

Fase 1 – Capacitação Tecnológica

Trilha 3 — Ciência de Dados

Relatório Técnico:

Implementação e Análise de Classificação com Redes Convolucionais

Luana Oliveira da Silva - polo Juazeiro

Elai Emylle Matos de Lima - polo Vitória da Conquista

01 de dezembro de 2024



INSTITUIÇÃO EXECUTORA



PARCEIRA



COORDENADORA



APOIO



Resumo

Este projeto explorou a aplicação de redes neurais convolucionais (CNNs) para classificar imagens do dataset CUHK Face Sketch Database (CUFS) com base no sexo biológico (masculino ou feminino). Nesse contexto, utilizando aprendizado de máquina e visão computacional, o trabalho abrangeu desde a preparação dos dados até a avaliação do modelo, abordando desafios como o desbalanceamento de classes e o tamanho limitado do dataset. Mais especificamente, a preparação incluiu etapas como a rotulação das imagens, o redimensionamento e a normalização das cores. Adicionalmente, o dataset foi dividido em 50% para treinamento, 30% para validação e 20% para teste, totalizando 188 imagens. Para lidar com o desbalanceamento (134 imagens masculinas e 54 femininas), técnicas como o oversampling com SMOTE foram utilizadas. Além disso, a arquitetura da CNN incluiu camadas convolucionais e densas, regularizadas com Dropout para evitar o overfitting. Concomitantemente, o Data Augmentation foi aplicado para aumentar a variabilidade dos dados, e também foi realizada validação cruzada com early stopping. Entretanto, os resultados mostraram desafios significativos. Por exemplo, a acurácia foi de 50%, similar à de uma classificação aleatória, enquanto o F1-score geral foi de 0,5221. Por outro lado, a matriz de confusão indicou maior precisão na classe masculina (71%) e maior recall na classe feminina (69%), refletindo um desempenho desequilibrado devido ao desbalanceamento do dataset. Ademais, a curva ROC revelou uma AUC de 0,57, indicando baixa capacidade do modelo em distinguir entre as classes. Portanto, as limitações do dataset e o desbalanceamento entre classes impactaram negativamente o aprendizado do modelo. Apesar de ajustes como a atribuição de pesos para classes e a aplicação de Data Augmentation, o desempenho permaneceu insatisfatório. Assim sendo, para superar esses desafios, são recomendadas melhorias no dataset, ajustes na arquitetura do modelo e experimentos adicionais com hiperparâmetros.

Sumário

1. Introdução	4
2. Metodologia	5
2.1 Preparação dos dados	5
2.2 Modelo CNN	7
3. Resultados	8
3.1 Acurácia e perda por Época	8
3.2 Matriz de confusão	9
3.3 Avaliando o modelo	11
3.4 Curva ROC e Área sob a Curva (AUC-ROC)	11
4. Discussão	11
5. Conclusões e trabalhos futuros	14
6. Referências	14

1. Introdução

A visão computacional é uma tecnologia que permite às máquinas reconhecer e descrever imagens de forma automática e eficiente. Atualmente, sistemas de computador acessam um grande volume de imagens e vídeos provenientes de smartphones, câmeras de trânsito, sistemas de segurança e outros dispositivos. As aplicações de visão computacional utilizam inteligência artificial (IA) e aprendizado de máquina (ML) para processar esses dados de maneira precisa, realizando tarefas como identificação de objetos, reconhecimento facial, classificação, recomendação, monitoramento e detecção (AMAZON WEB SERVICES, 2024).

Com o surgimento das imagens digitais, surgiu a necessidade de técnicas para processar essas imagens, criando um novo campo de estudo chamado Processamento Digital de Imagens (PDI). O PDI envolve a análise e manipulação de imagens digitais para identificar, extrair informações ou transformar uma imagem. Ele inclui uma variedade de softwares, hardwares e fundamentos teóricos essenciais para o tratamento de imagens (RODRIGUES, 2018).

