

Visão Computacional Aplicada à Acessibilidade: Classificação de Vestuário com Redes Neurais Convolucionais para Pessoas com Deficiência Visual

Jacqueline Navarro da Silva

Universidade Federal Rural de Pernambuco

Programa de Pós Graduação em Informática Aplicada

Aprendizado Profundo para Visão Computacional

Rua Manuel de Medeiros, s/n, Dois Irmãos, Recife-PE, CEP 52171-900

Email: jacqueline.navarro@ufrpe.br

Resumo—O desenvolvimento de sistemas assistivos baseados em Visão Computacional é essencial para ampliar a autonomia de Pessoas com Deficiência Visual (PDV), especialmente em tarefas cotidianas como a identificação de vestuário. Este artigo apresenta uma avaliação comparativa de quatro arquiteturas de Redes Neurais Convolucionais (GoogLeNet, ResNet-50, MobileNet-v2 e EfficientNet-B0), aplicando Transfer Learning ao dataset DeepFashion-1 para a classificação de 17 categorias de roupas. O modelo EfficientNet-B0 obteve o melhor desempenho inicial, com acurácia próxima de 71% no conjunto de teste. Em uma segunda fase, técnicas de otimização como Class Weighting e Fine-Tuning foram aplicadas, mas resultaram em aumento do overfitting e leve queda na acurácia, evidenciando que a ponderação de perda isolada não é suficiente para lidar com o desbalanceamento severo das classes. Concluímos que o EfficientNet-B0 é a arquitetura mais promissora para sistemas assistivos, mas ressaltamos a necessidade de estratégias mais robustas de balanceamento de dados, como oversampling, para melhorar a generalização em classes minoritárias e garantir maior confiabilidade em aplicações reais.

I. INTRODUÇÃO

A visão computacional deixou de ser apenas uma disciplina técnica para tornar-se uma força transformadora na sociedade. Impulsionada pelos avanços das Redes Neurais Convolucionais (DCNNs) [1], ela hoje sustenta aplicações que vão do diagnóstico médico à mobilidade urbana. Mas talvez seu impacto mais humano esteja nas tecnologias assistivas, capazes de ampliar a autonomia de pessoas com deficiência visual (PDV) [2].

Entre os desafios cotidianos enfrentados por esse público, a identificação correta de roupas, categoria e cor, é mais do que uma questão estética: trata-se de inclusão social e independência [3]. Este projeto busca responder a essa necessidade com soluções de baixo custo e alta eficiência, explorando o poder do Transfer Learning para classificação de vestuário em tempo real.

Este estudo compara quatro arquiteturas contemporâneas de DCNNs pré-treinadas (GoogLeNet [4], ResNet-50 [5], MobileNet-v2 [6] e EfficientNet-B0 [7]) aplicadas ao dataset de referência DeepFashion-1 disponibilizado na plataforma Kaggle [8]. Mais do que avaliar métricas de acurácia, a

contribuição central reside em discutir o equilíbrio entre desempenho e eficiência, aspecto decisivo para a implementação em dispositivos com recursos computacionais restritos. Adicionalmente, investigamos estratégias de balanceamento de classes, evidenciando os desafios inerentes ao Fine-Tuning em cenários desbalanceados. Ao articular rigor técnico com relevância social, este trabalho posiciona a visão computacional não apenas como uma ferramenta de classificação, mas como um recurso efetivo de inclusão e autonomia para pessoas com deficiência visual.

II. TRABALHOS RELACIONADOS

A classificação de roupas na visão computacional fundamenta-se em dois pilares principais: a eficácia do Transfer Learning com DCNNs e a análise comparativa entre diferentes arquiteturas de rede.

O primeiro pilar refere-se ao uso de DCNNs pré-treinadas. O Transfer Learning consiste em aproveitar os pesos de modelos treinados em grandes bases de dados de referência, como o ImageNet, e adaptá-los a novos problemas por meio do reajuste das camadas finais [5]. Essa abordagem acelera o processo de treinamento, reduz o risco de overfitting e gera modelos mais robustos, sendo amplamente adotada em cenários com restrições de tempo e recursos computacionais.

