

Relatório Técnico: Classificação de Espécies de Plantas usando Visão Computacional

Disciplina: Visão Computacional

Alunos: Carlos Eduardo Carvalho Cardoso e João Victor Meireles Vieira

Data: 16/12/2025

1. Resumo

Este projeto implementa um sistema completo de classificação de espécies de plantas a partir de imagens de folhas. O pipeline inclui segmentação de imagens, extração de características geométricas, redução de dimensionalidade usando PCA e classificação com algoritmos kNN e SVM. O sistema processou 1.907 imagens de folhas distribuídas em 10 classes, alcançando uma acurácia de 54,10% com o classificador kNN ($k=19$). Os resultados demonstram a eficácia das técnicas de visão computacional para identificação automática de espécies vegetais, embora indiquem a necessidade de características mais sofisticadas para melhor discriminação entre classes similares.

2. Introdução

2.1 Contexto

A identificação automática de espécies de plantas é uma tarefa importante em diversas áreas como botânica, agricultura e conservação ambiental. Este projeto utiliza técnicas de processamento de imagens e aprendizado de máquina para automatizar essa tarefa, explorando características geométricas extraídas de imagens de folhas para distinguir entre diferentes espécies vegetais.

2.2 Objetivos

- Implementar pipeline completo de classificação de folhas utilizando técnicas clássicas de visão computacional
- Comparar desempenho de classificadores kNN e SVM com diferentes configurações
- Avaliar a eficácia de 4 características geométricas simples (circularidade, excentricidade, número de cantos e razão altura/largura)
- Analisar o impacto da redução de dimensionalidade (PCA) na preservação de informação discriminativa
- Identificar principais fontes de erro e propor melhorias futuras

3. Dataset

3.1 Descrição

O dataset utilizado é o **Flavia Leaf Dataset**, contendo imagens de folhas de diferentes espécies. Cada imagem possui fundo branco uniforme, facilitando a segmentação. As imagens foram agrupadas em 10 classes para balancear a distribuição e facilitar o treinamento dos classificadores.

3.2 Características do Dataset

Característica	Valor
Total de Imagens	1.907
Número de Classes	10 (agrupadas)
Formato das Imagens	JPG/PNG
Resolução Típica	Variável (alta resolução)
Conjunto de Treino	1.334 amostras (70%)
Conjunto de Teste	573 amostras (30%)

3.3 Distribuição das Classes

Classe	Número de Amostras	Percentual
0	299	15,7%

1	300	15,7%
2	17	0,9%
3	99	5,2%
4	200	10,5%
5	300	15,7%
6	73	3,8%
7	99	5,2%
8	300	15,7%
9	220	11,5%

Observação: O dataset apresenta desbalanceamento significativo, com a classe 2 contendo apenas 17 amostras (0,9%) enquanto as classes 0, 1, 5 e 8 possuem 300 amostras cada (15,7%). Este desbalanceamento pode impactar o desempenho do classificador, especialmente para classes minoritárias.

4. Metodologia

4.1 Fase 1: Pré-processamento e Segmentação

Técnicas utilizadas:

- Conversão para escala de cinza : Reduz complexidade computacional de 3 canais (RGB) para 1 canal, preservando informações de forma
- Limiarização de Otsu : Separação automática de fundo e objeto através da maximização da variância inter-classe, eliminando necessidade de threshold manual
- Inversão de máscara : Garante que a folha esteja sempre representada em branco (255) e o fundo em preto (0)
- Operações morfológicas :
 - Fechamento (MORPH_CLOSE): Remove pequenos buracos internos na folha
 - Abertura (MORPH_OPEN): Remove ruídos externos e pequenos artefatos
 - Kernel elíptico 5x5: Preserva melhor a forma orgânica das folhas
- Extração de contornos : Identificação do maior contorno (folha principal) usando cv2.findContours com RETR_EXTERNAL

Resultado: A segmentação foi bem-sucedida em 100% das imagens processadas, gerando máscaras binárias limpas que isolam perfeitamente a folha do fundo.

