Relatório Técnico: Implementação e Análise de Classificação com Redes Convolucionais e o dataset CUFS

Lucca Fernandes Trancoso Nolasco, Rodrigo Ferreira Bento Aguiar

December 3, 2024

Resumo

Este relatório técnico apresenta o objetivo, metodologia e principais resultados de um projeto de análise com foco na classificação de imagens de rostos baseadas no sexo via uma Rede Neural Convolucional. Primeiramente, é realizada uma contextualização do problema e descrição do conjunto de dados. Após isso, são demonstrados dois modelos e os resultados do modelo superior.

Introdução

O objetivo deste relatório é apresentar uma documentação e explicação da modelagem e treino de um modelo de Rede Neural Convolucional para a classificação de imagens entre "Sexo Masculino" e "Sexo Feminino" do dataset CUHK Face Sketch Database (CUFS). O presente trabalho consiste numa avaliação proposta pelo CEPEDI na Reseidência em Software Restic36, para a trilha de Ciência de Dados. A opção pelo uso das Redes Neurais Convolucionais se deu pela sua grande aplicação em processamento digital da imagem, o que é um conteúdo de grande interesse da dupla. Dessa forma, foi possível obter grande aprendizado e avalaiação do modelo, bem como um feedback construtivo.

0.1 Contexto do Problema

O objetivo do modelo proposto é classificar a imagem do rosto de uma pessoa entre sexo masculino e sexo feminino. A classificação de imagens é uma tarefa central no campo de visão computacional, com aplicações que vão desde segurança e biometria até sistemas de recomendação e marketing. A capacidade de identificar automaticamente características específicas de imagens é crucial para o desenvolvimento de soluções inteligentes em diversas áreas.

No entanto, a classificação de imagens apresenta desafios significativos, como a variação de iluminação, a presença de ruídos, diferentes ângulos e expressões faciais. Esses fatores tornam a tarefa complexa para abordagens tradicionais de aprendizado de máquina.

É nesse contexto que as redes neurais convolucionais (CNNs) se destacam. As CNNs são modelos de aprendizado profundo projetados para lidar com dados estruturados em forma de grade, como imagens, e são especialmente eficazes para identificar padrões locais em diferentes escalas e contextos. Ao automatizar o processo de extração de características, as CNNs permitem que o modelo aprenda representações hierárquicas das imagens, tornando-o robusto a variações como escala, rotação e distorções. Esse poder de generalização das CNNs tem sido fundamental para o sucesso de sistemas de reconhecimento de imagens, tornando-as a abordagem preferida em tarefas como a classificação de rostos, identificação de objetos e segmentação semântica.

0.2 Descrição do Conjunto de Dados

O conjunto de dados utilizado neste estudo é o *CUHK Face Sketch Database (CUFS)*, disponível no Kaggle através do link: https://www.kaggle.com/datasets/arbazkhan971/cuhk-face-sketch-database-cufs. Esse conjunto foi originalmente desenvolvido para tarefas de correspondência entre esboços faciais e fotografias de rostos, mas foi adaptado para a tarefa de classificação binária entre sexo masculino e sexo feminino.

O conjunto é composto por um total de 188 imagens de rostos de indivíduos masculinos e femininos, com uma distribuição desigual entre as classes. As imagens possuem variações em termos de expressão facial, iluminação e ângulo, o que torna o conjunto interessante para treinamento de modelos de aprendizado profundo.

Para a utilização nas redes neurais, as imagens foram inicialmente processadas e organizadas em um *DataFrame Pandas*, que contém as seguintes colunas:

- **filename**: Contém o nome do arquivo de cada imagem;
- **sex**: Label associada à imagem, onde 0 representa homens e 1 representa mulheres;
- **image_array**: Representação da imagem em formato de array *Numpy*, após conversão e normalização dos valores RGB para um intervalo de [0, 1], o que facilita o processamento e treinamento do modelo.

Esse conjunto de dados, apesar de pequeno, proporciona uma base sólida para a experimentação de redes neurais convolucionais em tarefas de classificação facial, oferecendo um ponto de partida para modelos de maior escala e complexidade.