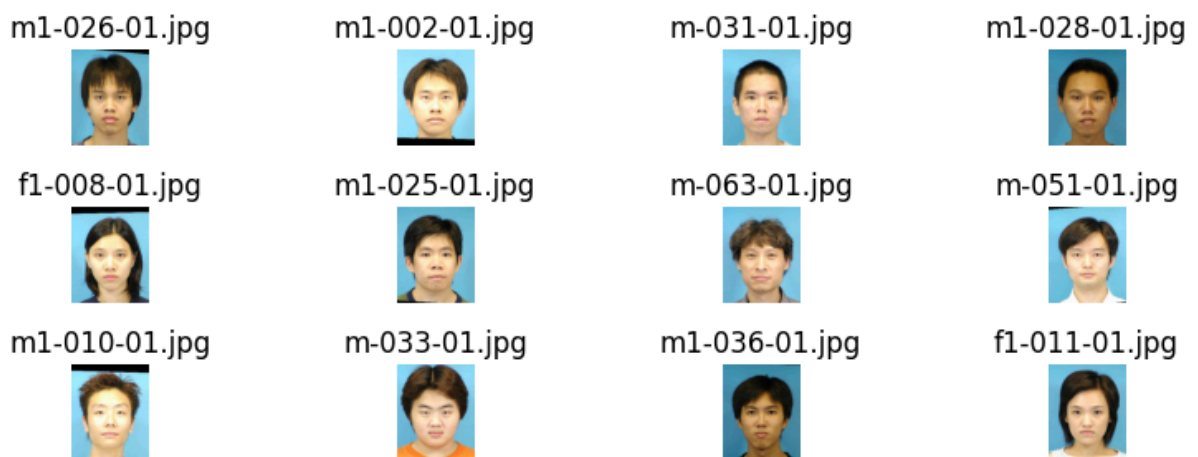
As redes neurais convolucionais (Convolutional Neural Networks ou CNNs) são uma classe de aprendizado de máquina amplamente utilizada em tarefas de classificação e visão computacional. Elas são eficazes em identificar padrões complexos nas imagens, tornando-as ideais para diversas aplicações (IBM, 2024).

Neste projeto, exploramos o uso de CNNs para classificar imagens de rostos no dataset CUHK Face Sketch Database (CUFS). Originalmente criado para mapear fotos de rostos para esboços artísticos, o CUFS contém uma rica coleção de imagens que podem ser usadas para várias tarefas de classificação. Utilizamos o dataset para classificar imagens de acordo com o sexo biológico (masculino ou feminino), aproveitando as capacidades das CNNs para extrair características visuais complexas.

O objetivo principal foi implementar um modelo CNN, enfrentando os desafios de desenvolver uma solução personalizada, desde a anotação e preparação dos dados até a análise dos resultados. Este relatório detalha cada etapa do processo, incluindo a escolha dos hiperparâmetros, a arquitetura da rede, o treinamento e a validação do modelo. Também discutimos os desafios enfrentados, como a variabilidade nas características faciais e as possíveis soluções para melhorar a precisão da classificação.

A seguir, é apresentada uma imagem com 12 exemplos aleatórios de rostos do dataset, organizados em 3 linhas e 4 colunas:

Figura 1 — Exemplos de Imagens do Dataset CUFS



Esta imagem ilustra como o modelo CNN processa e classifica as imagens de rostos, diferenciando entre rostos masculinos e femininos. Cada linha e coluna da imagem representa uma amostra classificada, evidenciando a capacidade do modelo em lidar com diferentes características faciais e sua precisão na distinção de sexo biológico.

2. Metodologia

2.1 Preparação dos dados

O dataset **CUFS** contém 188 imagens localizadas na pasta **photos**. As etapas de preparação envolveram:

1. Rotulação dos dados

Os rótulos de classe para cada arquivo de imagem foram definidos com base nos nomes dos arquivos, onde nomes iniciados com 'm' foram rotulados como 0 (Masculino) e nomes iniciados com 'f' foram rotulados como 1 (Feminino). Esses rótulos foram armazenados em uma lista chamada `class_labels`.