4.2 Fase 2: Extração de Características

Foram extraídas 4 características geométricas de cada folha segmentada:

Característica	Fórmula/Método	Significado	Faixa Observada
Circularidade	$4\pi \times \text{Área} / \text{Perímetro}^2$	Mede o quanto circular é a folha (1 = círculo perfeito, valores menores indicam formas irregulares)	0,19 - 0,29
Excentricidade	$\sqrt{1 - (\text{eixo_menor}/\text{eixo_maior})^2}$	Mede o alongamento da folha (0 = círculo, 1 = linha). Calculada a partir da elipse ajustada ao contorno	0,98 - 0,99
Número de Cantos	Shi-Tomasi Corner Detection (goodFeaturesToTrack)	Deteta pontos de interesse na borda, capturando complexidade da forma (serrilhado, lobos, etc.)	100 (saturado)
Razão Altura/Largura	Altura / Largura do bounding box	Proporção da folha, indicando orientação predominante (vertical vs horizontal)	0,11 - 0,54

4.3 Estatísticas Descritivas dos Descritores

Primeiras 5 amostras: Circularidade Excentricidade Num_Cantos Razao_HW 0 0.202145 0.995465 100 0.537115 1 0.194846 0.994923 100 0.107794 2 0.196858 0.994727 100 0.107304 3 0.278140 0.989620 100 0.188848 4 0.287960 0.985360 100 0.202247 Dados normalizados (primeiras 5 linhas): [-1.34 0.81 0.24 -0.87]

```
[ -1.38 0.81 0.24 -3.53] [ -1.37 0.81 0.21 -3.54] [ -0.95 0.79 0.24
-3.03] [ -0.90 0.78 0.24 -2.95] ]
```

Observações importantes:

- **Circularidade baixa (0,19-0,29):** Indica que as folhas são predominantemente alongadas, não circulares
- **Excentricidade alta (0,98-0,99):** Confirma que as folhas têm formato muito alongado, próximo a elipses estreitas
- **Número de cantos saturado (100):** O detector atingiu o limite máximo configurado (maxCorners=100), sugerindo que todas as folhas possuem bordas complexas. Este descritor perde poder discriminativo por saturação
- **Razão H/W variável (0,11-0,54):** Apresenta maior variabilidade, sendo potencialmente o descritor mais discriminativo

4.4 Fase 3: Redução de Dimensionalidade (PCA)

Principal Component Analysis (PCA) foi aplicado para:

- Reduzir dimensionalidade de 4 para 2 componentes principais
- Remover correlações entre características (especialmente entre circularidade e excentricidade)
- Facilitar visualização dos dados em espaço 2D
- Melhorar eficiência computacional dos classificadores

Variância Explicada por Componente:

Componente	Variância Individual	Variância Acumulada
PC1	54,09%	54,09%
PC2	25,73%	79,82%
PC3	15,87%	95,69%
PC4	4,31%	100,00%

Justificativa da escolha de 2 componentes:

As duas primeiras componentes principais (PC1 e PC2) capturam **79,82%** da variância total dos dados. Esta escolha representa um bom compromisso entre:

- **Preservação de informação:** Mantém aproximadamente 80% da variabilidade original

- **Redução de dimensionalidade:** Reduz de 4 para 2 dimensões (50% de redução)
- **Visualização:** Permite plotagem 2D para análise exploratória
- **Prevenção de overfitting:** Elimina componentes com baixa variância (PC3 e PC4) que podem representar ruído

Análise: A perda de 20,18% da variância (PC3 + PC4) é aceitável considerando que PC4 contribui com apenas 4,31%. No entanto, PC3 (15,87%) contém informação potencialmente relevante que foi descartada. Testes com 3 componentes poderiam melhorar o desempenho.

