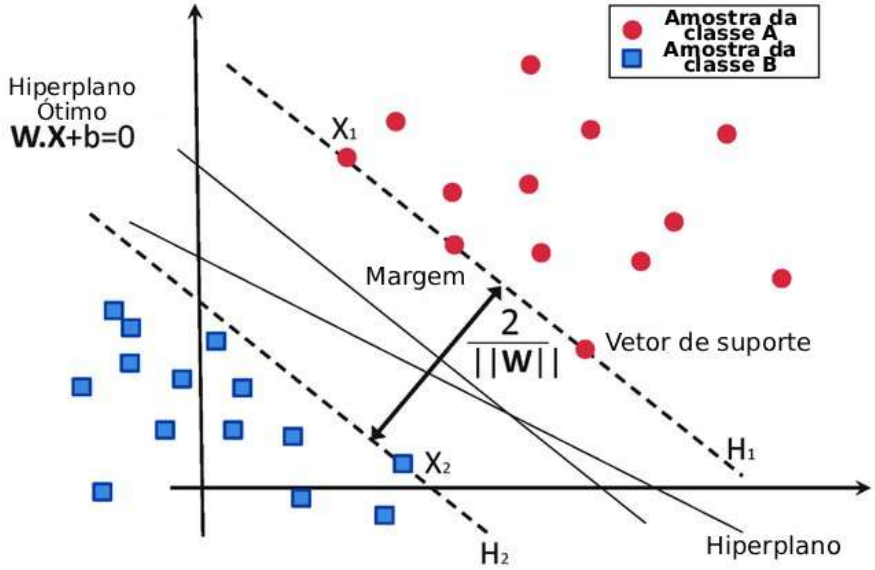


Figura 4 – Representação do hiperplano ótimo e das margens em um classificador SVM, destacando os vetores de suporte das classes A e B.

Para estender a aplicação da SVN a dados de treinamento não separáveis ou com erros, foi introduzido o conceito de margem suave. A margem suave permite erros e desvios ($\xi_i > 0$) nos dados de treinamento, e o problema de otimização passa a ser a minimização de uma função que equilibra a maximização da margem e a penalização dos erros.

O parâmetro C no funcional de minimização permite controlar o trade-off entre a complexidade da regra de decisão, ou seja, a função que determina a classificação de novos exemplos, e a frequência de erro, sendo essencial para o controle da capacidade de generalização da máquina de aprendizado.

O método do Kernel resolve o problema técnico de trabalhar com espaços de características de dimensão extremamente alta de forma computacionalmente eficiente. A função de classificação $f(\mathbf{x})$ de um vetor de entrada $\mathbf{x}$ depende apenas dos produtos escalares entre o vetor de entrada transformado $\phi(\mathbf{x})$ e os vetores de suporte transformados $\phi(\mathbf{x}_i)$.

Em vez de realizar explicitamente essa transformação de alta dimensão, o produto escalar $\phi(\mathbf{u}) \cdot \phi(\mathbf{v})$ é substituído por uma função Kernel $K(\mathbf{u}, \mathbf{v})$, calculada diretamente no espaço de entrada de menor dimensão.

Ao variar a função Kernel, a SVN se torna uma máquina de aprendizado universal, capaz de implementar diferentes redes de aprendizado, como classificadores polinomiais de grau arbitrário ou máquinas de Função de Base Radial (RBF). O processo de otimização da SVN é um problema de programação quadrática

que é resolvido de forma eficiente, determinando a matriz $D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$.

Na matriz, y_i e y_j representam os rótulos das amostras i e j, enquanto $K(\mathbf{x}_i, \mathbf{x}_j)$ mede a similaridade entre essas amostras no espaço definido pelo kernel. Assim, D_{ij} combina informação sobre as classes e a semelhança entre os dados, ao servir como termo quadrático permitindo que a SVN encontre a margem máxima entre as classes.

2.4.4 Logistic Regression

A Logistic Regression [36] é uma metodologia estatística desenhada para analisar situações onde as observações, representadas por Y_i, tomam apenas dois valores, tipicamente denotados como 0 e 1. O objetivo é modelar a dependência da probabilidade de um resultado ser 1, denotada $\theta_i = \text{pr}(Y_i = 1)$, em função de uma ou mais variáveis independentes, X_i.

A formulação padrão reconhece que uma relação linear direta entre a probabilidade θ_i e a variável independente X_i é inadequada, exceto em intervalos estreitos, visto que θ_i deve necessariamente permanecer restrita ao intervalo. Portanto, a forma mais adequada e matematicamente tratável para representar essa relação é a lei logística:

$$\text{logit } \theta_i = \log\left\{\frac{\theta_i}{1-\theta_i}\right\} = \alpha + \beta X_i$$

Nesta formulação, logit θ_i é a transformação logarítmica da razão de chances, e a relação linear é estabelecida entre o logit da probabilidade e a variável independente X_i.

O parâmetro β é o coeficiente de regressão, que mede a inclinação dessa dependência. O objetivo primário é fazer inferência sobre β, tratando α como um parâmetro de perturbação. A interpretação de β é que, se θ_i for pequena, β representa o aumento fracionário em θ_i por unidade de aumento em X_i; se $1 - \theta_i$ for pequeno, β representa a diminuição fracionária em $1 - \theta_i$ por unidade de aumento em X_i.

O processo de classificação e inferência muitas vezes se concentra na distribuição do estatístico suficiente conjunto para os parâmetros α e β, que são $Y = \sum Y_i$ (o número total de 1's) e $X = \sum Y_i X_i$. Para fazer inferência sobre β separadamente, a análise é feita condicionalmente no valor observado de Y (o número total de 1's), eliminando assim o parâmetro de perturbação α.

Em casos mais complexos como na Figura 5 [37], com múltiplas variáveis independentes, a lei logística é generalizada de forma natural. Para testes de hipóteses nulas ausência de regressão, $\beta = 0$, os testes desenvolvidos são não paramétricos, pois a lei logística auxilia apenas na derivação do critério de teste, mas não na distribuição amostral sob a hipótese nula.

Para fins de estimação, especialmente em amostras grandes, o método de máxima verossimilhança ou o método de mínimo logit χ^2 são as abordagens recomendadas, resultando em cálculos de regressão múltipla, que podem ser iterativos ou não iterativos, respectivamente.

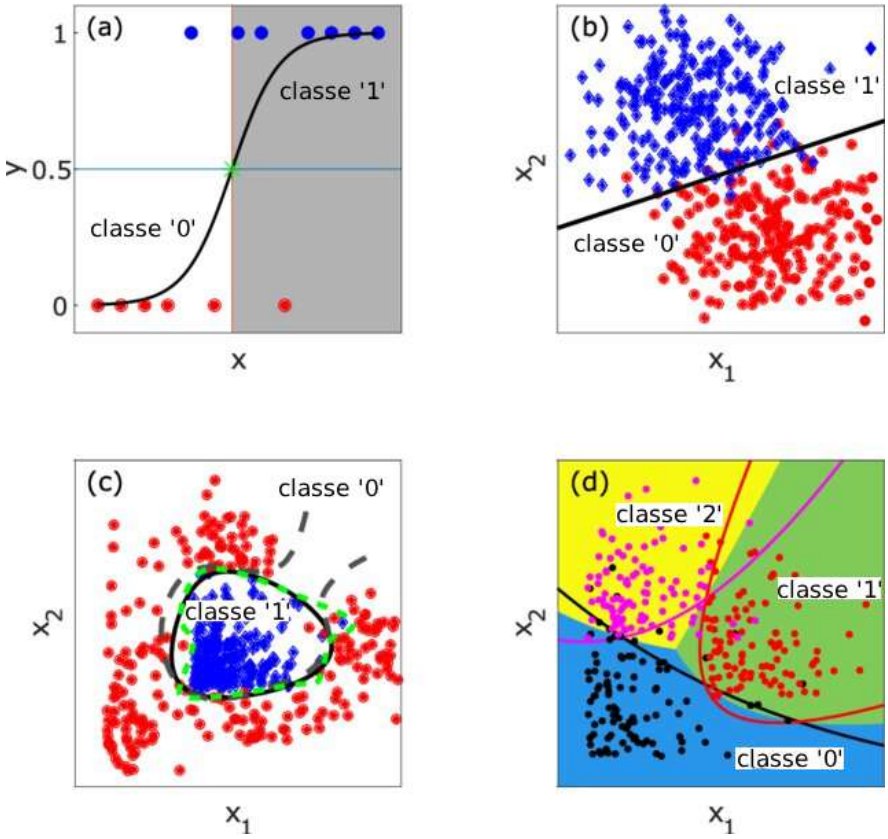


Figura 5 – Exemplos de classificação por regressão logística em diferentes cenários: unidimensional, linear, não linear e multiclasse.

2.4.5 XGBoost

O XGBoost [38] (eXtreme Gradient Boosting) é um sistema escalável de reforço de árvores que se destaca como um método altamente eficaz e amplamente utilizado em aprendizado de máquina, demonstrado na Figura 6 [39]. Ele se baseia nos algoritmos de reforço de gradiente de árvores, técnica que constrói um modelo preditivo em uma maneira aditiva. O modelo final de ensemble de árvores utiliza K funções aditivas para prever o resultado, sendo que a previsão final é a soma das pontuações de cada árvore.

O destaque do XGBoost é a sua função objetivo regularizada. Para um dado conjunto de dados com n exemplos, o modelo de ensemble de árvores busca minimizar a seguinte função objetivo regularizada:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

onde l é uma função de perda convexa e diferenciável que mede a diferença entre a previsão $\hat{y}_i$ e o alvo y_i. O termo $\Omega(f_k)$ é o termo de regularização, que penaliza a complexidade do modelo. Essa regularização adicional, $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$, onde T é o número de folhas da árvore f_k, $\mathbf{w}$ o vetor de pesos das folhas da árvore e λ o hiperparâmetro que controla a penalização, ajuda a suavizar os pesos finais aprendidos para evitar o *overfitting*. Quando o termo de regularização é zerado, o objetivo retorna ao método tradicional de gradient tree boosting.

O modelo é treinado de forma aditiva, o que significa que, em cada iteração t, uma nova função de árvore (f_t) é adicionada para otimizar o objetivo, dada a previsão do passo anterior $\hat{y}_i^{(t-1)}$. Para otimizar

rapidamente o objetivo em um cenário geral, o XGBoost utiliza uma aproximação de segunda ordem da função de perda. Após remover os termos constantes, o objetivo simplificado em cada passo t depende apenas dos estatísticos de gradiente de primeira ordem (g_i) e de segunda ordem (h_i) da função de perda.

Para uma estrutura de árvore $q(\mathbf{x})$ fixa, a equação simplificada permite calcular o peso ótimo w_j^* de cada folha j, bem como uma pontuação de qualidade da estrutura da árvore L_{split}. Essa pontuação atua como uma métrica de impureza similar à usada em árvores de decisão, mas é derivada para uma gama mais ampla de funções objetivo. Um algoritmo guloso é usado para encontrar a melhor divisão na árvore, que maximiza a redução de perda dada por L_{split}.

