

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**  
**FACULDADE DE CIÊNCIAS - CAMPUS BAURU**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

VINICIUS RODRIGUES DE SOUZA FIEDLER GARCIA

**IDENTIFICAÇÃO BIOMÉTRICA DE CÃES POR MEIO DO  
RECONHECIMENTO FACIAL UTILIZANDO TRANSFORMADORES**

BAURU  
Novembro/2024

VINICIUS RODRIGUES DE SOUZA FIEDLER GARCIA

**IDENTIFICAÇÃO BIOMÉTRICA DE CÃES POR MEIO DO  
RECONHECIMENTO FACIAL UTILIZANDO TRANSFORMADORES**

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Orientador: Prof. Dr. Aparecido Nilceu Marana

Coorientador: Prof. Victor Hugo Braguim Canto

BAURU  
Novembro/2024

G216i	<p>Garcia, Vinicius IDENTIFICAÇÃO BIOMÉTRICA DE CÃES POR MEIO DO RECONHECIMENTO FACIAL UTILIZANDO TRANSFORMADORES / Vinicius Garcia. -- Bauru, 2024 45 p. : il., tabs., fotos</p> <p>Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru</p> <p>Orientador: Aparecido Nilceu Marana Coorientador: Victor Hugo Braguim Canto</p> <p>1. Identificação Biométrica de Cães. I. Título.</p>
-------	--

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Vinicius Rodrigues de Souza Fiedler Garcia

# **IDENTIFICAÇÃO BIOMÉTRICA DE CÃES POR MEIO DO RECONHECIMENTO FACIAL UTILIZANDO TRANSFORMADORES**

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

---

**Prof. Dr. Aparecido Nilceu Marana**

Orientador

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

---

**Profa. Dra. Simone das Graças**

**Domingues Prado**

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

---

**Prof. Dr. Kelton Augusto Pontara da  
Costa**

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Bauru, 13 de Novembro de 2024.

*Dedico este trabalho a todos que estiveram presentes em minha jornada.  
À minha família — Heitor, Daniela e Lara — que me apoiaram em todos os momentos  
de dúvida.  
Aos meus amigos, à minha república e a todos que me acompanharam ao longo dessa  
caminhada.  
E ao meu orientador, Nilceu, que me permitiu explorar um dos campos pelos quais já  
era apaixonado.*

# Agradecimentos

Este trabalho representa o esforço conjunto de muitos que, de forma direta ou indireta, contribuíram para que ele se tornasse realidade. A todos, deixo meu mais sincero agradecimento.

À Universidade Estadual Paulista (UNESP), por fornecer um ambiente de aprendizado e crescimento, onde tive a oportunidade de expandir meus conhecimentos e me preparar para novos desafios.

Aos meus pais, que, com amor incondicional e apoio constante, estiveram ao meu lado em cada etapa deste caminho. Vocês sempre acreditaram em mim, mesmo nos momentos em que eu duvidei de mim mesmo. Como dizia Sêneca, “É nas adversidades que se descobre quem são os verdadeiros amigos.” Vocês foram meu pilar de força em todas as dificuldades.

Ao grupo de programação competitiva, pela troca de conhecimento e pelo constante incentivo ao aprimoramento. Em especial, agradeço ao professor Wilson, pela dedicação e orientação que me inspiraram a enfrentar desafios com determinação, e aos meus companheiros de grupo, Alex e Kaio, pela parceria e pelas muitas horas de prática e aprendizado compartilhadas.

Ao meu orientador, professor Nilceu, pela paciência e orientação ao longo deste trabalho, e ao meu co-orientador, Victor, cujo apoio e conhecimento foram fundamentais para o desenvolvimento deste projeto.

Aos meus amigos Lucas, João Z., João B., Gabriel, Luca, Rafael, que compartilharam comigo essa jornada, celebrando as vitórias e apoiando-me nas dificuldades. A companhia e amizade de cada um de vocês tornaram essa fase da minha vida muito mais leve e significativa.

Finalmente, agradeço a todos aqueles que, de alguma forma, contribuíram para que este trabalho se concretizasse, oferecendo ajuda, motivação e ensinamentos preciosos.

*"Foi o tempo que dedicaste à tua rosa que a fez tão importante."*

# Resumo

O reconhecimento biométrico de cães, utilizando visão computacional e aprendizagem de máquina, apresenta-se como uma solução atual e bastante relevante, tanto do ponto de vista social quanto econômico, em aplicações como o cadastro e a identificação automática desses animais para fins de gerenciamento e controle da população canina, a localização dos tutores dos animais perdidos e a prevenção de fraudes nos atendimentos em clínicas veterinárias públicas e privadas. Este trabalho propõe uma abordagem para a identificação biométrica automática de cães que utiliza o modelo YOLO na detecção automática das cabeças dos cães, durante a etapa de segmentação das imagens digitais dos animais, e o modelo de transformador visual na etapa de extração das características faciais dos cães. A abordagem proposta foi incorporada a um sistema biométrico projetado e implementado neste trabalho para ser executado em um servidor, hospedado em um ambiente de nuvem, de forma integrada a um aplicativo móvel que visa facilitar a captura e o envio das imagens ao servidor diretamente pelo usuário.

**Palavras-chave:** Identificação Biométrica de Cães; YOLO; Transformador Visual; Visão Computacional; Aprendizado Profundo; Aplicação Móvel.

# Abstract

Dog biometric recognition, using computer vision and machine learning, emerges as a contemporary and highly relevant solution from both social and economic perspectives. Applications include registering and automatically identifying dogs for population management, locating lost pet owners, and preventing fraud in veterinary clinics. This study proposes an approach for automatic dog biometric identification that leverages the YOLO model for automatic detection of dog heads during the image segmentation phase and a vision transformer model for extracting facial features. The proposed approach has been integrated into a biometric system designed and implemented as part of this work. This system runs on a server hosted in a cloud environment and is integrated with a mobile application to facilitate image capture and submission directly by users.

**Keywords:** Dog biometric identification; YOLO; Visual Transformer; Computer Vision; Deep Learning; Mobile Application.

# Listas de figuras

Figura 1 – Diagrama do Processo AutoDistill. . . . .	19
Figura 2 – Estrutura do mecanismo de atenção com múltiplas cabeças. . . . .	20
Figura 3 – Arquitetura do transformador, composta por múltiplos blocos de codificador (à esquerda) e decodificador (à direita). . . . .	22
Figura 4 – Arquitetura do Transformador Visual. . . . .	24
Figura 5 – Processo de detecção de objetos com o YOLO. Onde são selecionadas as melhores caixas, resultando nas detecções finais. . . . .	25
Figura 6 – Arquitetura geral do YOLO. . . . .	26
Figura 7 – Amostras da base de dados DogFaceNet (MOUGEOT; LI; JIA, 2019). .	29
Figura 8 – Fluxo de transformações realizadas pela técnica <i>RandomResizedCrop</i> no processo de <i>data augmentation</i> nas imagens faciais dos cães do conjunto de treinamento. . . . .	31
Figura 9 – Gráficos das métricas de avaliação: (a) Acurácia, (b) AUC, (c) Precision, (d) Recall, (e) F1-Score. . . . .	36
Figura 10 – Visão Geral da Aplicação. . . . .	38
Figura 11 – Estrutura do servidor na nuvem. . . . .	38
Figura 12 – Etapas do processamento no Flask. . . . .	39
Figura 13 – Exemplo de uso do aplicativo. Funcionalidade: Identificação de um animal. . . . .	42
Figura 14 – Exemplo de uso do aplicativo. Funcionalidade: Exibição de todos os animais cadastrados no sistema. . . . .	42
Figura 15 – Diagrama do funcionamento da aplicação. . . . .	44

# **Lista de Quadros**

Quadro 1 – Resultados obtidos pelo modelo YOLO para segmentação das imagens dos cães. . . . .	33
Quadro 2 – Resultados do modelo de transformadores visuais para extração de características. . . . .	35

# Lista de abreviaturas e siglas

CNN	<i>Convolutional Neural Network</i>
FPN	<i>Feature Pyramid Network</i>
IoU	<i>Intersection over Union</i>
LSTM	<i>Long-Short Term Memory</i>
NMS	<i>Non-Maximum Suppression</i>
PAN	<i>Path Aggregation Network</i>
R-CNN	<i>Region-based Convolutional Neural Network</i>
RNC	Rede Neural Convolucional
SAM	<i>Segment Anything Model</i>
ViT	<i>Vision Transformer</i>
YOLO	<i>You Only Look at Once</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	<b>Problemática</b>	14
1.2	<b>Justificativa</b>	15
1.3	<b>Objetivos</b>	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	<b>Destilação</b>	17
2.1.1	Autodestilação	18
2.1.2	AutoDistill	18
2.2	<b>Transformadores</b>	<b>19</b>
2.2.1	Arquitetura do Modelo	19
2.2.2	Mecanismo de Atenção	21
2.2.3	Codificação Posicional	21
2.2.4	Vantagens e Desvantagens	21
2.3	<b>Transformadores Visuais</b>	<b>23</b>
2.4	<b>YOLO</b>	<b>24</b>
2.4.1	Arquitetura	25
2.4.2	Processo de Treinamento e Inferência	26
2.4.3	Vantagens e Desvantagens	27
2.4.4	Evolução	27
<b>3</b>	<b>MATERIAL E MÉTODOS</b>	<b>29</b>
3.1	<b>Base de Dados</b>	<b>29</b>
3.1.1	<i>Data Augmentation</i>	30
3.2	<b>Métricas de Avaliação</b>	<b>31</b>
3.3	<b>Ambiente Utilizado</b>	<b>32</b>
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS</b>	<b>33</b>
4.1	<b>Treinamento do modelo YOLO</b>	<b>33</b>
4.2	<b>Modelo Transformadores Visuais</b>	<b>34</b>
<b>5</b>	<b>APLICAÇÃO</b>	<b>37</b>
5.1	<b>Introdução</b>	<b>37</b>
5.2	<b>Estrutura da aplicação</b>	<b>37</b>
5.2.1	Servidor	37
5.2.2	Aplicativo	40

<b>5.3</b>	<b>Funcionamento da Aplicação</b>	<b>41</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>45</b>
<b>6.1</b>	<b>Aplicativo Móvel</b>	<b>45</b>
<b>6.2</b>	<b>Modelos</b>	<b>45</b>
<b>6.3</b>	<b>Trabalhos Futuros</b>	<b>46</b>
	<b>REFERÊNCIAS</b>	<b>48</b>

# 1 Introdução

A identificação de animais tem sido realizada por meio de técnicas invasivas, como *transponders* e brincos eletrônicos, que podem afetar o bem-estar dos animais, e também por meio de métodos não invasivos, como biometria da íris e identificação por DNA, que enfrentam desafios relacionados a custos e praticidade (NAZARENO; RONCADA; SILVA, 2014). Avanços recentes no aprendizado de máquina passaram a possibilitar o uso de características biométricas que podem ser coletadas à distância e por sensores mais baratos, como o reconhecimento facial, promovendo alternativas bastante robustas, eficazes e não invasivas para identificação de animais dentro de uma mesma espécie, como a identificação de cães proposta por Canto et al. (2023).

Em paralelo a essas inovações, os Transformadores Visuais (*ViT*, do inglês *Vision Transformers*) (DOSOVITSKIY et al., 2020) trouxeram avanços no campo de visão computacional ao superar as RNCs em diversas tarefas de visão computacional. No entanto, os ViT normalmente requerem maiores quantidades de dados de treinamento para alcançar um desempenho comparável ao das RNCs.

Assim como as pesquisas realizadas por Canto (2023), este trabalho teve como objetivo o uso de tecnologias de reconhecimento facial, com um foco particular em transformadores visuais, para a identificação biométrica de cães. O objetivo foi adaptar essas técnicas de visão computacional para desenvolver um método confiável, não invasivo e prático de identificação canina. Ademais, durante este trabalho foi implementado um aplicativo que permite o cadastro de animais em uma base de dados para a posterior identificação ou autenticação biométrica desses animais por meio do método desenvolvido.

