Classificação de Sneakers Utilizando Pytorch

1st Carlos Emanuel Silva e Melo Oliveira
Departamento de Engenharia de Software
Pontifícia Universidade Católica (PUC)
Belo Horizonte, Brasil
1405439@sga.pucminas.br

2nd Giovanni Bogliolo Sirihal Duarte
Departamento de Engenharia de Software
Pontifícia Universidade Católica (PUC)
Belo Horizonte, Brasil
939967@sga.pucminas.br

3rd Gustavo Andrade Alves
Departamento de Engenharia de Software
Pontifícia Universidade Católica (PUC)
Belo Horizonte, Brasil
1405661@sga.pucminas.br

4th Vitor Nunes Calhau
Departamento de Engenharia de Software
Pontifícia Universidade Católica (PUC)
Belo Horizonte, Brasil
1380847@sga.pucminas.br

***Abstract*—With the increasing demand for automated product identification in e-commerce and retail inventory systems, computer vision techniques have become essential for improving user experience and operational efficiency. This study explores the use of GoogLeNet [1], a convolutional neural network based on the Inception architecture, for the image-based classification of sneakers. The model was trained using a publicly available dataset containing over 5,000 labeled sneaker images and implemented with the PyTorch framework. The proposed solution achieved a validation accuracy of 74.72%, demonstrating effective generalization to unseen data. Additionally, a confusion matrix analysis highlighted the model's ability to distinguish among visually similar classes. A simple graphical interface was developed to showcase the practical application of the classifier. The results suggest that GoogLeNet is a viable architecture for sneaker classification tasks and can be integrated into real-world retail solutions such as visual search engines and automatic cataloging systems.**

***Resumo*—Com o aumento da demanda por identificação automatizada de produtos em sistemas de e-commerce e controle de estoque no varejo, técnicas de visão computacional têm se tornado essenciais para aprimorar a experiência do usuário e a eficiência operacional. Este estudo investiga o uso da GoogLeNet [1], uma rede neural convolucional baseada na arquitetura Inception, para a classificação de imagens de tênis (sneakers). O modelo foi treinado utilizando um conjunto de dados público com mais de 5.000 imagens rotuladas e implementado com a biblioteca PyTorch. A solução proposta alcançou uma acurácia de validação de 74,72%, demonstrando boa capacidade de generalização para dados não vistos. A análise da matriz de confusão evidenciou a habilidade do modelo em distinguir entre classes visualmente semelhantes. Além disso, foi desenvolvida uma interface gráfica simples para demonstrar a aplicação prática do classificador. Os resultados indicam que a GoogLeNet é uma arquitetura viável para tarefas de classificação de sneakers e pode ser integrada a soluções reais no varejo, como mecanismos de busca visual e sistemas de catalogação automática.**

***Index Terms*—GoogleNet, Deep Learning, Sneakers Classification, Convolutional Neural Network, PyTorch**

I. Introdução

Com o crescimento acelerado do comércio eletrônico e o aumento da competitividade entre marketplaces, soluções baseadas em inteligência artificial têm ganhado destaque na otimização de processos relacionados à experiência do usuário e à gestão de estoques. Uma das demandas mais recorrentes nesse cenário é a identificação automática de produtos a partir de imagens, especialmente em categorias com grande diversidade visual, como calçados.

No caso específico dos tênis esportivos (*sneakers*), a variedade de modelos, cores, marcas e estilos representa um desafio significativo para sistemas tradicionais de busca por texto ou código de produto. Usuários frequentemente têm dificuldade em localizar um item específico com base apenas em palavras-chave, o que pode resultar em experiências frustrantes e impactar negativamente as vendas. Da mesma forma, o controle de inventário em lojas físicas ou centros de distribuição requer métodos eficientes de catalogação automática, capazes de lidar com grandes volumes de dados visuais.

Nesse contexto, o presente trabalho propõe o uso da arquitetura Inception, implementada na forma do modelo GoogLeNet [1], para treinar uma rede neural convolucional capaz de realizar a classificação automática de sneakers a partir de imagens. A base de dados utilizada para o treinamento é o conjunto disponibilizado por Gegenava [2] na plataforma Kaggle, que reúne imagens de tênis categorizadas por tipo e modelo.

A arquitetura GoogLeNet foi originalmente projetada para maximizar o desempenho em competições como a *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), sendo reconhecida por sua eficiência computacional e capacidade de extração multiescalar de características visuais por meio dos chamados *Inception Modules*. Contudo, seu uso em contextos especializados como o reconhecimento de calçados requer adaptações, principalmente no ajuste de pesos e na reconfiguração de camadas para melhor atender às especificidades do novo domínio.

