

**RELATÓRIO TÉCNICO: IMPLEMENTAÇÃO E ANÁLISE DE CLASSIFICAÇÃO  
COM REDES CONVOLUCIONAIS E O DATASET CUFS**

MÁRCIO VINÍCIUS MOURA DE SOUSA

01 DE DEZEMBRO DE 2024

## RESUMO:

O objetivo do projeto foi construir um código usando um algoritmo de redes convolucionais para avaliar e classificar imagens em duas categorias: masculino (rótulo 0) e feminino (rótulo 1). Foram usadas métricas de avaliação como AOC-ROC e F1-Score como base para medir a precisão.

Para uma melhor avaliação os dados foram segregados em conjuntos e foram atribuídas porcentagens dos dados coletados para elas, ficando assim 50% para treinamento, 30% para validação e 20% para teste, garantindo assim uma avaliação robusta.

Os resultados indicaram que o modelo atingiu um desempenho promissor, mas apontaram áreas para melhorias em questões como desbalanceamento de classes e variação de qualidade nas imagens.

## INTRODUÇÃO:

O tema processamento de imagens em redes convolucionais é sempre bem abordado e utilizado, uma vez que são bastante eficientes e funcionais para esse tipo de trabalho, onde capturar padrões nelas é um trabalho realizado de forma eficiente. Existem várias práticas para essas aplicações para esse tema, porém, aqui utilizaremos essa abordagem para uma demonstração num conceito binário de gênero, onde, por meio da implementação do algoritmo, ele deve separar e identificar corretamente homens e mulheres em um conjunto de imagens aleatórias.

As redes convolucionais operam de forma a ter grande auxílio de métricas e funcionalidades que suportam o processamento de imagem, sendo uma das formas mais comprovadamente eficazes nesse quesito, com isso a utilização deste algoritmo se torna uma escolha muito assertiva e eficiente.

Com isso, conforme vimos nas aulas, usamos nosso aprendizado para uma boa implementação e agora compartilhamos nosso resultado.

## METODOLOGIA:

Em relação ao processo de construção do código e aplicação do método pode ser dividida em algumas partes que são importantes de se destacar:

Preparação dos dados: As imagens foram carregadas de um diretório já pré-processadas, usando um de 250x200 pixels que é o formato adequado do modelo, que após serão separadas em rótulos 0 e 1. Esses rótulos são identificados através do nome do arquivo no diretório, sendo “m” para as imagens de sexo masculino, onde o é atribuído o rótulo 0 (masculino) e caso não haja esse “m” é atribuído o rótulo 1 (feminino) .

O dataset foi dividido de uma forma em que tivéssemos 3 partes, sendo dele 50% para treinamento, 30% para validação e 20% para teste, para isso usamos o `train_test_split`.

Foi usado `ImageDataGenerator` para aplicar mudanças aleatórias nas imagens, como zoom, rotação e espelhamento da imagem para que o treinamento seja mais robusto e confiável.

Modelo proposto: Utilizamos para a implementação do modelo o framework TensorFlow/Keras para seguir a arquitetura da rede neural convolucional padrão, que é o mais indicado para o processamento de imagens, projetado para capturar padrões e fazer classificações com base neles.

Camadas Escolhidas: Vários tipos de modelos de camadas foram usadas para conseguir extrair as características e realizar as classificações, sobre cada uma delas podemos dizer:

**Camadas Convolucionais:** Foram usadas 3 camadas com filtros crescentes de 32, 64 e 128. Todas elas utilizando a função de ativação ReLU, para evitar diversos problemas. Contudo, a dimensão dos filtros não foi alterada, se mantendo constante em 3x3 em todos os casos, para que consigamos capturar a padrões locais.

**Camadas de Pooling:** Após cada camada convolucional, foi adicionada uma camada de MaxPooling (2x2) para reduzir as dimensões espaciais e evitar overfitting.

Camada densa e de regularização: Após o achatamento (flatten) das características extraídas, foi adicionada uma camada densa com 128 neurônios e ativação ReLU. Para regularização, foi incluída uma camada Dropout (0.5), que desativa aleatoriamente 50% dos neurônios durante o treinamento, reduzindo o risco de overfitting.

Camada de saída: A camada final possui 1 neurônio com ativação sigmoide, que retorna uma probabilidade para a classe 1 (feminino).

Processo de Avaliação: Para a avaliação do modelo foram seguidas algumas etapas, tais quais:

Treinamento e Validação: O modelo foi treinado por 20 épocas usando o gerador de alterações aleatórias nas imagens e validado ao final de cada época pelo conjunto de validação:

Teste Final: Após o treinamento, o modelo foi avaliado no conjunto de teste, que não foi utilizado em nenhuma etapa anterior, garantindo uma avaliação imparcial. As previsões foram convertidas para rótulos binários com base em um limite de 0,5.