O segundo pilar envolve a comparação entre arquiteturas. O estudo que serviu de referência para este trabalho, voltado à classificação de roupas para pessoas com deficiência visual (PDV) [9], identificou a GoogLeNet (Inception v1) como a arquitetura mais adequada. Entretanto, desde sua introdução, novas topologias mais eficientes foram propostas. A família ResNet [1] revolucionou o campo ao introduzir conexões residuais, permitindo o treinamento de redes substancialmente mais profundas e mitigando o problema do vanishing gradient. Posteriormente, surgiram arquiteturas otimizadas para dispositivos móveis, como a MobileNet-v2 [6], que emprega convoluções separáveis em profundidade (depthwise separable convolutions) para reduzir drasticamente o número de parâmetros sem comprometer a acurácia. Mais recentemente, o EfficientNet [7] estabeleceu um novo paradigma ao aplicar

compound scaling, equilibrando largura, profundidade e resolução da rede para alcançar maior eficiência e precisão.

Diante desse cenário, torna-se relevante comparar arquiteturas clássicas e modernas em relação ao modelo de referência GoogLeNet, os 3 modelos de aprendizado profundo selecionados para este estudo foram ResNet-50, MobileNet-v2 e EfficientNet-B0. Essa análise busca identificar qual topologia contemporânea oferece o melhor equilíbrio entre desempenho e custo computacional na tarefa específica de classificação de vestuário.

III. METODOLOGIA

A metodologia de classificação foi dividida em três fases: preparação do dataset, definição e treinamento comparativo das arquiteturas base, e otimização do melhor modelo.

A. Dataset e Pré-Processamento

O dataset utilizado foi o DeepFashion-1 [8], composto por imagens de vestuário em diferentes ângulos e contextos. Foram extraídas 11.484 imagens dos diretórios de treino e teste, categorizadas em 17 classes de vestuário, Figura 1.



Figura 1. Mosaico das primeiras 10 imagens do dataset DeepFashion-1.

1) *Desbalanceamento e Divisão dos Dados*: O principal desafio deste dataset é o severo desbalanceamento de classes. Conforme ilustrado na Figura 2, a classe majoritária, Tees_Tanks, possui 2.885 amostras (25.12% do total), enquanto a classe minoritária, Suiting, possui apenas 8 amostras (0.07%), resultando em uma proporção de 360:1.

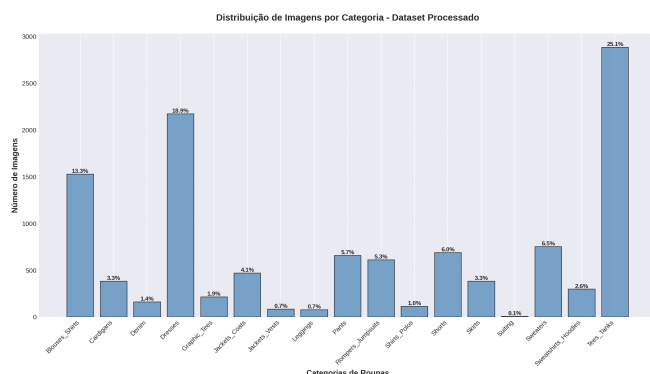


Figura 2. Distribuição de imagens por categoria do dataset processado, evidenciando o severo desbalanceamento de classes.

Embora a divisão estratificada fosse o ideal para este dataset, foi adotada a divisão aleatória simples para a baseline e a

otimização, reconhecendo a complexidade de garantir a representatividade de classes extremamente raras (como Suiting) em todos os splits do código inicial:

- Treinamento: 70% (8.038 imagens)
- Validação: 15% (1.722 imagens)
- Teste: 15% (1.724 imagens)

A proporção de amostras em cada split é fundamental para o Transfer Learning e para evitar que a complexidade do dataset se tornasse um obstáculo insuperável no tempo limite do projeto.