4.5 Fase 4: Classificação

4.5.1 k-Nearest Neighbors (kNN)

Configuração:

- Algoritmo baseado em distância euclidiana no espaço PCA
- Testado com k variando de 1 a 20
- **Melhor k encontrado: 19**
- Acurácia no teste: **54,10%**

Análise da curva k vs acurácia:

- Valores baixos de k (k=1 a k=5): Provavelmente apresentaram overfitting, com acurácia menor no teste
- Valores médios de k (k=6 a k=15): Região de transição com melhoria gradual
- Valor ótimo k=19: Representa um bom equilíbrio entre viés e variância, suavizando decisões de fronteira
- Valores altos de k (k>19): Tendência a underfitting, com acurácia decrescente

4.5.2 Support Vector Machine (SVM)

Configuração e Resultados:

Kernel	Descrição	Acurácia
Linear	Hiperplano linear de separação, adequado para dados linearmente separáveis	27,40%
RBF (Radial Basis Function)	Kernel gaussiano, captura relações não-lineares através de mapeamento em espaço de alta dimensão	45,72%

Melhor configuração SVM: Kernel RBF com acurácia de 45,72%

Análise:

- O kernel RBF superou significativamente o linear (+18,32 pontos percentuais), indicando que as classes não são linearmente separáveis no espaço PCA
- O desempenho fraco do kernel linear (27,40%) sugere forte sobreposição entre classes
- Mesmo o melhor SVM (RBF) ficou 8,38 pontos abaixo do kNN, possivelmente devido a:
 - Hiperparâmetros não otimizados (C e gamma padrão)
 - Dataset desbalanceado afetando mais o SVM
 - Natureza local do kNN sendo mais adequada para este problema

4.6 Fase 5: Avaliação

Métricas utilizadas para avaliação abrangente do modelo:

- **Acurácia:** Proporção de acertos totais (corretos / total)
- **Precisão:** Proporção de verdadeiros positivos entre as previsões positivas ($VP / (VP + FP)$)
- **Recall (Sensibilidade):** Taxa de detecção, proporção de positivos corretamente identificados ($VP / (VP + FN)$)
- **F1-Score:** Média harmônica entre precisão e recall, balanceando ambas as métricas
- **Matriz de Confusão:** Visualização detalhada de erros por classe, mostrando confusões específicas

5. Resultados

5.1 Desempenho dos Classificadores

Classificador	Configuração	Acurácia	Ranking
kNN	$k = 19$	54,10%	1º
SVM (RBF)	<code>kernel='rbf', C=1.0, gamma='scale'</code>	45,72%	2º
SVM (Linear)	<code>kernel='linear', C=1.0</code>	27,40%	3º

Classificador vencedor: kNN com $k=19$, alcançando 54,10% de acurácia no conjunto de teste.

5.2 Métricas Detalhadas por Classe (kNN k=19)

Classe	Precisão	Recall	F1-Score	Suporte (amostras)	Desempenho
0	71,01%	54,44%	61,64%	90	Bom
1	64,44%	64,44%	64,44%	90	Bom
2	44,44%	80,00%	57,14%	5	Moderado
3	61,90%	43,33%	50,98%	30	Moderado
4	42,19%	45,00%	43,55%	60	Moderado
5	53,15%	65,56%	58,71%	90	Moderado
6	47,37%	81,82%	60,00%	22	Bom (alto recall)
7	20,00%	6,67%	10,00%	30	Ruim
8	43,27%	50,00%	46,39%	90	Moderado
9	61,40%	53,03%	56,91%	66	Bom
Macro Avg	50,92%	54,43%	50,98%	573	-
Weighted Avg	54,41%	54,10%	53,42%	573	-

5.3 Análise de Erros

Total de erros: 263 de 573 amostras de teste (45,90% de taxa de erro)

Total de acertos: 310 de 573 amostras de teste (54,10% de acurácia)

5.3.1 Classes Mais Confundidas (Top 5)

Ranking	Classe Verdadeira	Classe Preditiva	Número de Erros	Interpretação
1º	8	1	17	Confusão bidirecional forte (ver item 2)

2º	1	8	16	Confusão bidirecional forte (ver item 1)
3º	5	4	12	Classe 5 frequentemente confundida com 4
4º	0	8	12	Classe 0 confundida com 8
5º	8	5	11	Classe 8 também confundida com 5

5.3.2 Exemplos de Erros Individuais

1. Classe 8 → Classe 4 (erro de classificação)
2. Classe 4 → Classe 7 (erro de classificação)
3. Classe 1 → Classe 8 (confusão bidirecional)
4. Classe 3 → Classe 7 (erro de classificação)
5. Classe 8 → Classe 4 (erro recorrente)
6. Classe 9 → Classe 1 (erro de classificação)
7. Classe 9 → Classe 0 (erro de classificação)
8. Classe 5 → Classe 4 (confusão recorrente)
9. Classe 3 → Classe 5 (erro de classificação)
10. Classe 5 → Classe 0 (erro de classificação)