Metodologia

Desenvolvemos dois modelos para a classificação binária de imagens de rostos entre sexo masculino e sexo feminino, e concluímos que o segundo modelo foi superior. No primeiro modelo, realizamos a normalização dos dados, mas não fizemos um balanceamento das classes. Ou seja, não inserimos amostras adicionais para a classe minoritária (sexo feminino). Para tentar equilibrar o desempenho, aplicamos pesos às classes e ajustamos o limiar de decisão. No segundo modelo, optamos por inserir imagens na classe minoritária até que as classes estivessem equilibradas. Essa abordagem resultou em um modelo mais eficaz.

0.3 Preparação dos dados

As imagens foram obtidas do *Kaggle* e baixadas para uma pasta no *Google Colab*. Utilizando algumas funções do *Python*, as imagens foram redimensionadas para 200 pixels de largura e 250 de altura e convertidas em arrays *Numpy*. Além disso, seus valores RGB foram normalizados, dividindo-os por 255. Para a tarefa de classificação, era necessário que as imagens possuíssem uma etiqueta, mas a base de dados fornecida não as incluía. A princípio, a solução seria etiquetar as imagens manualmente, mas logo foi percebido que os nomes dos arquivos das imagens de rostos masculinos começavam com "m" e os de rostos femininos com "f". Uma função foi então criada para gerar as etiquetas de forma automatizada, com base nesses prefixos.

Nessa primeira etapa, foi desenvolvida uma função para dividir os dados na porcentagem recomendada (50% para treino, 30% para validação e 20% para teste), e com a seed recomendada de 23.

0.4 Modelo Inicial

Neste modelo, foi desenvolvida uma arquitetura mais simples, composta por três camadas convolucionais, com uma quantidade crescente de filtros (16, 32 e 64), cada um com tamanho 3x3. A escolha pela quantidade crescente de filtros visou evitar a complexidade excessiva e a carga de memória elevada, mantendo o modelo leve e eficiente. Contudo, ao longo do desenvolvimento, percebeu-se que um modelo mais complexo seria necessário para a tarefa proposta.

Além das camadas convolucionais, foram utilizadas camadas de *Max Pooling* com tamanho 2x2, responsáveis por capturar o valor máximo de cada região da imagem, o que ajuda a reduzir suas dimensões e a extrair características importantes. Após cada camada de pooling, foi inserida uma camada de *Batch Normalization*, que ajusta as entradas para terem média zero e desvio padrão igual a um, contribuindo para uma convergência mais rápida e estável durante o treinamento.

Ao final das camadas convolucionais, utilizou-se uma camada *Flatten* para transformar a saída multidimensional em um vetor unidimensional, possibilitando que os dados fossem passados para a camada totalmente conectada. Para prevenir o overfitting, foi incorporada uma camada de *Dropout*, que desativa aleatoriamente uma fração das unidades durante o treinamento, aumentando a generalização do modelo.

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 198, 248, 16)	448
max_pooling2d (MaxPooling2D)	(None, 99, 124, 16)	0
batch_normalization (BatchNormalization)	(None, 99, 124, 16)	64
conv2d_1 (Conv2D)	(None, 97, 122, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 48, 61, 32)	0
batch_normalization_1 (BatchNormalization)	(None, 48, 61, 32)	128
conv2d_2 (Conv2D)	(None, 46, 59, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 23, 29, 64)	0
batch_normalization_2 (BatchNormalization)	(None, 23, 29, 64)	256
flatten (Flatten)	(None, 42688)	0
dense (Dense)	(None, 32)	1,366,048
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33
Total params		**1,390,113 (5.30 MB)**
Trainable params		**1,389,889 (5.30 MB)**
Non-trainable params		**224 (896.00 B)**

Table 1: Arquitetura do modelo inicial com camadas e parâmetros

Em seu treinamento, foi utilizado o *Early Stopping*, monitorando o desempenho no conjunto de validação. Além disso, foram atribuídos pesos às classes, com o objetivo de equilibrá-las sem a necessidade de inserir novos dados. Após algumas épocas de treino, o processo foi interrompido devido à estagnação na acurácia de validação. As métricas obtidas foram insatisfatórias, e a **AUC-ROC** indicou que o modelo estava basicamente classificando as imagens de forma aleatória, sem discernimento adequado entre as classes. A baixa **acurácia** também sugere que o modelo foi capaz de acertar uma quantidade muito pequena de previsões.

