

# 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.

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

## 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

80 compound scaling, equilibrando largura, profundidade e resolução da rede para alcançar maior eficiência e precisão.  
 81  
 82 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

90 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

95 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.

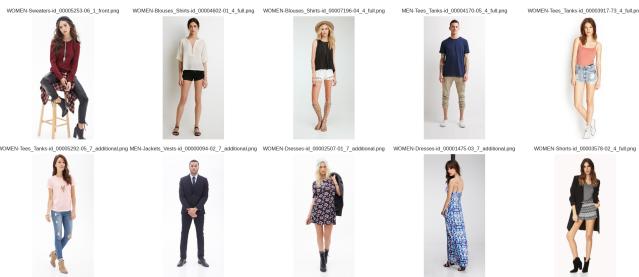


Figura 1. Mosaico das primeiras 10 imagens do dataset DeepFashion-1.

98  
 99 1) Desbalanceamento e Divisão dos Dados: O principal  
 100 desafio deste dataset é o severo desbalanceamento de classes. Conforme ilustrado na Figura 2, a classe majoritária,  
 101 Tees\_Tanks, possui 2.885 amostras (25.12% do total), en-  
 102 quanto a classe minoritária, Suiting, possui apenas 8 amostras  
 103 (0.07%), resultando em uma proporção de 360:1.

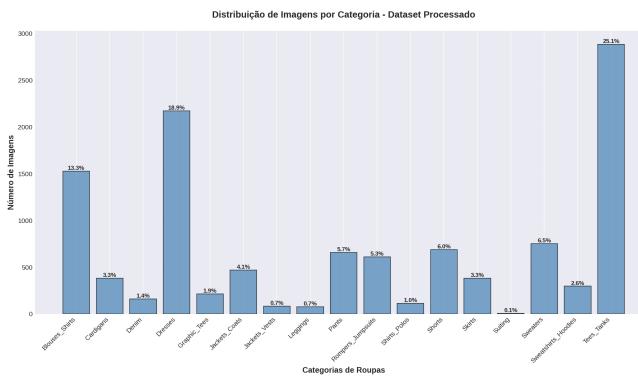


Figura 2. Distribuição de imagens por categoria do dataset processado, evidenciando o severo desbalanceamento de classes.

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

107 otimização, reconhecendo a complexidade de garantir a repre-  
 108 sentatividade de classes extremamente raras (como Suiting)  
 109 em todos os splits do código inicial:

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

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

#### B. Pré-Processamento e Aumento de Dados

117 O pré-processamento das imagens seguiu as diretrizes de  
 118 Transfer Learning para modelos pré-treinados no ImageNet:  
 119

- Redimensionamento e Augmentation: As imagens fo-  
 120 ram redimensionadas para 224 × 224 pixels. Para au-  
 121 mentar a robustez do conjunto de treinamento e simu-  
 122 lar variações do mundo real, foram aplicadas técnicas  
 123 de Aumento de Dados (Data Augmentation), incluindo  
 124 RandomHorizontalFlip, RandomRotation (até  
 125 15°), e ColorJitter (ajuste aleatório de brilho, con-  
 126 traste e saturação).
- Normalização: Todas as imagens, tanto de treino quanto  
 127 de teste/validation, foram normalizadas utilizando os va-  
 128 lores de média e desvio-padrão do ImageNet.

#### C. Arquiteturas e Treinamento Comparativo (Baseline)

131 Esta fase estabeleceu a *baseline* de desempenho, avaliando  
 132 quatro arquiteturas DCNNs (Deep Convolutional Neural Net-  
 133 works), todas inicializadas com pesos pré-treinados do Image-  
 134 Net (*Transfer Learning*): GoogLeNet, ResNet-50, MobileNet-  
 135 v2 e EfficientNet-B0.