2. Pré-processamento

Para garantir que todas as imagens tenham dimensões uniformes, todas foram redimensionadas para 200 x 250 pixels (largura x altura). Além disso, os valores RGB das imagens foram normalizados para o intervalo [0, 1], dividindo os valores por 255. Isso ajuda a melhorar o desempenho do modelo de aprendizado de máquina.

Para exibição das imagens, foi desenvolvida uma rotina para carregar e exibir todas as imagens do dataset, verificando os tamanhos das imagens e identificando quantas estavam no formato desejado de 200 x 250 pixels. Além disso foram selecionadas aleatoriamente 12 imagens do dataset para exibição em uma grade de 3 linhas e 4 colunas, com o objetivo de ilustrar a diversidade das imagens, conforme a Figura 1.

3. Divisão do dataset

Os dados foram divididos em três conjuntos principais para o treinamento, validação e teste do modelo:

- Treinamento (50%): Inclui 94 imagens.
- Validação (30%): Inclui 56 imagens.
- Teste (20%): Inclui 38 imagens.

Inicialmente, 50% das imagens foram alocadas para o conjunto de treinamento. Os 50% restantes formaram um conjunto temporário de 94 imagens, que foi subsequentemente dividido em dois subconjuntos:

- Validação: Recebeu 60% do conjunto temporário, resultando em 56 imagens.
- Teste: Recebeu 40% do conjunto temporário, resultando em 38 imagens.

Os cálculos para essa divisão foram realizados da seguinte forma:

- O conjunto de validação recebeu 60% do conjunto temporário: $\frac{30\%}{(30\% + 20\%)} = \frac{3}{5}$
- O conjunto de teste recebeu 40% do conjunto temporário: $\frac{20\%}{(30\% + 20\%)} = \frac{2}{5}$

Para garantir a replicabilidade dos resultados, foi utilizado o parâmetro `random_state=seed` durante a partição dos dados, assegurando que a mesma divisão pudesse ser reproduzida em futuras execuções do experimento.

2.2 Modelo CNN

Para o desenvolvimento do modelo de rede neural convolucional (CNN), utilizou-se o framework `TensorFlow/Keras`. A arquitetura do modelo foi projetada da seguinte forma:

1. Balanceamento dos dados:

O dataset apresentava um desbalanceamento entre as classes, com mais imagens do sexo masculino do que sexo feminino. Para resolver isso, utilizou-se a técnica de `oversampling` com o método do `SMOTE` (Synthetic Minority Over-sampling Technique), que gera exemplos sintéticos para a classe minoritária.

2. Configuração da arquitetura do modelo:

O modelo CNN foi configurado com várias camadas convolucionais e densas, utilizando funções de ativação `ReLU` e regularização com `Dropout` para reduzir o overfitting¹. As camadas convolucionais incluíram filtros de diferentes tamanhos e camadas de `pooling` para reduzir a dimensionalidade e extrair características importantes das imagens. A camada de saída utilizou a ativação `sigmoide` para realizar a classificação binária.

3. Compilação do modelo

O modelo foi compilado utilizando o otimizador `Adam`, a função de perda `binary_crossentropy` e métricas de precisão, recall e AUC para avaliar o desempenho do modelo.

4. Treinamento do modelo

O treinamento do modelo foi realizado utilizando `data augmentation` para aumentar a variabilidade dos dados, melhorando o desempenho do modelo. Foi utilizada validação cruzada com 3 `folds` para avaliar a precisão do modelo. `Early`

¹ **Overfitting** é um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados.