Métrica	**Valor**
F1-Score	0.5098
Acurácia	0.3421
Precisão	0.3421
AUC-ROC	0.5231

Table 2: Métricas de Avaliação do Modelo

A figura 1 demonstra a capacidade de decisão do modelo mediana, dividindo de forma aleatória entre as classes. Calculou-se então um "limiar ótimo para o modelo", mas as métricas ainda assim estavam insatisfatórias.

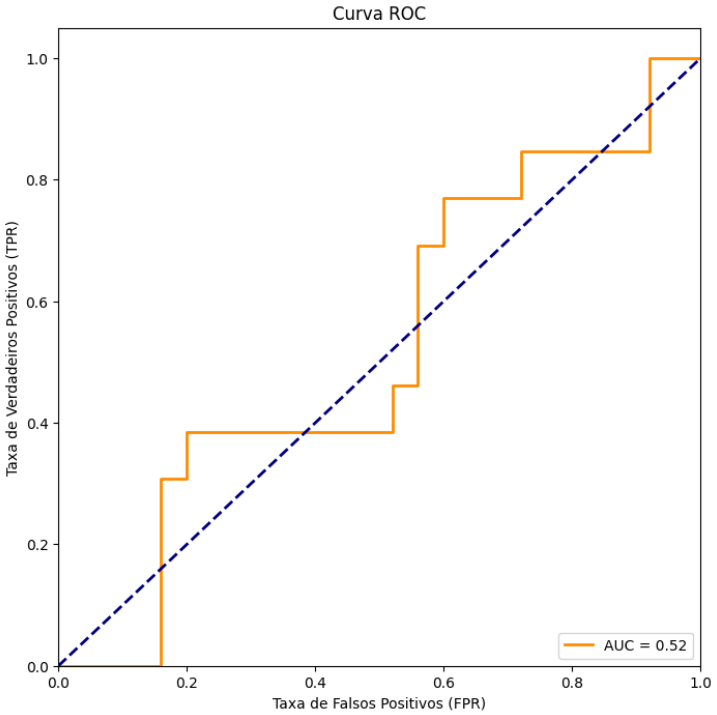

Figure 1: Curva Roc do Modelo Inicia. Fonte: Própria

Métrica	**Valor**
F1-Score com limiar ótimo	0.4348
Acurácia com limiar ótimo	0.6579
Precisão com limiar ótimo	0.5000
Recall com limiar ótimo	0.3846
AUC-ROC	0.5231

Table 3: Métricas do Modelo com Limiar Ótimo

0.5 Modelo Final

Antes do desenvolvimento de um novo modelo, foram criadas imagens adicionais para balancear a classe minoritária. Imagens de rostos femininos foram selecionadas aleatoriamente e, por meio de técnicas de aumento de dados, novas imagens foram geradas aplicando-se zoom, inclinação da imagem em um intervalo de graus, rotação horizontal, e leves variações na proporção e no preenchimento de pixels. As novas imagens também passaram pelo processo de redimensionamento e normalização.

Ao final, o novo dataset passou a contar com 134 imagens de homens e 134 imagens de mulheres, as quais foram divididas na mesma proporção solicitada, utilizando a mesma semente para garantir a replicabilidade. Além disso, foi assegurado que a quantidade de imagens das classes 0 e 1 fosse equilibrada dentro dos próprios conjuntos de dados. A nova divisão foi a seguinte:

- Treinamento: 67 imagens da classe 0 e 66 da classe 1;
- Validação: 41 imagens da classe 0 e 40 da classe 1;

- Teste: 27 imagens da classe 0 e 27 da classe 1.

O modelo final é mais profundo, possuindo 4 camadas. Nele, o número de filtros também é crescente, mas possui uma quantidade maior: 32, 64, 128, 256. Cada um possui tamanho de 3x3, e é capaz de aprender características mais complexas. Além disso, cada uma dessas camadas utiliza o **regularizador L2**, que ajuda a prevenir o *overfitting* ao penalizar pesos muito grandes. Isso ajuda o modelo a generalizar melhor, principalmente quando possui mais capacidade.