B. Pré-Processamento e Aumento de Dados

O pré-processamento das imagens seguiu as diretrizes de Transfer Learning para modelos pré-treinados no ImageNet:

- **Redimensionamento e Augmentation:** As imagens foram redimensionadas para 224×224 pixels. Para aumentar a robustez do conjunto de treinamento e simular variações do mundo real, foram aplicadas técnicas de Aumento de Dados (Data Augmentation), incluindo `RandomHorizontalFlip`, `RandomRotation` (até 15°), e `ColorJitter` (ajuste aleatório de brilho, contraste e saturação).
- **Normalização:** Todas as imagens, tanto de treino quanto de teste/validação, foram normalizadas utilizando os valores de média e desvio-padrão do ImageNet.

C. Arquiteturas e Treinamento Comparativo (Baseline)

Esta fase estabeleceu a *baseline* de desempenho, avaliando quatro arquiteturas DCNNs (Deep Convolutional Neural Networks), todas inicializadas com pesos pré-treinados do ImageNet (*Transfer Learning*): GoogLeNet, ResNet-50, MobileNet-v2 e EfficientNet-B0.

- Reajuste da Camada FC: Para cada modelo, a camada totalmente conectada (FC) final, originalmente dimensionada para 1.000 classes, foi substituída por uma camada linear adaptada para as 17 classes de saída do DeepFashion-1.
- Função de Perda: Foi utilizada a *Cross-Entropy Loss*.
- Otimizador e Scheduler: Utilizou-se o otimizador Adam (LR = 1×10^{-3}) por 10 épocas. Um *scheduler* ReduceLROnPlateau foi empregado, monitorando a perda de validação.

D. Otimização do Melhor Modelo (EfficientNet-B0)

O modelo de melhor desempenho na *baseline* foi o *EfficientNet-B0* (Acurácia de Teste: 70.79% / F1-Score: 70.26%), Tabela I. Esta fase focou na otimização utilizando técnicas de regularização e balanceamento, cujos detalhes são apresentados a seguir:

- 1) **Class Weighting:** Pesos de classe, calculados de forma inversamente proporcional à frequência da classe no conjunto de treinamento, foram aplicados à função de perda *Cross-Entropy Loss*. O objetivo principal era penalizar erros nas classes minoritárias e mitigar o severo desbalanceamento do dataset.

2) Fine-Tuning e Regularização: O otimizador foi ajustado para *AdamW* com um *Learning Rate* (LR) mais conservador de 1×10^{-4} e um *Weight Decay* (Regularização L2) de 5×10^{-4} . Para reforçar a prevenção contra *overfitting*, foi adicionado um *Dropout* de 0.2 na camada classificadora.

3) Early Stopping: Foi implementado com uma paciência de 5 épocas, resultando em um treinamento total de 17 épocas antes de ser interrompido. Esta técnica visa evitar a super-adaptação do modelo aos dados de treino, interrompendo o treinamento quando a métrica de validação (acurácia) não melhora.

4) Test-Time Augmentation (TTA): Esta técnica foi utilizada na avaliação final (após o treinamento), aumentando a robustez da predição. Consistiu em fazer a média das *softmax probabilities* da imagem original e da imagem espelhada horizontalmente.

O processo de Fine-Tuning e as técnicas de regularização, como o Dropout e o Weight Decay, foram aplicados em conjunto com o Early Stopping para monitorar o desempenho do modelo em tempo real. O monitoramento das curvas de Loss e Acurácia de treino e validação a cada época foi crucial para identificar o ponto de saturação da generalização e prevenir a super-adaptação do modelo aos dados de treinamento.