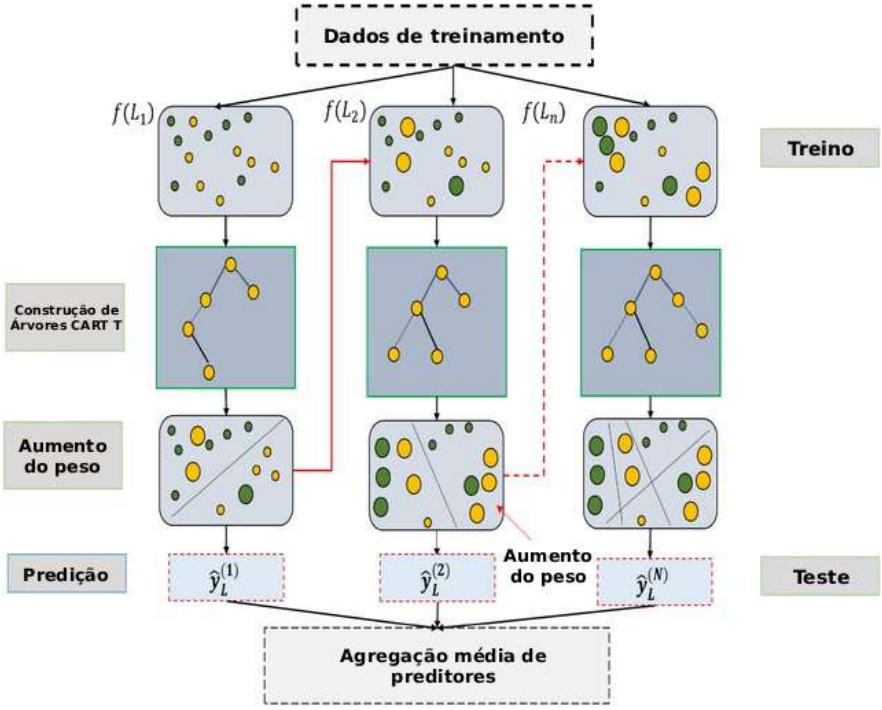


Figura 6 – Esquema gráfico do modelo XGBoost.

2.5 Trabalhos correlatos

Os trabalhos relacionados dispostos a seguir foram utilizados como embasamento para o presente estudo. Cada subseção aborda contribuições aplicadas a conjuntos de dados distintos.

2.5.1 Cross-View Contrastive Masked Autoencoders for Bird Species Classification and Mapping

O Cross-View Contrastive Masked Autoencoders for Bird Species Classification and Mapping [40] (BirdSAT), é um *framework* de Aprendizado Auto-Supervisionado que representa um avanço no campo da FGVC e no mapeamento ecológico. O modelo propõe aprender um espaço de representação unificado que é útil para ambas as tarefas, sendo particularmente relevante por enriquecer o espaço de *embedding* com metadados disponíveis nas imagens de pássaros ao nível do solo. A inclusão de metadados demonstrou ser muito eficaz para lidar com o desafio da classificação de espécies, caracterizado pela baixa variação interclasses e alta variação intraclasses.

O BirdSAT foi pré-treinado em um novo *dataset* global chamado Cross-View iNAT 2021 Birds Dataset, que é intrinsecamente multimodal e *cross-view*. Este *dataset* é composto por pares de imagens de pássaros ao nível do solo, imagens de satélite correspondentes e metadados de aquisição. As imagens de satélite fornecem ao modelo o contexto do ambiente e do *habitat* onde o pássaro pode ser encontrado, ilustrado na Figura 7 [40] exemplos de pares de imagens de satélite e de nível do solo de aves, juntamente com os metadados associados a cada par.

Figura 7 – Exemplo de dados do Cross-View iNAT-2021 Birds.

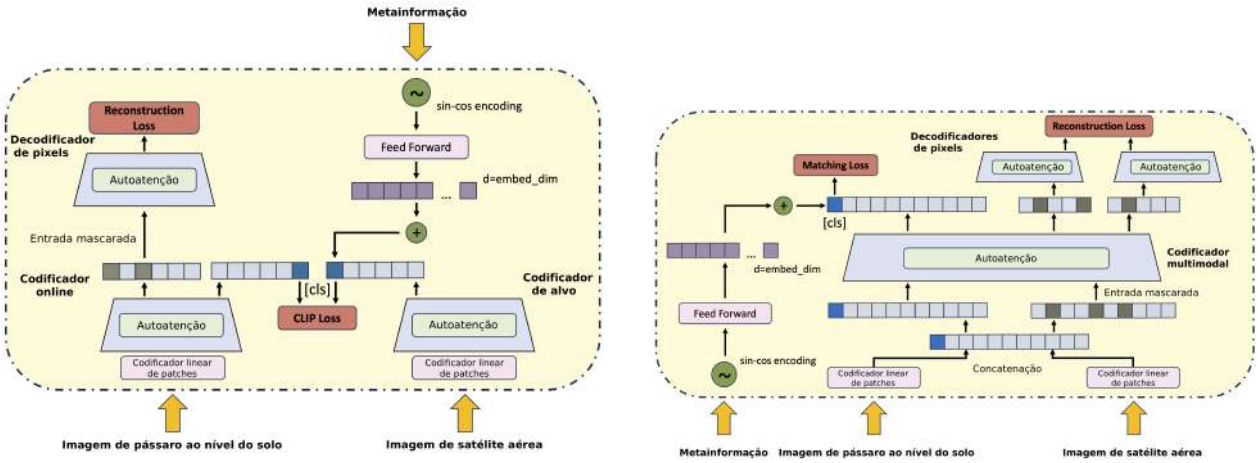
Os metadados de aquisição, cruciais para o BirdSAT, fornecem pistas adicionais que podem melhorar a interpretação e reduzir o número de classes possíveis, sendo utilizados os atributos de localização e tempo.

Os atributos numéricos foram codificados usando o método senoidal-cossenoidal e, em seguida, passados para uma camada *feed-forward* que gera um *embedding* adicionado ao *embedding* do *[cls] token* resultante dos *encoders*, antes da classificação. O uso da geolocalização e dados como detalhes adicionais que o modelo pode entender vem de estudos passados que mostram que adicionar informações sobre localização ajuda a performance na tarefa de FGVC.

O *framework* utiliza a arquitetura ViT para o pré-treinamento *cross-view*. Para alcançar um espaço de *embedding* comum para as tarefas de FGVC e mapeamento de espécies, o BirdSAT unifica as estratégias SSL de *Contrastive Learning* (CL) e *Masked Image Modeling* (MIM).

Duas abordagens arquiteturais foram propostas para a fusão das modalidades e metadados: o Cross-View Embed MAE (CVE-MAE), que é uma configuração uni-modal de fusão tardia que usa *encoders* transformadores separados para cada modalidade, e o Cross-View Metric MAE (CVM-MAE), que é uma configuração *cross-modal* de fusão precoce que emprega um único *encoder* transformador multimodal e *decoders* separados por modalidade.

As arquiteturas estão dispostas na Figura 8 [40] onde foi avaliado: (a) um pré-treinamento unimodal (fusão tardia) e (b) um pré-treinamento multimodal (fusão inicial) do ViT, incorporando metadados e objetivos de reconstrução mascarada e contraste. Os modelos que incorporam metadados alcançaram desempenho estado da arte na classificação fina de pássaros no iNAT-2021 Birds.



(a) Cross-View Embed MAE (b) Cross-View Metric MAE

Figura 8 – Frameworks propostos.

2.5.2 Visual WetlandBirds Dataset

O trabalho Visual WetlandBirds Dataset [6] foca na criação e disponibilização do primeiro *dataset* de vídeo de granularidade fina especificamente para a detecção de comportamento e classificação de espécies de pássaros em vídeos.

Devido à crise mundial de desaparecimento de espécies e ao alto preço do acompanhamento de animais, é muito importante criar sistemas automáticos que possam fornecer informações certas para a proteção dessas espécies. O escopo principal deste estudo é preencher uma lacuna notável na escassez de *datasets* de vídeo de pássaros com anotações detalhadas de comportamento, com exemplos mostrados na Figura 9 [6].

Figura 9 – Recortes de quadros de vídeo de espécies de pássaros realizando os sete comportamentos que compõem o conjunto de dados.

O *dataset* é composto por 178 vídeos gravados em pântanos espanhóis na região de Alicante, capturando 13 espécies diferentes e 7 classes de comportamento distintas. O que torna o Visual WetlandBirds especial é que ele dá informações sobre o tempo e o lugar em que os pássaros aparecem, no nível do quadro. Ele mostra qual é a espécie, onde o pássaro está e o que ele está fazendo a cada instante, indo além de apenas confirmar que a espécie está ali. Para a tarefa de classificação de espécies, o trabalho utilizou o YOLOv9 como *baseline*, e para a detecção de comportamento foram avaliados modelos baseados em Transformer e redes convolucionais.

Apesar de ser o primeiro a fornecer anotações de comportamento, espécie e localização no nível do quadro para aves em vídeo, o Visual WetlandBirds enfrenta limitações significativas, principalmente relacionadas à sua escala e metodologia de anotação, um desafio primário é a quantidade limitada de dados disponível para o treinamento de modelos complexos de aprendizado profundo.

O *dataset* totaliza apenas 178 vídeos, com uma duração total de aproximadamente 58 minutos e 53 segundos. Essa restrição de volume de dados é um fator que demonstra a necessidade de mais recursos para a captura de informações adicionais.

Outra limitação importante é o desequilíbrio de classes de comportamento. O total de vídeos feitos sobre comportamentos não é igual para todos, ações como voar e se limpar têm a menor quantidade de vídeos, porque acontecem menos vezes ou são complicadas de registrar com câmeras que ficam paradas.

O processo de anotação semi-automática impôs critérios que simplificam a complexidade do comportamento animal. Primeiramente, quando um pássaro realiza múltiplas atividades simultaneamente, o protocolo de anotação estipula que apenas um único comportamento pode ser atribuído por quadro. Nesses casos, o comportamento considerado ecologicamente mais relevante, como alimentar-se, é priorizado sobre comportamentos locomotores concomitantes, como andar ou nadar.

Para que um conjunto de movimentos seja classificado como um comportamento distinto, ele deve ter uma duração mínima de 30 quadros. Movimentos mais curtos são rotulados como sub-movimentos do comportamento principal, o que facilita a segmentação e classificação pelos modelos, mas simplifica o comportamento real.

2.5.3 A Multi-Path Feature Fusion and Spectral–Temporal Attention-Based Model for Bird Audio Classification

Lu et al. [41] propõem a Dual-path spectro–temporal Attention & Fusion Network (DuSAFNet) para capturar simultaneamente texturas espectrais locais e dependências temporais de longo alcance nos espectrogramas de entrada log Mel.

O modelo começa com um backbone compartilhado e é seguido pelo Módulo de Atenção Espectro-Temporal (STA). O STA recalibra adaptativamente a importância de cada banda de frequência e segmento de tempo, modelando pesos de Atenção separadamente nos eixos de frequência e tempo. Essa separação permite que a rede se concentre nas bandas e períodos mais discriminativos, superando as dificuldades das CNNs tradicionais em capturar informações de frequência absoluta e relações temporais de longo alcance.

O cerne da extração de recursos é o Módulo de Extração de Recursos de Caminho Duplo (DPFM), que opera em paralelo. A GrowthBranch utiliza unidades de crescimento densamente conectadas para capturar texturas locais de grão fino, que são sensíveis a variações de frequência de curto prazo.

Em contraste, a SkipBranch emprega uma estrutura de salto residual para refinar o contexto de longo alcance, fortalecendo a capacidade do modelo de capturar padrões temporais e cruzados de frequência, a fusão adaptativa dos recursos destas duas ramificações é realizada pelo Mapeamento de Fusão Controlada (GFM), um mecanismo de *gating* leve que ajusta dinamicamente a proporção do fluxo de informações, suprimindo recursos redundantes e realçando informações críticas para aumentar a eficiência da fusão.

Após a fusão inicial, é utilizado o Módulo de Fusão Temporal e Espacial, que tem duas partes: a LocalSpanAttention, que analisa as relações de tempo em uma área, e o MultiscaleAttentionModule, que ajusta as informações em diferentes escalas espaciais e de canais. O objetivo é melhorar a maneira como os dados são representados, tanto em relação às ligações temporais locais quanto ao ajuste em diferentes escalas no espaço e nos canais.

Para aumentar a diferença entre espécies que são um pouco diferentes, o DuSAFNet traz um Classificador ArcMarginProduct Multi-banda. O ArcMarginProduct é aplicado a cada banda com fatores de escala (s) e margens angulares (m) distintos para aumentar explicitamente a distância angular entre as classes.