O novo modelo usa **Global Average Pooling (GAP)** em vez de Flatten. O Global Average Pooling calcula a média de cada mapa de características (ao invés de achatá-los), o que reduz a dimensionalidade de forma mais eficiente e pode ajudar a prevenir overfitting, além de ser mais eficiente do que o flattening em termos de parâmetros. O Flatten na arquitetura anterior cria uma grande quantidade de parâmetros, o que pode tornar o modelo propenso a overfitting, especialmente quando há muitas camadas convolucionais. O Global Average Pooling gera um vetor com uma única média por mapa de características, reduzindo significativamente o número de parâmetros e ajudando a evitar esse problema. A preocupação com o overfitting se dá justamente devido a muitas amostras de dados serem versões modificadas de outras, e que podem ainda guardar grande semelhança.

A camada totalmente conectada é mais robusta, onde Dense(128) no novo modelo possui 128 unidades (neurônios), o que pode permitir ao modelo aprender representações mais complexas. O modelo anterior tinha apenas 32 unidades nesta camada, o que significa que o novo modelo tem mais capacidade para aprender padrões mais complexos nas informações extraídas pelas camadas convolucionais.

O dropout de 50% foi mantido, ajudando a regularizar e evitar o *overfitting*.

Layer (type)	**Output Shape**	**Param #**
conv2d_10 (Conv2D)	(None, 198, 248, 32)	896
max_pooling2d_10 (MaxPooling2D)	(None, 99, 124, 32)	0
batch_normalization_10 (BatchNormalization)	(None, 99, 124, 32)	128
conv2d_11 (Conv2D)	(None, 97, 122, 64)	18,496
max_pooling2d_11 (MaxPooling2D)	(None, 48, 61, 64)	0
batch_normalization_11 (BatchNormalization)	(None, 48, 61, 64)	256
conv2d_12 (Conv2D)	(None, 46, 59, 128)	73,856
max_pooling2d_12 (MaxPooling2D)	(None, 23, 29, 128)	0
batch_normalization_12 (BatchNormalization)	(None, 23, 29, 128)	512
conv2d_13 (Conv2D)	(None, 21, 27, 256)	295,168
max_pooling2d_13 (MaxPooling2D)	(None, 10, 13, 256)	0
batch_normalization_13 (BatchNormalization)	(None, 10, 13, 256)	1,024
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 256)	0
dense_6 (Dense)	(None, 128)	32,896
dropout_3 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 1)	129
Total params		**423,361 (1.61 MB)**
Trainable params		**422,401 (1.61 MB)**
Non-trainable params		**960 (3.75 KB)**

Table 4: Arquitetura do Modelo com Camadas Convolucionais e Densas

Embora o novo modelo possua quase três vezes menos parâmetros, ele apresenta maior capacidade em comparação ao anterior, devido a ser menos propenso ao *overfitting*. A maior profundidade e o aumento da quantidade de filtros contribuem para um maior

potencial de detecção de detalhes nas imagens. A regularização **L2** também desempenha um papel importante, ajudando a manter os pesos controlados e prevenindo o *overfitting*. A substituição da camada **Flatten** pela camada **Global Average Pooling (GAP)** foi a principal responsável pela drástica redução no número de parâmetros, o que favorece a generalização do modelo.

0.6 Avaliação

Para a avaliação do modelo, foram utilizadas as métricas de Acurácia, Precisão, Recall e F1-Score. Além disso, foi gerada a curva **ROC** e calculada a área sob a curva (AUC), a fim de avaliar o desempenho geral do modelo. A matriz de confusão também foi exibida para identificar os tipos de erros cometidos. A curva de perda durante o treino e a validação foi plotada para verificar indícios de *overfitting*.

Além disso, foram analisadas as imagens classificadas incorretamente. Gerou-se *feature map* do modelo, com o objetivo de identificar as áreas em que o modelo pode estar cometendo erros.

Resultados e Discussões

As novas métricas do modelo demonstram uma melhoria promissora, bem como a curva de perda do treino e validação indica uma baixa chance de estar ocorrendo *overfitting*.