Vale destacar a diferença entre as avaliações "Otimizado" e "Otimizado + TTA". No primeiro caso, o modelo apresenta sua performance no conjunto de teste sem a aplicação do Test-Time Augmentation (TTA), refletindo a acurácia padrão. Já no segundo, o mesmo modelo é avaliado com o uso do TTA apenas na fase de inferência. Essa comparação busca mostrar como o TTA contribui para tornar a predição final mais robusta e estável.

E. Análise dos Resultados da Otimização

Apesar da aplicação das técnicas de otimização, o modelo *baseline* não foi superado. A acurácia de teste final piorou em 1.65% e o *F1-Score* (Macro) em 14.74%. A técnica de *Class Weighting* (inversamente proporcional) provou ser muito agressiva para este *dataset*, penalizando excessivamente as classes majoritárias (que representam a maior parte dos dados) e levando a um aumento significativo no *overfitting* (salto de 18.68% para 31.42%), superando a capacidade de mitigação das técnicas de regularização, Tabela II.

F. Reprodutibilidade do Estudo

Para garantir a total transparência e reprodutibilidade dos resultados, o código completo do *pipeline* de processamento e treinamento, juntamente com todos os gráficos e dados gerados, estão publicamente disponíveis no repositório *GitHub* do autor [10].

IV. RESULTADOS E DISCUSSÃO

A avaliação do desempenho dos modelos foi realizada em duas etapas principais: a comparação entre quatro arquiteturas baseadas em Transfer Learning e a posterior otimização do modelo com melhor desempenho, conforme detalhado nas

subseções seguintes. Todo o processo de treinamento e testes foi executado na plataforma Kaggle, utilizando o ambiente com GPU NVIDIA Tesla T4 (CUDA) [11], com duração de 10 épocas para a fase comparativa.

A. Comparação das Arquiteturas Base (*GoogLeNet*, *ResNet-50*, *MobileNet-v2*, *EfficientNet-B0*)

Os quatro modelos foram treinados e avaliados no conjunto de teste, e seus resultados de acurácia, precisão, recall e F1-Score (média ponderada) são apresentados na Tabela I.

Tabela I
COMPARAÇÃO DE MÉTRICAS DE DESEMPENHO PARA OS QUATRO
MODELOS BASE (APÓS 10 ÉPOCAS).

Modelo	Train Acc (%)	Val Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1-Score (%)	Épocas	Tempo (s)
GOOGLENET	74.05	62.66	63.07	63.67	63.07	62.73	10	955.0
RESNET50	74.51	64.17	66.84	68.71	66.84	65.85	10	1060.0
MOBILENET_V2	80.17	66.20	67.48	68.59	67.48	67.22	10	921.6
EFFICIENTNET_B0	89.53	70.85	70.79	70.45	70.79	70.26	10	925.2

Conforme a Tabela I, o modelo *EfficientNet-B0* demonstrou o melhor desempenho em todas as métricas no conjunto de teste, atingindo 70.79% de acurácia e 70.26% de F1-Score. O *MobileNet-v2* e o *ResNet-50* apresentaram desempenho ligeiramente inferior, mas superior ao modelo de referência.

O modelo *GoogLeNet*, que obteve os melhores resultados no artigo de referência [9], apresentou o desempenho mais baixo em nossa replicação, com 63.07% de acurácia. Este resultado sugere que, para o *dataset DeepFashion-1* e as 17 categorias mantidas, modelos mais recentes e otimizados para eficiência, como o *EfficientNet-B0* estejam melhor adaptadas a tarefas que exigem maior precisão com menor custo computacional, o que pode ter favorecido seu desempenho neste cenário.

Analisando a diferença entre a Acurácia de Treino e a Acurácia de Validação, nota-se que o *EfficientNet-B0* apresentou o maior *overfitting* inicial ($89.53\% - 70.85\% = 18.68\%$), indicando que, embora tenha atingido o maior pico de desempenho, ele está mais propenso à super-adaptação aos dados de treinamento do que os demais modelos, justificando a escolha por técnicas de regularização e fine-tuning conservador na etapa de otimização.

