

# MareIA: Modelo de *deep learning* para detecção de raças de cães.

João Victor Magalhães Souza  
Universidade Federal de Viçosa  
[joao.souza5@ufv.br](mailto:joao.souza5@ufv.br)

## I. INTRODUÇÃO

A identificação precisa das raças de cães é uma tarefa de grande importância para médicos veterinários, pois cada raça possui características específicas que podem influenciar diretamente no diagnóstico e no tratamento de diversas condições de saúde. Motivado por essa necessidade, apresento uma Inteligência Artificial (IA) inovadora que visa auxiliar profissionais da área veterinária na detecção automática e precisa de raças de cães. Essa ferramenta promete otimizar o tempo dos profissionais e aumentar a precisão dos diagnósticos.

A metodologia empregada no desenvolvimento dessa IA é baseada em técnicas avançadas de *deep learning* [1], utilizando a arquitetura GoogLeNET [2] como motor principal. Essa rede neural convolucional, conhecida por sua eficácia e precisão em tarefas de reconhecimento de imagens, foi treinada com um vasto conjunto de dados contendo milhares de imagens de cães de diversas raças. A escolha do GoogLeNET se deu por sua capacidade de capturar nuances sutis nas imagens, o que é crucial para diferenciar raças com características visuais semelhantes. Os resultados esperados com essa IA são promissores. Espera-se que a ferramenta não apenas identifique corretamente a raça dos cães com uma alta taxa de precisão, mas também forneça informações detalhadas sobre características específicas de cada raça. Isso inclui predisposições genéticas a certas doenças, comportamentos típicos e necessidades de cuidados específicos. Ao integrar essa ferramenta ao cotidiano dos veterinários, vislumbramos um aumento significativo na qualidade dos atendimentos, permitindo diagnósticos mais rápidos e precisos, além de um melhor planejamento dos tratamentos.

Além dos benefícios clínicos, a IA pode ser utilizada em abrigos de animais, facilitando a identificação de raças e, consequentemente, ajudando na adoção de cães. Pode também ser integrada a aplicativos de identificação de animais de estimação, auxiliando donos e treinadores a conhecerem melhor seus mascotes. Além do viés científico deste trabalho, gostaria também de destacar as diversas possibilidades do seu uso prático em diversos campos do nosso cotidiano.

## II. MATERIAL E MÉTODOS

### A. Aquisição de dados para treinamento

Para o desenvolvimento do modelo de detecção de raças de cães, a primeira etapa crítica envolveu a aquisição de um conjunto de dados robusto e diversificado. A obtenção de

dados foi realizada através da plataforma Kaggle, utilizando o dataset “Dog Breed Identification” [3]. Este dataset foi escolhido devido à sua ampla representatividade e qualidade, contendo um total de 10.222 imagens de alta resolução distribuídas entre 120 diferentes raças de cães.

Outro ponto interessante sobre o uso de um dataset público é assegurar a reproduzibilidade e a confiabilidade dos artefatos desenvolvidos e dos resultados obtidos, uma vez que outros pesquisadores podem replicar o estudo utilizando os mesmos dados. Além disso, a diversidade das imagens presentes no conjunto de dados ajuda a garantir que o modelo tenha uma alta capacidade de generalização, podendo identificar raças de cães em diversas condições de iluminação, ângulos e contextos diferentes.

Por questões de limitação computacional, as abordagens posteriormente citadas foram aplicadas em apenas 65% de toda a base de imagens baixadas com o intuito de permitir a construção de um modelo em tempo hábil e testando o maior número de abordagens possíveis.