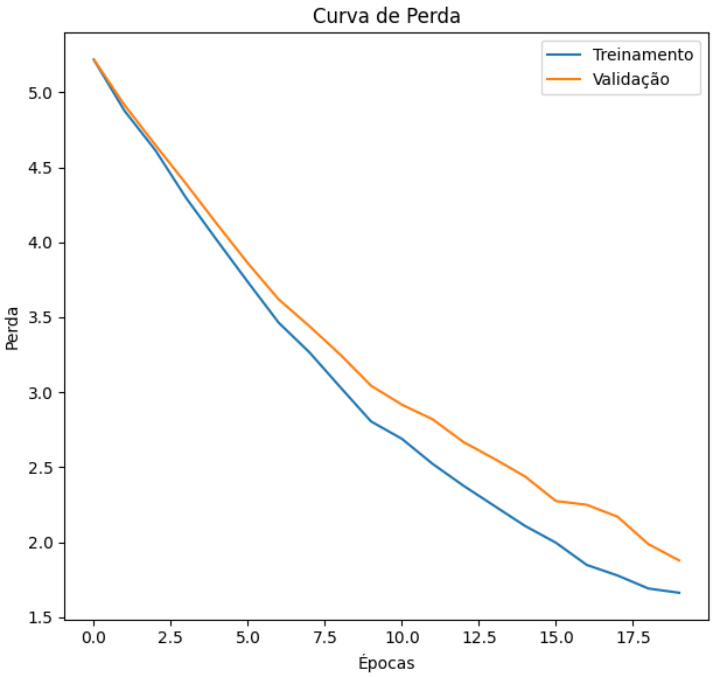

Figure 2: Curva de Perda Treino x Validação. Fonte: Própria

Métrica	**Valor**
Acurácia no Teste	0.8704
Precisão no Teste	1.0000
Recall no Teste	0.7407
F1-Score no Teste	0.8511
AUC-ROC	0.9383
Loss no Teste	1.8223

Table 5: Métricas de Avaliação do Modelo no Conjunto de Teste

A **acurácia** aponta que o modelo está acertando cerca de 87,04% das previsões do conjunto de teste, o que sugere que o modelo faz boas predições no geral.

A **precisão** de 100% aponta que, das imagens do sexo feminino que detectou, todas de fato pertenciam a essa classe. Isso poderia indicar que o modelo está deixando de

classificar corretamente algumas amostras, e acertando as poucas que classificou. Isso é melhor explorado no recall e na análise de feature maps.

O **recall** indica que conseguiu capturar cerca de 74,07% das amostras de imagens do sexo feminino. É um valor sólido, mas podemos buscar as razões e possíveis melhorias no modelo.

O **F1-Score** de 0.8511 é uma média harmônica entre precisão e recall, e um valor de 0.85 é excelente. Isso indica que o modelo tem um bom equilíbrio entre precisão e recall. Sugere que o modelo está bem ajustado para a tarefa, oferecendo boas previsões sem negligenciar excessivamente a capacidade de capturar a classe positiva. O valor é bom, especialmente considerando que há um leve desequilíbrio entre precisão e recall. Isso significa que, apesar de o modelo ter uma precisão perfeita (1.0), ele também consegue manter um bom nível de recall, equilibrando as duas métricas de maneira eficaz. A inserção de novos dados certamente contribuiu para uma melhora nessa métrica.

O **AUC-ROC (Área sob a Curva Característica de Operação do Receptor)** de 0.9383 indica que o modelo tem uma excelente capacidade de discriminação entre as classes. O AUC varia de 0 a 1, e quanto mais próximo de 1, melhor o modelo é para distinguir entre as classes positiva e negativa. Com 0.94, o modelo está muito bom em identificar quais amostras pertencem à classe positiva e quais pertencem à classe negativa, o que é uma excelente performance.