O objetivo deste artigo é investigar o desempenho da arquitetura GoogLeNet adaptada para o problema de classificação de *sneakers*, avaliando sua aplicabilidade em cenários de uso como buscas visuais em marketplaces e automação da catalogação de estoques em lojas. Espera-se, com isso, contri-

buir para o avanço de soluções baseadas em visão computacional voltadas ao setor varejista, oferecendo uma alternativa eficaz, escalável e de boa acurácia para tarefas de identificação de produtos por imagem.

II. Método

A. Ambiente de Execução

O treinamento e a execução do modelo foram realizados na plataforma Google Colab, que oferece infraestrutura computacional em nuvem com acesso gratuito a unidades de processamento gráfico (GPUs). Para este trabalho, utilizou-se uma GPU NVIDIA Tesla T4, com 16 GB de memória, que oferece um bom equilíbrio entre desempenho computacional e acessibilidade, sendo adequada para tarefas de treinamento de redes neurais convolucionais de porte médio. Essa escolha permitiu acelerar o processo de treinamento, reduzir o tempo de experimentação e viabilizar a utilização da arquitetura GoogLeNet sem a necessidade de infraestrutura local de alto desempenho.

O conjunto de dados utilizado neste estudo foi obtido a partir da plataforma Kaggle, por meio do repositório "Sneakers Classification" [2]. O dataset contém mais de 5.000 imagens de tênis, representando diferentes tipos de sneakers. As imagens são coloridas, com resolução variável, e foram organizadas em pastas conforme a classe correspondente. Esse dataset apresenta um bom grau de variação visual, incluindo diferentes ângulos, cores e estilos, o que torna o problema de classificação desafiador e ao mesmo tempo representativo de situações reais encontradas em marketplaces e estoques de varejo.

A arquitetura utilizada para o modelo de classificação foi a GoogLeNet, implementada com base na biblioteca PyTorch. Essa rede convolucional profunda é composta por múltiplos Inception Blocks, que realizam operações convolucionais em paralelo com diferentes tamanhos de *kernel*, permitindo a extração de características em múltiplas escalas. Essa abordagem foi originalmente proposta para maximizar a eficiência e precisão em grandes conjuntos de dados, como o ImageNet, e mostrou-se adequada para lidar com a diversidade visual presente nas imagens de calçados. Para o presente trabalho, a arquitetura foi ajustada de forma a adaptar sua camada de saída ao número de classes do dataset de sneakers.

Além do processo de treinamento e avaliação do modelo, foi desenvolvida uma interface simples de inferência. Essa interface permite carregar imagens de tênis e visualizar, de forma interativa, a predição da classe realizada pelo modelo treinado. Tal recurso foi útil para validar qualitativamente o desempenho da rede em situações não vistas durante o treinamento, aproximando o sistema de um cenário de uso real.

B. Metodologia

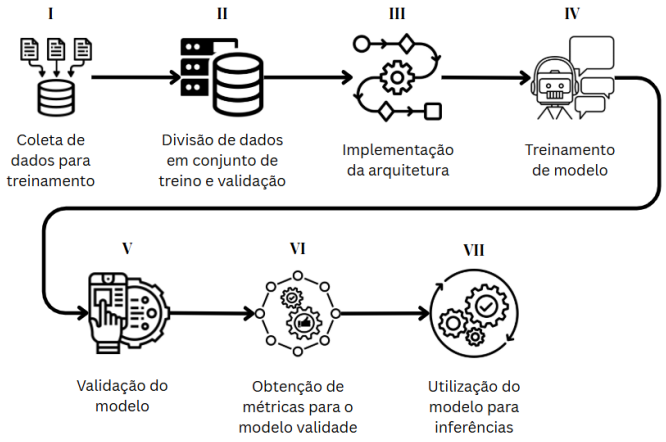

Figura 1. Diagrama da metodologia.

A metodologia adotada neste trabalho foi estruturada em sete etapas principais, conforme ilustrado na Figura 1 Cada fase foi pensada para garantir a construção de um modelo robusto, desde o preparo dos dados até a aplicação prática da rede treinada.

O processo teve início com a preparação das imagens, que foram organizadas em classes de acordo com a taxonomia presente no conjunto de dados. Essa organização permitiu estruturar os diretórios de forma compatível com ferramentas de carregamento de dados do PyTorch.

Em seguida, os dados foram divididos em subconjuntos de treinamento e validação, com o objetivo de possibilitar a avaliação do desempenho do modelo em dados não vistos. A divisão preservou a proporção entre as categorias, evitando viés de classe.