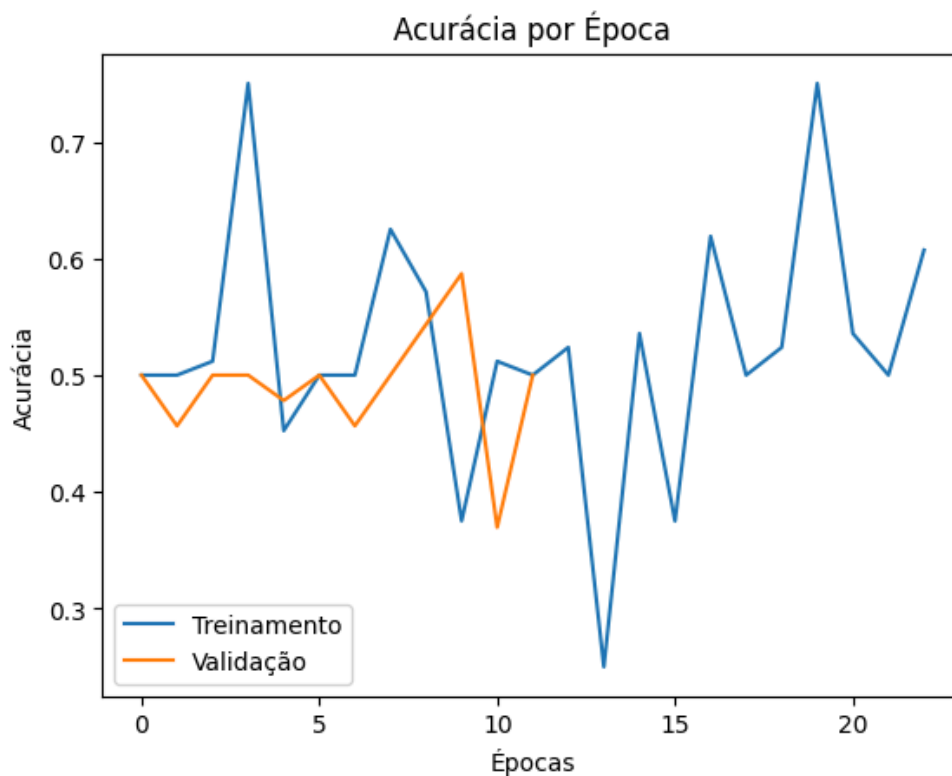
stopping foi implementado para interromper o treinamento quando o desempenho no conjunto de validação não melhorasse após um número determinado de épocas.

3. Resultados

3.1 Acurácia e perda por Época

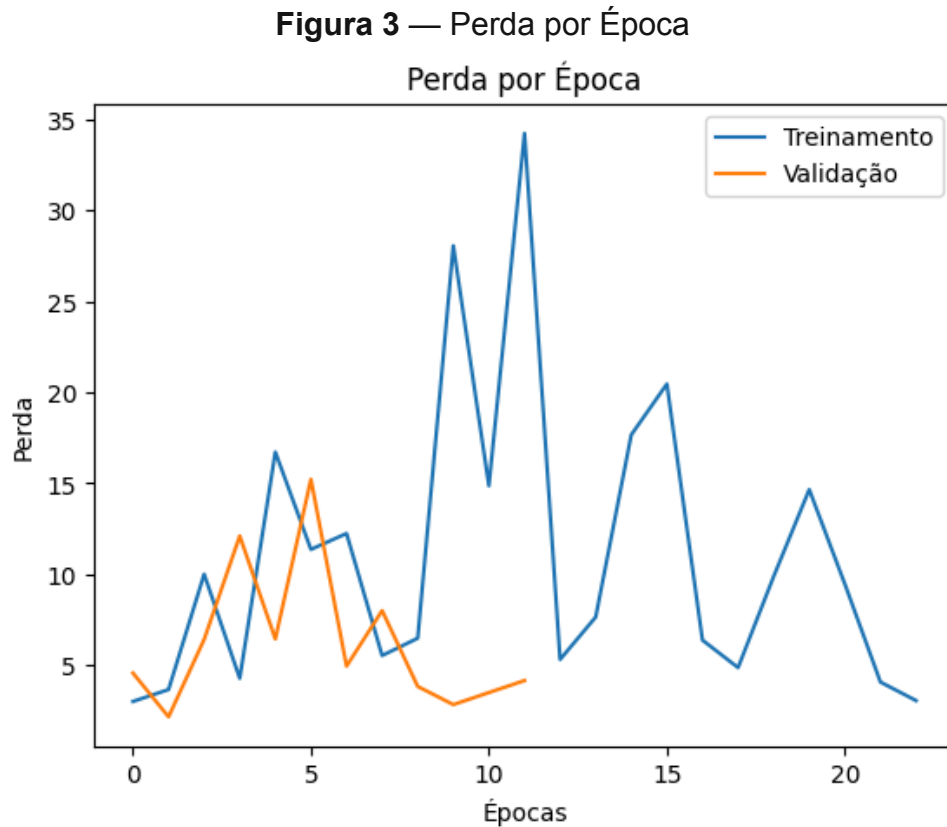
A Figura 2 apresenta o gráfico da acurácia (precisão) do modelo ao longo das épocas tanto para o conjunto de treinamento quanto para o conjunto de validação. A linha que representa a acurácia de treinamento (linha em azul) exibe flutuações significativas, enquanto a linha da validação (linha em laranja) segue o mesmo modo, mas abaixo da acurácia do treinamento, sugerindo possíveis problemas de overfitting.