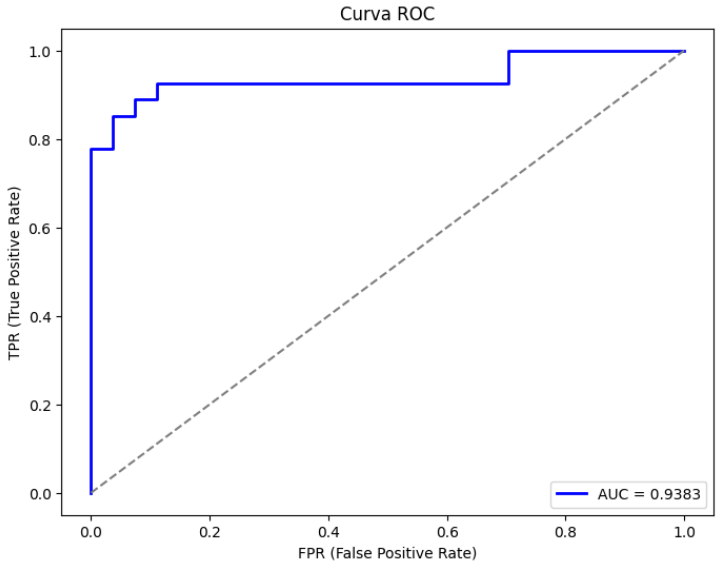

Figure 3: Curva ROC. Fonte: Própria

O **Loss (perda)** é uma medida da diferença entre as previsões do modelo e os valores reais, calculada com a função *binary crossentropy* (nesse caso). O valor de 1.8223 é relativamente alto, o que sugere que, apesar de a acurácia ser boa, o modelo ainda comete erros significativos nas previsões, especialmente em alguns casos de probabilidades próximas de 0.5 (onde o modelo tem mais incerteza). No entanto, o loss não deve ser considerado isoladamente, já que outras métricas, como acurácia e F1-score, são muito boas.

A matriz de confusão, comprova o que foi verificado no recall e na precisão: o modelo acertou todas as suas classificações 1, mas não classificou todas as imagens do sexo feminino corretamente.

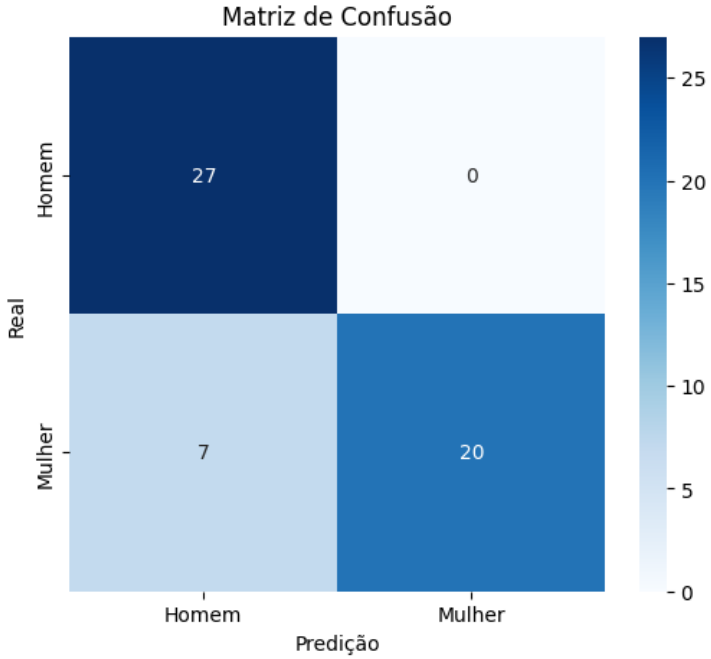

Figure 4: Matriz de Confusão. Fonte: Própria

Observamos que o modelo teve dificuldades principalmente com a classe que originalmente era minoritária. A limitação do dataset referente ao equilíbrio das classes teve um grande impacto nos resultados finais. Embora novas amostras tenham sido geradas artificialmente para balancear as classes, elas podem não ter introduzido diferenças significativas em relação às já existentes, o que comprometeu a capacidade do modelo de generalizar para a classe 1. Isso é especialmente relevante considerando que mais da metade das amostras foi gerada artificialmente. Ao analisar as imagens classificadas incorretamente, notamos que seis das sete imagens apresentam uma forte iluminação frontal, o que sugere que o modelo identificou essa característica como relevante para classificar a imagem como um rosto masculino.