### B. Pré processamento das imagens

O pré-processamento das imagens é uma etapa crucial no desenvolvimento de modelos de visão computacional, uma vez que essa etapa garante a qualidade e a uniformidade dos dados fornecidos à rede neural. Durante o desenvolvimento do MareAI, foram testadas várias abordagens de pré-processamento [4] para identificar a mais eficaz, tais como:

- **Desfoque Gaussiano:** Aplicação de um filtro de desfoque gaussiano com a intenção de suavizar as imagens e reduzir o ruído. O filtro utilizado foi de kernel 5x5;
- **Detecção de Bordas (Canny):** Utilização do algoritmo de Canny para detecção de bordas, com o objetivo de enfatizar os contornos das raças de cães e facilitar a distinção das características visuais;
- **Equalização de Histograma:** Aplicação da equalização de histograma na camada de luminância da imagem, para melhorar o contraste e a definição das características visuais.

Os processos supracitados tiveram os seguintes resultados:

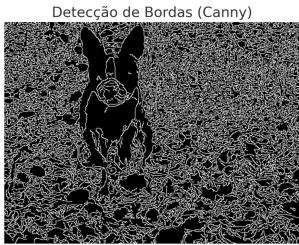


Figura 1: Filtros testados.

Após vários testes com diferentes filtros observados na Figura 1, observou-se que havia uma maior vantagem em treinar o modelo sem nenhum filtro específico. Entretanto, foi feito um redimensionamento das imagens para o 256x256 e foi aplicada uma camada de *Data Augmentation*, que será detalhada na próxima etapa.

### C. Data Augmentation

Visando aumentar o meu espaço de possibilidades, foi implementado uma etapa de *Data Augmentation* para criação de novos contextos de imagens a partir do *dataset* já existente. Como dito anteriormente, foram pegas 65% das imagens de maneira aleatória e, dessas imagens, foi aplicada a seguinte configuração:

- **Rotation Range:** Rotação das imagens em até 15 graus, para ensinar o modelo a reconhecer cães independentemente da orientação;
- **Width Shift e Height Shift:** Deslocamentos horizontais e verticais de até 10% da largura e altura da imagem, respectivamente, para simular mudanças de posição;
- **Shear Range:** Transformações de cisalhamento em até 10%, para simular distorções.
- **Zoom Range:** Zoom nas imagens em até 10%, para ensinar o modelo a reconhecer cães em diferentes escalas;
- **Horizontal Flip:** Espelhamento horizontal das imagens, para aumentar a variabilidade e ajudar o modelo a generalizar melhor;
- **Fill Mode:** Modo de preenchimento 'nearest', que preenche os pixels vazios gerados pelas transformações com os valores dos pixels mais próximos.

Dessa forma, cada época de treinamento recebe um padrão de imagem diferente, permitindo ao modelo observar diferentes ângulos de uma mesma raça de cão, aumentando seu poder de generalização.

Figura 2: Alguns resultados do *Data Augmentation* usado.

Portanto, tendo como base o conceito aplicado e observando a Figura 2, permitimos ao modelo captar conceitos de um cão em diferentes perspectivas visuais.

### D. Divisão da base

Outro importante passo é dividir a base em treino, validação e teste. Para isso, utilizamos a seguinte proporção de divisão: 80% para treino e 20% para teste (também conhecida como amostra fora do tempo). Ainda assim, do montante que representa os 80% de imagens para treino, dividimos 90% para realizar o treinamento do modelo e 10% para a validação das métricas em treinamento.

### E. Arquitetura do modelo

Para a etapa de desenvolvimento da arquitetura do modelo de *deep learning* a ser responsável por gerar a identificação da raça de cada cão, foi decidido usar o GoogLeNET (também conhecido como InceptionV3) por ser um modelo pré-treinado de uso geral desenvolvido pela Google e ter uma boa acurácia em problemas de detecção de imagens. Com isso, conseguimos aproveitar e transferir o conhecimento (*transfer learning* [5]) dessa rede treinada com um gigantesco conjunto de imagens. Além da arquitetura inicial do InceptionV3, foram aplicadas duas camadas densas com, respectivamente, 256 e 128 neurônios e uma camada de saída para predição da raça do cão observado. Devido ao tamanho e complexidade da arquitetura final, não será colocado o diagrama da rede nesta seção. Entretanto, o diagrama pode ser consultado no repositório do projeto que estará disponível na seção de Código-fonte.