## 1.1 Problemática

Atualmente, os métodos de identificação de animais dividem-se entre invasivos, como brincos eletrônicos e *transponders*, que podem comprometer o bem-estar dos animais domésticos, e não invasivos, como análises laboratoriais e reconhecimento de íris, que frequentemente são custosos e não tão eficientes (NAZARENO; RONCADA; SILVA, 2014). Isso acaba resultando em uma menor adesão por parte dos tutores, principalmente devido ao desconforto ou aos altos custos envolvidos nessas abordagens.

A demanda por soluções eficazes e baratas é importante, especialmente considerando que a população de cães é a predominante diante de todos os outros animais

domésticos (Instituto Pet Brasil, 2022). Esse cenário impacta diretamente diversos setores, como seguros, vigilância sanitária e saúde pública, além de influenciar programas de controle populacional e bem-estar animal. Assim, há uma necessidade crescente de métodos inovadores que sejam ao mesmo tempo práticos, acessíveis e seguros, garantindo a identificação confiável dos animais e facilitando a adesão por parte dos tutores.

## 1.2 Justificativa

Dada a situação apresentada, é essencial o desenvolvimento de soluções acessíveis e eficientes para a identificação de cães, sendo as técnicas de aprendizagem de máquina, como o transformador visual (ViT) (DOSOVITSKIY et al., 2020), uma abordagem promissora. A aplicação do ViT ao reconhecimento facial de cães permite a criação de um método não invasivo, eficiente e de fácil adoção pelos tutores, superando as limitações dos métodos atuais. No entanto, para tornar essa tecnologia amplamente utilizável, é necessário um meio de acesso simplificado e conveniente para o público em geral.

Nesse contexto, um aplicativo móvel torna-se indispensável, pois combina a ampla utilização de *smartphones* no Brasil com a possibilidade de se prover uma interface amigável para uso das técnicas de identificação biométrica. O aplicativo desenvolvido possibilita que os tutores identifiquem seus animais de forma prática, eliminando a necessidade de dispositivos invasivos ou caros (NAZARENO; RONCADA; SILVA, 2014). Além disso, ele facilita o cadastro dos animais em uma base de dados, contribuindo para a identificação em casos de perda, roubo ou até mesmo disputas de propriedade.

O aplicativo desenvolvido permite que qualquer pessoa possa operá-lo de maneira intuitiva, incluindo funções básicas como captura da foto do animal, processamento da imagem utilizando o modelo ViT, e exibição do resultado da identificação ou autenticação do animal. Esse *design* simplificado é crucial para garantir a acessibilidade e promover a adesão ao método de identificação biométrica, tornando-o uma solução prática e acessível para tutores, clínicas veterinárias, ONGs e outros envolvidos no cuidado e controle de cães.

## 1.3 Objetivos

O objetivo geral deste trabalho foi possibilitar a identificação biométrica de cães por meio do reconhecimento facial, via transformadores visuais, utilizando um aplicativo móvel projetado e desenvolvido especificamente para esta finalidade.

Os objetivos específicos deste trabalho foram:

- Obter bases de dados contendo imagens faciais de cães em diversas posições e obstruções;
- Pesquisar métodos atuais e eficientes para segmentação das faces dos cães em imagens, como por exemplo o YOLO;
- Pesquisar modelos de arquitetura de transformadores visuais mais apropriados para serem utilizados em aplicativos móveis;
- Desenvolver, treinar, validar e testar o modelo de transformador visual escolhido para a tarefa de reconhecimento facial de cães;
- Avaliar as ferramentas e linguagens de programação para desenvolvimento de aplicativos móveis que melhor se adequassem à este projeto;
- Desenvolver o aplicativo móvel com as ferramentas e a linguagem selecionada para o cadastro e a identificação biométrica de cães, utilizando os modelos escolhidos.

## 2 Fundamentação Teórica

Este capítulo apresenta os conceitos fundamentais abordados neste trabalho, como aprendizado de máquina, redes neurais avançadas e, em particular, os transformadores, com destaque ao transformador visual e ao YOLO, que desempenham papéis essenciais no método de reconhecimento facial de cães proposto.

### 2.1 Destilação

Destilação é um processo em que um modelo menor (alvo) aprende a partir de um modelo maior (base), resultando em modelos mais eficientes e compactos.

Os modelos base, conhecidos como modelos professores, são grandes, pré-treinados, possuem um vasto conhecimento e são capazes de realizar múltiplas tarefas com alta precisão. Esses modelos são utilizados para gerar rótulos e fornecer conhecimento para os modelos alvo (HINTON; VINYALS; DEAN, 2015).

Os modelos alvo, por sua vez, também chamados de modelos estudantes, são menores e mais otimizados. Eles aprendem a partir dos modelos base, utilizando técnicas de destilação que garantem a transferência eficiente do conhecimento, resultando em base rápidos e adaptáveis (HINTON; VINYALS; DEAN, 2015).

Os modelos base são modelos pré-treinados em grandes volumes de dados, utilizando abordagens de auto-supervisão ou semi-supervisão, e são capazes de capturar representações significativas que facilitam uma ampla gama de tarefas subsequentes. Esses modelos funcionam como um ponto de partida sólido para a adaptação a tarefas específicas, permitindo economizar recursos e aumentar a precisão dos modelos finais (modelos alvo).

Existem três tipos principais de modelos base: baseados em texto, baseados em visão e baseados em modalidades heterogêneas, que combinam dados de visão, texto e áudio. Para tarefas de visão computacional, os modelos baseados em visão são os mais relevantes, pois permitem o aprendizado de características visuais complexas de maneira genérica. Isso garante que o modelo possa ser posteriormente adaptado e refinado para contextos específicos (AWAIS et al., 2023).

Um aspecto importante dos modelos base é o treinamento em larga escala, que possibilita a construção de uma compreensão rica e generalizável sobre dados visuais. Por exemplo, o modelo SAM (KIRILLOV et al., 2023) foi treinado com mais de um bilhão de máscaras e onze milhões de imagens, permitindo sua adaptação para várias tarefas, desde segmentação geral até casos específicos, como segmentação

em imagens médicas. Esse tipo de treinamento garante que os modelos base sejam suficientemente robustos para serem usados em diferentes aplicações, onde a precisão e a capacidade de generalização são essenciais.

### 2.1.1 Autodestilação

Autodestilação é uma variação do processo de destilação tradicional onde o modelo utiliza suas próprias previsões para aperfeiçoar seu desempenho ao longo do tempo. Neste caso, o modelo serve simultaneamente como professor e estudante, refinando suas previsões a partir de sucessivas iterações de aprendizado. A autodestilação visa simplificar o processo de treinamento ao eliminar a necessidade de um modelo professor externo, permitindo que o próprio modelo adapte suas respostas e melhore sua acurácia em tarefas específicas (ZHANG et al., 2022).

Essa técnica é particularmente vantajosa quando se deseja melhorar continuamente um modelo já existente, otimizando-o para contextos específicos sem precisar desenvolver um novo modelo professor.

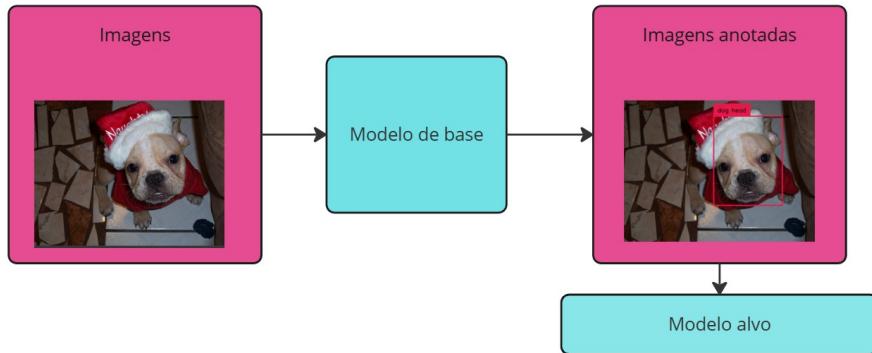
### 2.1.2 AutoDistill

*AutoDistill* é uma ferramenta que automatiza o processo de destilação, auxiliando na criação de modelos eficientes para aplicações que demandam otimização de hardware e alta precisão. A ferramenta utiliza métodos de busca de arquitetura, identificando automaticamente configurações que equilibram precisão e latência de inferência (ZHANG et al., 2022).

O uso do *AutoDistill* é vantajoso, uma vez que elimina a necessidade de anotadores humanos em tarefas complexas de visão computacional, automatizando o processo de destilação e tornando viável o desenvolvimento de modelos com grandes bases de dados. Além disso, inclui a técnica de *Flash Distillation*, que realiza uma transferência de conhecimento rápida para identificar modelos promissores com menor custo computacional.

Esse processo é composto por múltiplas iterações, onde a arquitetura dos modelos é ajustada de acordo com os resultados obtidos. O *AutoDistill* também leva em consideração o desempenho do hardware alvo, realizando medições precisas da latência e outras métricas que são utilizadas para guiar o processo de otimização. Dessa forma, a ferramenta se torna uma solução eficaz para a criação de modelos compactos e adaptados a diferentes contextos de aplicação. A Figura 1 mostra um diagrama do processo *AutoDistill*.

Figura 1 – Diagrama do Processo AutoDistill.



Fonte: Elaborada pelo autor.

## 2.2 Transformadores

Os transformadores, introduzidos por Vaswani et al. (2017), são uma arquitetura de redes neurais que revolucionaram o campo do aprendizado de máquina, principalmente no contexto de processamento de linguagem natural e visão computacional. Os transformadores foram projetados para superar as limitações das RNNs (Recurrent Neural Network) e CNNs em problemas de modelagem de sequência e transdução, substituindo mecanismos complexos de recorrência e convolução por um mecanismo baseado exclusivamente em atenção.

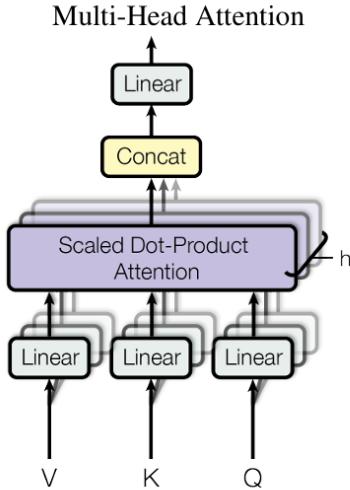
O principal diferencial dos transformadores é a eliminação do cálculo sequencial, permitindo a paralelização do processamento e, consequentemente, a redução significativa do tempo de treinamento. Em vez de depender de estruturas recorrentes, como as LSTMs (Long Short-Term Memory), os transformadores utilizam camadas de atenção auto-regressiva para modelar dependências globais entre os elementos de uma sequência, sendo capazes de processar todos os elementos simultaneamente.

### 2.2.1 Arquitetura do Modelo

A arquitetura do transformador é composta por um bloco de *encoder* e um bloco de *decoder*, ambos organizados em camadas similares empilhadas, como demonstrado na figura 3. O *encoder* recebe uma sequência de entrada e a mapeia para uma representação interna, enquanto o *decoder* utiliza essa representação para gerar a saída. Cada camada do *encoder* e do *decoder* possui duas subcamadas principais: uma subcamada de atenção com várias cabeças e uma rede totalmente conectada (*feed-forward*). As camadas utilizam conexões residuais seguidas de normalização

para facilitar o treinamento. A Figura 2 ilustra a estrutura do mecanismo de atenção com múltiplas cabeças.

Figura 2 – Estrutura do mecanismo de atenção com múltiplas cabeças.



Fonte: (VASWANI et al., 2017)

A subcamada de atenção com várias cabeças (*Multi-Head Attention*) consiste em várias cabeças de atenção funcionando em paralelo. Cada cabeça utiliza três matrizes de pesos: consultas ( $Q$ ), chaves ( $K$ ) e valores ( $V$ ). A operação de atenção é dada pela equação 2.1:

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

Cada cabeça de atenção gera uma representação diferente dos elementos da sequência, permitindo que o modelo capture diferentes aspectos das relações entre os elementos. As saídas de todas as cabeças são concatenadas e projetadas novamente para uma representação comum.