A arquitetura adotada foi baseada no modelo GoogLeNet, utilizando blocos Inception. A estrutura foi adaptada para o número de classes do problema, mantendo a modularidade característica do modelo original. As imagens foram redimensionadas para 300×300 pixels e submetidas a técnicas de aumento de dados, como inversão horizontal e variação de brilho e contraste.

A rede foi treinada por múltiplas épocas com ajustes progressivos nos hiperparâmetros, utilizando o otimizador Adam e a função de perda *CrossEntropyLoss*. Durante esse processo, a acurácia no conjunto de validação foi monitorada a cada época para acompanhar a evolução do desempenho do modelo.

Ao final do treinamento, foram realizadas inferências sobre os dados de validação com o objetivo de verificar a capacidade de generalização da rede. Embora análises detalhadas de erros não tenham sido conduzidas formalmente, a avaliação da acurácia serviu como referência para a escolha do modelo final.

Foram coletadas métricas quantitativas como a perda (*training loss*) e a acurácia no conjunto de validação, além da análise da matriz de confusão. Essas informações permitiram

acompanhar a convergência do modelo e embasaram a seleção da versão final para aplicação prática.

Por fim, o modelo treinado foi integrado a uma interface gráfica simples, permitindo ao usuário carregar imagens e obter classificações diretamente. Essa etapa visa demonstrar a aplicabilidade prática da solução proposta em contextos como catalogação de produtos ou busca por imagem.

III. Resultados

Após o treinamento, o modelo alcançou os seguintes resultados sobre o conjunto de validação.

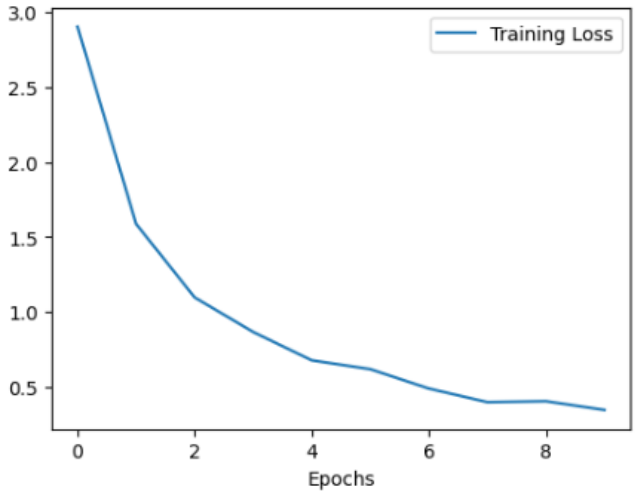

Figura 2. Grafico de Training Loss.

Conforme ilustrado na Figura 2, a função de perda apresentou uma redução significativa ao longo do processo de treinamento. Na primeira época, o valor inicial era de 2,9011, indicando um alto grau de erro nas previsões iniciais do modelo. Ao final do treinamento, esse valor caiu para 0,3494, o que demonstra que a rede foi capaz de aprender os padrões presentes nos dados de treinamento e reduzir substancialmente os erros cometidos. Esse comportamento é esperado em modelos bem treinados, pois a perda tende a diminuir gradualmente à medida que os pesos são ajustados para minimizar a discrepância entre as previsões e os rótulos verdadeiros.

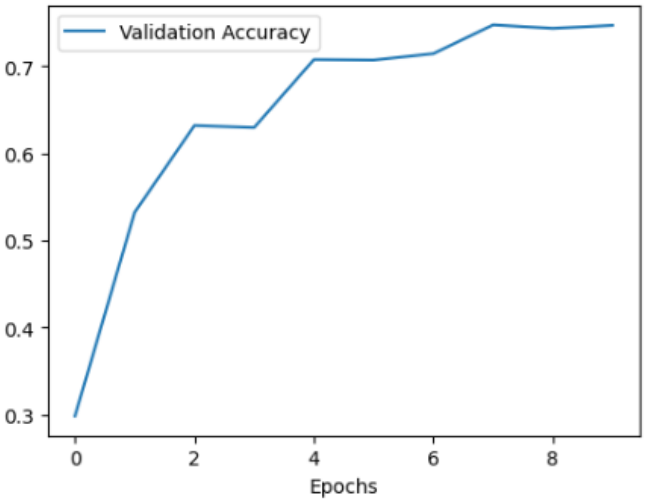

Figura 3. Grafico de Validation Accuracy.

Conforme ilustrado na Figura 3, a acurácia no conjunto de validação apresentou uma melhora considerável ao longo das épocas. Inicialmente, o modelo atingiu uma acurácia de 0,2984, o que indica um desempenho pouco superior ao acaso em um problema com múltiplas classes. Ao final do treinamento, a acurácia alcançou 0,7472, refletindo uma capacidade significativamente maior do modelo em classificar corretamente as imagens não vistas durante o treinamento. Esse aumento consistente é um indicativo de que o modelo não apenas aprendeu os padrões dos dados de treinamento, mas também foi capaz de generalizar para novos exemplos, dentro dos limites do conjunto de validação utilizado.