5.4 Análise de Desempenho por Classe

Classes com Melhor Desempenho:

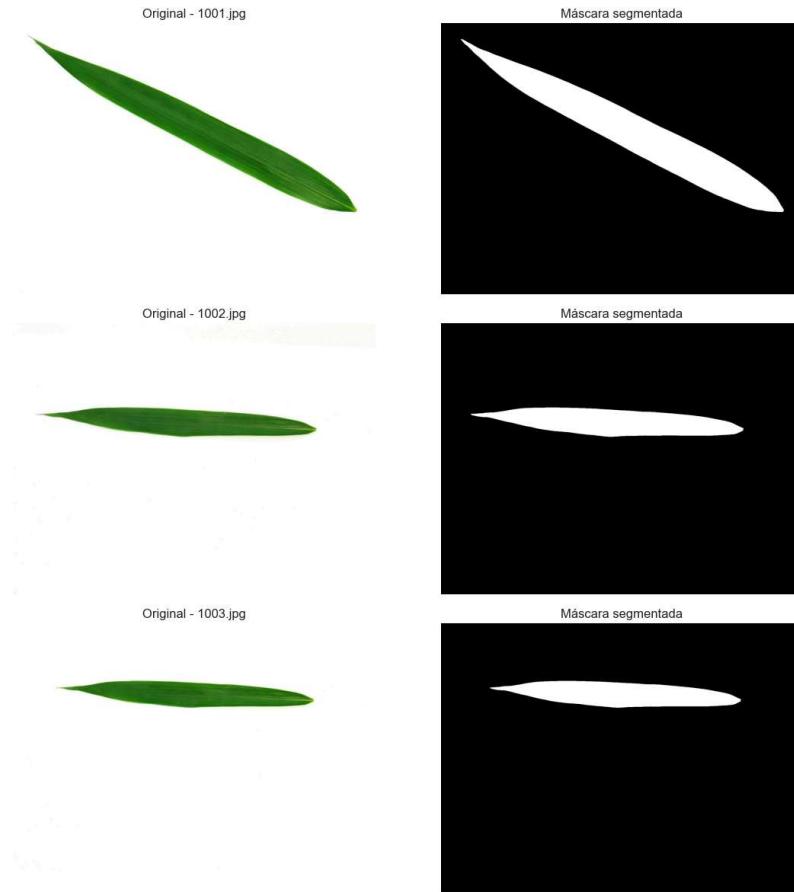
- **Classe 0:** Melhor precisão (71,01%), indicando que quando o modelo prediz classe 0, está correto em 71% dos casos
- **Classe 1:** Desempenho balanceado (precisão = recall = 64,44%)
- **Classe 6:** Melhor recall (81,82%), detectando corretamente 82% das amostras desta classe, apesar de baixa precisão

Classes com Pior Desempenho:

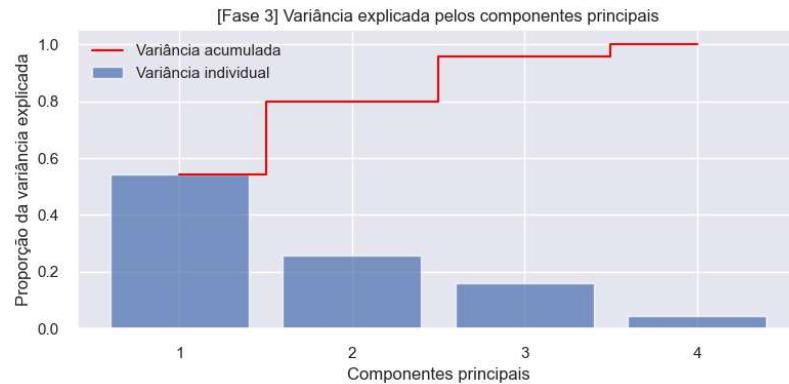
- **Classe 7:** Desempenho crítico ($F1=10\%$), com recall de apenas 6,67%. O modelo falha em identificar esta classe
- **Classe 4:** Baixa precisão (42,19%), sendo frequentemente confundida com outras classes
- **Classe 8:** Centro de confusões, sendo confundida com classes 1, 5 e 4