A fusão final dos logits de cada banda é feita usando pesos aprendíveis, o que permite ao modelo equilibrar automaticamente a importância de cada faixa de frequência durante o treinamento, toda arquitetura é observada na Figura 10 [41].

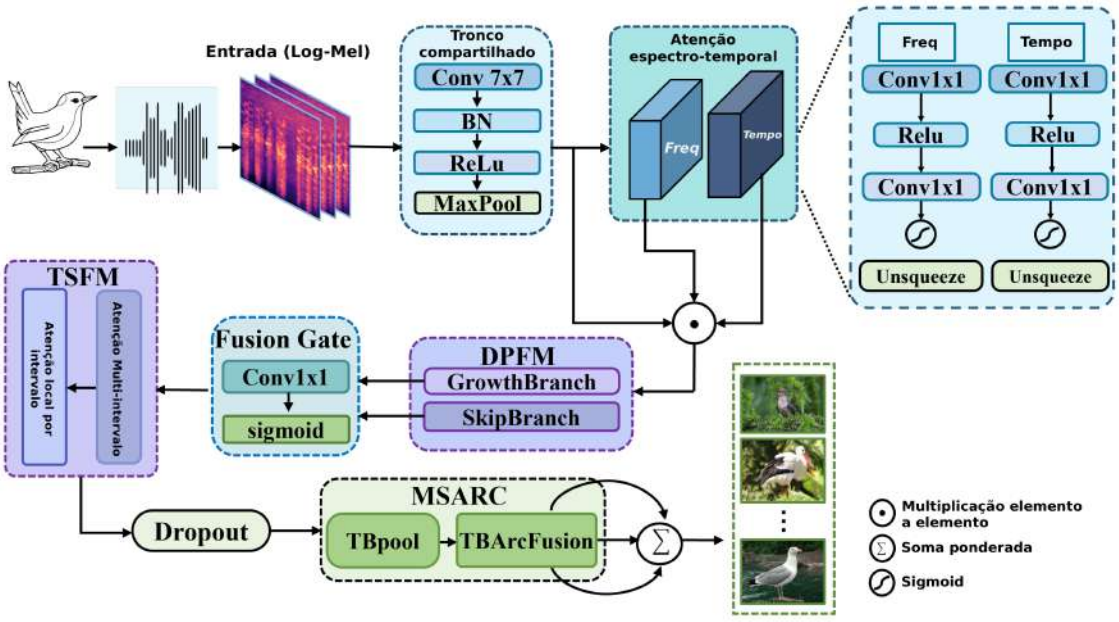


Figura 10 – Arquitetura geral do DuSAFNet.

3 MÉTODO DE PESQUISA

Os métodos adotados foram definidos com o objetivo de alcançar uma classificação multimodal das espécies selecionadas por estado, buscando avaliar o desempenho dos modelos a partir de métricas consolidadas na literatura. A abordagem apresenta caráter quantitativo e experimental, envolvendo a coleta sistemática dos dados, seu pré-processamento e a avaliação do desempenho das diferentes arquiteturas propostas. A Figura 11 ilustra de forma geral os processos aplicados.

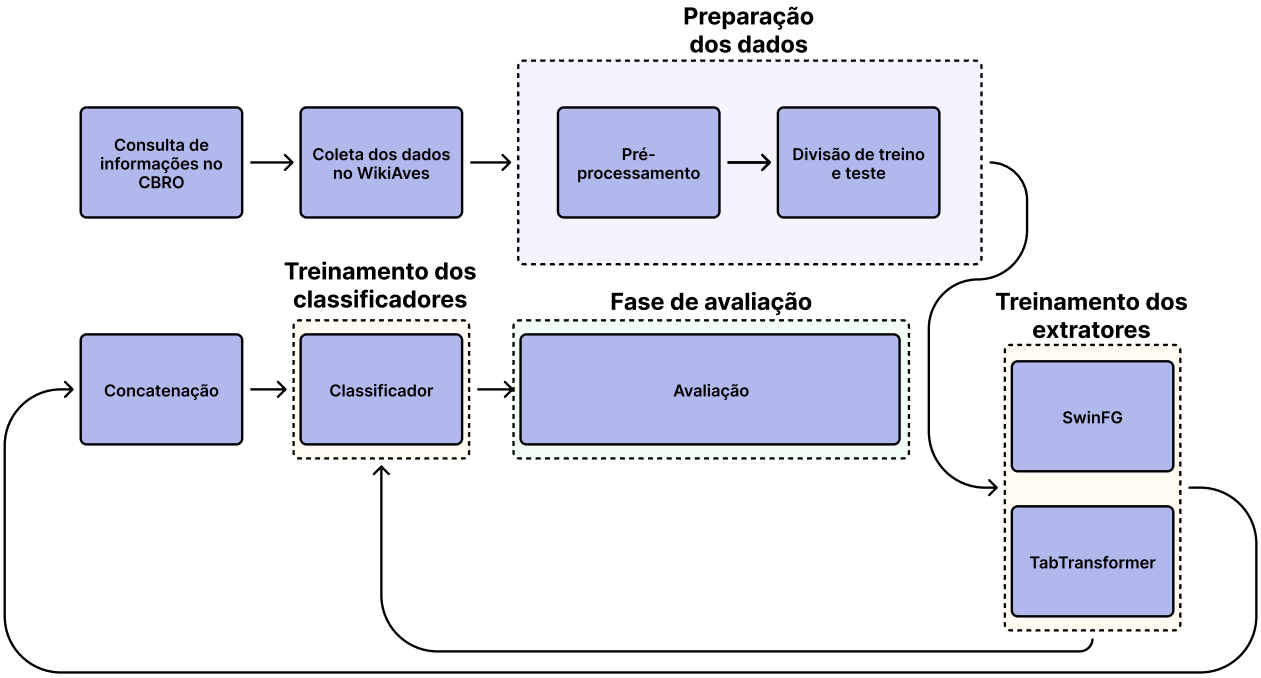


Figura 11 – Fluxo dos processos aplicados.

3.1 Conjunto de dados

A fim de obter a lista de aves do Brasil, foi consultado o website do Comitê Brasileiro de Registros Ornitológicos, este que, em sua 13ª edição, publicada em 2021, havia 1950 nomes populares válidos, com correspondência no WikiAves, onde era possível consultar as fotos e seus metadados correspondentes. A escolha dos nomes populares foi devida à mudança de nome de alguns táxons, estes que eram descritos de outra maneira no WikiAves.

A mediana de imagens correspondente para estas espécies válidas foi de 696 e a média, 2700. A mediana foi adotada como número máximo de imagens que deveriam ser obtidas para cada espécie, assim mitigando desbalanceamento entre a quantidade de imagens de cada classe.

Como algumas espécies tinham uma quantidade muito pequena de registros, a análise considerou espécies que tinham mais de 100 fotos, assim não obtendo aquelas que poderiam ser interpretadas como ruído pelo modelo. Dessa forma, obtivemos espécies com mais de 100 e até 696 registros. A quantidade final de espécies a serem obtidas foi de 1590.

Além disso, pela presença de um número muito grande de espécies por estado, a quantidade de espécies a ser utilizada foi limitada às cinco com maior ocorrência. Essa medida foi necessária para evitar grande desbalanceamento em um problema de classificação multiclasse.

A partir da quantidade final de espécies definidas no escopo, houve a coleta das imagens correspondentes a cada uma delas, juntamente com os seguintes metadados: id, autor, data, localização, endereço eletrônico original e o id da espécie. Estes dados serviram, respectivamente, para o treinamento do SwinFG e do TabTransformer.

3.2 Pré-processamento

Utilizando Python, todos os dados foram separados em suas respectivas pastas, contendo as imagens de cada indivíduo e um arquivo *JavaScript Object Notation* (JSON), formato de arquivo para armazenamento de dados estruturados. Também foi gerado um novo arquivo JSON que contém todos os metadados estruturados, a fim de facilitar o acesso.

Em busca de arquivos corrompidos, a etapa identificou quaisquer arquivos cujo tamanho era de 0 *kilobytes*, ou seja, que estavam corrompidos.

3.3 Divisão dos dados

Todos os dados, sejam eles imagens ou dados tabulares, foram divididos na proporção de 70% para treino e 30% para teste nos modelos utilizados para a extração. Todos os dados foram estratificados, a fim de mitigar o desbalanceamento nas amostras.

Já nos modelos utilizados para classificação a divisão foi de 60% para treino, 20% para validação e 20% para teste.

3.4 Modelagem

A abordagem aplicada foi *downstream*, a aplicação *dual-branch* resultou em uma abordagem onde duas arquiteturas foram responsáveis pelo pré-treinamento extraindo as características e cinco efetuando o treinamento, teste e validação final da classificação. No *branch* das imagens o SwinFG foi aplicado devido a sua natureza voltada para tarefas de FGVC que possibilita a extração do vetor de características do modelo de imagem, para os dados tabulares (metadados), o TabTransformer foi empregado, obtendo por vez os *embeddings* relacionados às informações geo-temporais e outros atributos.

Nos modelos empregados para o pré-treinamento, os hiperparâmetros de suas respectivas implementações originais foram mantidos, a fim de contornar limitações de tempo e recursos computacionais.

Após o processo inicial de pré-treino e classificação das modalidades de forma individual, houve a concatenação dos vetores responsáveis pelas características tabulares e as de imagem. Diferentes modelos foram avaliados como Cabeças de classificação, incluindo k-NN, Random Forest, Support Vector Machine, Logistic Regression e XGBoost, esses modelos receberam os *embeddings* das modalidades individuais e por fim, os *embeddings* concatenados.

3.5 Validação e avaliação

Na fase de treino e teste dos modelos extratores, foram utilizados 5 *folds* estratificados, de maneira a obter os benefícios da utilização do *k-fold* e, ao mesmo tempo, não aumentar o tempo de treinamento.

Os mesmos dados utilizados no treinamento e teste dos dados tabulares também foram utilizados no treinamento do modelo responsável pelas imagens. Isto é, cada um, em sua forma, seja tabular ou em formato de imagem, foi separado por um identificador correspondente, tornando possível a concatenação do vetor de características ao fim do processo, a métrica escolhida para avaliação foi a acurácia, por medir a proporção de classificações corretas sobre o total de exemplos, tendo uma avaliação direta e interpretável, representada como:

$$\text{Acurácia} = \frac{N_{\text{corretas}}}{N_{\text{total}}}.$$

onde N_{corretas} é o número de previsões corretas do modelo e N_{total} é o número total de exemplos avaliados.

4 EXPERIMENTOS

Esta seção apresenta os experimentos realizados para avaliar o desempenho da abordagem proposta, as seções seguem a ordem de realização das etapas do experimento buscando evidenciar diferenças entre os métodos aplicados ao problema.

4.1 Cenários de avaliação

Os experimentos foram realizados em três diferentes cenários, sendo o primeiro deles baseado na utilização de metadados. Nesse conjunto estão englobadas informações como localização, data da observação, atributos taxonômicos (reino, filo, classe, ordem e família), atributos temporais derivados (ano, mês e dia), além de identificadores de espécie e nome popular para classificação. Um exemplo de conjunto de atributos utilizados na classificação pode ser visto na Figura 12[1].

Foi empregado o TabTransformer para extrair o vetor de características dos metadados, cuja saída foi posteriormente processada por diferentes classificadores de arquiteturas distintas, de modo a evidenciar possíveis vantagens ou desvantagens de cada um nesta aplicação.

No segundo cenário o SwinFG serviu como extrator do vetor de características das respectivas imagens associadas aos metadados, a sua saída passou de forma individual pelos mesmos classificadores listados no texto.