B. Otimização do *EfficientNet-B0* (*Baseline* vs. *Otimizado*)

Com base nos resultados da Tabela I, o *EfficientNet-B0* foi selecionado como o modelo *baseline* para a otimização. Foram aplicadas as seguintes técnicas em uma abordagem conservadora: *Class Weighting* (para mitigar o desbalanceamento das classes), *Learning Rate* baixo (10^{-4}), *Weight Decay* (5×10^{-4}), *Dropout* (0.2) e *Early Stopping* (paciência de 5). A avaliação final incluiu o *Test-Time Augmentation* (TTA) simplificado.

Os resultados da otimização foram inversos ao esperado. Apesar do emprego de regularização, observa-se uma queda de aproximadamente 1.65% na Acurácia de Teste e uma redução significativa no F1-Score macro (de 70.26% para 55.52%).

O histórico de treinamento do modelo otimizado, ilustrado na Figura 3, fornece a comprovação visual do problema de

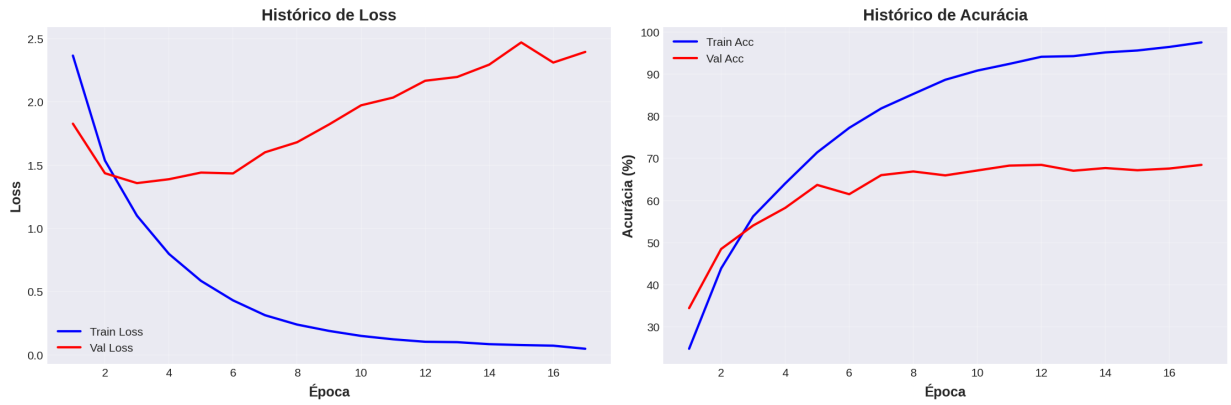


Figura 3. Histórico de treinamento do modelo EfficientNet-B0 Otimizado. À esquerda, a curva de Loss mostra a divergência entre a perda de treino (azul) e a perda de validação (vermelho) a partir da Época 4. À direita, a curva de Acurácia ilustra a saturação da acurácia de validação, confirmando o aumento do *overfitting*.

Tabela II
COMPARAÇÃO DO EFFICIENTNET-B0 (ORIGINAL) COM O MODELO OTIMIZADO E TTA.

Modelo	Train Acc (%)	Val Acc (%)	Test Acc (%)	F1-Score (%)
EfficientNet-B0 (Original)	89.53	70.85	70.79	70.26
EfficientNet-B0 (Otimizado)	99.89	68.47	69.20	56.13
EfficientNet-B0 (Otimizado + TTA)	99.89	68.47	69.14	55.52

generalização. A curva de perda de validação (Val Loss, em vermelho) para de cair e começa a subir a partir da Época 4, enquanto a acurácia de validação (Val Acc, em vermelho) se estabiliza. Esta acentuada divergência entre as curvas de treino e validação confirma visualmente o aumento do *overfitting*, justificando a interrupção do processo pelo Early Stopping na Época 17.