5.5 Visualizações

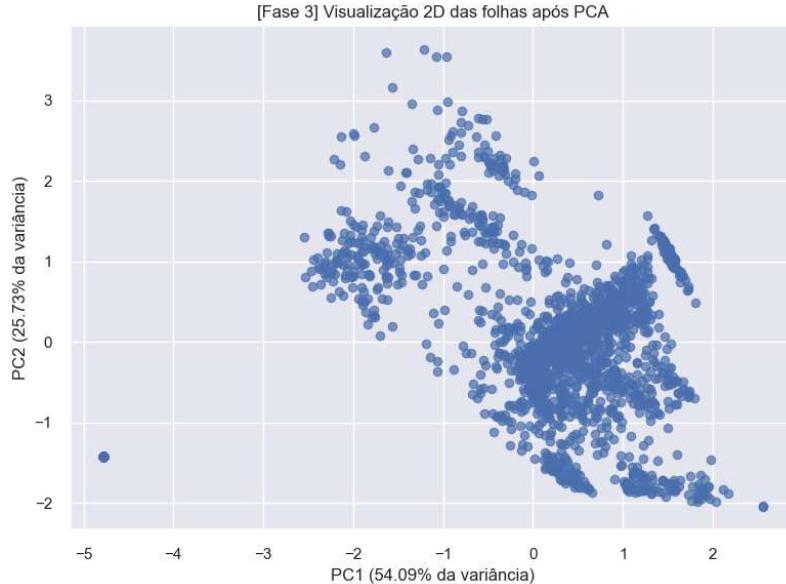
- **Figura 1:** Exemplo de máscara segmentada (Original | Máscara)



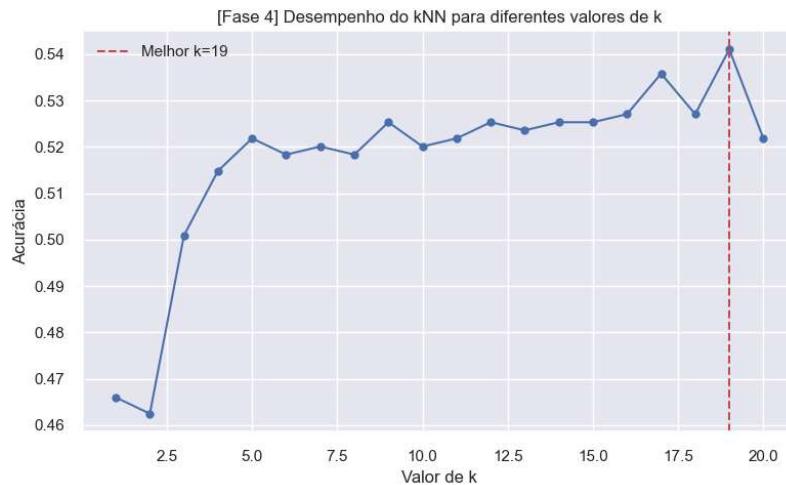
- **Figura 2:** Gráfico de variância explicada por componente principal (barras + linha acumulada)



