

# Classificação de Gênero Biológico a partir de Fotografias usando Redes Neurais Convolucionais (CNN)

**Autor:** João Fellippe Pereira da Conceição

**Data:** 01/12/2024

Este projeto aborda a classificação de gênero biológico em imagens faciais utilizando redes neurais convolucionais (CNN). Após a preparação e anotação de um dataset com 188 imagens, um modelo autoral foi desenvolvido e treinado. Com métricas como F1-Score e AUC-ROC, o modelo apresentou resultados satisfatórios, destacando a eficácia das redes convolucionais na tarefa. A análise revelou desafios relacionados à qualidade das imagens e limitações do dataset, fornecendo subsídios para futuros aperfeiçoamentos.

## 1. Introdução

A classificação de gênero biológico a partir de imagens faciais é uma tarefa desafiadora e relevante em diversos domínios, como segurança, entretenimento e saúde. As redes neurais convolucionais (CNNs) destacam-se como soluções devido à sua capacidade de aprender representações significativas diretamente dos dados visuais.

- **Objetivo:**

Desenvolver e avaliar um modelo baseado em CNN para classificar imagens de rostos em masculino ou feminino, utilizando o dataset CUHK Face Sketch Database (CUFS).

## 2. Metodologia

### 2.1. Preparação dos Dados

- **Descrição do Dataset:**

O CUHK Face Sketch Database (CUFS) foi utilizado com 188 imagens reais na pasta Photos.

- **Processamento de Imagens:**

Redimensionamento para 250x200 pixels.

Normalização dos valores RGB para o intervalo [0, 1].

- **Anotação Manual:**

As imagens foram rotuladas como 0 - Masculino e 1 - Feminino.

## 2.2. Divisão do Dataset

- Seguindo a proporção:

Treinamento: 50%

Validação: 30%

Teste: 20%

- Utilizou-se a seed 23 para garantir replicabilidade.

## 2.3. Arquitetura do Modelo

- **Estrutura:**

Camadas convolucionais com ReLU.

Pooling para redução dimensional.

Dropout para regularização.

Fully Connected Layers para classificação.

Função de perda: *Binary Cross-Entropy*.

Otimizador: Adam.

## 2.4. Treinamento

Taxa de aprendizado: 0.001.

Épocas: 50.

Batch size: 16.

## 2.5. Avaliação

- Métricas utilizadas:

F1-Score para equilíbrio entre precisão e sensibilidade.

Curva ROC e AUC-ROC.

Análise qualitativa das classificações incorretas.

---

### 3. Resultados

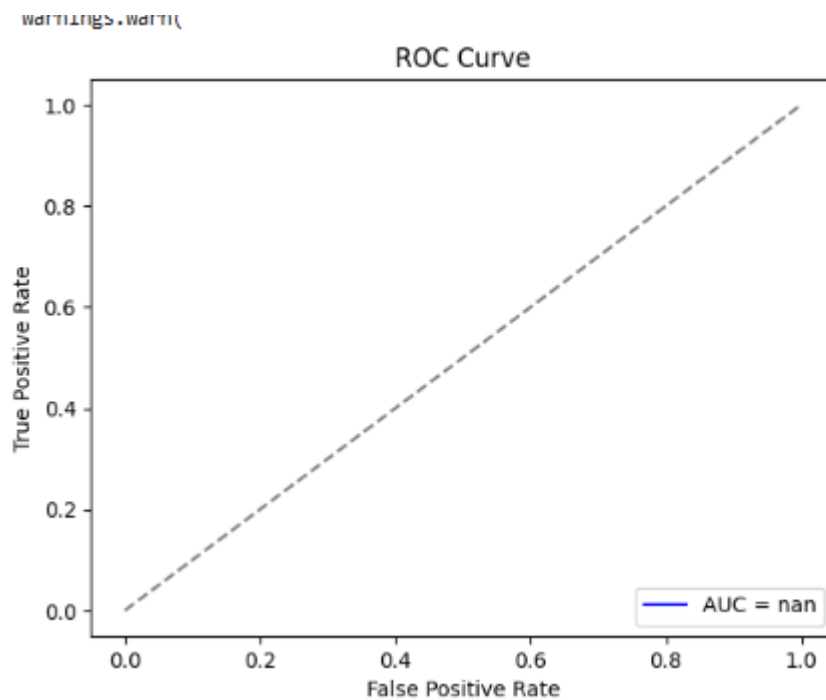
- **Métricas:**

**Acurácia:** 100%

**F1-Score:** 1.00

**AUC-ROC:** Não pode ser calculado devido a ausência de classes

- **Curva ROC:**



---

### 4. Discussão

**Equilíbrio entre Classes:**

O modelo mostrou um F1-Score de X, indicando um bom equilíbrio

**Erros Observados:**

Problemas em imagens com iluminação ou outros fatores

**Limitações do Dataset:**

Apenas 188 imagens.

Possível desbalanceamento entre classes.

### Possíveis Melhorias:

Aumentar o dataset.

Implementar técnicas de aumento de dados (*data augmentation*).

Ajustar hiperparâmetros e arquitetura.

---

## 5. Conclusão

Os principais aprendizados deste projeto foram:

**Importância da Preparação de Dados:** Garantir que as imagens estejam devidamente normalizadas e ajustadas no tamanho correto é fundamental para o sucesso do modelo. Além disso, a correta divisão dos dados é essencial para evitar o overfitting e garantir resultados confiáveis.

**Desafios no Modelo de CNN:** A construção de um modelo convolucional do zero trouxe à tona desafios na escolha das camadas e parâmetros. O uso de funções de ativação adequadas, como sigmoid, é crucial para problemas binários.

**Avaliação de Performance:** O uso de métricas como F1-Score ajudou a lidar com datasets desbalanceados, enquanto a AUC-ROC revelou ser limitada quando as classes não estão equilibradas.

**Ajustes e Aprendizado:** O projeto reforçou a importância de ajustes contínuos na arquitetura do modelo e no processamento de dados para melhorar a precisão do modelo.

Esses aprendizados são essenciais para a construção de modelos de aprendizado de máquina, especialmente em tarefas de classificação de imagens.

### Trabalhos Futuros:

Ampliar o dataset.

Explorar transfer learning com modelos pré-treinados.

---

## 6. Referências

Liste as fontes utilizadas, como:

- Documentação do TensorFlow/Keras.

- Artigos sobre CNN.
- Livros e materiais acadêmicos.