A subcamada de Rede Totalmente Conectada é aplicada a cada posição da sequência de forma independente. Ela consiste em duas camadas lineares separadas por uma função de ativação não-linear (usualmente a função ReLU). A operação é dada apartir da equação 2.2:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (2.2)$$

em que  $W_1$  e  $W_2$  são matrizes de pesos aprendíveis, e  $b_1$  e  $b_2$  são vetores de vieses aprendíveis. Essa subcamada adiciona uma capacidade não-linear ao transformador, permitindo que ele capture características mais complexas dos dados de entrada.

Além dessas subcamadas, as conexões residuais são aplicadas em cada subcamada, seguidas por uma camada de normalização. Em outras palavras, a saída de cada subcamada é somada à sua respectiva entrada, e o resultado passa por uma normalização, que pode ser observada na equação 2.3:

$$\text{Saída} = \text{LayerNorm}(x + \text{Subcamada}(x)) \quad (2.3)$$

As conexões residuais ajudam a evitar problemas como o desaparecimento do gradiente e facilitam o treinamento, garantindo que o fluxo de informação seja preservado ao longo das camadas, mesmo em redes muito profundas.

Essas subcamadas trabalham em conjunto para permitir que o transformador processe sequências de maneira eficiente e capture as dependências globais entre os elementos, resultando em um modelo capaz de realizar tarefas complexas, como tradução automática e sumarização de texto.

### 2.2.2 Mecanismo de Atenção

O mecanismo de atenção é o cerne dos transformadores, permitindo que cada elemento de uma sequência se conecte diretamente a todos os outros elementos, independentemente da sua posição. A “Atenção Escalonada por Produto Escalar” (*Scaled Dot-Product Attention*) é calculada utilizando consultas, chaves e valores. O resultado da atenção é obtido como uma soma ponderada dos valores, sendo os pesos definidos pela compatibilidade entre as consultas e as chaves. Para evitar problemas de gradientes muito pequenos, o produto escalar é dividido pela raiz quadrada da dimensão das chaves.

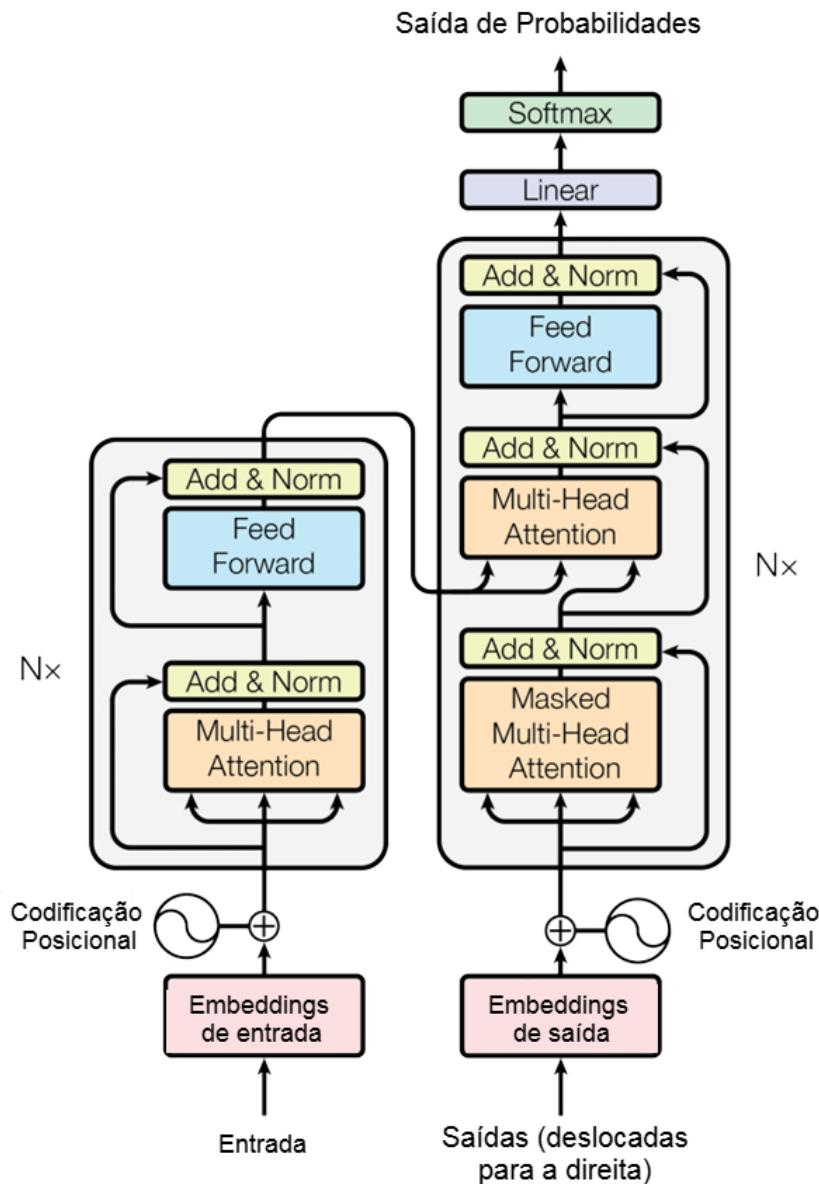
### 2.2.3 Codificação Posicional

Como os transformadores não têm uma estrutura sequencial inata, é necessário fornecer informação sobre a ordem dos elementos na sequência. Para isso, é adicionada uma “codificação posicional” (*positional encoding*) aos *embeddings* de entrada. Essa codificação utiliza funções seno e cosseno de diferentes frequências, permitindo ao modelo identificar a posição relativa dos elementos (VASWANI et al., 2017).

### 2.2.4 Vantagens e Desvantagens

Os transformadores revolucionaram tarefas de transdução de sequência, como tradução automática e resumo de texto, ao introduzirem um mecanismo de atenção que permite capturar relações globais entre os elementos de uma sequência, independentemente da distância entre eles. Diferentemente dos modelos baseados em RNNs

Figura 3 – Arquitetura do transformador, composta por múltiplos blocos de codificador (à esquerda) e decodificador (à direita).



Fonte: Adaptado de Vaswani et al. (2017)

e LSTMs, os transformadores são mais paralelizáveis, o que reduz significativamente o tempo de treinamento. Além disso, eles se destacam em tarefas que exigem a modelagem de dependências de longo alcance, uma vez que o mecanismo de autoatenção facilita a captura dessas relações complexas de forma eficiente (VASWANI et al., 2017).

Ao evitar o uso de recorrência, os transformadores são capazes de processar toda a sequência de entrada simultaneamente, ao invés de depender de processamento sequencial como as RNNs. Isso resulta em uma maior eficiência computacional em termos de paralelização e treinamento mais rápido, especialmente em dispositivos com

hardware robusto.

No entanto, apesar dessas vantagens, os transformadores apresentam algumas desvantagens. Um dos principais desafios é o alto custo computacional associado ao mecanismo de autoatenção, cuja complexidade cresce de forma quadrática com o comprimento da sequência. Esse fator gera uma demanda elevada por recursos de memória e poder de processamento, tornando os transformadores menos adequados para tarefas com sequências extremamente longas ou para dispositivos com recursos limitados, como smartphones ou sistemas embarcados.

Uma limitação significativa dos transformadores é a sua dependência de grandes volumes de dados para atingir um desempenho ideal. Sem um pré-treinamento adequado em grandes conjuntos de dados, o modelo pode não alcançar os melhores resultados. Isso dificulta a aplicação dos transformadores em domínios com dados limitados, onde a coleta de grandes quantidades de informação é custosa ou inviável (HENRY; EMEBO; OMONGHINMIN, 2024).

## 2.3 Transformadores Visuais

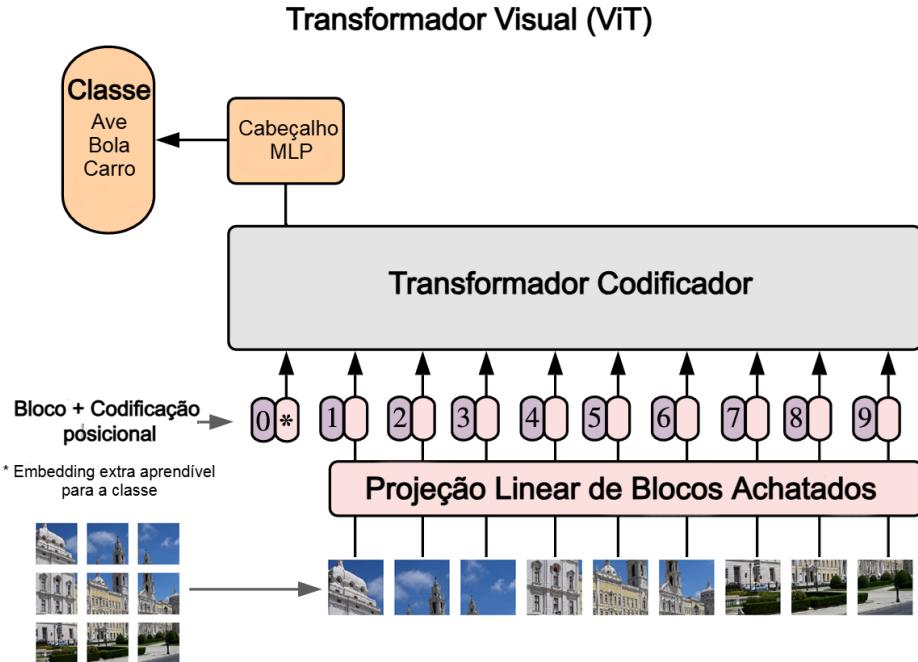
Os transformadores visuais são uma adaptação da arquitetura de transformadores, agora aplicada ao domínio da visão computacional. Ao dividir uma imagem em blocos de tamanho fixo e tratá-los como uma sequência de *tokens*, um transformador pode ser treinado para realizar tarefas de classificação de imagens de maneira eficiente.

O transformador visual divide a imagem de entrada em blocos de tamanho fixo, por exemplo, 16x16 pixels, que são achados e linearmente projetados em vetores de dimensão fixa. Cada bloco, portanto, é tratado como um *token*, similar ao que ocorre no processamento de palavras em modelos de NLP descrito em . Em seguida, esses vetores são alimentados ao transformador, juntamente com um *token* de classificação adicional que serve para gerar a representação da imagem completa ao final do processamento. Esse *token* de classificação é inserido no início da sequência de blocos e, ao final do processo de atenção e camadas de *feed-forward*, é utilizado como a representação final da imagem, que será então passada para uma camada de classificação, esse processo pode ser observado na representação da Figura 4.

Uma codificação posicional também é adicionada a esses *embeddings* de blocos, permitindo que o modelo capture informações sobre a posição espacial dos elementos da imagem, da mesma forma que descrito na subseção 2.2.3.

Assim como nos transformadores convencionais, o transformador visual utiliza um mecanismo de atenção, especificamente o *Multi-Head Self-Attention*, que permite que o modelo integre informações de diferentes partes da imagem de forma simultânea.

Figura 4 – Arquitetura do Transformador Visual.



Fonte: Adaptado de Dosovitskiy et al. (2020)

Cada bloco pode interagir diretamente com todos os outros, possibilitando que o modelo capture relações globais entre os elementos da imagem, sem a limitação de um campo receptivo fixo, como nas CNNs. Essa característica faz com que o transformador visual tenha um potencial expressivo significativo, permitindo que ele aprenda características complexas e globais das imagens.