Figura 2 — Acurácia por Época



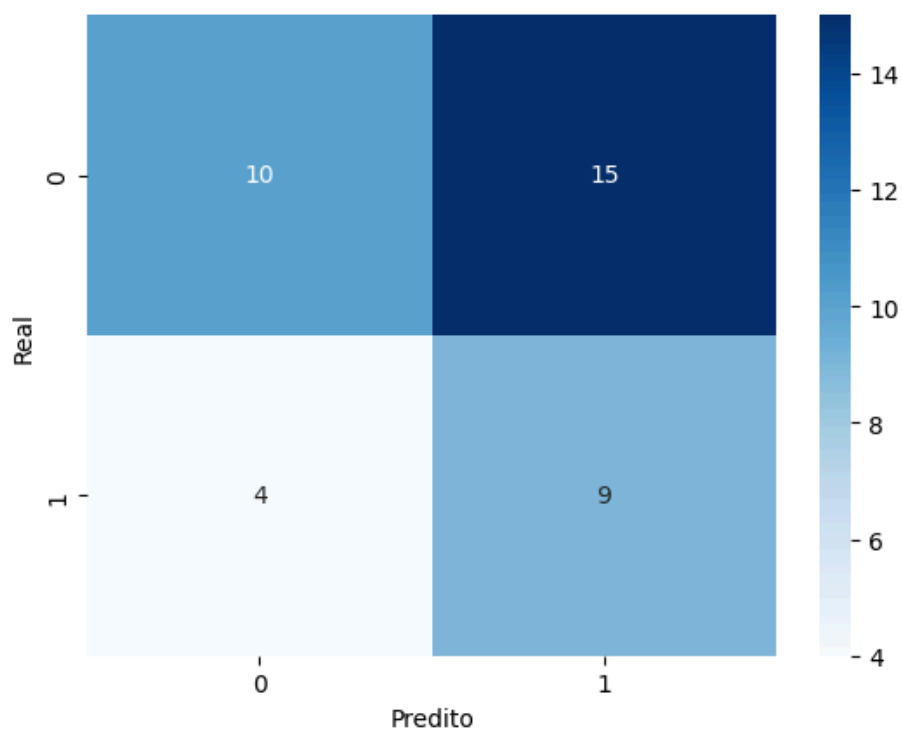
Da mesma forma, a Figura 3 ilustra a perda do modelo ao longo das épocas para os conjuntos de treinamento e validação. A linha azul representa a perda no treinamento, enquanto a linha laranja representa a perda por validação. A perda no

treinamento mostra flutuações significativas, enquanto a perda na validação exibe uma tendência geral de diminuição com menos flutuações.