## III. RESULTADOS

Após o treinamento do modelo com os dados processados e com a arquitetura supracitada (executando o arquivo *pipeline\_train\_inception.py*), tivemos o seguinte resultado:

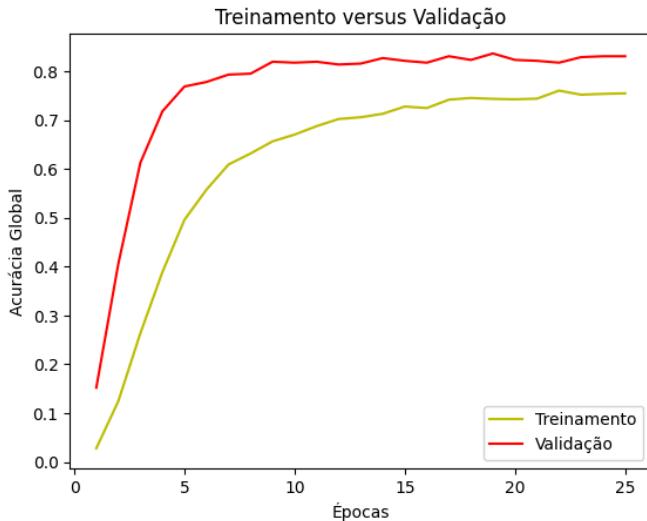


Figura 3: Resultado do modelo em treinamento e validação.

Com base na Figura 3, podemos observar que o modelo converge para uma acurácia de 83% em validação após aproximadamente 20 épocas de treinamento. Analisando o gráfico, podemos dizer que o modelo apresenta uma boa capacidade de generalização, dado que a acurácia do modelo em treinamento está substancialmente menor que a acurácia em validação.

Quando olhamos para a amostra fora do tempo, isto é, a parte dedicada para teste, temos uma acurácia de 85%. Isso nos diz que a cada 100 imagens, 85 ele identifica corretamente, o que é um resultado extremamente interessante.

A fim de testar o modelo produzido em termos práticos cotidianos, foram reunidas algumas fotos de cães domésticos que não estavam na base de treino, validação e teste a fim de verificar a acurácia final do modelo. Após submeter as fotos dos cães para a predição do modelo (usando o arquivo *pipeline\_predict.py*), obtivemos o seguinte resultado:

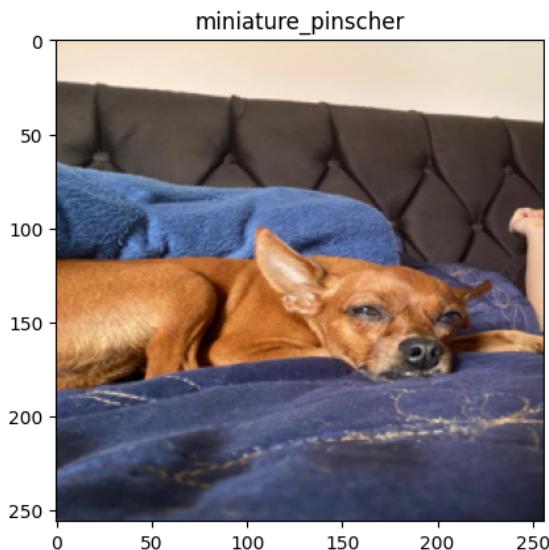


Figura 4: MareAI previu o cão como da raça *pinscher*.

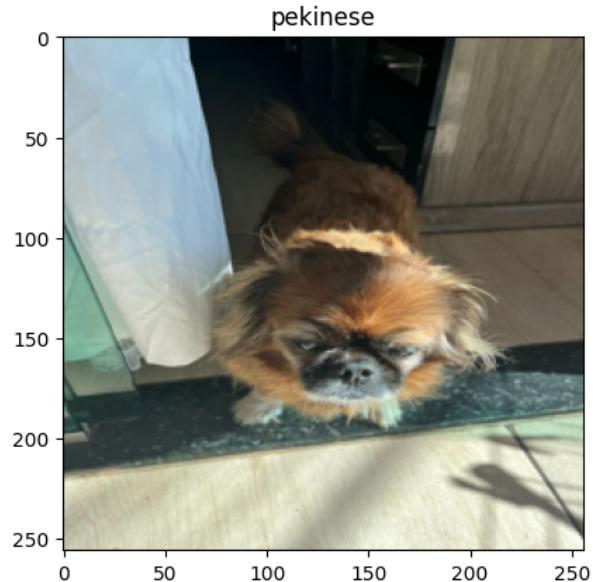


Figura 5: MareAI previu o cão como da raça *pequinês*.