Aumento do Overfitting: O *overfitting* aumentou de 18.68% para 31.42% (99.89% – 68.47% resultando os 31.42%). A combinação de Class Weighting e Learning Rate muito baixo em um dataset desbalanceado provavelmente concentrou o treinamento nas classes minoritárias (como Suiting, com apenas 8 amostras), forçando o modelo a memorizar esses poucos exemplos e resultando em um desempenho de treino muito alto (99.89%), mas uma generalização pior.

Desempenho por Classe (F1-Score): O F1-Score por classe (Figura 4) confirma que, embora o Class Weighting tenha elevado o desempenho em algumas classes raras, a penalização excessiva nas classes majoritárias (como Tees_Tanks) comprometeu drasticamente o F1-Score macro total. A classe Suiting, por exemplo, permaneceu com 0.00% de F1-Score, evidenciando que a estratégia de class weighting isolada não foi suficiente para tratar as classes com menos de 10 amostras.

Logo, a tentativa de balancear o aprendizado em um dataset severamente desbalanceado resultou em uma perda de poder preditivo nas categorias mais frequentes e uma dificuldade persistente em prever as classes mais raras.

Reforçando os resultados, a matriz de confusão do modelo EfficientNet-B0 otimizado revela padrões importantes de acerto e erro entre as classes (Figura 5). Observa-se que as

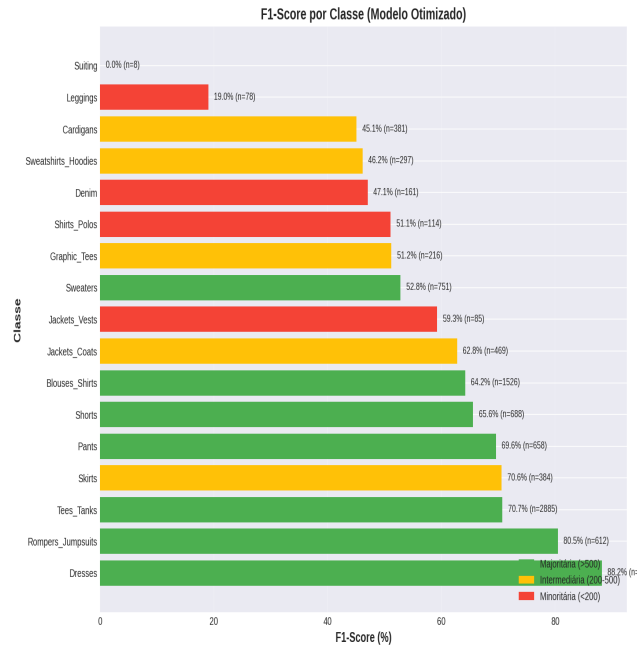


Figura 4. F1-Score por Classe (Modelo EfficientNet-B0 Otimizado), mostrando o desequilíbrio de performance entre classes majoritárias (verde) e minoritárias (vermelho).

classes com maior volume de amostras concentram os maiores valores na diagonal, indicando alta taxa de acerto: Tees_Tanks (313 acertos), Dresses (292 acertos) e Blouses_Shirts (138 acertos).

Por outro lado, classes minoritárias apresentam dispersão significativa de predições incorretas. Esse comportamento sugere que o modelo tende a confundir peças com características visuais sobrepostas, especialmente quando há escassez de exemplos representativos. Por exemplo:

- As duas amostras de Suiting (n=2) resultaram em 0% de acurácia, sendo incorretamente preditas como Pants (1 amostra) e Sweatshirts_Hoodies (1 amostra).