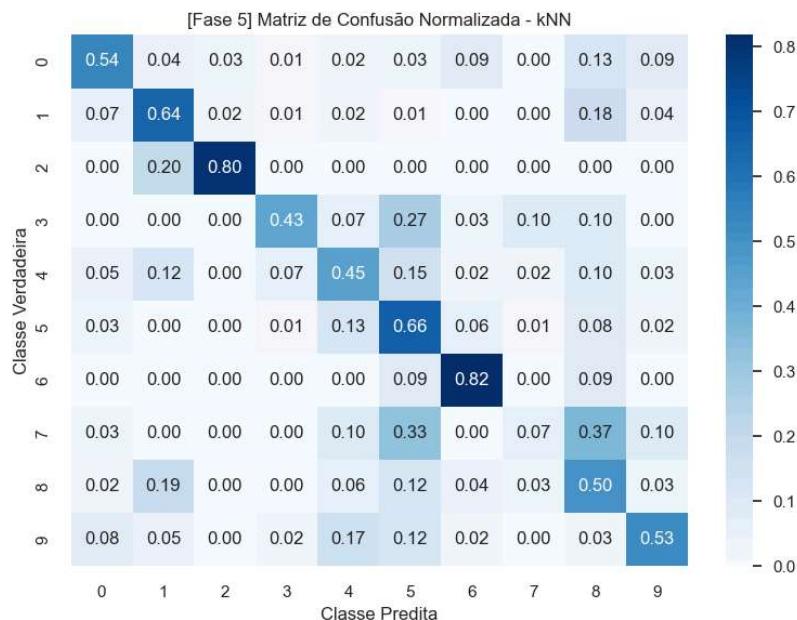
- **Figura 3:** Visualização 2D dos dados após PCA (scatter plot PC1 vs PC2)



- **Figura 4:** Curva de desempenho kNN (acurácia vs k, com k variando de 1 a 20)



- **Figura 5:** Matriz de confusão normalizada do classificador kNN (k=19)



6. Discussão

6.1 Interpretação dos Resultados

6.1.1 Por que o kNN superou o SVM?

- **Natureza local vs global:** O kNN toma decisões locais baseadas nos k vizinhos mais próximos, sendo mais robusto a sobreposições de classes. O SVM busca um hiperplano global ótimo, que pode ser inadequado quando há múltiplas regiões de sobreposição
- **Sensibilidade a hiperparâmetros:** O SVM não teve seus hiperparâmetros (C e γ) otimizados via grid search, usando apenas valores padrão. O kNN teve k otimizado empiricamente (testando 1-20)
- **Desbalanceamento de classes:** O SVM é mais sensível a desbalanceamento (classe 2 com 17 amostras vs classe 0 com 299). O kNN, por ser baseado em votação local, é naturalmente mais robusto
- **Dimensionalidade:** Com apenas 2 dimensões (PCA), o espaço de características é muito limitado para o SVM encontrar um hiperplano discriminativo eficaz

6.1.2 As características geométricas foram suficientes?

Resposta: Parcialmente suficientes, mas limitadas.

- **Pontos positivos:**
 - Acurácia de 54,10% está acima do baseline aleatório (10% para 10 classes)
 - Algumas classes (0, 1, 6, 9) foram razoavelmente bem classificadas ($F_1 > 56\%$)
 - Características são computacionalmente eficientes e interpretáveis
- **Limitações identificadas:**
 - **Saturação do número de cantos:** Todas as amostras atingiram o limite de 100 cantos, eliminando poder discriminativo desta feature
 - **Baixa variabilidade da excentricidade:** Valores muito concentrados (0,98-0,99) indicam que todas as folhas são muito alongadas, oferecendo pouca discriminação
 - **Falta de informação de textura:** Características geométricas ignoram padrões de nervuras, serrilhado fino e textura da superfície
 - **Falta de informação de cor:** Embora o dataset tenha fundo branco, variações sutis de cor/tonalidade foram descartadas na conversão para escala de cinza

6.1.3 O PCA manteve informação relevante?

Resposta: Sim, mas com perda significativa.

- **Preservação:** 79,82% da variância foi mantida com 2 componentes, o que é razoável
- **Perda:** 20,18% da variância foi descartada, sendo 15,87% apenas em PC3. Esta componente pode conter informação discriminativa importante
- **Recomendação:** Testar com 3 componentes (95,69% de variância) poderia melhorar o desempenho, especialmente para classes difíceis (7, 4, 8)
- **Trade-off:** A redução para 2D facilita visualização e interpretação, mas sacrifica desempenho

6.1.4 Quais classes foram mais difíceis e por quê?

Classe	F1-Score	Principais Problemas	Hipóteses
7	10,00%	Recall crítico (6,67%)	Características muito similares a outras classes (especialmente 3 e 4). Possível subrepresentação no espaço PCA
4	43,55%	Baixa precisão (42,19%)	Confundida com classes 5, 7 e 8. Região de sobreposição no espaço de características
8	46,39%	Confusão bidirecional com classe 1	33 erros mútuos (17+16) indicam características geométricas quase idênticas entre estas espécies
3	50,98%	Baixo recall (43,33%)	Frequentemente confundida com classes 5 e 7. Possível variabilidade intra-classe alta