- Reajuste da Camada FC: Para cada modelo, a camada totalmente conectada (FC) final, originalmente dimensi-  
 137 onada para 1.000 classes, foi substituída por uma ca-  
 138 mada linear adaptada para as 17 classes de saída do  
 139 DeepFashion-1.
- Função de Perda: Foi utilizada a *Cross-Entropy Loss*.
- Otimizador e Scheduler: Utilizou-se o otimizador Adam  
 140 (LR =  $1 \times 10^{-3}$ ) por 10 épocas. Um *scheduler* Redu-  
 141 ceLROnPlateau foi empregado, monitorando a perda de  
 142 validação.

#### D. Otimização do Melhor Modelo (EfficientNet-B0)

147 O modelo de melhor desempenho na *baseline* foi o  
 148 EfficientNet-B0 (Acurácia de Teste: 70.79% / F1-Score:  
 149 70.26%), Tabela I. Esta fase focou na otimização utilizando  
 150 técnicas de regularização e balanceamento, cujos detalhes são  
 151 apresentados a seguir:

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

- 159 2) Fine-Tuning e Regularização: O otimizador foi ajustado  
 160 para AdamW com um *Learning Rate* (LR) mais conser-  
 161 vador de  $1 \times 10^{-4}$  e um *Weight Decay* (Regularização  
 162 L2) de  $5 \times 10^{-4}$ . Para reforçar a prevenção contra  
 163 *overfitting*, foi adicionado um *Dropout* de 0.2 na camada  
 164 classificadora.
- 165 3) Early Stopping: Foi implementado com uma paciência  
 166 de 5 épocas, resultando em um treinamento total de 17  
 167 épocas antes de ser interrompido. Esta técnica visa evitar  
 168 a super-adaptação do modelo aos dados de treino, inter-  
 169 rompendo o treinamento quando a métrica de validação  
 170 (acurácia) não melhora.
- 171 4) Test-Time Augmentation (TTA): Esta técnica foi utili-  
 172 zada na avaliação final (após o treinamento), aumen-  
 173 tando a robustez da predição. Consistiu em fazer a  
 174 média das *softmax probabilities* da imagem original e  
 175 da imagem espelhada horizontalmente.

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

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

### 191 E. Análise dos Resultados da Otimização

192 Apesar da aplicação das técnicas de otimização, o modelo  
 193 *baseline* não foi superado. A acurácia de teste final piorou  
 194 em 1.65% e o *F1-Score* (Macro) em 14.74%. A técnica de  
 195 *Class Weighting* (inversamente proporcional) provou ser muito  
 196 agressiva para este *dataset*, penalizando excessivamente as  
 197 classes majoritárias (que representam a maior parte dos dados)  
 198 e levando a um aumento significativo no *overfitting* (salto de  
 199 18.68% para 31.42%), superando a capacidade de mitigação  
 200 das técnicas de regularização, Tabela II.

### 201 F. Reprodutibilidade do Estudo

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

## 207 IV. RESULTADOS E DISCUSSÃO

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

subseções seguintes. Todo o processo de treinamento e testes  
 212 foi executado na plataforma Kaggle, utilizando o ambiente  
 213 com GPU NVIDIA Tesla T4 (CUDA) [11], com duração de  
 214 10 épocas para a fase comparativa.  
 215

### 216 A. Comparação das Arquiteturas Base (*GoogLeNet*, *ResNet-50*, 217 *MobileNet-v2*, *EfficientNet-B0*)

218 Os quatro modelos foram treinados e avaliados no conjunto  
 219 de teste, e seus resultados de acurácia, precisão, recall e F1-  
 220 Score (média ponderada) são apresentados na Tabela I.  
 221

222 **Tabela I**  
 223 COMPARAÇÃO DE MÉTRICAS DE DESEMPEÑO PARA OS QUATRO  
 224 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

225 Conforme a Tabela I, o modelo EfficientNet-B0 demonstrou  
 226 o melhor desempenho em todas as métricas no conjunto de  
 227 teste, atingindo 70.79% de acurácia e 70.26% de F1-Score.  
 228 O MobileNet-v2 e o ResNet-50 apresentaram desempenho  
 229 ligeiramente inferior, mas superior ao modelo de referência.  
 230

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

240 Analisando a diferença entre a Acurácia de Treino e a  
 241 Acurácia de Validação, nota-se que o EfficientNet-B0 apresen-  
 242 tou o maior overfitting inicial (89.53% - 70.85% = 18.68%),  
 243 indicando que, embora tenha atingido o maior pico de desem-  
 244 penho, ele está mais propenso à super-adaptação aos dados de  
 245 treinamento do que os demais modelos, justificando a escolha  
 246 por técnicas de regularização e fine-tuning conservador na  
 247 etapa de otimização.