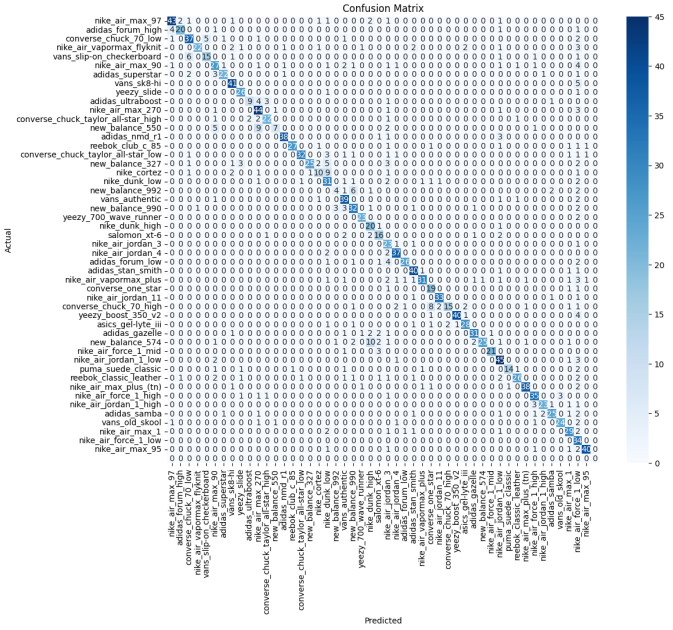

Figura 4. Matriz de confusão.

A Figura 4 exibe a matriz de confusão gerada a partir das predições do modelo no conjunto de validação. A matriz

apresenta a contagem de classificações reais (linhas) versus as classificações previstas pelo modelo (colunas), permitindo uma análise detalhada da distribuição de acertos e erros por classe.

Observa-se uma concentração acentuada de valores na diagonal principal, o que indica que o modelo acertou corretamente a maioria das previsões. Esse padrão sugere que a rede foi bem-sucedida em aprender os padrões distintivos das diferentes classes de tênis presentes no conjunto de dados.

No entanto, também é possível identificar alguns casos de confusões entre classes semelhantes. Por exemplo, alguns modelos da linha Nike Air Max ou Jordan apresentaram pequenas taxas de confusão entre si, possivelmente devido a semelhanças visuais nos designs ou cores predominantes. Esses erros são esperados em tarefas de classificação multiclasse com alta similaridade entre categorias.

A matriz também confirma a distribuição equilibrada das amostras entre as classes, o que corrobora a validade dos resultados obtidos. Essa análise reforça a importância da matriz de confusão como ferramenta complementar às métricas globais, fornecendo uma visão mais granular da performance do modelo.

IV. Conclusão

Os resultados obtidos neste trabalho demonstram o potencial da arquitetura GoogLeNet, implementada com a biblioteca PyTorch, para a tarefa de classificação automática de tênis (*sneakers*) a partir de imagens. O modelo alcançou uma acurácia de validação de aproximadamente 74,72%, indicando uma capacidade significativa de generalização, mesmo diante da diversidade visual presente no conjunto de dados.

A redução consistente da função de perda ao longo das épocas e a predominância de acertos na diagonal principal da matriz de confusão reforçam a eficácia da abordagem adotada. Embora ainda existam desafios relacionados à distinção entre classes visualmente semelhantes, os resultados apontam para a viabilidade do uso de redes neurais convolucionais em aplicações práticas, como sistemas de busca visual em *marketplaces* ou ferramentas de catalogação automatizada de produtos no varejo.

Como trabalhos futuros, recomenda-se explorar arquiteturas mais recentes, como EfficientNet ou Vision Transformers (ViT), bem como aplicar técnicas de fine-tuning mais agressivas e estratégias avançadas de aumento de dados. Além disso, a avaliação em cenários reais e a ampliação da base de dados podem contribuir para a consolidação e robustez da solução proposta.

Referências

[1] C. Szegedy et al., ”Going deeper with convolutions,”2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594. keywords: Computer architecture;Convolutional codes;Sparse matrices;Neural networks;Visualization;Object detection;Computer vision,

[2] Gegenava,Nikolas. “Popular Sneakers Classification” Kaggle. Ultima atualização em: 02 de Maio de 2025. Dataset. Disponível em: [https://www.kaggle.com/datasets/nikolasgegenava/sneakers-classification]. Acesso em: 14 de maio de 2025.