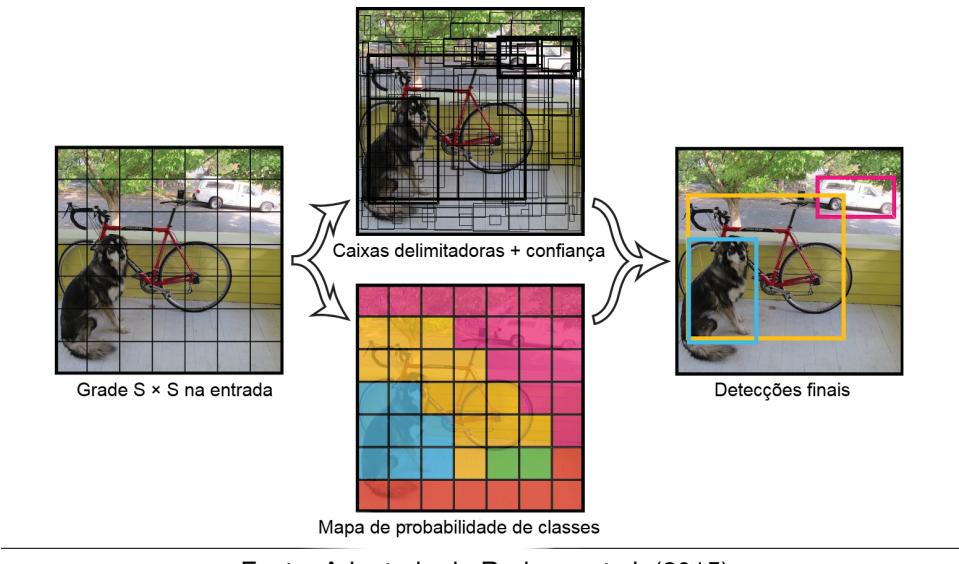
No entanto, para alcançar um bom desempenho, é necessário um pré-treinamento em larga escala, com grandes volumes de dados. Estudos demonstraram que o ViT alcança resultados competitivos com os modelos convolucionais mais avançados quando pré-treinado em grandes conjuntos de dados, como o ImageNet-21k e o JFT-300M. O fato de o último *layer* ser retirado e substituído por uma camada de classificação específica para a tarefa desejada permite que o modelo seja adaptado para diferentes problemas com relativa facilidade, desde que seja realizado um *fine-tuning* adequado (DOSOVITSKIY et al., 2020).

## 2.4 YOLO

O YOLO (You Only Look at Once) é uma das abordagens mais populares para detecção de objetos em visão computacional, introduzido por Redmon et al. (2015), o YOLO reformulou o problema de detecção de objetos como uma tarefa de regressão, em que a rede neural prevê diretamente as caixas delimitadoras e as probabilidades

de classe de uma única vez a partir de uma imagem completa. Esse processo é realizado sem a necessidade de realizar classificações múltiplas em diferentes regiões da imagem, como era feito em abordagens anteriores, tais como a R-CNN e seus derivados (REDMON et al., 2015). A Figura 5 ilustra o processo de detecção de objetos com o YOLO.

Figura 5 – Processo de detecção de objetos com o YOLO. Onde são selecionadas as melhores caixas, resultando nas detecções finais.



Fonte: Adaptado de Redmon et al. (2015)

O YOLO é um sistema unificado que permite o treinamento ponta-a-ponta, facilitando a otimização e trazendo ganhos significativos de velocidade, sendo capaz de processar até 45 quadros por segundo em tempo real. Um dos grandes diferenciais do YOLO é o fato de considerar a imagem inteira durante o treinamento e a inferência, em vez de se concentrar em regiões específicas. Dessa forma, é capaz de raciocinar globalmente sobre a presença dos objetos e a relação entre eles, permitindo uma detecção mais precisa em relação ao contexto da imagem.

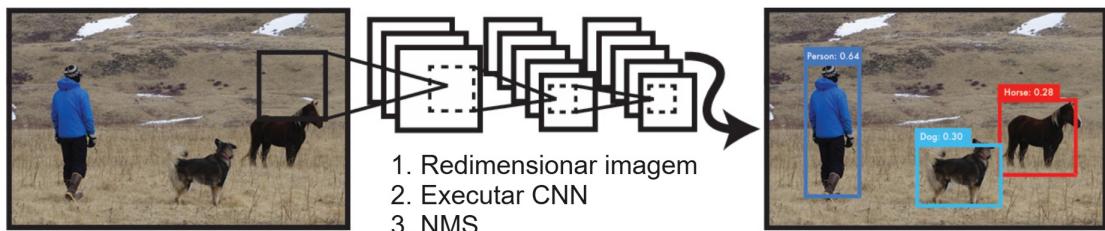
#### 2.4.1 Arquitetura

A arquitetura do YOLO baseia-se em uma rede convolucional profunda, inspirada nas redes utilizadas para classificação de imagens, como a rede GoogLeNet (SZEGEDY et al., 2014). O YOLO divide a imagem de entrada em uma grade de células de  $S \times S$ , onde cada célula da grade é responsável pela detecção dos objetos cujo centro está localizado naquela célula. Cada célula prevê múltiplas caixas delimitadoras (*bounding boxes*) e suas respectivas pontuações de confiança, além das probabilidades de classe condicionais a cada caixa.

A saída do YOLO consiste em um tensor de dimensões  $S \times S \times (B \times 5 + C)$ , onde  $S$  é o número de células da grade,  $B$  é o número de caixas delimitadoras previstas por célula,  $C$  é o número de classes possíveis. O valor 5 refere-se às 4 coordenadas da caixa delimitadora (centro, largura, altura) e a pontuação de confiança.

A pontuação de confiança reflete a precisão da previsão da caixa delimitadora e a probabilidade de que um objeto esteja presente naquela caixa. As células da grade que não contêm o centro de um objeto têm uma pontuação de confiança próxima de zero, enquanto as células que contêm o centro de um objeto fornecem as coordenadas da caixa e as probabilidades de classe correspondentes. A arquitetura do YOLO pode ser observada na Figura 6

Figura 6 – Arquitetura geral do YOLO.



Fonte: Adaptado de Redmon et al. (2015)

#### 2.4.2 Processo de Treinamento e Inferência

Durante o treinamento, o YOLO minimiza uma função de perda que considera tanto o erro nas coordenadas das caixas delimitadoras quanto a classificação incorreta das classes. A função de perda é composta por três termos principais: o erro de regressão das caixas delimitadoras, que penaliza as diferenças entre as caixas previstas e as caixas reais, o erro de confiança, que penaliza a diferença entre a pontuação de confiança prevista e a presença real de um objeto, e o erro de classificação, que penaliza a discrepância entre as classes previstas e as classes reais para os objetos detectados.

O modelo utiliza a técnica NMS (*Non-Maximum Suppression*) durante a inferência para filtrar previsões redundantes, garantindo que apenas as caixas com maior confiança sejam mantidas. O funcionamento da NMS ocorre em três etapas principais. Primeiramente, a caixa com a maior pontuação de confiança é selecionada, sendo aquela que o modelo considera mais precisa tanto em relação à posição quanto à presença do objeto. Em seguida, são avaliadas todas as outras caixas que se sobreponhem a essa primeira, utilizando o critério de IoU (*Intersection over Union*), que mede a área de interseção entre duas caixas, e se o valor de sobreposição entre elas for maior do

que um limite predefinido, a caixa com menor confiança é descartada. Por fim, esse processo é repetido até que todas as caixas redundantes sejam removidas, mantendo apenas aquelas com alta confiança e baixa sobreposição (SUBRAMANYAM, 2021).

### 2.4.3 Vantagens e Desvantagens

O modelo YOLO possui diversas vantagens em relação a métodos anteriores de detecção de objetos. Uma de suas principais vantagens é a velocidade. A abordagem unificada permite que a detecção seja realizada em uma única etapa, o que reduz significativamente o tempo necessário para processar cada imagem. Além disso, o modelo oferece uma detecção global, considerando a imagem inteira em vez de focar apenas em regiões específicas. Isso possibilita uma compreensão mais ampla do contexto da imagem, o que contribui para a redução de falsas detecções, especialmente em situações onde objetos diferentes estão próximos uns dos outros. Outra vantagem importante é a simplicidade. O treinamento ponta-a-ponta facilita tanto a implementação quanto a otimização do modelo, eliminando a necessidade de múltiplos estágios de processamento, como ocorre em abordagens que utilizam propostas de regiões.

Entretanto, o YOLO também apresenta algumas limitações. Um dos desafios é a dificuldade com objetos pequenos. Como a imagem é dividida em uma grade relativamente grande, pode acontecer de objetos pequenos serem ignorados ou não detectados corretamente, já que a célula da grade responsável pela detecção pode não capturar adequadamente suas características. Além disso, o modelo apresenta limitações de precisão. Em situações onde é necessária uma detecção de alta acurácia, sua precisão pode ser inferior à de métodos como o Faster R-CNN (REN et al., 2016). Isso ocorre devido à natureza do compromisso que o modelo faz entre velocidade e precisão, priorizando a eficiência em tempo real.

### 2.4.4 Evoluções

Desde a introdução do YOLO original, várias versões e melhorias foram propostas, incluindo o YOLOv2, YOLOv3, e versões mais recentes. Cada versão trouxe otimizações que aumentaram a precisão e a capacidade de generalização, mantendo a eficiência em tempo real.

O YOLOv8 foi treinado em um conjunto de dados maior e mais diversificado do que as versões anteriores, incluindo uma combinação do dataset COCO e outros conjuntos de dados, resultando em um desempenho superior em uma variedade de imagens. Ele também inclui uma ferramenta de anotação chamada *RoboFlow Annotate*, que facilita o processo de anotação, incluindo recursos como anotação automática e atalhos personalizáveis.

Essas melhorias fazem com que o YOLOv8 alcance uma precisão média superior e um desempenho geral melhor em relação a outras versões, especialmente na detecção de objetos pequenos ou em condições desafiadoras, como aqueles que se misturam ao fundo ou que apresentam alta variância de escala e rotação (REIS et al., 2024).

# 3 Material e Métodos

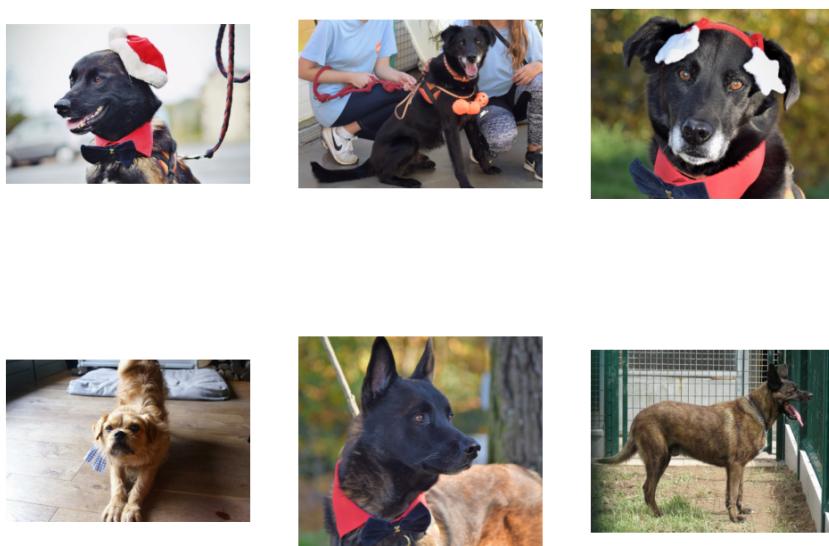
Este capítulo apresenta o conjunto de imagens utilizado nos experimentos realizados para avaliar os métodos de segmentação de faces (YOLO) e de extração de características faciais (ViT) para a identificação biométrica facial de cães. Apresenta também as métricas adotadas neste trabalho para avaliação dos resultados, bem como o ambiente computacional utilizado nos experimentos.

## 3.1 Base de Dados

Neste trabalho, foi utilizada a base de dados DogFaceNet, proposta por Mouceot, Li e Jia (2019), que consiste em um conjunto de dados desenvolvido com o objetivo de fornecer uma base robusta para pesquisas de identificação biométrica de cães.

A base de dados DogFaceNet é composta por 8.363 imagens, distribuídas em 1.393 classes, onde cada classe representa um cão individual e possui, no mínimo, duas imagens. Para assegurar uma cobertura adequada, as imagens foram coletadas de diferentes fontes da internet, abrangendo uma diversidade significativa de raças, ângulos e variações de captura. Essa diversidade de imagens é fundamental para garantir a robustez dos modelos empregados para a identificação dos indivíduos, especialmente considerando-se arquiteturas baseadas em transformadores visuais, como a ViT apresentada na Seção 2.3. A Figura 7 apresenta amostras de imagens de cães da base de dados DogFaceNet.

Figura 7 – Amostras da base de dados DogFaceNet (MOUGEOT; LI; JIA, 2019).