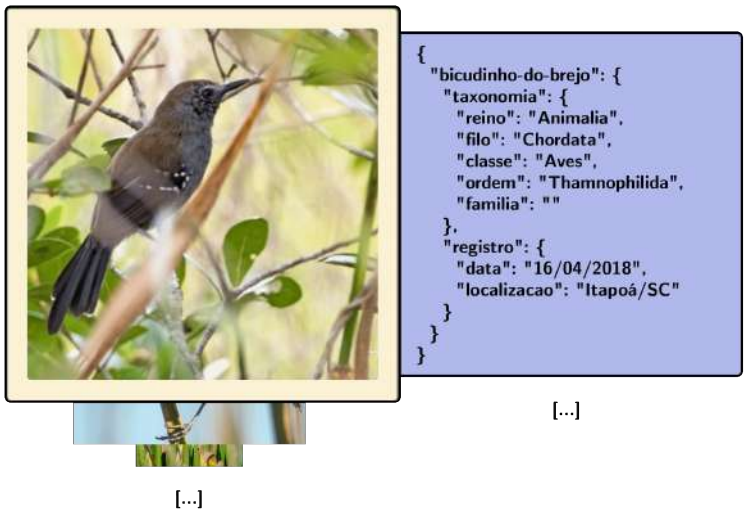


Figura 12 – Exemplo de amostra.

Já no terceiro cenário o vetor de características de ambos os modelos foram concatenados e passaram pelos exatos mesmos classificadores, isto é, buscando possíveis diferenças presentes nos resultados ao utilizar os dois vetores juntos, podendo extrair ou não informações relevantes para a classificação. A Figura 12 ilustra exemplos de entrada onde a fotografia da espécie serve de entrada para o SwinFG e os dados tabulares para o TabTransformer.

[1] https://www.wikiaves.com.br/2965931

O treinamento de ambos seguiu a mesma configuração, com *batch size* reduzido, *gradient accumulation*, otimizador *Adam*, *CrossEntropyLoss*, *early stopping* e *5-fold cross-validation*. A Tabela 1 demonstra as modificações realizadas nos hiperparâmetros onde as motivações foram respectivamente: manter simplicidade, evitar *overfitting*, controle temporal, melhor convergência e por fim melhor performance e maior robustez ao *overfitting*; os demais valores não listados foram mantidos em seus valores padrão.

Classificador	**Principais Alterações**	**Configurado**	**Default**
***k*-NN**	Nenhuma	n_neighbors=5	n_neighbors=5
RandomForest	max_depth=10	Profundidade limitada	Sem limite
SVM	max_iter=1000	Iterações limitadas	Ilimitado
LogisticRegression	max_iter=1000	Mais iterações	max_iter=100
XGBoost	n_estimators=50, max_depth=4, learning_rate=0.1	Conservador	Agressivo

Tabela 1 – Configurações dos classificadores testados.

4.2 Protocolo de execução

A partir das métricas vistas em na Subseção 3.5 os valores de acurácia representam a porção de teste, os dados foram organizados a partir do *k*-fold estratificado de 5 partições, reduzindo o risco de *overfitting*.

Todo processo sequencial foi realizado com *seed* calculada a partir do *hash* da sigla de seu estado correspondente e normalizada para 32 *bits*, garantindo que reprodução determinística das modalidades durante o tempo de execução, a partir da aplicação dos folds nos modelos utilizados no pré-treinamento ambos receberam as mesmas amostras na mesma ordem, tornando possível a concatenação final dos vetores de características.

Os dados utilizados para o treinamento dos modelos de classificação foram aplicados com uma divisão de dados de 60% para treino, 20% para validação e 20% para teste, a partir da quantidade maior de repartições foi possível uma avaliação mais consistente da generalização dos modelos.

5 RESULTADOS

A Figura 13 apresenta a distribuição de registros por estado considerando a soma de todas as espécies, onde a coluna “Total” indica o número geral de amostras obtidas, a coluna “Com Imagem” representa aquelas para as quais foram possíveis obter e processar ao menos uma imagem, e a coluna “Filtrados” corresponde às amostras que possuem um nome popular válido e uma imagem processável. Para este trabalho, foram utilizados os dados categorizados como “Filtrados”.

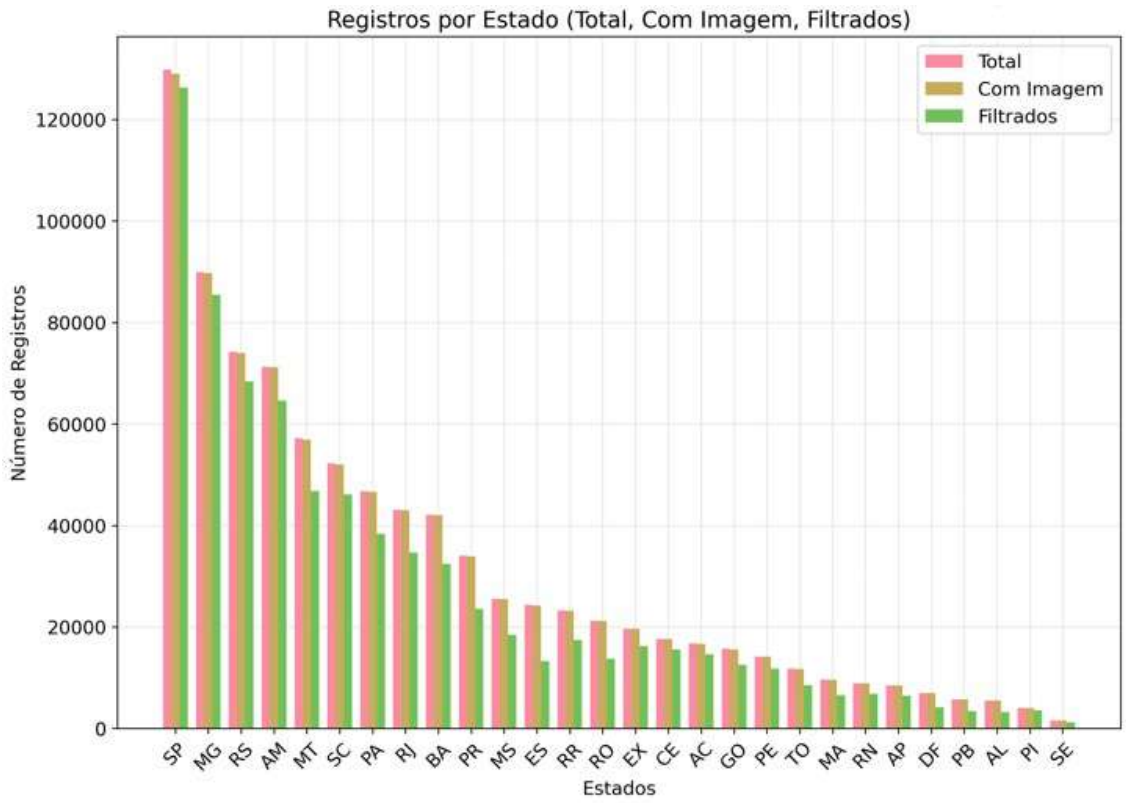


Figura 13 – Registros por estado.

A Figura 15 contém a distribuição de espécies válidas por estado. São Paulo, Minas Gerais e Mato Grosso concentram a maior quantidade de registros, enquanto estados como Alagoas, Piauí e Sergipe possuem valores menores, essa variação mostra tanto o trabalho feito na coleta de amostras quanto a presença de pessoas que observam em certas áreas, afetando diretamente o equilíbrio dos dados. Dessa forma, foram aplicados filtros, selecionando apenas as espécies mais representativas em cada estado, conforme descrito na Seção 3, o comportamento das distribuições pode ser visto na Figura 14.

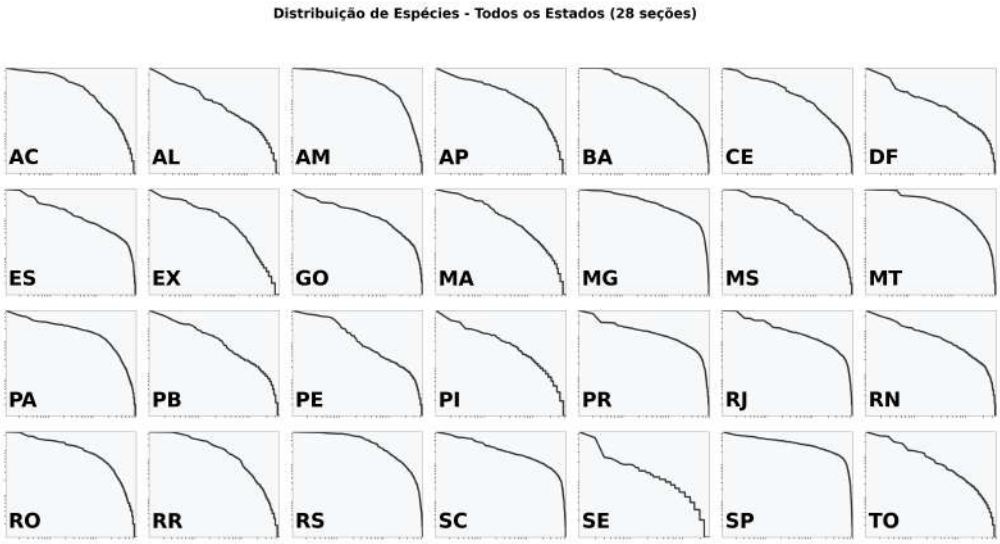


Figura 14 – Distribuição das espécies por estado, de maior a menor ocorrência.

Com base nos dados apresentados nas Figuras 11 e 15, a análise selecionou as cinco espécies com maior ocorrência em cada estado, devido à elevada diversidade de classes no conjunto de dados. Essa escolha visa mitigar problemas decorrentes do grande número de amostras, resultando em um conjunto de dados mais balanceado para os experimentos realizados.

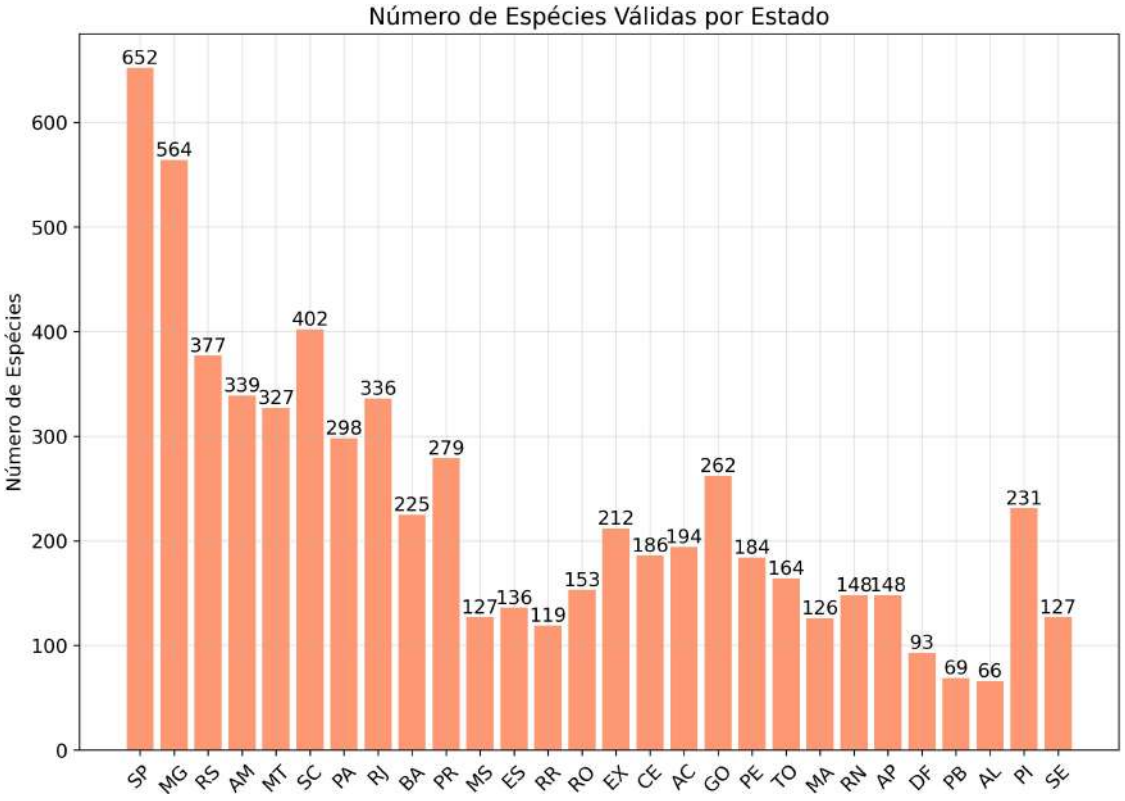


Figura 15 – Número de espécies válidas por estado.