3.2 Matriz de confusão

A matriz de confusão, Figura 4, fornece uma visão detalhada do desempenho do modelo ao prever as classes masculino e feminino no conjunto de teste. Assim como a contagem de predições corretas e incorretas feitas pelo modelo. No eixo x temos as predições (Predito) e no eixo y os valores reais (Real). Este gráfico é crucial para avaliar a precisão e o recall do modelo.

Figura 4 — Matriz de Confusão

Utilizando o método `classification_report` da biblioteca `sklearn.metrics`, é obtida a Tabela 1. É possível ver que todas as métricas estão presentes neste resultado do método para cada classe (0 para masculino e 1 para feminino). Tem-se a acurácia, precisão, recall e f1-score. A acurácia global foi de 50%, com uma precisão e recall mais alta para a classe masculina.

Tabela 1 — Tabelas de métricas de avaliação do modelo

	precision	recall	f1-score	support
0	0,71	0,40	0,51	25
1	0,38	0,69	0,49	13
accuracy			0,50	38
macro avg	0,54	0,55	0.50	38
weighted avg	0,60	0,50	0.50	38

3.3 Avaliando o modelo

Para avaliação mais detalhada, foram calculadas as métricas adicionais com F1-Score, perda e acurácia.

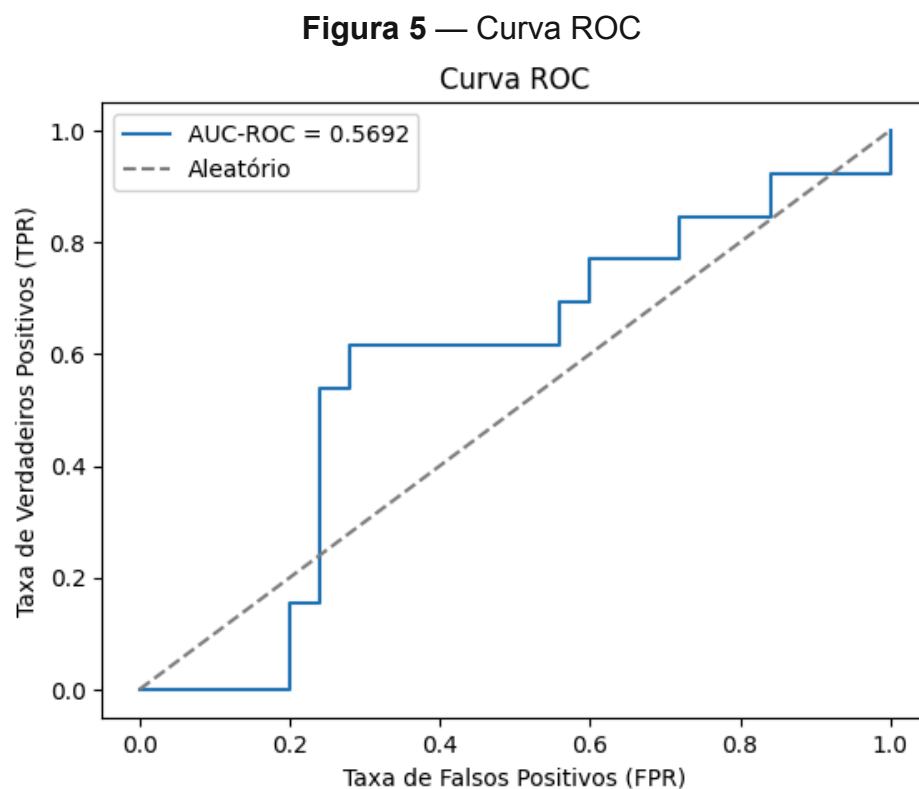
F1-score: 0,5221

Perda: 191,78%

Acurácia: 50,00%

3.4 Curva ROC e Área sob a Curva (AUC-ROC)

A área sob a curva ROC obtida foi de aproximadamente 0,57, conforme expresso na Figura 5 abaixo.