Fonte: Elaborada pelo autor

### 3.1.1 Data Augmentation

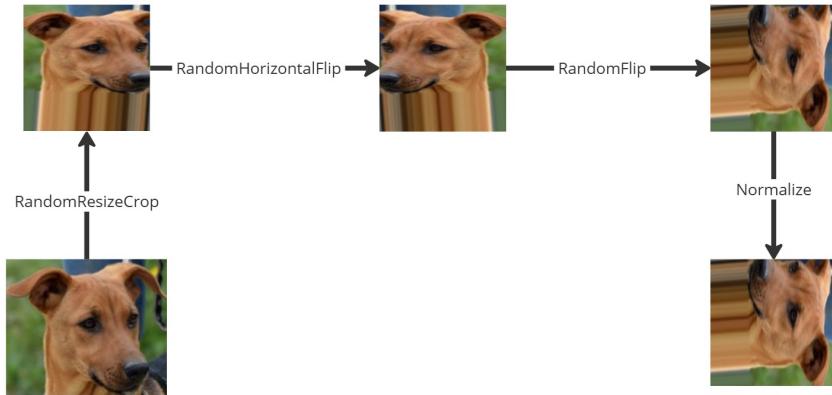
A eficácia dos métodos atuais de aprendizado de máquina, como os baseados em transformadores, depende da utilização de grandes volumes de dados rotulados de treinamento e que apresentam variações que possibilitam capturar as relações existentes entre os elementos da imagem e também permitam lidar com diferentes cenários de captura. Fatores como variações de iluminação, ângulos não uniformes e o uso de diferentes sensores — comuns em dispositivos móveis — são desafios que os modelos precisam superar. Ao incluir essa gama de variações nos conjuntos de dados de treinamento, os modelos podem aprender a generalizar melhor, tornando-se capazes de realizar uma identificação biométrica precisa independentemente das condições do ambiente ou do dispositivo utilizado para a captura.

O treinamento robusto e diverso é essencial para o sucesso do reconhecimento facial de cães em um contexto real de uso. Ainda que o conjunto de dados DogFaceNet seja relativamente grande, neste trabalho foi utilizada uma estratégia de *data augmentation* com o objetivo de aumentar a diversidade das amostras deste conjunto de dados durante o treinamento. O emprego de *data augmentation* permite que o modelo ViT aprenda a generalizar, aumentando sua capacidade de lidar com variações nas imagens que podem ocorrer no mundo real, como mudanças de ângulo, iluminação e dispositivos de captura.

Observa-se que a estratégia de *data augmentation* foi aplicada apenas no subconjunto de dados de treinamento do modelo ViT, composto por imagens faciais dos cães, conjunto este obtido com a segmentação das imagens da base de dados DogFaceNet utilizando o modelo YOLO, uma vez que o modelo YOLO obteve êxito no processo de detecção das faces dos cães mesmo sem o uso de *data augmentation*.

As transformações de *data augmentation* foram realizadas utilizando-se a técnica *RandomResizedCrop*, na qual a imagem é recortada aleatoriamente em diferentes tamanhos e proporções, seguida por redimensionamento para um tamanho fixo. A *RandomHorizontalFlip* espelha horizontalmente as imagens com uma certa probabilidade, permitindo que o modelo se torne mais robusto a diferentes orientações. A *RandomRotation* aplica rotações aleatórias nas imagens dentro de um intervalo definido, garantindo que o modelo seja capaz de reconhecer os objetos independentemente do ângulo de rotação. Por fim, a *Normalize* realiza a normalização da imagem utilizando a média e o desvio padrão previamente calculados para o dataset, assegurando que as características dos dados permaneçam em uma faixa de valores que facilite o aprendizado do modelo. A Figura 8 mostra o fluxo das transformações realizadas nas imagens faciais dos cães, do conjunto de treinamento, por meio da técnica *RandomResizedCrop*.

Figura 8 – Fluxo de transformações realizadas pela técnica *RandomResizedCrop* no processo de *data augmentation* nas imagens faciais dos cães do conjunto de treinamento.



Fonte: Elaborada pelo autor

### 3.2 Métricas de Avaliação

Para avaliar o desempenho do modelo ViT para a identificação biométrica dos cães, foram utilizadas as seguintes métricas, que são consolidadas e amplamente utilizadas na área de Reconhecimento de Padrões:

**Precision:** A precision (precisão) mede a proporção de previsões corretas entre todas as previsões positivas feitas pelo modelo, como observado na equação 3.1:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.1)$$

onde  $TP$  é o número de verdadeiros positivos e  $FP$  o número de falsos positivos.

**Recall:** O recall (revocação), também conhecido como sensibilidade, mede a capacidade do modelo de identificar corretamente as instâncias positivas, como observado na equação 3.2:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.2)$$

onde  $TP$  é o número de verdadeiros positivos e  $FN$  o número de falsos negativos.

**Area Under the Curve (AUC):** A métrica AUC mede a área sob a curva ROC, que é a relação entre a taxa de verdadeiros positivos (TPR) e a taxa de falsos positivos (FPR). A Equação da AUC pode ser observada apartir da equação 3.3.

$$\text{AUC} = \int_0^1 \text{TPR}(x) d\text{FPR}(x) \quad (3.3)$$

**F1-Score:** O F1-Score é a média harmônica entre *Precision* e *Recall*, proporcionando um balanço entre as duas métricas. Esta métrica é descrita pela equação 3.4.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.4)$$

### 3.3 Ambiente Utilizado

Para o desenvolvimento dos experimentos e treinamento do modelo ViT foi utilizado o ambiente Google Colab, uma plataforma amplamente utilizada para pesquisas em aprendizado de máquina devido à sua capacidade de processamento acelerado e facilidade de acesso a recursos de hardware.

No contexto deste trabalho, foi utilizado um ambiente de execução com as seguintes especificações:

- **GPU:** NVIDIA A100, com 40 GB de memória VRAM
- **Memória RAM:** 83 GB
- **Armazenamento:** 500 GB
- **Processador:** CPU Intel Xeon

O Colab permitiu a realização de experimentos para treinar o modelo baseado em transformadores visuais (ViT). Além disso, o acesso à memória expandida e ao armazenamento foi essencial para lidar com as exigências dos transformadores.

# 4 Resultados Experimentais

Este capítulo apresenta os resultados obtidos com os experimentos realizados com a aplicação do método YOLO, utilizado para segmentação das imagens (detecção das faces) da base de dados DogFaceNet, e a posterior identificação biométrica dos cães, por meio do reconhecimento facial, realizada pelo modelo ViT, baseado em transformadores.

## 4.1 Treinamento do modelo YOLO

O modelo YOLO v9 foi utilizado para realizar a segmentação das imagens de cães, com o objetivo de identificar a cabeça dos animais, etapa que precede a extração das características visuais. A Tabela 1 apresenta as métricas de desempenho do modelo, onde observa-se que a precisão alcançou 83%, indicando que a maioria das detecções realizadas foram corretas. O valor de *recall*, que reflete a capacidade do modelo de identificar corretamente os objetos relevantes, foi de aproximadamente 72%. Essas métricas, somadas ao *F1-Score* e à AUC, sugerem que, apesar de a segmentação ser satisfatória, o modelo ainda pode ser otimizado para melhorar a detecção de objetos em cenários mais complexos.

Métrica	Valor
Acurácia	0,7610
AUC	0,7871
Precisão	0,8314
Recall	0,7239
F1-Score	0,7746

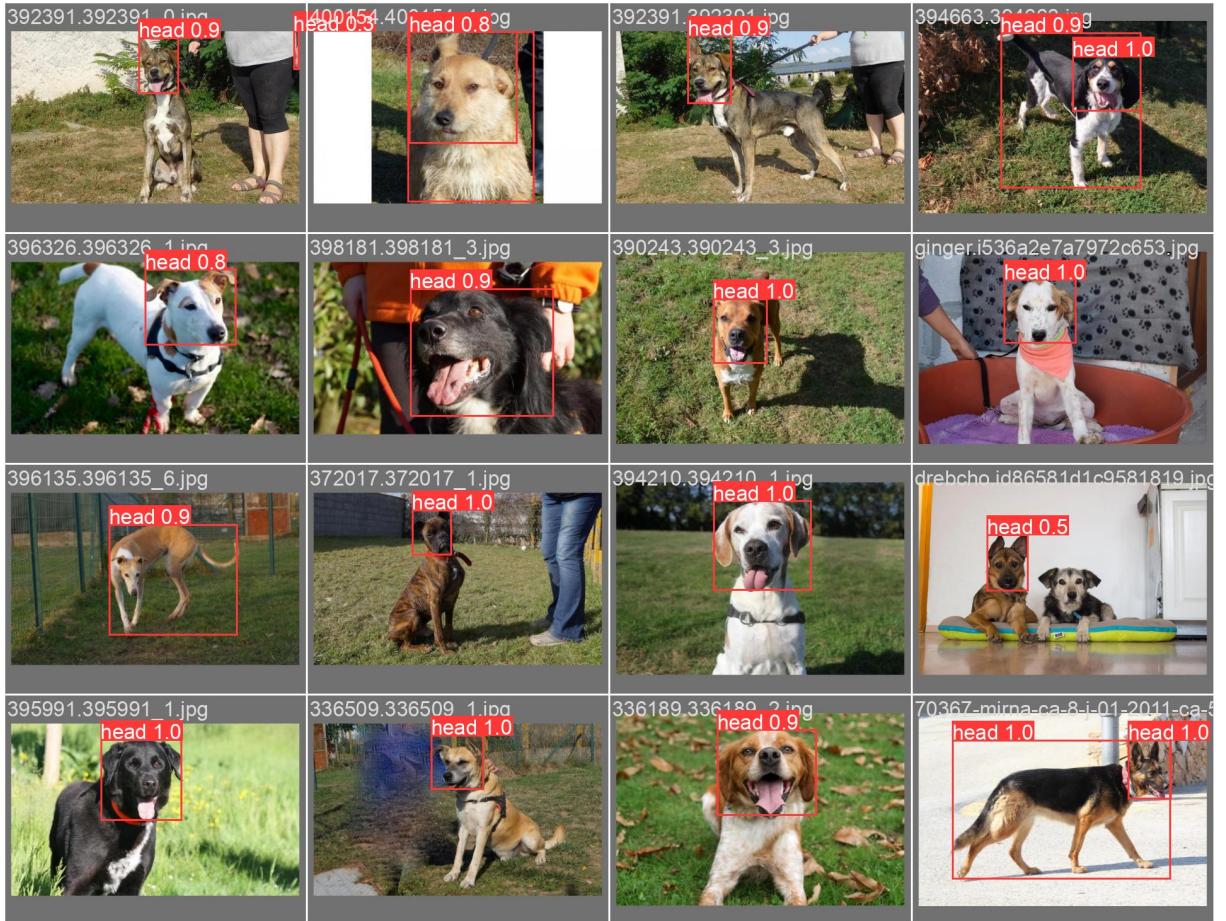
Fonte: Elaborada pelo autor.

Quadro 1 – Resultados obtidos pelo modelo YOLO para segmentação das imagens dos cães.

A Figura 1 ilustra inferências realizadas pelo modelo YOLO no conjunto de teste, demonstrando sua habilidade em segmentar corretamente a cabeça dos cães nas imagens.

Esses resultados indicam que o modelo YOLO apresentou uma segmentação eficaz nas imagens de cães, mas melhorias adicionais podem ser realizadas, especialmente no aumento da taxa de *recall*, que permitiria uma detecção mais abrangente dos objetos presentes nas imagens.

Tabela 1 – Inferências realizadas em indivíduos do conjunto de teste da base de dados DogFaceNet, utilizando o modelo YOLO.



Fonte: Elaborada pelo autor

## 4.2 Modelo Transformadores Visuais

Durante os experimentos, foi utilizado o modelo base *vit-base-patch16-224 da Google*<sup>1</sup>, o *primeiro modelo de transformadores visuais (ViT)*, que foi proposto por Dosovitskiy et al. (2020). Este modelo foi pré-treinado no conjunto de dados ImageNet-21k, que contém 14 milhões de imagens de 21.843 classes, com resolução de  $224 \times 224$ , e ajustado no conjunto de dados ImageNet 2012, que contém 1 milhão de imagens de 1.000 classes, também com resolução de  $224 \times 224$ .

Este modelo de ViT, pré-treinado em um conjunto de dados genérico, como já mencionado, foi submetido a um processo de *fine-tuning* no conjunto de dados DogFaceNet, descrito na Seção 3.1, visando a transferência de aprendizagem para o domínio de identificação facial de cães.