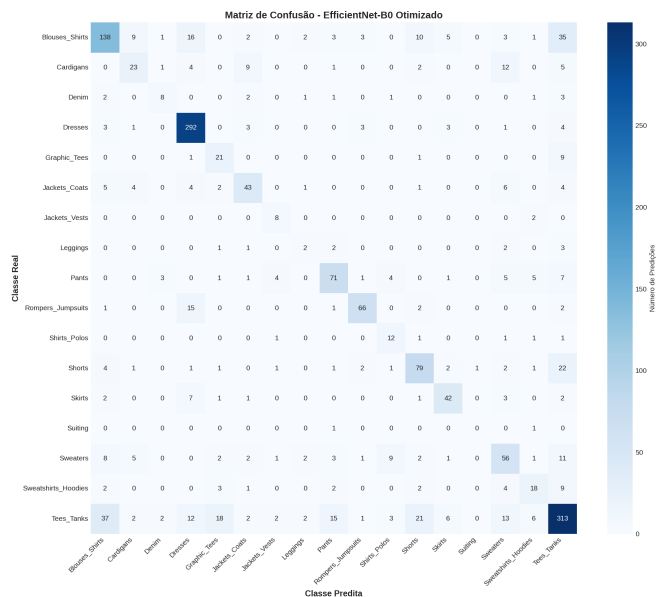


Figura 5. Matriz de Confusão do EfficientNet-B0 Otimizado. Observa-se a concentração de erros de classificação nas classes minoritárias (ex: Leggings e Suiting).

- A classe Rompers_Jumpsuits ($n = 87$) sofreu sua confusão notável com Dresses, onde 15 amostras foram preditas erroneamente.

Além disso, a matriz de confusão evidencia que o desequilíbrio de classes não apenas afeta o desempenho global, mas também compromete a capacidade do modelo de aprender fronteiras decisórias claras entre categorias específicas. Isso reforça a necessidade de estratégias complementares, como data augmentation direcionado ou oversampling seletivo (aumenta artificialmente o número de instâncias da classe minoritárias para equilibrar o conjunto de dados) para melhorar a representatividade das classes raras e reduzir a ambiguidade nas previsões.

V. CONCLUSÃO

A. Conclusão

O presente projeto de disciplina alcançou o objetivo de aplicar e comparar diferentes arquiteturas de Deep Learning para a classificação de categorias de roupas, uma etapa fundamental no desenvolvimento de um sistema assistivo para pessoas com deficiência visual. A avaliação comparativa, que envolveu a replicação de um modelo de referência (GoogLeNet) e a comparação com três modelos escolhidos que foram apreciados nas aulas (ResNet-50, MobileNet-v2 e EfficientNet-B0), demonstrou a superioridade do EfficientNet-B0.

O EfficientNet-B0 (Baseline) atingiu a melhor performance inicial, com 70.79% de Acurácia de Teste e 70.26% de F1-Score. Este resultado valida a eficácia do Transfer Learning na tarefa de classificação de roupas e sugere que a otimização de parâmetros de eficiência (Compound Scaling) empregada pelo EfficientNet-B0 é vantajosa.

Contudo, a etapa de otimização revelou um desafio metodológico crucial. As técnicas de balanceamento, como o Class Weighting, combinadas com o Fine-Tuning em um Learning Rate baixo, resultaram em um aumento de overfitting (de 18.68% para 31.42%) e uma queda de 1.65% na Acurácia de Teste (passando para 69.14% com TTA). Este resultado crítico indica que o desequilíbrio severo de classes do dataset DeepFashion-1 não pode ser corrigido eficientemente apenas pela ponderação de perda, necessitando de intervenções diretas na distribuição dos dados de treinamento.

Em suma, o projeto demonstrou a viabilidade da classificação de categorias de roupas com o EfficientNet-B0 e expôs a importância da calibração das estratégias de balanceamento em cenários de classificação multiclasse e desbalanceamento.