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

249 Com base nos resultados da Tabela I, o EfficientNet-B0  
 250 foi selecionado como o modelo baseline para a otimização.  
 251 Foram aplicadas as seguintes técnicas em uma abordagem  
 252 conservadora: Class Weighting (para mitigar o desbalancea-  
 253 mento das classes), Learning Rate baixo ( $10^{-4}$ ), Weight Decay  
 254 ( $5 \times 10^{-4}$ ), Dropout (0.2) e Early Stopping (paciente de 5).  
 255 A avaliação final incluiu o Test-Time Augmentation (TTA)  
 256 simplificado.

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

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

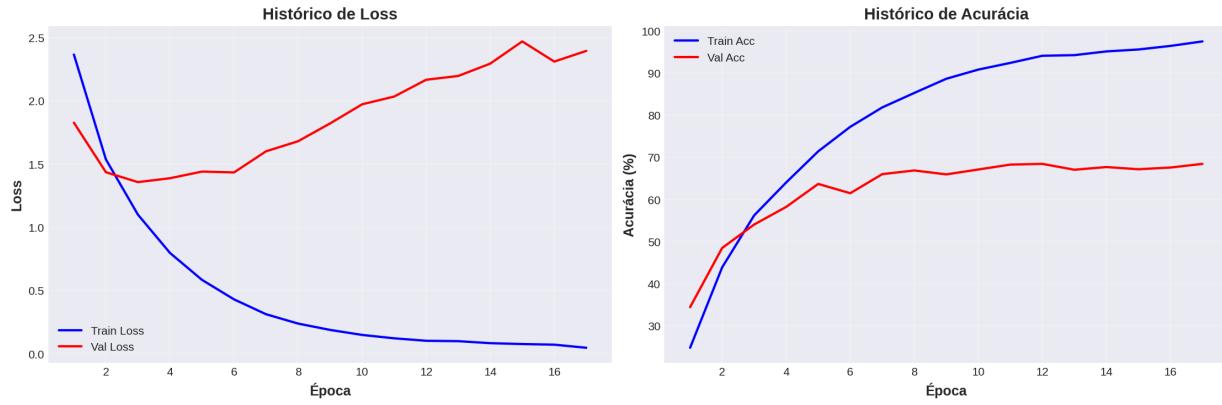


Figura 3. Histórico de treinamento do modelo EfficientNet-B0 Otimizado. À esquerda, a curva de Loss mostra a divergência entre a perda de treino (azul) e a perda de validação (vermelho) a partir da Época 4. À direita, a curva de Acurácia ilustra a saturação da acurácia de validação, confirmando o aumento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

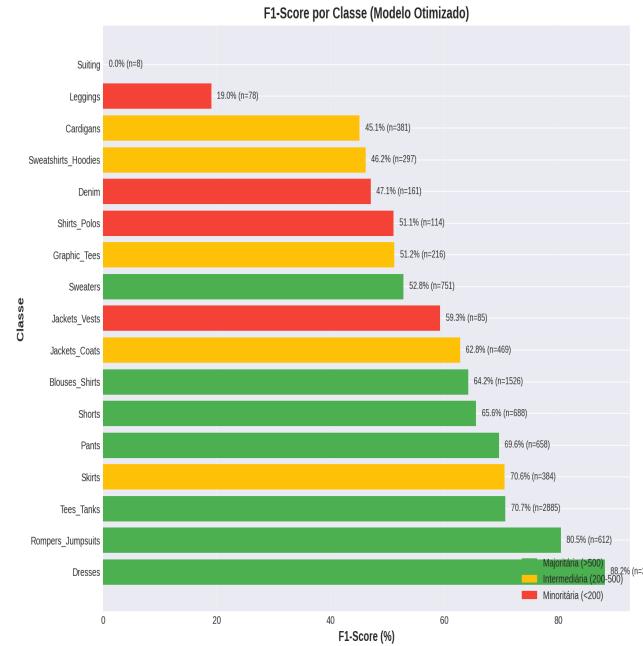


Figura 4. F1-Score por Classe (Modelo EfficientNet-B0 Otimizado), mostrando o desequilíbrio de performance entre classes majoritárias (verde) e minoritárias (vermelho).