Neste novo treinamento, realizado em 100 épocas, foi aplicado o critério de parada antecipada (*early stopping*) baseado no monitoramento da métrica de F1-Score

<sup>1</sup> <https://huggingface.co/google/vit-base-patch16-224>

no conjunto de validação. Com este critério, o treinamento foi interrompido na época 97, tendo em vista que a métrica de F1-Score parou de melhorar significativamente nas últimas 10 épocas consecutivas, indicando que o modelo atingiu um ponto de estagnação.

Como pode ser observado na Figura 9, os gráficos apresentam os principais resultados do modelo após o treinamento, incluindo acurácia, AUC, F1-Score, precisão e *recall*. Apesar de o modelo ter mostrado bons resultados, o conjunto de dados relativamente pequeno pode ter impactado a capacidade do modelo de aprender de maneira mais robusta e alcançar métricas superiores.

O treinamento foi realizado com alguns hiperparâmetros ajustados para maximizar o desempenho do modelo. O *learning rate* foi definido como  $1 \times 10^{-5}$ , permitindo uma adaptação suave ao novo conjunto de dados, tendo sido utilizada uma razão de *warmup* de 10%, para garantir que a taxa de aprendizado aumentasse gradualmente no início do treinamento.

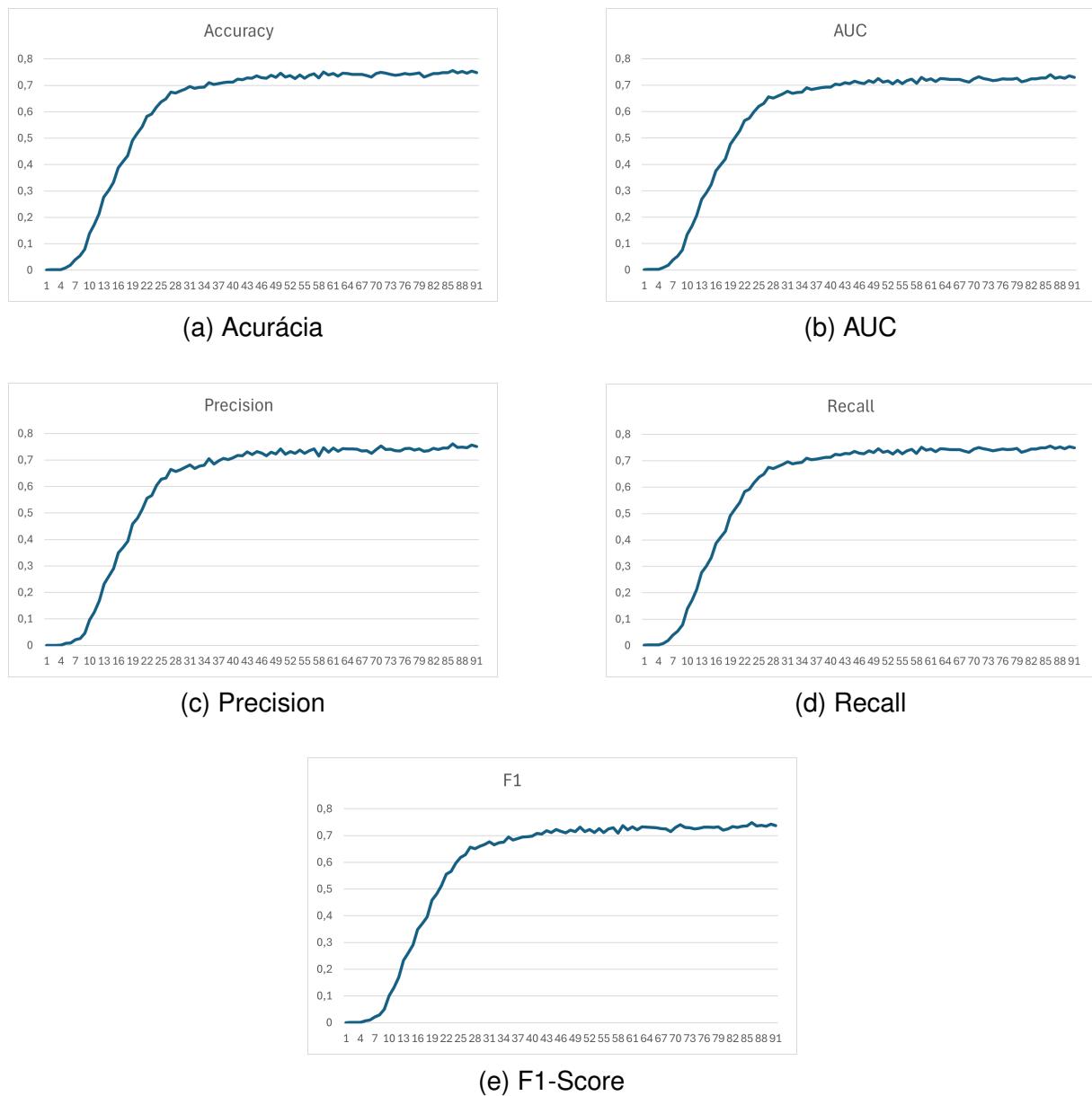
Os valores da melhor época estão presentes na Tabela 2. Esses valores mostram que o modelo mantém um desempenho consistente entre as épocas, contudo o valor é aquém do esperado para transformadores visuais.

Métrica	Valor
Acurácia	0,7610
AUC	0,78
Precisão	0,7114
Recall	0,7111
F1-Score	0,7112

Quadro 2 – Resultados do modelo de transformadores visuais para extração de características.

Apesar de aceitáveis, esses resultados evidenciam que há espaço para melhorias, especialmente no que diz respeito à quantidade e diversidade dos dados usados para o treinamento. Um conjunto de dados mais extenso e balanceado poderia contribuir para uma melhor generalização do modelo, resultando em um aumento das métricas, como acurácia, F1-Score e AUC.

Figura 9 – Gráficos das métricas de avaliação: (a) Acurácia, (b) AUC, (c) Precision, (d) Recall, (e) F1-Score.



Fonte: Elaborada pelo autor

# 5 Aplicação

Este Capítulo apresenta o aplicativo para dispositivos móveis desenvolvido neste trabalho para a identificação biométrica facial de cães, utilizando transformadores.

## 5.1 Introdução

Este trabalho visa, além da avaliação experimental do método YOLO, para a detecção das faces dos cães em imagens digitais, e do modelo ViT, para a extração de características faciais dos cães, a construção de um aplicativo para dispositivos móveis que seja capaz de permitir os seus usos de forma mais prática e simples para a identificação de cães em situações reais do cotidiano.

O aplicativo foi projetado para rodar em qualquer dispositivo móvel, sendo o processamento das imagens realizado em ambiente de nuvem, proporcionando acesso remoto e simplificado a partir de qualquer dispositivo conectado à internet. Essa abordagem oferece maior flexibilidade e escalabilidade, permitindo que os usuários accessem suas funcionalidades de forma eficiente, sem a necessidade de manutenção local ou instalações complexas. Com foco em processamento de imagens, a aplicação integra modelos de aprendizado profundo para segmentação e análise de imagens, garantindo alta performance e eficiência no processamento intensivo de dados.

O objetivo da aplicação desenvolvida é permitir que os usuários capturem e enviem imagens diretamente para um servidor central, onde serão processadas por meio de modelos avançados de visão computacional. O fluxo começa com o envio de uma imagem pelo aplicativo, que será automaticamente segmentada e analisada no servidor, que, por sua vez, retorna ao aplicativo a identidade do cão, conforme ilustra a Figura 10.

## 5.2 Estrutura da aplicação

### 5.2.1 Servidor

O servidor da aplicação foi projetado para rodar em nuvem, aproveitando componentes como Nginx<sup>1</sup>, Gunicorn<sup>2</sup> e Flask<sup>3</sup> para garantir um processamento eficiente e seguro das requisições. O Nginx é responsável por atuar como um servidor intermediá-

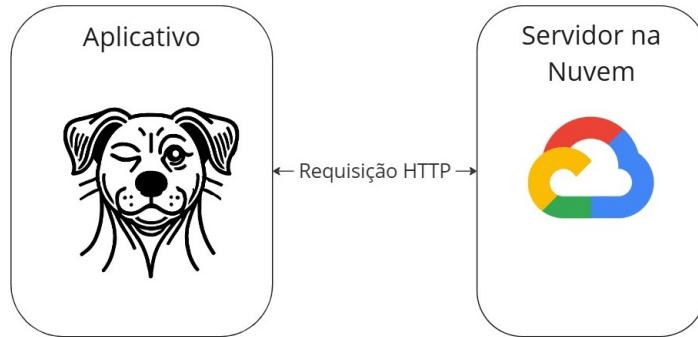
---

<sup>1</sup> <https://nginx.org/en/>

<sup>2</sup> <https://gunicorn.org/>

<sup>3</sup> <https://flask.palletsprojects.com/en/stable/>

Figura 10 – Visão Geral da Aplicação.

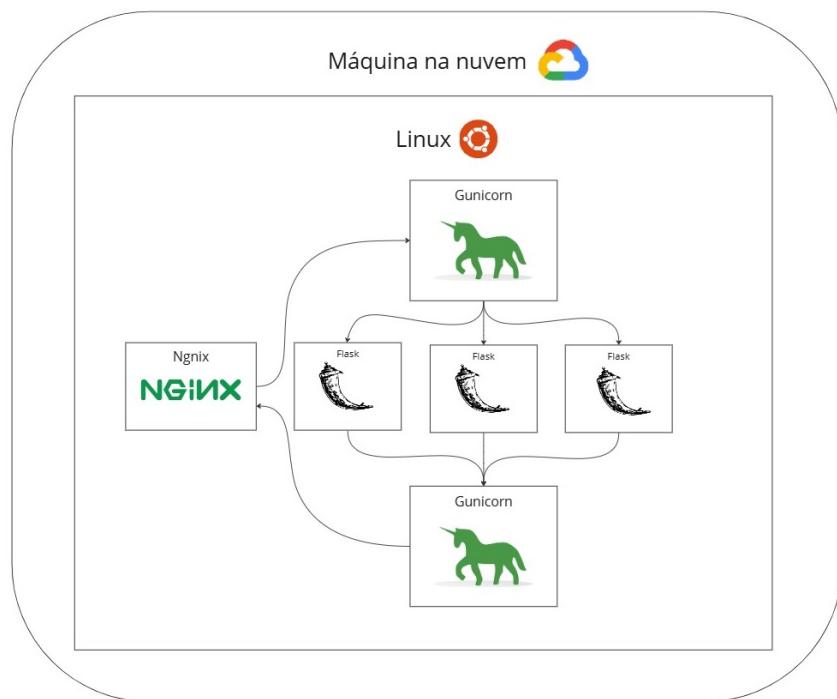


Fonte: Elaborada pelo autor.

rio, recebendo e redirecionando as requisições para a aplicação interna. O Gunicorn gerencia os processos do Flask, que é o microframework responsável pela lógica de negócios e processamento das requisições.

Quando um usuário envia uma imagem por meio do aplicativo, a requisição HTTP é direcionada inicialmente para o Nginx, que a recebe na porta 80. O Nginx verifica e encaminha essa requisição para o Gunicorn, que, por sua vez, distribui as requisições para um dos processos Flask disponíveis, conforme mostra o diagrama da Figura 11. O Flask interpreta a rota acessada, como /registrar ou /identificar, e prepara a requisição para processamento, convertendo a imagem enviada e validando sua extensão.