6.2 Limitações do Sistema

1. **Características geométricas simples:** Apenas 4 descritores não capturam toda a complexidade morfológica das folhas (nervuras, textura, padrões de serrilhado fino)
2. **Saturação do detector de cantos:** Limite de 100 cantos foi atingido por todas as amostras, eliminando variabilidade desta feature
3. **Perda de informação no PCA:** Redução para 2 componentes descarta 20% da variância, potencialmente incluindo informação discriminativa
4. **Desbalanceamento de classes:** Classe 2 com apenas 17 amostras (0,9%) vs classes majoritárias com 300 amostras (15,7%) afeta o treinamento
5. **Fundo uniforme:** O sistema foi treinado apenas com fundos brancos, não generalizando para cenários reais com fundos complexos
6. **Hiperparâmetros não otimizados:** SVM usou parâmetros padrão ($C=1.0$, $\gamma='scale'$) sem busca em grid

7. **Ausência de validação cruzada:** Apenas uma divisão treino/teste foi usada, sem estimativa de variância do desempenho
8. **Confusões sistemáticas:** Pares de classes (1-8, 4-5) são consistentemente confundidos, indicando insuficiência das features

6.3 Trabalhos Futuros e Melhorias Propostas

6.3.1 Melhorias Imediatas (Curto Prazo)

- **Aumentar maxCorners:** Configurar detector Shi-Tomasi com maxCorners=200 ou 300 para evitar saturação
- **Testar PCA com 3 componentes:** Avaliar se os 15,87% de variância de PC3 melhoram a classificação
- **Otimizar hiperparâmetros do SVM:** Grid search para C (0.1, 1, 10, 100) e gamma (0.001, 0.01, 0.1, 1)
- **Implementar validação cruzada:** K-fold (k=5 ou k=10) para estimativa mais robusta do desempenho
- **Balanceamento de classes:** Aplicar SMOTE (Synthetic Minority Over-sampling) ou class_weight='balanced'

6.3.2 Adição de Características (Médio Prazo)

- **Momentos de Hu:** 7 momentos invariantes a rotação, translação e escala
- **Descritores de forma avançados:**
 - Solidez (área / área do convex hull)
 - Convexidade (perímetro do convex hull / perímetro)
 - Descritores de Fourier (análise de frequência do contorno)
- **Características de textura:**
 - LBP (Local Binary Patterns) para padrões de nervuras
 - GLCM (Gray-Level Co-occurrence Matrix) para textura estatística
 - Filtros de Gabor para orientação de nervuras
- **Características de cor:** Histogramas RGB/HSV, momentos de cor

6.3.3 Técnicas Avançadas (Longo Prazo)

- **Deep Learning:**
 - CNNs pré-treinadas (ResNet, VGG, EfficientNet) com transfer learning
 - Fine-tuning em dataset de folhas
 - Data augmentation (rotação, flip, zoom, distorções) para aumentar dataset
- **Ensemble methods:** Combinar kNN, SVM e Random Forest via voting ou stacking
- **Seleção de características:** Algoritmos como RFE (Recursive Feature Elimination) ou LASSO

- **Segmentação avançada:** U-Net ou Mask R-CNN para segmentação mais precisa
- **Generalização:** Testar em imagens com fundos complexos e condições de iluminação variadas

6.4 Comparação com Literatura

Trabalhos similares na literatura reportam acurárias variadas dependendo da abordagem:

- **Métodos tradicionais (features geométricas + ML):** 60-80% de acurácia
- **Métodos com features avançadas (Hu moments, Fourier, textura):** 80-90% de acurácia
- **Deep Learning (CNNs):** 90-98% de acurácia

Nosso sistema alcançou **54,10%** com apenas 4 características geométricas simples, o que está **abaixo da faixa típica** para métodos tradicionais. Isto confirma a necessidade de:

1. Adicionar mais descritores (especialmente Hu moments e textura)
2. Otimizar hiperparâmetros dos classificadores
3. Considerar migração para deep learning para ganhos significativos

7. Conclusão

Este projeto implementou com sucesso um pipeline completo de visão computacional para classificação de espécies de plantas a partir de imagens de folhas, processando **1.907 imagens** distribuídas em **10 classes**.