4. Discussão

O conjunto de dados possui 188 imagens, com um desequilíbrio significativo (54 mulheres e 134 homens). Este tamanho limitado tornou o problema desafiador devido ao risco de *overfitting* e baixa generalização, que foi o que ocorreu. A fim de

amenizar isso, foi utilizado a técnica de Data Augmentation, para aumentar a diversidade dos dados em cenários com poucos exemplos.

Esse desequilíbrio mencionado afetou diretamente a capacidade do modelo de prever adequadamente a classe minoritária, as mulheres. Por isso, tentou-se realizar o uso de pesos para as classes, no intuito de que o treinamento fosse mais justo, e o F1-score para fazer a avaliação ao penalizar erros na classe minoritária.

Sendo assim, os resultados obtidos neste trabalho refletem os desafios enfrentados durante o desenvolvimento do modelo de classificação, especialmente considerando as limitações do dataset e as condições impostas para o experimento.

O F1-score global alcançado foi de 0,5221, indicando um desempenho apenas moderado. Este resultado sugere que o modelo enfrenta dificuldades para equilibrar precisão e *recall*, especialmente devido ao desbalanceamento das classes no dataset, com uma representação significativamente menor da classe feminina. Essa discrepância provavelmente impactou a capacidade do modelo de aprender padrões relevantes para a classe minoritária, resultando em previsões menos assertivas para essa categoria.

Já a menor perda registrada foi de 191,78%, um valor bastante elevado que aponta para problemas na minimização da função de custo durante o treinamento. Após muito tempo realizando ajustes no modelo, após chegar a perdas de mais de 1000%, este foi o menor valor que obtivemos durante o período de execução do projeto. Este cenário pode ser explicado por diversos fatores, como a quantidade limitada de dados disponíveis, o desbalanceamento das classes, ou ainda uma configuração inadequada de hiperparâmetros. Além disso, pode-se inferir que a arquitetura do modelo não foi suficientemente otimizada para capturar as características distintivas do dataset.

Além disso, a acurácia de 50,00% reflete um desempenho semelhante ao de uma classificação aleatória. Esse resultado sugere que o modelo não conseguiu identificar padrões robustos para discriminar as classes de forma eficaz, indicando que ajustes mais profundos são necessários tanto nos dados quanto na modelagem.

Em adição a isso, analisando os resultados apresentados na matriz de avaliação indica-se que o modelo possui um desempenho desequilibrado entre as classes, refletindo as dificuldades impostas pelo desbalanceamento do dataset e pela capacidade limitada do modelo em aprender padrões consistentes.

Para a classe 0, que é a mais representada no conjunto de dados, o modelo alcançou uma *precision* de 0,71, o que significa que, entre as previsões feitas para essa classe, 71% estavam corretas. Contudo, o *recall* foi de apenas 0,40, indicando que o modelo foi capaz de identificar corretamente apenas 40% dos exemplos reais dessa classe, com muitos deles sendo erroneamente classificados como pertencentes à classe 1. O f1-score, que pondera o equilíbrio entre *precision* e *recall*, foi de 0,51, refletindo um desempenho moderado.

Por outro lado, para a classe 1, que é menos representada no conjunto de dados, o modelo apresentou uma *precision* de 0,38, sugerindo que, ao prever essa classe, o modelo não foi muito confiável, com apenas 38% de acertos. Por outro lado, o *recall* foi relativamente alto, com 69% dos exemplos reais dessa classe identificados corretamente. Isso demonstra que o modelo tem uma tendência maior a identificar exemplos da classe 1, embora isso venha às custas de muitas previsões incorretas. O f1-score para essa classe foi de 0,49, o que reflete esse equilíbrio instável.