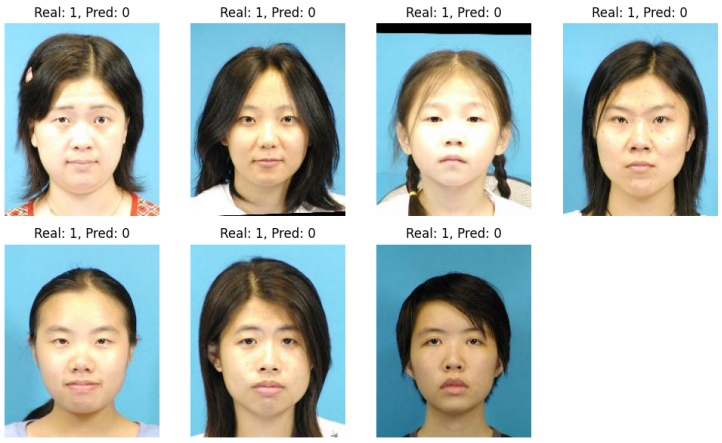

Figure 5: Imagens classificadas incorretamente. Fonte: Própria

Além disso, ao examinar os **feature maps**, observamos que muitos filtros estavam focados na parte superior da cabeça, o que indica que o modelo associou essa região ao cabelo. A sétima imagem classificada incorretamente apresenta uma região superior da

cabeça com formato arredondado – uma característica comum em rostos masculinos com cabelo curto – em contraste com o formato ligeiramente afundado observado em outros rostos, causado pelo tipo de cabelo.

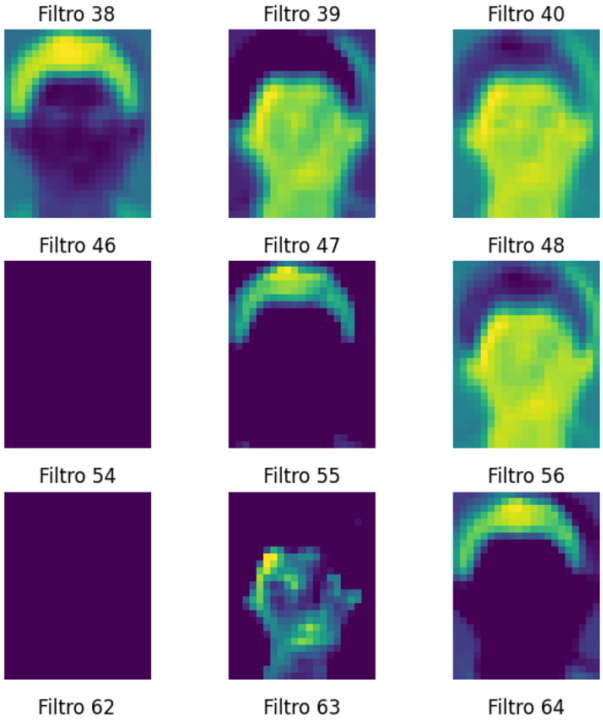

Figure 6: Alguns filtros da camada mais profunda. Fonte: Própria

Conclusão e Trabalhos Futuros

A comparação entre o modelo inicial e o final destacou a importância de um bom equilíbrio entre as classes, como evidenciado pela significativa diferença no **F1-Score** e na **AUC-ROC**. Outra descoberta importante foi o impacto da substituição da camada *Global Average Pooling* pela camada *Flatten*. Essa alteração reduziu consideravelmente a dimensionalidade e a quantidade de parâmetros, o que contribuiu para a diminuição do *overfitting*.

Acreditamos que, com uma base de dados maior e sem a necessidade de inserção artificial de amostras, o modelo poderia generalizar melhor e apresentar resultados superiores na detecção de faces femininas. Além disso, a exposição a imagens mais variadas — com diferentes ângulos, faixas etárias mais diversas e características, como a presença ou ausência de pelos faciais — poderia tornar o modelo ainda mais robusto e capaz de detectar uma maior diversidade de rostos.

Este trabalho abre portas para novas abordagens e experimentos que podem levar a um modelo mais capaz e aplicável em contextos do mundo real, como sistemas de reconhecimento facial em ambientes diversos. Com a evolução do modelo e a utilização de dados mais variados, será possível alcançar um desempenho ainda mais expressivo e garantir que ele seja eficaz em uma variedade maior de situações.

Bibliography

[1] PRATHAP, Prajeesh. *The Secret to Understanding CNNs: Convolution, Feature Maps, Pooling and Fully Connected Layers!* Medium, 2021. Disponível em: https://medium.com/@prajeeshprathap/the-secret-to-understanding-cnns-convolution-feature-maps-pooling-and-fully-co Acesso em: 30 nov. 2024.