A Tabela 9, disponibilizada nos Apêndices devido à sua extensão, demonstra as espécies a serem

classificadas e suas respectivas quantidades para cada estado.

Na Tabela 2, os classificadores que utilizaram apenas a imagem, obtiveram resultados muito semelhantes nos diversos estados da região, indicando que os atributos visuais extraídos não foram suficientes para uma classificação de maior acurácia, em cenários onde a concatenação foi realizada observamos uma piora nos resultados, exceto no estado de Santa Catarina. Os resultados tabulares apresentam um comportamento consistente.

O desempenho relativamente baixo do modelo que utilizou as características extraídas da imagem sugere que as informações obtidas não carregam atributos discriminantes que melhorem a classificação.

Tabela 2 – Acurácia de Teste por Estado - Região Sul

Estado	**Tabular**	**Imagem**	**Concatenado**
PR	0.9118	0.3162	0.9044
RS	0.7223	0.3308	0.7132
SC	1.0000	0.3728	1.0000
Média	**0.8780**	**0.3399**	**0.8725**

A região demonstrada na Tabela 3 apresenta que diferentemente da região vista na Tabela 2 houve um modelo que performou uma acurácia inferior a 0,70. Outro aspecto importante é que, nesta região a concatenação obteve o mesmo resultado ou apresentou ganhos.

Além da média dos resultados serem superiores à da tabela da região Sul, as informações extraídas do modelo de imagem do Sudeste foram discriminantes o suficiente para obter melhoras em termos de acurácia ao realizar a concatenação, isto é, as informações obtidas tiveram características discriminativas que não estavam presentes apenas no *embedding* extraído do modelo tabular, essa característica pode estar associada ao número diferente de amostras em cada região.

Tabela 3 – Acurácia de Teste por Estado - Região Sudeste

Estado	**Tabular**	**Imagem**	**Concatenado**
ES	0.8462	0.4487	0.8897
MG	1.0000	0.3260	1.0000
RJ	0.6346	0.3107	0.6761
SP	0.9121	0.3919	0.9121
Média	**0.8482**	**0.3693**	**0.8695**

No Norte representado pela Tabela 4, há uma variação maior entre os estados, indicando possível diferença na distribuição ou na qualidade dos dados. O modelo concatenado apresentou ganhos sutis em alguns estados, o modelo de imagem nesta região de forma isolada continua demonstrando dificuldade para extrair informações relevantes perante a complexidade visual das amostras.

Tabela 4 – Acurácia de Teste por Estado - Região Norte

Estado	Tabular	Imagem	Concatenado
AC	0.6465	0.2323	0.6364
AM	0.7140	0.3706	0.6937
AP	0.6692	0.3154	0.6923
PA	0.8300	0.3374	0.8645
RO	0.8365	0.2644	0.8173
RR	0.8080	0.2834	0.8173
TO	1.0000	0.3397	1.0000
Média	**0.7863**	**0.3062**	**0.7888**

O Nordeste, presente na Tabela 5 por sua vez manteve o mesmo padrão observado em outras regiões, no estado do Piauí que figurava dentre os de maior acurácia, ao realizar a concatenação apresentou um resultado de menor acurácia quando comparado ao modelo que utilizou apenas os dados tabulares. Em três estados a acurácia atingiu o valor máximo, sugerindo que os atributos tabulares capturaram os padrões necessários para que a classificação performasse de forma ótima no cenário apresentado.

Tabela 5 – Acurácia de Teste por Estado - Região Nordeste

Estado	Tabular	Imagem	Concatenado
AL	0.8605	0.3430	0.8663
BA	1.0000	0.3009	1.0000
CE	0.8893	0.3811	0.8955
MA	1.0000	0.3568	1.0000
PB	0.8050	0.3648	0.8050
PE	0.8333	0.3485	0.8428
PI	0.9114	0.3165	0.8861
RN	1.0000	0.4309	1.0000
SE	0.8421	0.4474	0.8421
Média	**0.9046**	**0.3655**	**0.9042**

No Centro-Oeste retratado na Tabela 6 novamente há tendência no melhor desempenho dos modelos tabulares sobre os modelos de imagem, reforçando a dificuldade de aprendizado visual isolado diante dos padrões apresentados.

Tabela 6 – Acurácia de Teste por Estado - Região Centro-Oeste

Estado	Tabular	Imagem	Concatenado
DF	0.9727	0.4044	0.9563
GO	0.8950	0.3591	0.9061
MS	0.8830	0.3333	0.8883
MT	0.8603	0.2615	0.8443
Média	**0.9028**	**0.3396**	**0.8988**

Por outro lado, o exterior representado na Tabela 7 vemos que, a região foi a que mais apresentou ganho na concatenação, isso demonstra que os dois *embeddings* de forma individual possuem características únicas e que, quando concatenados, conseguem discriminar melhor as características e obter um melhor resultado.

Tabela 7 – Acurácia de Teste por Estado - Região Exterior

Estado	Tabular	Imagem	Concatenado
EX	0.6575	0.3950	0.7459
Média	**0.6575**	**0.3950**	**0.7459**

Por fim, a Tabela 8 detalha os resultados gerais considerando todas regiões abordadas, a mesma ilustra a diferença entre a acurácia das diferentes modalidades em diferentes métricas.

Tabela 8 – Estatísticas Descritivas por Modalidade - Todas as Regiões

Modalidade	Média	Mediana	Desvio Padrão	Mínimo	Máximo
Tabular	0.8550	0.8604	0.1159	0.6346	1.0000
Imagem	0.3458	0.3414	0.0533	0.2323	0.4487
Concatenado	0.8606	0.8762	0.1087	0.6364	1.0000

As Figuras 16 e 17 demonstram, de forma visual, a diferença entre os melhores resultados obtidos em cada modalidade em todas áreas e a diferença entre a acurácia ao realizar as concatenações. Na Figura 16 indica que, são poucos os modelos classificadores que receberam apenas a imagem e que conseguem ultrapassar o limiar de 0,4 no gráfico.

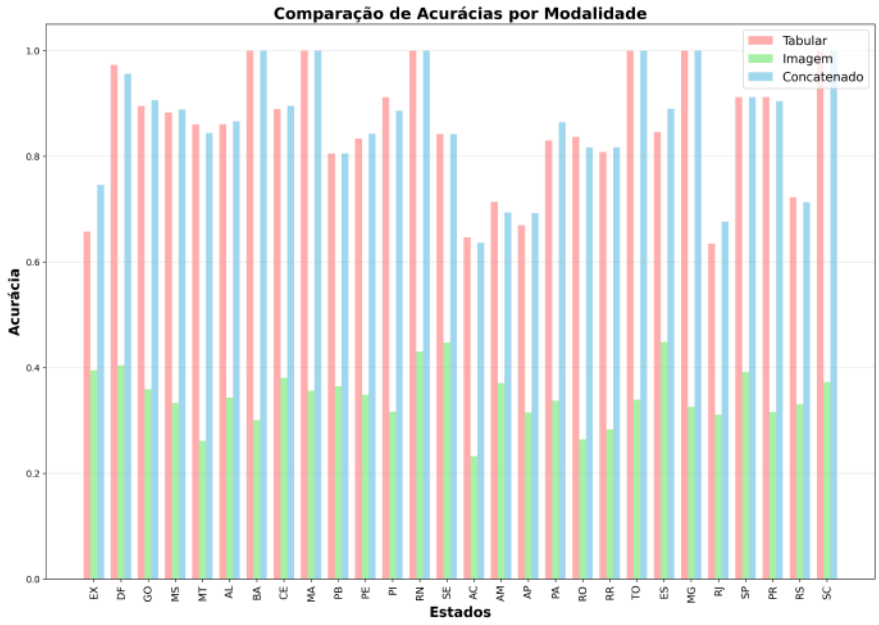


Figura 16 – Melhores acurácias obtidas nas regiões analisadas.

Na Figura 17 os valores indicam que a maioria dos modelos que receberam a concatenação obtiveram resultados iguais ou melhores, ocorrendo em ambas modalidades.

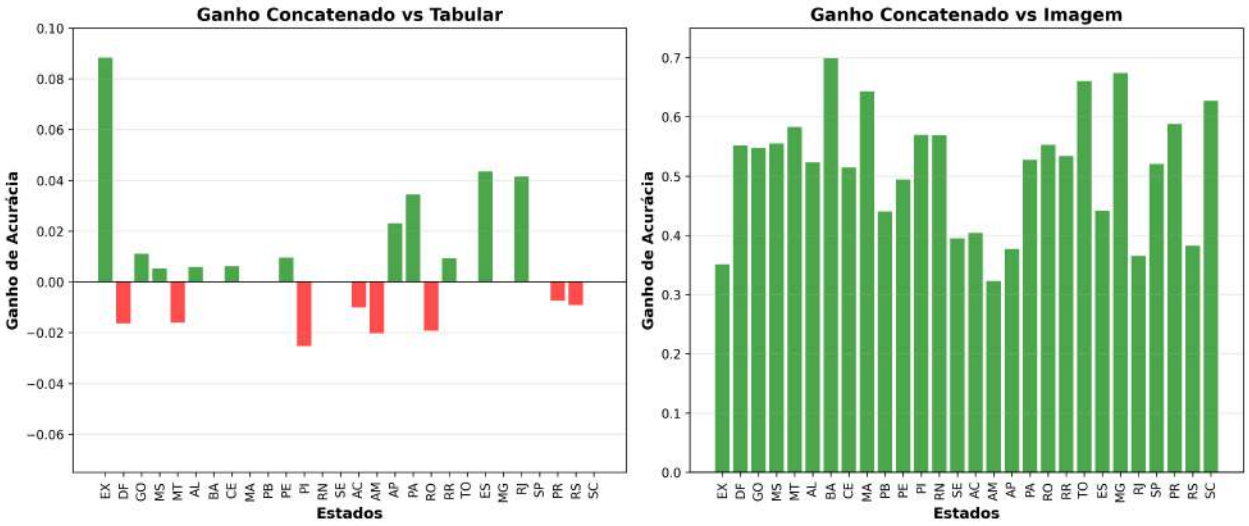


Figura 17 – Comparativo entre os ganhos obtidos através da concatenação.

O resultado das acurácias detalhadas por modelo podem ser visualizado na Tabela 10 presente nos Apêndices, por conter grande quantidade de informações a mesma foi movida para manter a continuidade do texto.

A Figura 18 ilustra a relação entre volume amostral e o desempenho da classificação por modalidades, temos independência notável entre essas variáveis, correlações desprezíveis para dados tabulares e concatenados e correlação fraca para imagens.

A modalidade de imagem mesmo tendo acurácias consistentemente inferiores ainda continua a apresentar um comportamento que não depende da quantidade de dados. Isso mostra que há limites naturais nas imagens e não no número de exemplos. O Acre, sendo um caso diferente na forma combinada, destaca que certas características da região têm mais impacto no resultado do que o número de amostras.