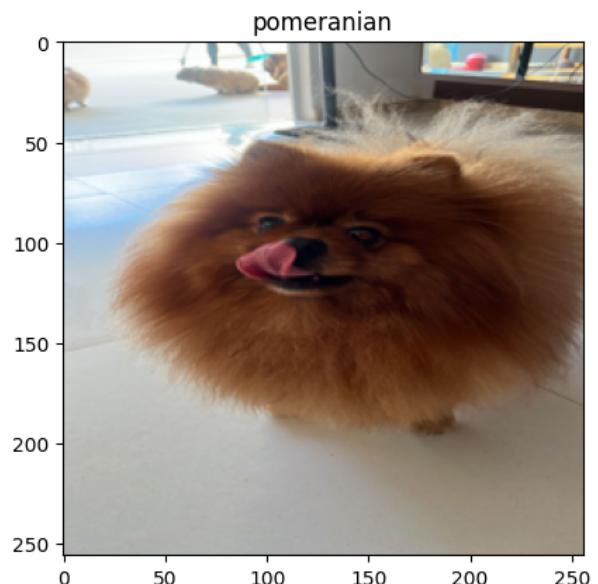


Figura 6: MareAI previu o cão como da raça *pomeriano*.

Após uma pesquisa das classes preditas, viu-se que MareAI conseguiu acertar todas as raças dos de todos os cães submetidos à sua predição, cumprindo satisfatoriamente bem seu propósito.

#### IV. CONCLUSÕES

O desenvolvimento do modelo MareIA para a detecção de raças de cães com base em técnicas avançadas de deep learning demonstrou resultados promissores e aplicações práticas relevantes. A utilização da arquitetura GoogLeNET, treinada com um vasto conjunto de dados diversificado,

permittiua alcançar uma acurácia de 85% na identificação correta das raças de cães. Este nível de precisão é significativo e valida a eficácia do modelo proposto e abre um leque de oportunidades para seu uso em diversas áreas da medicina veterinária. A metodologia empregada, que incluiu a aquisição de dados robustos, pré-processamento das imagens e técnicas de *Data Augmentation*, foi crucial para o sucesso das métricas encontradas pelo MareIA.

Os resultados obtidos não só cumprem com o objetivo inicial deste trabalho mas também abrem espaço para a criação de um *software open source* para a utilização do modelo, assim como apresenta oportunidades para melhorias nos resultados obtidos nessa primeira versão.

#### CÓDIGO-FONTE

O repositório deste projeto pode ser acessado em:  
<https://github.com/JoaovictorMagalhaesSouza/deep-learning-dog-breed-detection>

#### REFERENCES

- [1] Yann Lecun, Yoshua Bengio, Geoffrey Hinton. Deep learning. *Nature*, 2015, 521 (7553), pp.436-444. ff10.1038/nature14539ff. ffhal-04206682
- [2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. *arXiv*, 2014. ff<https://arxiv.org/pdf/1409.4842ff.pdf>
- [3] Will Cukierski. (2017). Dog Breed Identification. *Kaggle*. <https://kaggle.com/competitions/dog-breed-identification>
- [4] A.Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow*. 2nd Edition. 2019. Chapter 13: Loading and Preprocessing Data with TensorFlow and Chapter 14: Deep Computer Vision Using Convolutional Neural Networks.
- [5] Torrey, Lisa, and Jude Shavlik. "Transfer learning." *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010. 242-264.