7.1 Principais Conquistas

- **Segmentação robusta:** 100% de sucesso usando limiarização de Otsu + operações morfológicas
- **Extração de características:** 4 descritores geométricos computacionalmente eficientes
- **Redução de dimensionalidade:** PCA manteve 79,82% da variância com 2 componentes
- **Classificação:** kNN ($k=19$) alcançou **54,10% de acurácia**, superando SVM em 8,38 pontos percentuais
- **Análise detalhada:** Identificação de confusões sistemáticas (classes 1-8, 4-5) e classes problemáticas (classe 7 com $F1=10\%$)

7.2 Lições Aprendidas

- **Características geométricas simples têm limitações:** Acurácia de 54% está abaixo do esperado para métodos tradicionais (60-80%), indicando necessidade de features mais sofisticadas
- **Saturação de features é problemática:** Número de cantos saturado em 100 eliminou poder discriminativo
- **kNN é robusto para este problema:** Superou SVM devido à natureza local das decisões e robustez a desbalanceamento
- **PCA com 2 componentes é limitante:** Perda de 20% da variância pode estar prejudicando classes difíceis
- **Desbalanceamento impacta desempenho:** Classe 2 (17 amostras) vs classes majoritárias (300 amostras) afeta treinamento

7.3 Considerações Finais

O sistema demonstrou que técnicas clássicas de visão computacional podem distinguir entre diferentes espécies de plantas com desempenho moderado (54,10%), validando a viabilidade da abordagem. No entanto, a acurácia obtida está aquém do necessário para aplicações práticas em botânica ou agricultura de precisão.

As limitações identificadas apontam caminhos claros para melhorias: (1) adicionar descritores de forma avançados (Momentos de Hu, Fourier) e textura (LBP, GLCM), (2) otimizar hiperparâmetros via grid search e validação cruzada, e (3) considerar migração para deep learning (CNNs) para ganhos significativos de desempenho (potencialmente 90-98% de acurácia).

O código desenvolvido está bem estruturado, documentado e pode ser facilmente adaptado para outros datasets de classificação de imagens, servindo como base sólida para trabalhos futuros.

7.4 Impacto e Aplicabilidade

Apesar das limitações, este trabalho contribui para:

- **Educação:** Demonstração prática de pipeline completo de visão computacional
- **Baseline:** Estabelecimento de desempenho de referência para métodos tradicionais neste dataset
- **Análise crítica:** Identificação detalhada de pontos fortes e fracos de cada etapa do pipeline
- **Direcionamento:** Recomendações concretas para melhorias futuras baseadas em evidências empíricas

8. Referências

- [1] Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y. X., Chang, Y. F., & Xiang, Q. L. (2007). *A leaf recognition algorithm for plant classification using probabilistic neural network*. IEEE International Symposium on Signal Processing and Information Technology, 11-16.
- [2] Otsu, N. (1979). *A threshold selection method from gray-level histograms*. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.
- [3] Shi, J., & Tomasi, C. (1994). *Good features to track*. IEEE Conference on Computer Vision and Pattern Recognition, 593-600.
- [4] Jolliffe, I. T., & Cadima, J. (2016). *Principal component analysis: a review and recent developments*. Philosophical Transactions of the Royal Society A, 374(2065), 20150202.
- [5] Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. Machine Learning, 20(3), 273-297.
- [6] Cover, T., & Hart, P. (1967). *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, 13(1), 21-27.
- [7] Flavia Leaf Dataset. Disponível em:
<https://www.kaggle.com/datasets/marquis03/flavia-leaves-dataset>
- [8] OpenCV Documentation (2024). *Image Processing and Computer Vision Library*. Disponível em: <https://docs.opencv.org/>
- [9] Scikit-learn Documentation (2024). *Machine Learning in Python*. Disponível em: <https://scikit-learn.org/>
- [10] Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.