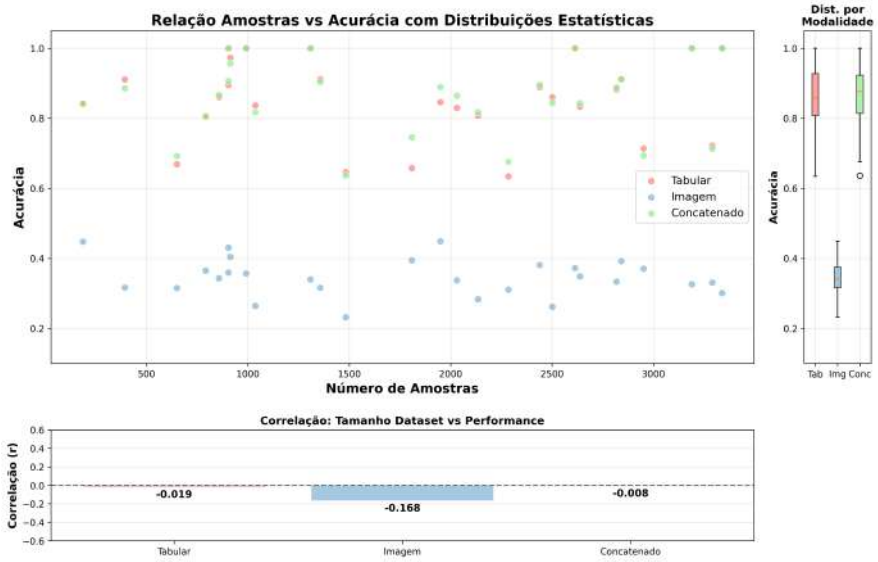


Figura 18 – Análise híbrida da relação entre tamanho de dataset e desempenho de classificação por modalidade.

A Figura 19 apresenta dois comparativos que relacionam métricas de distribuição de espécies por estado com a acurácia do classificador por imagem. No painel esquerdo, o índice de Shannon de diversidade/igualdade é realizado contra a acurácia, valores baixos indicam ambientes onde a distribuição de abundâncias é muito desigual, já valores altos representam abundâncias mais uniformes, a relação negativa mostra que locais com mais variedade e distribuição mais uniforme costumam ter resultados menores no modelo de imagens.

No painel direito, o índice de dominância, isto é, quando poucas espécies predominam mostra correlação positiva com a acurácia, sinalizando que algumas relações têm resultados melhores quando há poucas espécies envolvidas.

As relações sugerem que a composição da comunidade influencia fortemente o desempenho do classificador. Quando há uma ou poucas espécies que são mais comuns, o modelo consegue identificar padrões de maneira mais clara. Por outro lado, quando a comunidade tem mais variedade e está mais equilibrada, fica mais complicado realizar essa tarefa.

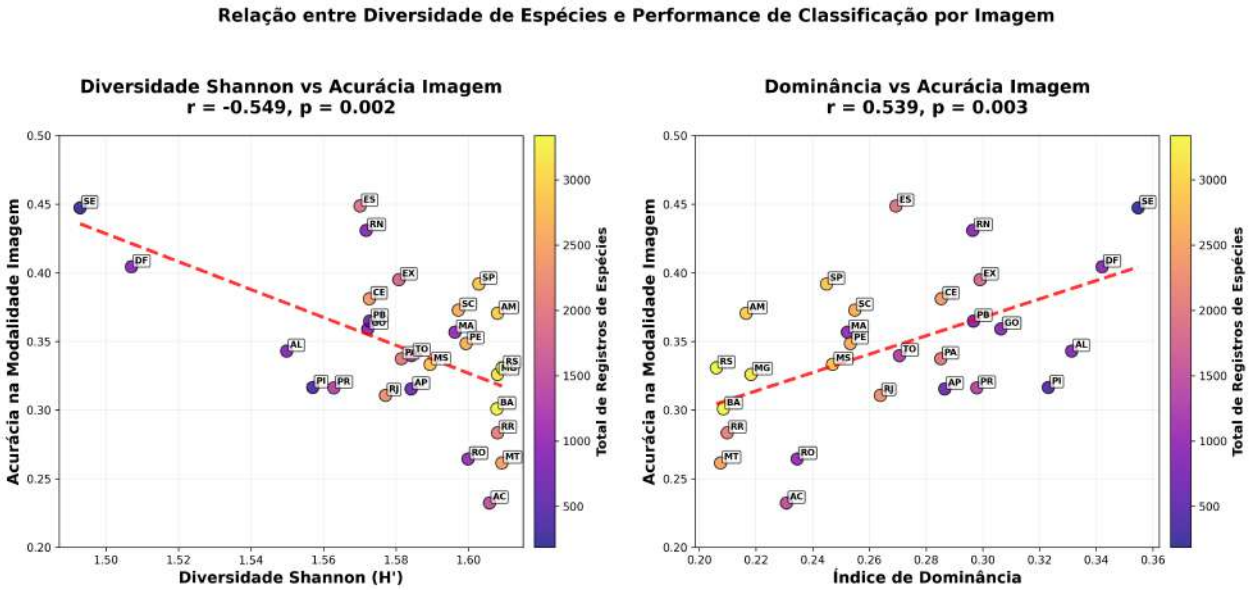


Figura 19 – Relação entre amostras e acurácia dos classificadores por modalidade.

A distribuição de gaps entre algoritmos por modalidade está disposta na Figura 20, apresentando diferenças na robustez à seleção dos algoritmos entre os três domínios do problema. A modalidade tabular fica próxima a zero, a escolha do algoritmo de classificação exerce impacto mínimo no desempenho final, 83,6% dos gaps estão abaixo de 5%, sugerindo que os algoritmos apresentam desempenhos similares quando aplicados aos dados tabulares.

O domínio de imagem demonstra uma distribuição mais dispersa e com formato indicativo de alta variabilidade. Em 35% dos casos os gaps foram superiores a 15%, alcançando valores de até 37,3%, a escolha do modelo é um fator determinante para o resultado.

Na modalidade onde os dados foram concatenados tem um comportamento intermediário, a distribuição é concentrada similar ao visto nos dados tabulares mas com a presença de alguns outliers. Tendo 82,2% gaps abaixo de 5% a abordagem é relativamente robusta à escolha do algoritmo, obtendo benefício da

concatenação dos dados. A integração indica que a concatenação dos dados mitiga a sensibilidade à escolha do algoritmo, oferecendo maior estabilidade preditiva.

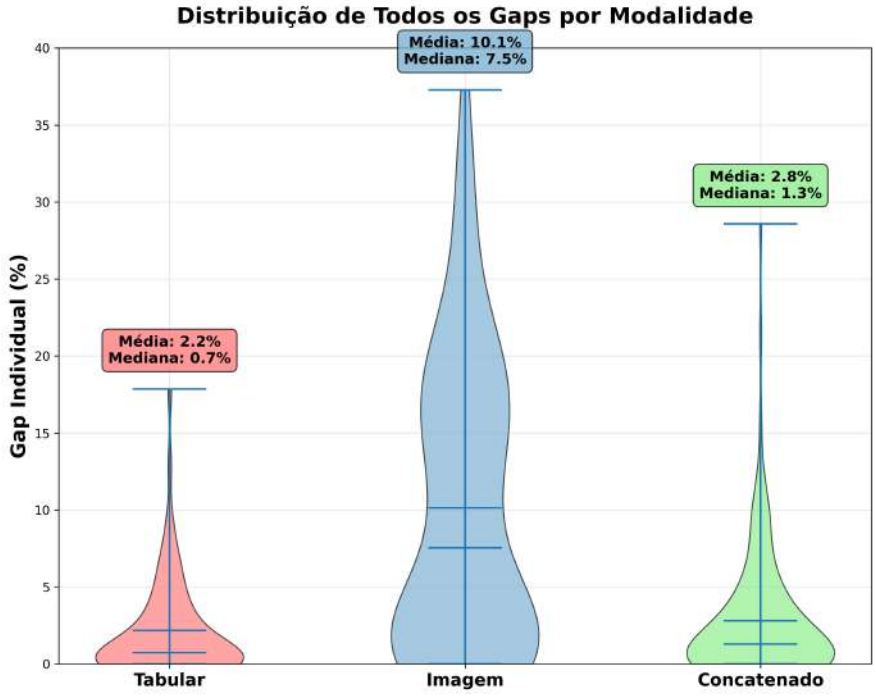


Figura 20 – Distribuição de gaps de desempenho entre algoritmos por modalidade de dados.

A distribuição dos melhores classificadores por modalidade demonstra que para cada domínio do problema apresentado uma modalidade de classificador se mostra mais especializada, conforme demonstrado na Figura 21.

O k-Nearest Neighbors em 32,1% dos estados acaba por performar melhor em dados tabulares, isto é, a eficácia da similaridade local tende a funcionar melhor, o XGBoost apresenta melhor desempenho em 35,7% dos dados de imagem, indicando que métodos mais robustos tendem a funcionar melhor ao tentar capturar complexidades visuais presentes nos *embeddings*.

Na modalidade concatenada existe uma distribuição equilibrada entre LogisticRegression, Random Forest e XGBoost, a fusão multimodal criou um espaço de características híbrido explorável por diferentes paradigmas, algo que não ocorria antes, cada tipo de representação possui características distintivas que favorecem estratégias classificatórias específicas.

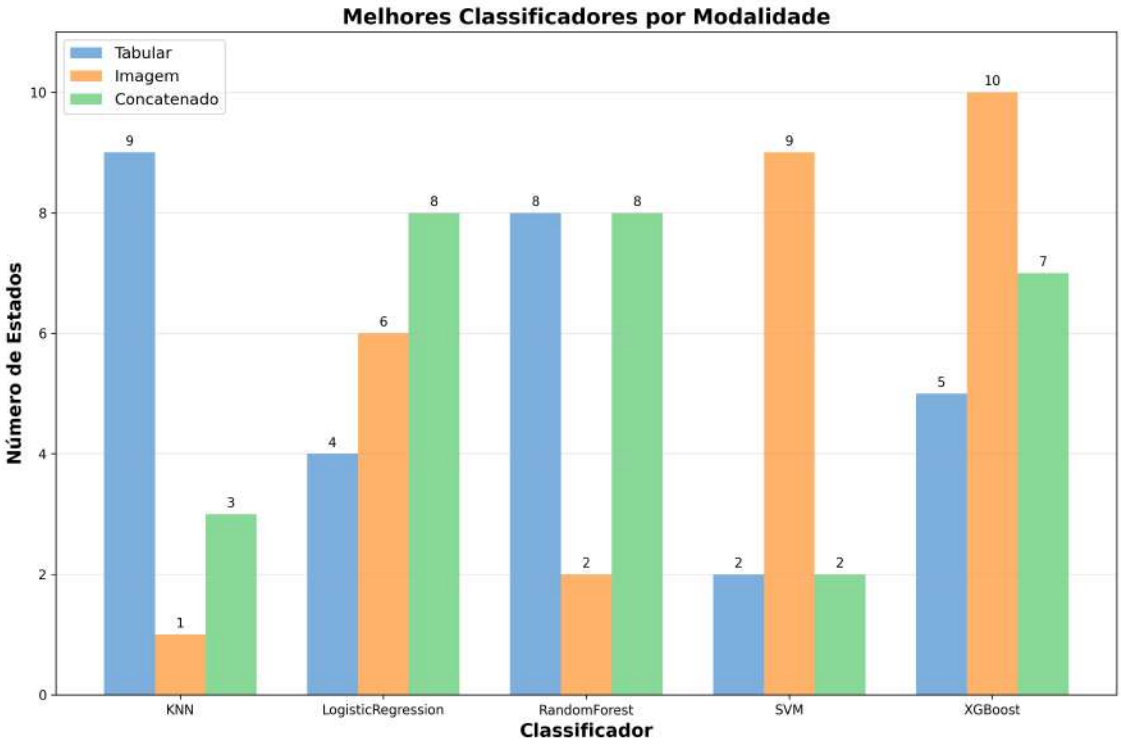


Figura 21 – Melhor classificador por modalidade.