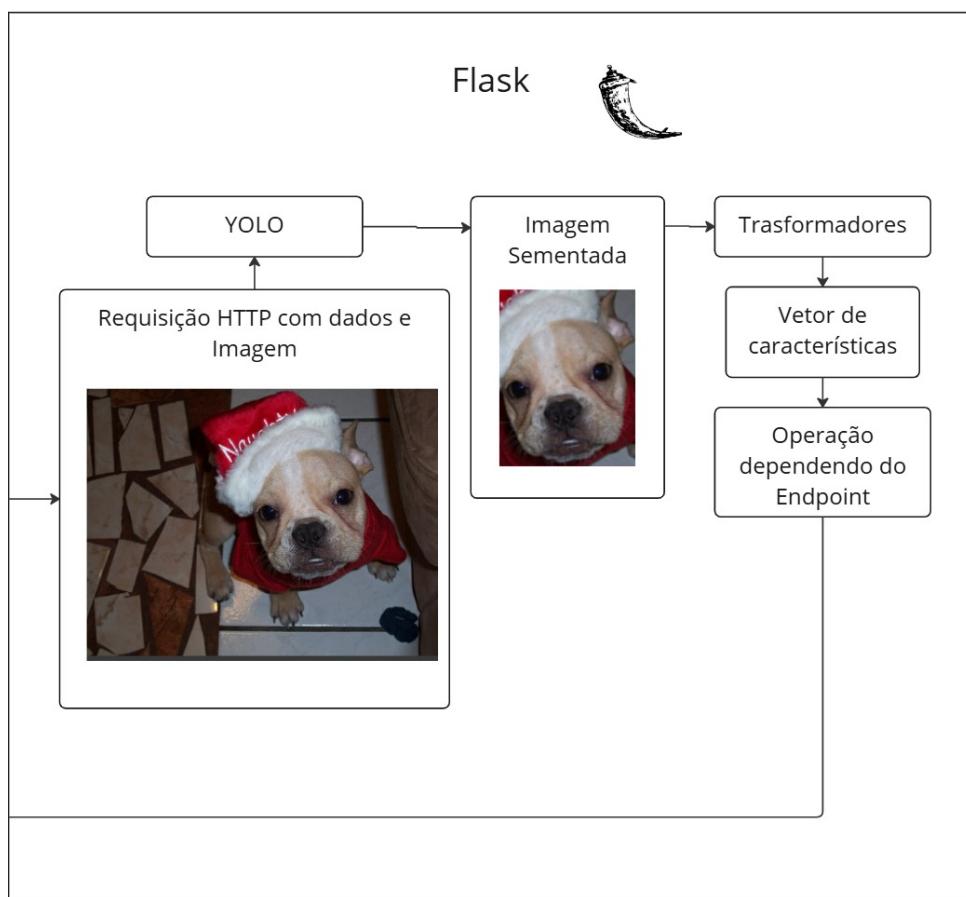
Figura 11 – Estrutura do servidor na nuvem.



Fonte: Elaborada pelo autor.

Após essa etapa, a imagem é passada para o modelo YOLO v9, que realiza a segmentação, verificando se há a presença de objetos de interesse, como a face de um cachorro. Em caso afirmativo, a imagem é recortada e enviada para o modelo de transformadores visuais, que extrai um vetor de características, representando a informação visual relevante da imagem. Caso nenhum cachorro seja detectado, o servidor retorna uma mensagem de erro informando essa situação. A Figura12 mostra um diagrama das etapas do processamento no Flask.

Figura 12 – Etapas do processamento no Flask.



Fonte: Elaborada pelo autor.

No caso da rota /register, o vetor extraído é salvo junto com um identificador único (UUID) e o nome fornecido pelo usuário. Já na rota /identify, o vetor é comparado com outros previamente armazenados, utilizando a distância euclidiana para identificar o registro mais próximo. Se houver correspondência, a aplicação retorna o ID e nome do registro encontrado.

A resposta JSON<sup>4</sup> gerada pelo Flask é então encaminhada de volta pelo Gunicorn e enviada ao Nginx, que a devolve ao cliente por meio de HTTP. Durante todo

<sup>4</sup> <https://www.json.org/json-en.html>

o processo, *logs* são mantidos tanto pelo Nginx quanto pelo Gunicorn, garantindo a rastreabilidade e o monitoramento contínuo das requisições.

### 5.2.2 Aplicativo

O aplicativo foi desenvolvido utilizando *Flutter*<sup>5</sup>, uma tecnologia que permite a criação de interfaces nativas para Android e iOS a partir de um único código-fonte. O objetivo principal é proporcionar uma experiência fluida e intuitiva ao usuário, com foco na captura e envio de imagens para identificação, além de oferecer uma navegação simples entre diferentes funcionalidades.

O fluxo de interação envolve a captura de uma imagem por meio de uma interface integrada ao aplicativo. Após a captura, a imagem é enviada para o servidor por meio de uma requisição HTTP para [www.caoapi.com](http://www.caoapi.com), onde é processada e retornada ao aplicativo. A interface foi projetada para apresentar as informações obtidas de forma clara e objetiva, exibindo o resultado da análise do servidor diretamente ao usuário.

As telas foram organizadas de forma modular, permitindo que cada uma desempenhe uma função específica. Assim, o usuário é guiado por uma sequência de interações simples e diretas, desde a captura da imagem até a exibição das informações processadas. As principais funcionalidades envolvem o envio e a identificação de imagens, o gerenciamento do acesso por meio de *login* e cadastro, e o fornecimento de configurações básicas, como a opção de sair do aplicativo.

Os dados recebidos do servidor são apresentados ao usuário em *layouts* que destacam a informação visual e os metadados associados, como data, horário e, eventualmente, localização. Por exemplo, ao capturar e enviar uma imagem de um cachorro, o usuário recebe como resposta uma análise que inclui informações relevantes sobre o animal. Caso não seja possível identificar o objeto na imagem, o aplicativo informa o usuário imediatamente sobre a falha na identificação.

Para garantir uma experiência fluida, o aplicativo utiliza indicadores visuais para mostrar o progresso de operações que demandam mais tempo, como o envio e processamento de imagens. A navegação entre as funcionalidades é feita por meio de rotas internas, facilitando a transição rápida entre as diferentes seções do aplicativo.

O uso de *Flutter* permitiu a criação de um aplicativo eficiente e responsivo, garantindo uma integração direta com funcionalidades nativas do dispositivo, como a câmera. A comunicação com o servidor ocorre de forma fluida por meio de requisições HTTP, garantindo que o usuário tenha uma experiência completa e integrada desde a captura da imagem até a exibição dos resultados processados.

---

<sup>5</sup> <https://flutter.dev/>

## 5.3 Funcionamento da Aplicação

A aplicação projetada e implementada neste trabalho é composta pelo servidor e pelo aplicativo móvel, oferecendo uma solução prática para captura, envio e análise de imagens visando o cadastro, a identificação e a autenticação de cães por meio da biometria facial.

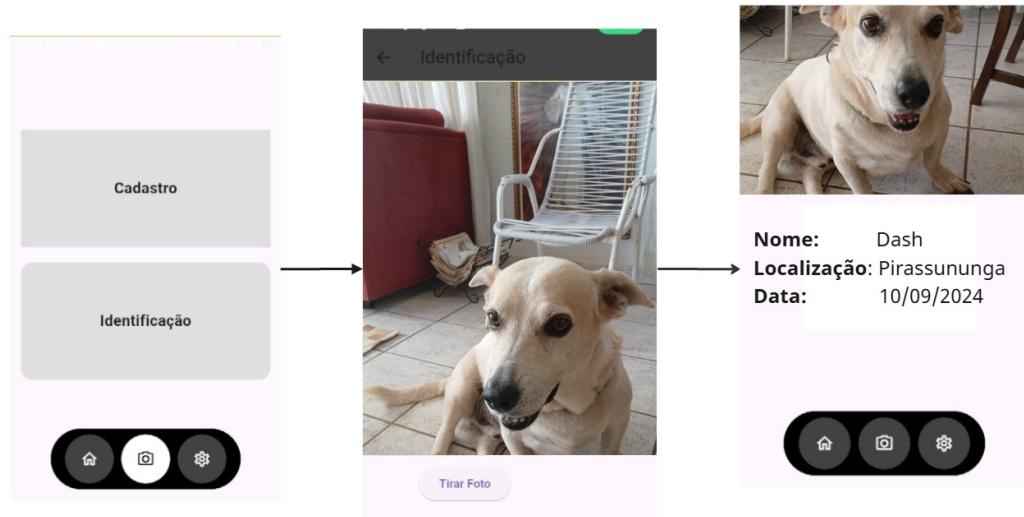
O cadastro, identificação e autenticação de cães são três funcionalidades principais da aplicação. As duas primeiras são acessadas na tela principal do aplicativo, onde estão disponíveis dois botões: um para cadastro e outro para identificação. A autenticação, por sua vez, é realizada ao selecionar um cão já cadastrado e, em seguida, o ícone da câmera fotográfica.

A Figura 13 exemplifica o processo de identificação de um cão. Neste caso, após selecionar a opção Identificação, o usuário deve capturar uma fotografia do animal usando a câmera fotográfica do dispositivo móvel. A imagem é enviada ao servidor, que a processa e, caso o animal seja encontrado nos registros armazenados da aplicação, a aplicação retorna ao usuário: a identidade do animal, a imagem armazenada que mais se assemelha à imagem de busca, a localização e a data do cadastro daquele animal. A identidade do animal é determinada pela menor distância Euclidiana entre a imagem de consulta e todas as imagens de animais armazenadas nos registros da aplicação. Caso a menor distância obtida entre todas as comparações seja maior do que um limiar pré-estabelecido, o usuário é informado que o animal em questão não foi cadastrado no sistema e, portanto, não pode ser identificado.

A autenticação, permite verificar se um novo registro de imagem pertence ao mesmo indivíduo que um registro existente. Nesse modo, se a menor distância Euclidiana entre o vetor de características da nova imagem e o vetor armazenado estiver abaixo de um limiar específico, o sistema considera que as imagens representam o mesmo animal, confirmando sua identidade ao usuário. Caso contrário, ele é considerado como um indivíduo diferente, mesmo que pertença a uma classe semelhante.

Caso o usuário queira saber quais são todos os animais cadastrados no sistema, basta acessar a Tela de Cadastros, que permite visualizar todos os cadastros realizados pelo usuário, com as imagens e algumas informações obtidas no momento do cadastro. A Figura 14 mostra a utilização desta funcionalidade do aplicativo.

Figura 13 – Exemplo de uso do aplicativo. Funcionalidade: Identificação de um animal.



Fonte: Elaborada pelo autor.

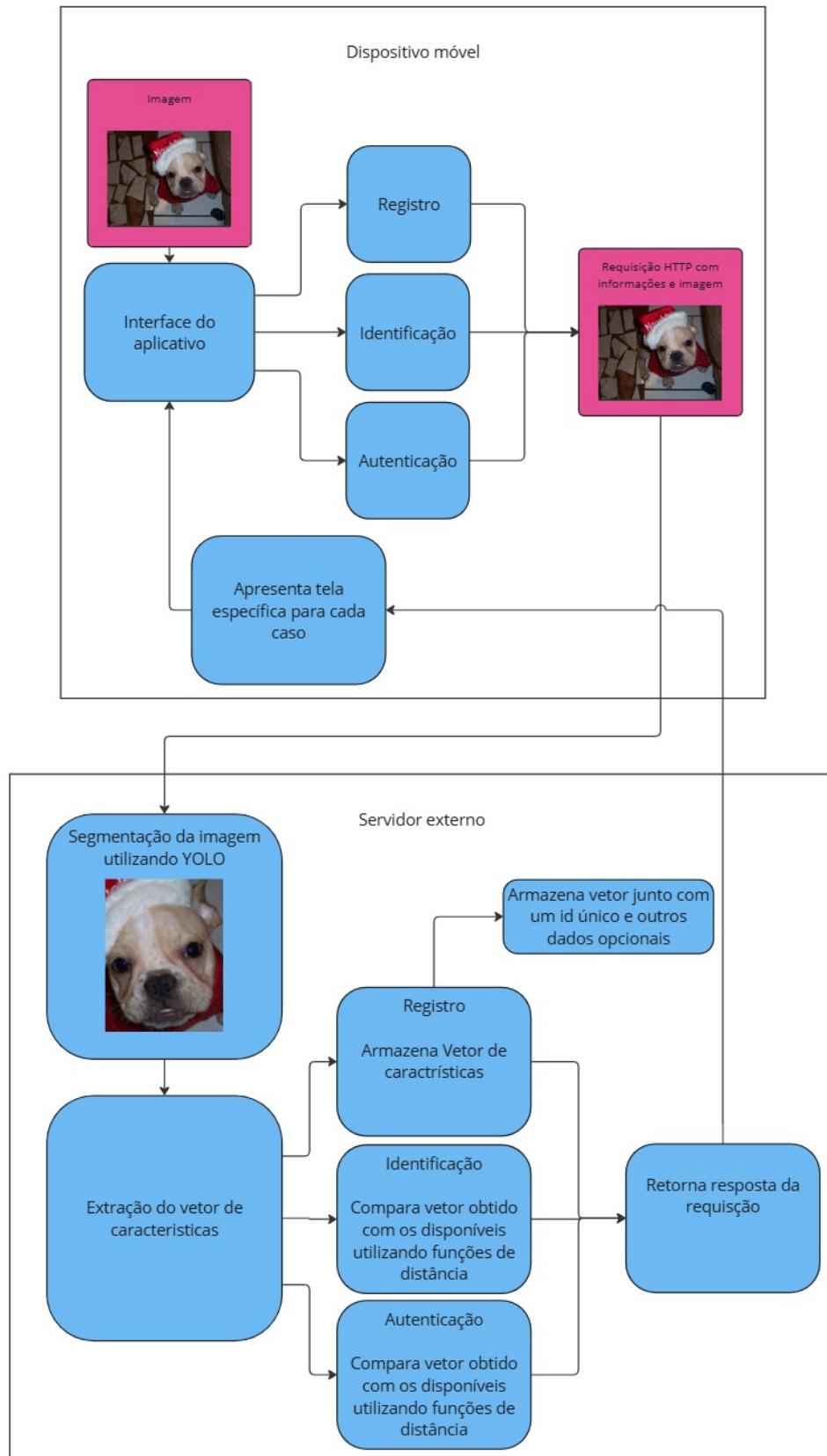
Figura 14 – Exemplo de uso do aplicativo. Funcionalidade: Exibição de todos os animais cadastrados no sistema.