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 previsões 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).

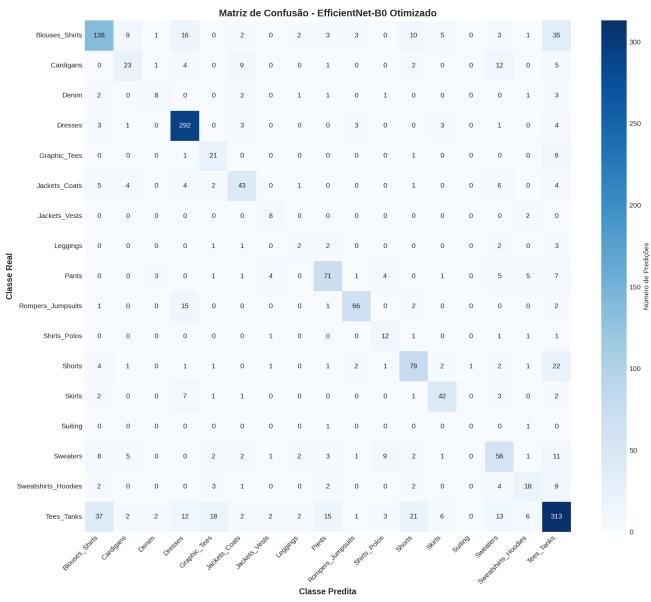


Figura 5. Matriz de Confusão do EfficientNet-B0 Otimizado. Observa-se a concentração de erros de classificação nas classes minoritárias (ex: Leggings e Suiting).

- 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 minoritária 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, “Mobi-  
395 leNetV2: Inverted residuals and linear bottlenecks,” in *Proceedings of the  
396 IEEE conference on computer vision and pattern recognition (CVPR)*.  
397 IEEE, 2018, pp. 4510–4520.
- 398 [7] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling  
399 for Convolutional Neural Networks,” in *Proceedings of the 36th  
400 International Conference on Machine Learning (ICML 2019)*, 2019.  
401 [Online]. Available: <https://arxiv.org/abs/1905.11946>
- 402 [8] V. B. Sadanand, “DeepFashion-1 Dataset.” Kaggle, 2025, acesso em:  
403 19 de Outubro de 2025. [Online]. Available: <https://www.kaggle.com/404/datasets/vishalbsadanand/deepfashion-1/data>
- 405 [9] D. Rocha, F. Soares, E. Oliveira, and V. Carvalho, “Blind People:  
406 Clothing Category Classification and Stain Detection Using Transfer  
407 Learning,” *Appl. Sci.*, vol. 13, no. 3, p. 1925, 2023. [Online]. Available:  
408 <https://doi.org/10.3390/app13031925>
- 409 [10] J. N. d. Silva, “Visão computacional aplicada à acessibilidade:  
410 Classificação de vestuário com redes neurais convolucionais  
411 para pessoas com deficiência visual,” GitHub repository, 2025,  
412 acesso em: 19 de Outubro de 2025. [Online]. Avai-  
413 lable: [https://github.com/jacquelinanavarro/jacquelinanavarro-Projeto\414\\_Roupas\AprendizadoProfundo\VisaoComputacional\V4](https://github.com/jacquelinanavarro/jacquelinanavarro-Projeto\414_Roupas\AprendizadoProfundo\VisaoComputacional\V4)
- 415 [11] Kaggle, “Kaggle: Your Home for Data Science,” Web Site, 2025,  
416 acesso em: 19 de Outubro de 2025. [Online]. Available: [https://www.kaggle.com/](https://417/www.kaggle.com/)