6 CONCLUSÃO

A utilização de dados de ciência aberta embora tenha possibilitado o estudo apresentou disparidade na distribuição dos dados por estado, tal fenômeno pode ser ligado à ocorrência desses próprios avistamentos por entusiastas, isto é, por mais que a ciência cidadã forneça mais amostras os experimentos ligados a ela ainda são dependentes da distribuição geográfica dos contribuintes. A distribuição das espécies indica que a maior quantidade de indivíduos está concentrada no primeiro e último quartil, especialmente no último, indicando que poucas espécies dominam as ocorrências enquanto muitas são incomuns, a delimitação de escopo é dependente do domínio do problema.

Com relação a performance de cada arquitetura, a quantidade de amostras não apresentou alta correlação com a acurácia obtida. A diferença de performance foi mais atribuída a escolha da arquitetura, principalmente no cenário onde apenas as imagens foram utilizadas como entrada, no cenário onde utilizamos apenas os dados tabulares ou a concatenação das modalidades obtivemos resultados muito próximos, indicando que a escolha do modelo não tem grande peso nos resultados na ótica da métrica escolhida.

Os modelos classificadores atingiram no cenário de teste uma média de acurácia de 0,8550 para tabular, 0,3458 para imagem e 0,8606 para concatenado, a modalidade tabular por si só demonstrou ser uma escolha viável para contornar um problema de Classificação Visual Fina que apresentou ser mais difícil de resolver no domínio das imagens, a partir dos resultados de relação entre amostras e acurácia, os modelos tendem a acertar a classe majoritária inflando a acurácia geral.

A concatenação demonstra ganho de acurácia na maioria dos cenários em que foi aplicada, principalmente no domínio de imagem, este que apresentou menor desempenho.

Por fim, a concatenação dos dois *embeddings* nesse modelo *downstream* oferece a possibilidade de um trade-off, por mais que a acurácia não apresente ganhos significativamente grandes, a abertura de um novo espaço de características híbrido combina a estabilidade do tabular com a expressividade das imagens, isso garante menor sensibilidade à escolha do algoritmo e boa capacidade de representação. Essa característica abre possibilidades para um classificador com maior desempenho computacional e resultados próximos ao de melhor performance.

Em geral, o presente trabalho apresentou uma abordagem para classificação multimodal de pássaros em território nacional, oferecendo uma *baseline* para o estudo da aplicação de diferentes arquiteturas no cenário de um problema *downstream* de Classificação Visual Fina.

6.1 Trabalhos futuros

Os pontos aqui abordados podem ser utilizados para melhorar a acurácia e o desempenho computacional de classificadores utilizados no monitoramento de espécies ou em outras iniciativas de conservação. Por meio de arquiteturas baseadas em Transformer e de sua dinâmica de concatenação de *embeddings*, surge a viabilidade de explorar dados de ciência cidadã, aproveitando a distribuição de pessoas próximas aos *habitats* dessas espécies.

Considerando as limitações e resultados dispostos, é viável que trabalhos futuros explorem as espécies presentes em outras áreas da distribuição utilizando outras métricas, verificando se o comportamento de

relação das amostras e acurácia continua o mesmo na medida que diminuímos as amostras disponíveis para o modelo, também fica aberta a possibilidade da aplicação de métodos de *Hyperparameter tuning*.

Outro ponto a ser investigado é a substituição das arquiteturas presentes nas *branches*. Principalmente a de imagem a fim de realizar comparações na extração de características presentes nas imagem das espécies dispostas.

Finalmente, a utilização de um modelo monolítico single-branch é uma possibilidade a se explorar visando obter maior explicabilidade dos resultados perante a extração das características devido ao menor número de arquiteturas envolvidas.

REFERÊNCIAS

[1] FUZESSY, L. et al. Loss of species and functions in a deforested megadiverse tropical forest. *Conservation Biology*, v. 38, n. 4, p. e14250, 2024. Disponível em: <https://conbio.onlinelibrary.wiley.com/doi/abs/10.1111/cobi.14250>.

[2] PINHEIRO, B. T.; ALMEIDA, S. M.; SANTOS, M. P. D. The impact of rare and common species on the functional diversity of forest birds in a palm-dominated landscape in the eastern amazon. *Acta Oecologica*, v. 126, p. 104060, 2025. ISSN 1146-609X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1146609X25000049>.

[3] PATRóN, A. L. The ruins of a steel mill: Planetary urbanization in the brazilian amazon. *Journal of Urban History*, v. 50, n. 3, p. 541–562, 2024. Disponível em: <https://doi.org/10.1177/00961442231209298>.

[4] SOUTO, R. D. et al. *Anais da VII Jornada de Geotecnologias do Estado do Rio de Janeiro (JGEOTEC 2024)*. Rio de Janeiro: Editora IVIDES, 2024., 2024. Disponível em: <https://doi.org/10.5281/zenodo.14428169>.

[5] PACHECO, J. F. et al. Annotated checklist of the birds of brazil by the brazilian ornithological records committee—second edition. *Ornithology Research*, v. 29, n. 2, p. 94–105, jun 2021. ISSN 2662-673X. Disponível em: <https://doi.org/10.1007/s43388-021-00058-x>.

[6] RODRIGUEZ-JUAN, J. et al. *Visual WetlandBirds Dataset: Bird Species Identification and Behavior Recognition in Videos*. 2025. Disponível em: <https://arxiv.org/abs/2501.08931>.

[7] VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.

[8] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986. Disponível em: <https://doi.org/10.1038/323533a0>.

[9] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.

[10] SPECHT, D. A general regression neural network. *IEEE Transactions on Neural Networks*, v. 2, n. 6, p. 568–576, 1991.

[11] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

[12] DOSOVITSKIY, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] MAURíCIO, J.; DOMINGUES, I.; BERNARDINO, J. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, v. 13, n. 9, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/9/5521>.

[14] TAKAHASHI, S. et al. Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review. *Journal of Medical Systems*, v. 48, n. 1, p. 84, set. 2024. ISSN 1573-689X. Systematic review on ViTs vs CNNs in medical imaging. Disponível em: <https://doi.org/10.1007/s10916-024-02105-8>.

[15] SHAFIK, W. et al. A novel hybrid inception-xception convolutional neural network for efficient plant disease classification and detection. *Scientific Reports*, v. 15, n. 1, p. 3936, jan. 2025. ISSN 2045-2322. Proposes a hybrid Inception-Xception CNN for plant disease detection across multiple datasets including PlantVillage, Turkey Disease, and Plant Doc. Disponível em: <https://doi.org/10.1038/s41598-024-82857-y>.

[16] CHEN, C.; Mat Isa, N. A.; LIU, X. A review of convolutional neural network based methods for medical image classification. *Computers in Biology and Medicine*, v. 185, p. 109507, 2025. ISSN 0010-4825. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010482524015920>.

[17] MA, Z. et al. Swinfg: A fine-grained recognition scheme based on swin transformer. *Expert Systems with Applications*, v. 244, p. 123021, 2024. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417423035236>.

[18] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359–366, 1989. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/0893608089900208>.

[19] QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, n. 1, p. 81–106, mar. 1986. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116251>.

[20] BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, out. 2001. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1010933404324>.

[21] FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189 – 1232, 2001. Disponível em: <https://doi.org/10.1214/aos/1013203451>.

[22] RANA, P. S. et al. Comparative analysis of tree-based models and deep learning architectures for tabular data: Performance disparities and underlying factors. In: *2023 International Conference on Advanced Computing Communication Technologies (ICACCTech)*. [S.l.: s.n.], 2023. p. 224–231.

[23] BORISOV, V. et al. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, v. 35, n. 6, p. 7499–7519, 2024.

[24] HUANG, X. et al. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[25] ATREY, P. K. et al. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, v. 16, n. 6, p. 345–379, 2010. ISSN 1432-1882. Disponível em: <https://doi.org/10.1007/s00530-010-0182-0>.

[26] SNOEK, C. G. M.; WORRING, M.; SMEULDERS, A. W. M. Early versus late fusion in semantic video analysis. In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2005. (MULTIMEDIA '05), p. 399–402. ISBN 1595930442. Disponível em: <https://doi.org/10.1145/1101149.1101236>.

[27] SRIVASTAVA, N.; SALAKHUTDINOV, R. R. Multimodal learning with deep boltzmann machines. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf>.

[28] KIM, W.; SON, B.; KIM, I. Vilt: Vision-and-language transformer without convolution or region supervision. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 5583–5594.

[29] RADFORD, A. et al. Learning transferable visual models from natural language supervision. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 8748–8763.

[30] JAEGLE, A. et al. Perceiver: General perception with iterative attention. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 4651–4664.

[31] COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27, 1967.

[32] TAHA, K. Big data analytics in iot, social media, nlp, and information security: trends, challenges, and applications. *Journal of Big Data*, v. 12, n. 1, p. 150, 2025. ISSN 2196-1115. Disponível em: <https://doi.org/10.1186/s40537-025-01192-9>.

[33] DIMITRIADIS, S. I.; LIPARAS, D.; INITIATIVE for the A. D. N. How random is the random forest? random forest algorithm on the service of structural imaging biomarkers for alzheimer's disease: from alzheimer's disease neuroimaging initiative (adni) database. *Neural Regeneration Research*, v. 13, n. 6, 2018. ISSN 1673-5374. Disponível em: <https://journals.lww.com/nrronline/fulltext/2018/13060/how_random_is_the_random_forest___random_forest.4.aspx>.

[34] CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00994018>.

[35] GARCíA-GONZALO, E. et al. Hard-rock stability analysis for span design in entry-type excavations with learning classifiers. *Materials*, v. 9, n. 7, 2016. ISSN 1996-1944. Disponível em: <https://www.mdpi.com/1996-1944/9/7/531>.

[36] COX, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, v. 20, n. 2, p. 215–232, 12 2018. ISSN 0035-9246. Disponível em: <https://doi.org/10.1111/j.2517-6161.1958.tb00292.x>.

[37] PAULA, R. A. de et al. Mitigation of nonlinear phase noise in single-channel coherent 16-qam systems employing logistic regression. *Optical and Quantum Electronics*, v. 53, n. 9, p. 508, 2021. ISSN 1572-817X. Disponível em: <https://doi.org/10.1007/s11082-021-03149-7>.

[38] CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939785>.

[39] ALI, Z. H.; BURHAN, A. M. Hybrid machine learning approach for construction cost estimation: an evaluation of extreme gradient boosting model. *Asian Journal of Civil Engineering*, v. 24, n. 7, p. 2427–2442, 2023. ISSN 2522-011X. Disponível em: <https://doi.org/10.1007/s42107-023-00651-z>.

[40] SASTRY, S. et al. Birdsat: Cross-view contrastive masked autoencoders for bird species classification and mapping. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2024. p. 7136–7145.

[41] LU, Z. et al. Dusafnet: A multi-path feature fusion and spectral–temporal attention-based model for bird audio classification. *Animals*, v. 15, n. 15, 2025. ISSN 2076-2615. Disponível em: <https://www.mdpi.com/2076-2615/15/15/2228>.

Apêndices

Tabela 9 – Registros completos de espécies por estado após filtragens.

Estado	Espécie	Quantidade
AC	agulha-de-garganta-branca	342
	periquito-de-cabeça-suja	303
	ariramba-castanha	292
	ariramba-da-capoeira	274
	anambé-de-cara-preta	271
AL	saíra-pintor	284
	picapauzinho-de-pernambuco	190
	papa-taoca-de-pernambuco	140
	maria-de-barriga-branca	123
	anumará	120
AM	araçari-negro	639
	galo-da-serra	605
	capitão-de-bigode-carijó	595
	papagaio-da-várzea	559
	rabo-de-arame	553
AP	uirapuru-vermelho	186
	caboclinho-lindo	136
	caraxué	115
	formigueiro-de-cabeça-preta	107
	iratauá-grande	105
BA	beija-flor-de-gravata-vermelha	696
	arara-azul-de-lear	696
	gravatazeiro	695
	saíra-pérola	650
	anambé-de-asa-branca	601
CE	soldadinho-do-araripe	696
	cara-suja	596
	vira-folha-cearense	417
	maria-do-nordeste	366
	jacucaca	364
DF	capacetinho-do-oco-do-pau	312
	maria-preta-do-nordeste	226
	limpa-folha-do-brejo	183
	pula-pula-de-sobrancelha	102
	bacurau-de-rabo-maculado	89

Tabela 9 – Registros completos de espécies por estado após filtragens.