No nível agregado, a acurácia geral do modelo foi de 50%, um resultado comparável a uma classificação aleatória. As médias macro (médias simples de *precision*, *recall* e *f1-score*) foram de 0,54, 0,55 e 0,50, respectivamente, destacando o desempenho desequilibrado entre as classes. Já as médias ponderadas, que levam em conta o tamanho de cada classe, indicaram valores de 0,60 para *precision*, e 0,50 tanto para *recall* quanto para *f1-score*. Esses números mostram que o modelo favorece a classe majoritária, o que é esperado em situações de desbalanceamento de classes.

Outra métrica utilizada para o desempenho do modelo, a Curva ROC acima, ilustra o desempenho do modelo em termos de sua capacidade de distinguir entre as classes 0 e 1. A área sob a curva (AUC-ROC), calculada em 0,5692, indica que o modelo tem um desempenho apenas ligeiramente superior ao de um classificador aleatório, cuja AUC seria 0,5. Isso evidencia que o modelo atual possui dificuldades significativas em separar corretamente as classes.

Ao analisar a curva, observa-se que ela permanece próxima da linha diagonal, que representa um comportamento aleatório. Esse padrão sugere que o modelo não conseguiu aprender padrões consistentes o suficiente para discriminar adequadamente as classes. O valor da AUC-ROC, embora acima do nível aleatório, ainda é bastante baixo, indicando que o modelo tem apenas 56,92% de chance de classificar corretamente um exemplo positivo como mais provável do que um exemplo negativo.

Essa análise é consistente com as métricas previamente discutidas. A acurácia de 50% e o f1-score em torno de 0,50 para ambas as classes reforçam a ideia de que o modelo atual apresenta limitações na tarefa de classificação. O desbalanceamento do dataset, já identificado, parece ter influenciado fortemente esses resultados, uma vez que a classe minoritária (classe 1) apresenta maior dificuldade de identificação, prejudicando o desempenho geral do modelo.

Esses resultados evidenciam as limitações do modelo atual, principalmente sua dificuldade em generalizar adequadamente e em lidar com o desbalanceamento das classes. Embora ele tenha um desempenho ligeiramente melhor para a classe 0, não consegue ser confiável ao prever a classe 1, o que compromete sua aplicabilidade.

5. Conclusões e trabalhos futuros

Foram aplicados os conhecimentos obtidos na trilha tecnológica de Ciência de Dados para montar um modelo de classificação de imagens utilizando uma rede neural convolucional.

Houve overfitting no modelo, apesar do uso de métodos como o SMOTE (*Synthetic Minority Oversampling Technique*), para ajudar a equilibrar melhor o conjunto de dados e melhorar o aprendizado do modelo, técnicas de regularização, como Dropout e L2 Regularization, e validação cruzada.

Com isso, recomenda-se revisar a arquitetura do modelo, explorando ajustes como o aumento ou redução da complexidade com a inclusão ou exclusão de mais camadas convolucionais ou filtros. Bem como, testar outros valores para parâmetros.

Por fim, é essencial ampliar o dataset, tanto em quantidade quanto em diversidade. A inclusão de novas imagens que capturem diferentes variações de iluminação, ângulos e outros fatores contextuais pode ajudar no treinamento do modelo.

6. Referências

AMAZON Web Services. **O que é visão computacional?** Disponível em: <https://aws.amazon.com/pt/what-is/computer-vision/>. Acesso em: 30 nov. 2024.

RODRIGUES, Diego Alves. **Deep Learning e Redes Neurais Convolucionais: Reconhecimento Automático de Caracteres em Placas de Licenciamento Automotivo**. João Pessoa: Universidade Federal da Paraíba, 2018. Monografia (Bacharelado em Ciência da Computação) – Centro de Informática, Universidade Federal da Paraíba.

IBM. **O que são Redes Neurais Convolucionais?** Disponível em: <https://www.ibm.com/br-pt/topics/convolutional-neural-networks>. Acesso em: 30 nov. 2024.