B. Trabalhos Futuros

O trabalho aqui apresentado estabelece uma base preliminar para a pesquisa de Mestrado, que se concentrará em refinar a precisão e expandir a utilidade prática do sistema. As seguintes direções de pesquisa são propostas:

- Classificação Multi-Label (Categoria e Cor): O objetivo principal é evoluir o sistema para a classificação simultânea da categoria e da cor da roupa. Isso exigirá a curadoria ou anotação de um dataset mais robusto ou a implementação de uma arquitetura Multi-label capaz de gerar previsões independentes para ambos os atributos.
- Estratégias de Balanceamento de Dados: Para resolver o problema de overfitting e a performance ruim nas classes minoritárias (ex: Suiting), sugere-se explorações técnicas que atuam diretamente na amostragem:
 - Oversampling/Undersampling: Utilização de métodos como SMOTE ou Random Over-Sampling para aumentar o volume das classes raras no conjunto de treino.
 - Curadoria Focada: Análise e potencial descarte de classes com contagens extremamente baixas (como Suiting com $n = 8$) ou agregação em categorias mais amplas para melhorar a generalização.
- Desenvolvimento de Aplicação Acessível: Como aplicação final da pesquisa, espera-se ser desenvolvida uma prova de conceito de um aplicativo móvel ou web que integra o modelo otimizado, fornecendo feedback de áudio em tempo real para o usuário com deficiência visual, concretizando a motivação inicial do projeto.

REFERÊNCIAS

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [2] A. A. Khan Ashar, A. Abrar, and J. Liu, "A Survey on Deep Learning-based Smart Assistive Aids for Visually Impaired Individuals," in *Proceedings of the 2023 7th International Conference on Information System and Data Mining (ICISDM '23)*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 90–95. [Online]. Available: <https://doi.org/10.1145/3603765.3603775>
- [3] K. Chemnad and A. Othman, "Digital accessibility in the era of artificial intelligence-Bibliometric analysis and systematic review," *Front. Artif. Intell.*, vol. 7, p. 1349668, 2024. [Online]. Available: <https://doi.org/10.3389/frai.2024.1349668>

- 387 [4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov,
388 D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper
389 with Convolutions," *arXiv preprint arXiv:1409.4842*, 2014. [Online].
390 Available: <https://arxiv.org/abs/1409.4842>
- 391 [5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image
392 recognition," in *Proceedings of the IEEE conference on computer vision
393 and pattern recognition (CVPR)*. IEEE, 2016.
- 394 [6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the
395 IEEE conference on computer vision and pattern recognition (CVPR)*.
396 IEEE, 2018, pp. 4510–4520.
- 397 [7] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling
398 for Convolutional Neural Networks," in *Proceedings of the 36th
399 International Conference on Machine Learning (ICML 2019)*, 2019.
400 [Online]. Available: <https://arxiv.org/abs/1905.11946>
- 401 [8] V. B. Sadanand, "DeepFashion-1 Dataset," Kaggle, 2025, acesso em:
402 19 de Outubro de 2025. [Online]. Available: [https://www.kaggle.com/
403 datasets/vishalbsadanand/deepfashion-1/data](https://www.kaggle.com/datasets/vishalbsadanand/deepfashion-1/data)
- 404 [9] D. Rocha, F. Soares, E. Oliveira, and V. Carvalho, "Blind People:
405 Clothing Category Classification and Stain Detection Using Transfer
406 Learning," *Appl. Sci.*, vol. 13, no. 3, p. 1925, 2023. [Online]. Available:
407 <https://doi.org/10.3390/app13031925>
- 408 [10] J. N. d. Silva, "Visão computacional aplicada à acessibilidade:
409 Classificação de vestuário com redes neurais convolucionais
410 para pessoas com deficiência visual," GitHub repository, 2025,
411 acesso em: 19 de Outubro de 2025. [Online]. Avail-
412 able: [https://github.com/jacquelinenavarro/jacquelinenavarro-Projeto\
413 _Roupas_AprendizadoProfundo_VisaoComputacional_V4](https://github.com/jacquelinenavarro/jacquelinenavarro-Projeto_Roupas_AprendizadoProfundo_VisaoComputacional_V4)
- 414 [11] Kaggle, "Kaggle: Your Home for Data Science," Web Site, 2025,
415 acesso em: 19 de Outubro de 2025. [Online]. Available: [https:
416 //www.kaggle.com/](https://www.kaggle.com/)
- 417