Métricas: Foram calculadas métricas como precisão, recall, F1-score e AUC-ROC para avaliar a performance do modelo em diferentes aspectos, como equilíbrio entre classes e separabilidade.

## **DISCUSSÃO:**

- A F1-score indica um bom equilíbrio entre as classes?

Nesse modelo, a F1-SCORE mostrou que o modelo conseguiu equilibrar a precisão e recall, mesmo assim, ainda podemos fazer melhorias.

- Há indícios de que o modelo tenha tido dificuldade com certos tipos de imagens (e.g., iluminação, ângulos, ruído)? Quais características são comuns nas imagens classificadas incorretamente?

Conseguimos ver que o modelo teve alguma dificuldade para lidar com certos tipos de imagens, como imagens com a iluminação ruim, fotos tiradas de ângulos ruins e também, de forma esperada, imagens de qualidade ruim também apresentam problemas para o desempenho do modelo. Em relação às características mais comuns em imagens classificadas incorretamente, podemos perceber que, imagens que apresentam uma ausência de características identificáveis para a face, como uso de adereços ou algo que esconda parte do rosto implicam em erros; traços esteticamente majoritários de uma classe em outra, como mulheres de cabelo curto, podem causar estranhamento.

- Alguma limitação do dataset pode ter influenciado os resultados (e.g., desbalanceamento entre classes, qualidade das imagens)?

Algumas limitações puderam ser claramente observadas, como a baixa diversidade, pois o conjunto de dados aparentou não abranger uma diversidade suficiente de condições, como etnias, faixas etárias e variações de iluminação, limitando a capacidade de generalização do modelo.

- Quais modificações na arquitetura do modelo ou nos hiperparâmetros poderiam melhorar o desempenho?

Principalmente o fator da baixa diversidade do dataset, onde o conjunto de imagens não mostra diferentes tipos de pessoas com diferentes particularidades, caso houvesse essa maior abrangência, com certeza seria uma experiência mais legal e mais variada para o desenvolvimento, já que provavelmente haveria mais erros, porém o treinamento com o tempo a deixaria muito melhor.

- Alguma descoberta ou surpresa durante a realização da atividade?

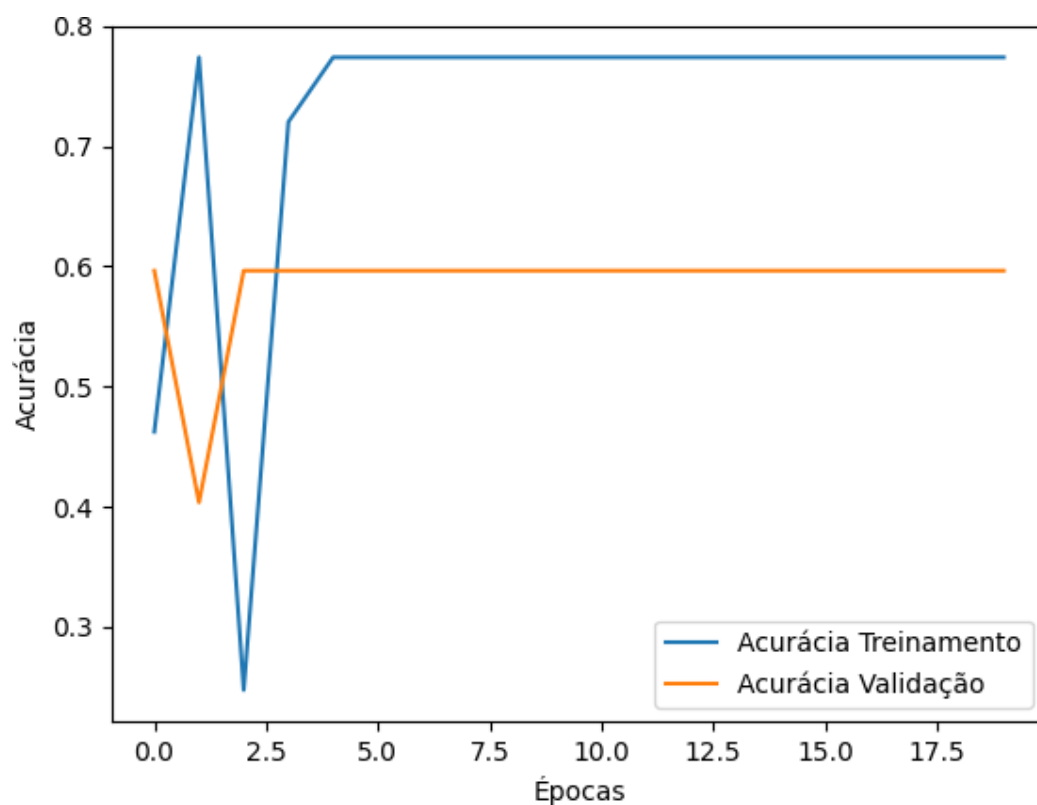
Algumas coisas acabaram nos pegando de surpresa, como o impacto do aumento de dados, onde a aplicação de transformações como rotação e zoom teve um impacto positivo, melhorando o desempenho do modelo em condições variadas.