Estado	Espécie	Quantidade
ES	mutum-de-bico-vermelho	525
	rabo-branco-mirim	500
	tiriba-de-orelha-branca	354
	chauá	334
	furriel	236
EX	pelicano	541
	ganso-de-magalhães	345
	garça-moura-europeia	315
	tesoura-do-campo	305
	marreca-oveira	303
GO	tiriba-do-paranã	277
	pato-corredor	182
	cardeal-do-araguaia	170
	papagaio-galego	139
	vite-vite-de-cabeça-cinza	136
MA	garça-tricolor	250
	chupa-dente-de-capuz	211
	rabo-branco-do-maranhão	200
	aracuã-de-sobrancelhas	174
	araponga-do-nordeste	156
MG	beija-flor-de-gravata-verde	696
	pato-mergulhão	645
	andarilho	641
	maxalalagá	612
	rolinha-do-planalto	595
MS	tiriba-fogo	696
	rapazinho-do-chaco	679
	periquito-de-cabeça-preta	569
	arapaçu-do-campo	441
	jacutinga-de-garganta-azul	433
MT	tiriba-do-madeira	519
	cujubi	502
	saíra-de-cabeça-azul	502
	capitão-de-cinta	494
	jacu-de-barriga-castanha	484

Tabela 9 – Registros completos de espécies por estado após filtragens.

Estado	Espécie	Quantidade
PA	jacupiranga	579
	tiriba-de-hellmayr	432
	ararajuba	392
	asa-de-sabre-de-cauda-escura	323
	arapaçu-de-listras-brancas-do-leste	304
PB	saíra-pintor	235
	chororó-didi	178
	papa-taoca-de-pernambuco	141
	gavião-gato-do-nordeste	122
	maria-de-barriga-branca	116
PE	atobá-de-pé-vermelho	668
	grazina	540
	juruviara-de-noronha	508
	trinta-réis-preto	465
	rabo-de-palha-de-bico-laranja	456
PI	arapaçu-do-nordeste	127
	chupa-dente-de-capuz	81
	canário-do-amazonas	75
	caneleiro-enxofre	55
	asa-de-telha-pálido	55
PR	bicudinho-do-brejo	404
	cisqueiro	339
	arredio-oliváceo	206
	tico-tico-de-costas-cinza	205
	gralha-picaça	202
RJ	formigueiro-de-cabeça-negra	603
	formigueiro-do-litoral	595
	papa-moscas-estrela	381
	vite-vite	374
	saudade	332
RN	chorozinho-de-papo-preto	268
	picapauzinho-da-caatinga	197
	joão-xique-xique	171
	maçarico-de-costas-brancas	148
	caneleiro-enxofre	120

Tabela 9 – Registros completos de espécies por estado após filtragens.

Estado	Espécie	Quantidade
RO	gaturamo-de-bico-grosso	243
	periquito-de-cabeça-suja	238
	curica-de-bochecha-laranja	196
	maria-da-praia	190
	picapauzinho-dourado	169
RR	periquito-de-bochecha-parda	448
	joão-pinto-amarelo	444
	choca-de-crista-preta	441
	papa-capim-cinza	413
	téu-téu-da-savana	389
RS	joão-da-palha	678
	caminheiro-de-unha-curta	664
	caminheiro-de-espora	659
	boininha	653
	batuíra-de-coleira-dupla	637
SC	maria-catarinense	666
	papagaio-charão	566
	aracuã-escamoso	472
	flamingo-dos-andes	467
	tapaculo-ferreirinho	442
SE	chorozinho-de-papo-preto	66
	jandaia-verdadeira	52
	pipira-preta	24
	papa-taoca-da-bahia	23
	maçarico-branco	21
SP	bicudinho-do-brejo-paulista	696
	topetinho-verde	569
	papagaio-de-cara-roxa	543
	maria-leque-do-sudeste	520
	não-pode-parar	514
TO	pica-pau-da-taboca	354
	pato-corredor	283
	chororó-de-goiás	275
	cardeal-do-araguaia	203
	garça-da-mata	193

Tabela 10 – Acurácia de Teste - Todos os Classificadores (Todos os Estados)

Estado	Modalidade	k-NN	LogReg	RandomForest	SVM	XGBoost
AC	Tabular	0.5859	0.5926	0.6364	0.5926	0.6465

Tabela 10

Estado	**Modalidade**	***k*-NN**	**LogReg**	**RandomForest**	**SVM**	**XGBoost**
AC	Imagem	0.2121	0.2323	0.2222	0.2222	0.2222
AC	Concatenado	0.6094	0.6364	0.6195	0.5724	0.6195
AL	Tabular	0.8140	0.8605	0.8547	0.8605	0.8547
AL	Imagem	0.2151	0.3430	0.2733	0.3372	0.2733
AL	Concatenado	0.7791	0.8663	0.8140	0.8663	0.8256
AM	Tabular	0.6751	0.6193	0.7140	0.5888	0.6734
AM	Imagem	0.2893	0.3316	0.3587	0.3249	0.3706
AM	Concatenado	0.6108	0.6650	0.6497	0.6328	0.6937
AP	Tabular	0.6615	0.6692	0.6538	0.6615	0.6462
AP	Imagem	0.2308	0.2308	0.2615	0.3154	0.2615
AP	Concatenado	0.6385	0.6154	0.6538	0.6692	0.6923
BA	Tabular	1.0000	1.0000	1.0000	1.0000	1.0000
BA	Imagem	0.2530	0.2754	0.2964	0.2799	0.3009
BA	Concatenado	1.0000	1.0000	1.0000	1.0000	1.0000
CE	Tabular	0.8668	0.8893	0.8893	0.8730	0.8832
CE	Imagem	0.2930	0.3811	0.3730	0.3607	0.3730
CE	Concatenado	0.8402	0.8873	0.8852	0.8770	0.8955
DF	Tabular	0.9617	0.9727	0.9672	0.9454	0.9672
DF	Imagem	0.3333	0.3880	0.3279	0.4044	0.3443
DF	Concatenado	0.9290	0.9563	0.9454	0.9399	0.9563
ES	Tabular	0.8462	0.8154	0.8385	0.8282	0.8462
ES	Imagem	0.3590	0.3513	0.4333	0.3590	0.4487
ES	Concatenado	0.8487	0.8564	0.8897	0.8590	0.8846
EX	Tabular	0.6326	0.6547	0.6243	0.6575	0.6381
EX	Imagem	0.2845	0.3757	0.3343	0.3950	0.3343
EX	Concatenado	0.7155	0.7459	0.6961	0.7403	0.6878
GO	Tabular	0.8674	0.8729	0.8950	0.8508	0.8840
GO	Imagem	0.2541	0.2873	0.3204	0.2983	0.3591
GO	Concatenado	0.8619	0.8840	0.9006	0.8453	0.9061
MA	Tabular	1.0000	1.0000	1.0000	1.0000	1.0000
MA	Imagem	0.2462	0.3568	0.2714	0.3417	0.3015
MA	Concatenado	0.9950	1.0000	1.0000	1.0000	1.0000
MG	Tabular	1.0000	1.0000	1.0000	1.0000	0.9969
MG	Imagem	0.2712	0.2680	0.3260	0.2853	0.3260
MG	Concatenado	0.9984	1.0000	1.0000	1.0000	0.9969
MS	Tabular	0.8759	0.8741	0.8812	0.8652	0.8830
MS	Imagem	0.2589	0.2748	0.3245	0.2287	0.3333
MS	Concatenado	0.8617	0.8723	0.8723	0.8652	0.8883

Tabela 10

Estado	**Modalidade**	***k*-NN**	**LogReg**	**RandomForest**	**SVM**	**XGBoost**
MT	Tabular	0.8603	0.8463	0.8483	0.8044	0.8503
MT	Imagem	0.2315	0.2255	0.2575	0.2116	0.2615
MT	Concatenado	0.8044	0.8383	0.8144	0.7964	0.8443
PA	Tabular	0.8177	0.7857	0.8202	0.7882	0.8300
PA	Imagem	0.2611	0.3251	0.3079	0.3374	0.2906
PA	Concatenado	0.8547	0.8645	0.8498	0.8547	0.8645
PB	Tabular	0.8050	0.7862	0.7987	0.7925	0.7862
PB	Imagem	0.3019	0.3522	0.3082	0.3648	0.3585
PB	Concatenado	0.7610	0.7736	0.7862	0.8050	0.8050
PE	Tabular	0.8201	0.8068	0.8295	0.8239	0.8333
PE	Imagem	0.2973	0.3371	0.3277	0.3295	0.3485
PE	Concatenado	0.8314	0.8314	0.8428	0.8239	0.8314
PI	Tabular	0.8734	0.8481	0.9114	0.8481	0.8608
PI	Imagem	0.3165	0.3165	0.2658	0.3038	0.2532
PI	Concatenado	0.6962	0.7975	0.8861	0.6329	0.8101
PR	Tabular	0.8787	0.9118	0.9081	0.8787	0.9007
PR	Imagem	0.2794	0.3162	0.3088	0.3162	0.2610
PR	Concatenado	0.8750	0.9044	0.8824	0.8787	0.8934
RJ	Tabular	0.6083	0.5886	0.6346	0.5733	0.6258
RJ	Imagem	0.2713	0.3042	0.2757	0.3020	0.3107
RJ	Concatenado	0.6433	0.6761	0.6455	0.6608	0.6652
RN	Tabular	1.0000	1.0000	1.0000	1.0000	0.9945
RN	Imagem	0.3370	0.4309	0.3757	0.4309	0.3923
RN	Concatenado	0.9834	1.0000	1.0000	1.0000	0.9945
RO	Tabular	0.8077	0.7933	0.8173	0.7788	0.8365
RO	Imagem	0.2212	0.1827	0.2644	0.1731	0.2163
RO	Concatenado	0.8173	0.8125	0.8125	0.8029	0.8125
RR	Tabular	0.8056	0.8080	0.8080	0.8080	0.8080
RR	Imagem	0.2553	0.2834	0.2810	0.2623	0.2717
RR	Concatenado	0.8080	0.8103	0.8173	0.8033	0.7963
RS	Tabular	0.7102	0.6692	0.7223	0.5933	0.7086
RS	Imagem	0.2792	0.3171	0.3232	0.3308	0.3247
RS	Concatenado	0.6449	0.7011	0.6904	0.6616	0.7132
SC	Tabular	1.0000	1.0000	1.0000	1.0000	0.9981
SC	Imagem	0.3098	0.3155	0.3690	0.3040	0.3728
SC	Concatenado	0.9943	1.0000	1.0000	1.0000	0.9981
SE	Tabular	0.7368	0.8421	0.8158	0.8158	0.7895

Tabela 10

Estado	**Modalidade**	***k*-NN**	**LogReg**	**RandomForest**	**SVM**	**XGBoost**
SE	Imagem	0.4211	0.4474	0.2895	0.4474	0.3421
SE	Concatenado	0.7368	0.8421	0.7105	0.7632	0.7632
SP	Tabular	0.8770	0.9104	0.9121	0.8787	0.9069
SP	Imagem	0.3111	0.3919	0.3497	0.3620	0.3814
SP	Concatenado	0.8805	0.9121	0.8893	0.8840	0.8946
TO	Tabular	1.0000	1.0000	1.0000	1.0000	0.9962
TO	Imagem	0.3015	0.3206	0.3206	0.2481	0.3397
TO	Concatenado	1.0000	1.0000	1.0000	1.0000	0.9962