Fonte: Elaborada pelo autor.

O diagrama do fluxo de funcionamento completo da aplicação é apresentado na Figura 15. O processo se inicia no aplicativo, que envia a imagem de entrada para processamento no servidor, via requisição HTTP. No servidor, a imagem é segmentada utilizando o modelo YOLO e, caso seja detectado o objeto de interesse (a face do cão), é extraída da imagem segmentada o vetor de características que representa a face, via modelo de Transformadores visuais (ViT). O vetor de características é então utilizado para cadastro de um novo animal, ou para busca do animal nos registros armazenados na aplicação, neste caso, comparando-se o vetor de características da imagem segmentada com todos os vetores de características faciais dos animais cadastrados no sistema, retornando, ao final, o resultado ao aplicativo.

Figura 15 – Diagrama do funcionamento da aplicação.



Fonte: Elaborada pelo autor.

# 6 Considerações Finais

Este trabalho objetivou a construção de uma aplicação para identificação de cães, por meio de biometria facial e o uso de transformadores visuais, combinando um aplicativo móvel e processamento em nuvem.

Embora a aplicação tenha sido completamente implementada e finalizada, e suas funcionalidades tenham apresentado êxito na execução, os resultados experimentais obtidos na identificação de cães, utilizando-se a base de dados DogFaceNet e transformadores visuais para a obtenção de vetores de características faciais, não atingiram totalmente as expectativas em relação à precisão e generalização dos modelos quando comparados aos resultados obtidos por outros trabalhos encontrados na literatura. Fatores como o tamanho limitado do conjunto de dados e a complexidade das tarefas de visão computacional podem ter impactado diretamente o desempenho dos modelos, resultando em uma menor eficácia.

## 6.1 Aplicativo Móvel

O aplicativo móvel desenvolvido neste trabalho é uma solução escalável e de fácil manutenção, graças à sua implementação em um servidor remoto. A abordagem baseada em nuvem permite que o sistema seja facilmente ampliado para atender a uma demanda maior de usuários sem sobrecarregar o dispositivo do usuário final. No entanto, não foram implementadas técnicas robustas de segurança de dados, o que poderia ser considerado em versões futuras do aplicativo, visando garantir a privacidade e segurança das imagens e informações dos usuários.

Além disso, a ausência de um banco de dados para armazenar informações de forma estruturada foi uma escolha intencional para simplificar a arquitetura do sistema. Embora um banco de dados pudesse adicionar funcionalidades importantes, como o histórico de identificações e a rastreabilidade dos registros, a sua implementação aumentaria significativamente a complexidade do sistema. A inclusão de um banco de dados, aliado a técnicas de segurança e criptografia, poderia fortalecer ainda mais a solução proposta em termos de escalabilidade e confiabilidade.

## 6.2 Modelos

Apesar de terem sido utilizados modelos de aprendizado profundo, considerados estado-da-arte, a eficácia final não foi a esperada. A pequena quantidade de dados

disponíveis para o treinamento foi um dos principais limitadores, afetando negativamente a capacidade dos modelos de realizarem detecções precisas e generalizar para novos casos. Além disso, os resultados sugerem que a otimização do processo de segmentação e a extração de características poderiam ser melhorados com ajustes finos nos hiperparâmetros e maior diversidade nos dados de treino.

É importante destacar que, durante os testes realizados com o aplicativo, o modelo se mostrou suficientemente eficaz para as finalidades propostas, como catalogar e registrar uma quantidade pequena de cães. O sistema conseguiu identificar e gerar vetores de características para cães de maneira satisfatória. Contudo, ao considerar cenários mais complexos, como a presença de diversos cães cadastrados, o modelo estaria mais suscetível a falhas de reconhecimento principalmente no que se diz ao comparar diferentes vetores de características.

Portanto, embora o modelo atenda às necessidades para um uso mais básico, é provável que em aplicações de larga escala ou em cenários com múltiplas entidades, a performance sofra degradação. Futuros desenvolvimentos poderiam explorar conjuntos de dados mais amplos e diversificados, além de técnicas de *data augmentation* mais avançadas, para melhorar o desempenho geral do sistema e a robustez em situações mais desafiadoras.

Em resumo, este trabalho conseguiu demonstrar a viabilidade de um sistema de reconhecimento facial de cães, integrando um aplicativo móvel com um servidor em nuvem. Embora os resultados obtidos não tenham sido totalmente satisfatórios, as bases para futuras melhorias foram estabelecidas. Com ajustes na coleta de dados, aprimoramento dos modelos, o sistema pode se tornar uma ferramenta poderosa e escalável para aplicações práticas em reconhecimento facial de cães e até outros animais domésticos.

### 6.3 Trabalhos Futuros

Embora o DogFaceNet forneça uma base inicial para o reconhecimento facial de cães, futuras pesquisas poderiam explorar o treinamento de modelos utilizando conjuntos de dados mais amplos e diversificados, como o apresentado por (CANTO et al., 2023). Além disso, o treinamento do modelo com imagens de outros animais poderia ampliar o escopo da aplicação, permitindo sua adaptação para espécies distintas.

Outra possibilidade é o desenvolvimento de um modelo otimizado para identificar múltiplos cães em uma única imagem. Isso exigiria ajustes na segmentação e na extração de características, bem como a utilização de métricas específicas para medir a precisão e a robustez dessa funcionalidade.

Por fim, melhorias na segurança dos dados e no tempo de processamento também são aspectos relevantes para futuros desenvolvimentos. A implementação de criptografia avançada para proteger informações sensíveis e o uso de técnicas de compressão de vetores de características podem ser exploradas para aumentar a eficiência e segurança da aplicação. Esses avanços contribuiriam para tornar a aplicação mais robusta, escalável e acessível em contextos variados de uso.

# Referências

- AWAIS, M.; NASEER, M.; KHAN, S.; ANWER, R. M.; CHOLAKKAL, H.; SHAH, M.; YANG, M.-H.; KHAN, F. S. *Foundational Models Defining a New Era in Vision: A Survey and Outlook*. 2023. Disponível em: <https://arxiv.org/abs/2307.13721>.
- CANTO, V. H. B. *Identificação Biométrica de Animais Baseada em Aprendizado de Máquina*. Dissertação (Dissertação de Mestrado) — Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2023. Programa de Pós-Graduação em Ciência da Computação.
- CANTO, V. H. B.; MANESCO, J. R. R.; SOUZA, G. B. de; MARANA, A. N. Dog face recognition using vision transformer. In: NALDI, M. C.; BIANCHI, R. A. C. (Ed.). *Intelligent Systems*. Cham: Springer Nature Switzerland, 2023. p. 33–47. ISBN 978-3-031-45389-2.
- DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT, J.; HOULSBY, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, 2020. Disponível em: <https://arxiv.org/abs/2010.11929> . Acesso em: 03 de abril de 2024.
- HENRY, E. U.; EMEBO, O.; OMONHINMIN, C. A. *Vision Transformers in Medical Imaging: A Review*. 2024. Disponível em: <https://your-link-to-the-article> Acesso em: 19 de Outubro de 2024.
- HINTON, G.; VINYALS, O.; DEAN, J. *Distilling the Knowledge in a Neural Network*. 2015. Disponível em: <https://arxiv.org/abs/1503.02531> Acesso em: 27 de Outubro de 2024.
- Instituto Pet Brasil. *Censo Pet IPB: com alta recorde de 6% em um ano, gatos lideram crescimento de animais de estimação no Brasil*. 2022. Disponível em: [https://institutopetbrasil.com/fique-por-dentro/amor-pelos-animais-impulsiona-os-negocios-2-2/#:~:text=Os%20c%C3%A3es%20lideram%20o%20ranking,\(2%2C5%20milh%C3%B5es\)](https://institutopetbrasil.com/fique-por-dentro/amor-pelos-animais-impulsiona-os-negocios-2-2/#:~:text=Os%20c%C3%A3es%20lideram%20o%20ranking,(2%2C5%20milh%C3%B5es).). Acesso em: 4 de Abril de 2024.
- KIRILLOV, A.; MINTUN, E.; RAVI, N.; MAO, H.; ROLLAND, C.; GUSTAFSON, L.; AL. et. *Segment Anything Model (SAM)*. 2023. Disponível em: <https://arxiv.org/abs/2304.02643> . Acesso em: 3 de outubro de 2024.
- MOUGEOT, G.; LI, D.; JIA, S. A deep learning approach for dog face verification and recognition. In: NAYAK, A. C.; SHARMA, A. (Ed.). *PRICAI 2019: Trends in Artificial Intelligence*. Cham: Springer International Publishing, 2019. p. 418–430. ISBN 978-3-030-29894-4. Disponível em: <<https://github.com/GuillaumeMougeot/DogFaceNet>>. Acesso em: 14 out. 2024.
- NAZARENO, A. C.; RONCADA, L. P.; SILVA, I. J. O. d. Identificação eletrônica de animais: quais são as aplicabilidades desses métodos na produção de carne? *Journal*

*of Animal Behaviour and Biometeorology*, Universidade de São Paulo, Departamento de Engenharia de Biossistemas, v. 2, n. 4, p. 142–150, 2014. ISSN 2318-1265.

REDMON, J.; DIVVALA, S. K.; GIRSHICK, R. B.; FARHADI, A. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. Disponível em: <http://arxiv.org/abs/1506.02640> Acesso em: 4 de Abril de 2024.

REIS, D.; KUPEC, J.; HONG, J.; DAOUDI, A. *Real-Time Flying Object Detection with YOLOv8*. 2024. Disponível em: <<https://arxiv.org/abs/2305.09972>>. Acesso em: 08 de Setembro de 2024.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. Disponível em: <https://arxiv.org/abs/1506.01497> Acesso em: 27 de Outubro de 2024.

SUBRAMANYAM, V. S. *Non Max Suppression (NMS)*. 2021. Disponível em: <https://medium.com/analytics-vidhya/non-max-suppression-nms-6623e6572536> Acesso em: 19 de Outubro de 2024.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. *Going Deeper with Convolutions*. 2014. Disponível em: <https://arxiv.org/abs/1409.4842> Acesso em: 19 de Outubro de 2024.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. *Attention Is All You Need*. 2017.

ZHANG, X.; ZHOU, Z.; CHEN, D.; WANG, Y. E. *AutoDistill: an End-to-End Framework to Explore and Distill Hardware-Efficient Language Models*. 2022. Disponível em: <https://arxiv.org/abs/2201.08539> Acesso em: 27 de Outubro de 2024.