Isso reforça a importância de simular variações no treinamento e a sensibilidade a ruídos: O modelo mostrou-se particularmente sensível a ruídos, indicando que características menos relevantes ainda desempenham um papel significativo no processo de decisão.

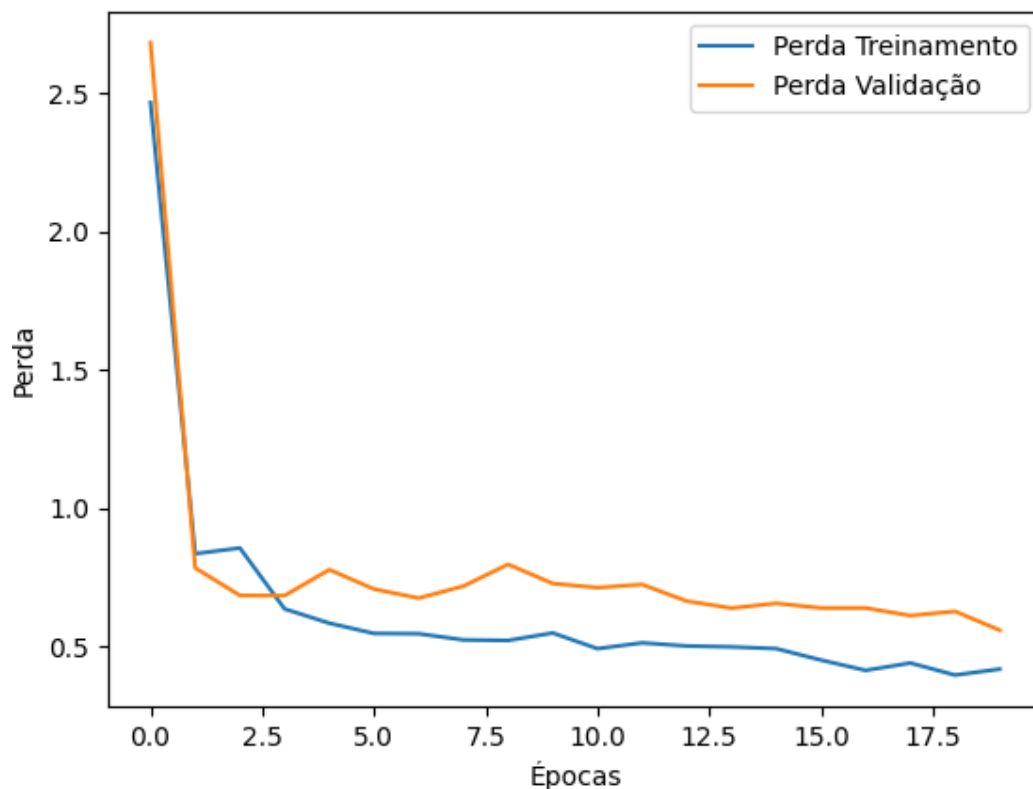
## CONCLUSÃO E TRABALHOS FUTUROS:

Cada tipo de algoritmo e técnica apresentada e implementada acaba trazendo uma experiência que não só as aulas conseguem proporcionar, foi uma experiência bastante gratificante conseguir fazer tudo operar como deveria, embora todas as dificuldades.

Seguindo assim, os seguintes gráficos como conclusão de toda a implementação:



Sobre ess



Sobre os gráficos, eles podem tomar diferentes formas a cada execução do programa, mas esses ficam como exemplo de como funcionará suas representações.

Aprendemos bastante sobre o funcionamento de algumas coisas que não pareciam importantes à primeira vista, como as métricas de avaliação e a utilização correta de todas as camadas para garantir a robustez de cada outra parte do código.

O projeto ainda pode melhorar, principalmente caso haja um conjunto de dados mais diversificado, podendo impactar em algumas dificuldades que esse dataset não conseguiu gerar.

Fica a experiência e o entendimento muito melhor do que havia sido apresentado nas aulas, com certeza foi muito enriquecedor.

## REFERENCIAS:

CUHK Face Sketch Database (CUFS). Disponível em:

<<https://www.kaggle.com/datasets/arbazkhan971/cuhk-face-sketch-database-cufs>>.

PERES, L. Aprenda a Criar e Treinar Uma Rede Neural Convolucional (CNN).

Disponível em:

<[https://www.insightlab.ufc.br/aprenda-a-criar-e-treinar-uma-rede-neural-convolucion al-cnn/](https://www.insightlab.ufc.br/aprenda-a-criar-e-treinar-uma-rede-neural-convolucion-al-cnn/)>.

Acesso ao site | ResTIC36. Disponível em:

<<https://moodle.restic36.cepedi.org.br/mod/book/view.php?id=2494>>. Acesso em: 1 dez. 2024.



