

# Classificação Foliar de Ervas Medicinais Utilizando Algoritmos de Inteligência Artificial

Kenedy F. S. Silva  
Departamento de Computacao - DC  
Universidade Federal Rural de Pernambuco - UFRPE  
Recife, Pernambuco, Brasil  
kfssufrpe@gmail.com

**Resumo**—Atualmente obter resultados satisfatórios para reconhecimento de imagens tem se tornado algo desafiador, cada vez mais estamos adotando ideias que possam exercer esse papel de modo a ter uma eficiência capaz de ajudar a todos e trazer benefícios a sociedade, este artigo propõe uma ideia de classificação foliar comumente utilizadas através de algoritmos classificadores bastante conhecidos, como Árvore Decisão, Naive Bayes, KNN e SVM, buscando eficiência através da base de dados criada e validando as classificações comparando os dados estatísticos com teste de hipótese de Holm-Sidak, a criação da base manual nos trazem recursos reais, que podem ser realizados por qualquer pessoa, isso mostra que se pode ter a atuação do ambiente real para reconhecimento de plantas. Onde foi identificado que a questão chave no final das contas é ter uma boa seleção de recursos de entrada de imagens, dando alto poder e eficiência a classificação do algoritmo, conseguindo obter uma acurácia para a classificação de 95%. Utilizando recursos de Machine Learning com bibliotecas da linguagem Python, o efeito que a entrada dos dados causa é surpreendente com ajuda de framework processamento de imagens e reconhecimento de padrões os métodos de classificação conseguem atuar de forma altamente eficiente sem muito recurso computacional, podendo realizar a classificação de mais de 500 folhas em poucos segundos.

**Palavras-chave**—reconhecimento foliar, ervas medicinais, classificação de imagens, inteligência artificial

## I. INTRODUÇÃO

O reconhecimento de plantas pode ajudar na identificação de características que visam ajudar independentemente de sua localização, sabendo que plantas existem em todas as partes a sua identificação hoje é algo bastante importante além de ser uma área de pesquisa totalmente ativa. Observando que toda planta tem um valor medicinal, pode-se ter as classificações de tipos de plantas que podem trazer benefícios e malefícios ao ser humano, então podendo assim identificar a espécie da planta que está a sua frente, isso pode ajudar a identificar o melhor modo de exploração da planta como recurso medicinal ou até mesmo não ser contaminado com uma planta tóxica. Devido a essa possibilidade, este artigo propõe o reconhecimento das espécies através da imagem da folha, com a utilização de dados que busquem nos identificar qual o tipo (benefício ou malefícios), a espécie, a informação característica anatômicas da imagem da planta coletada, e utilizar Machine Learning na identificação irá proporcionar a coleta de dados e

análises de recursos existentes de forma mais eficiente visando sua utilização, Machine Learning irá automatizar a identificação diminuindo a quantidade de erros que podem ser obtidas por falha humana. Cada informação sobre a planta pode tornar o projeto de identificação algo robusto e eficiente, tornando assim a pesquisa algo de bastante utilidade para a população, com isso ter as melhores e mais detalhadas informações características em base de dados para que se possa realizar o reconhecimento de cada uma das folhas que possuam interesses medicinais.

## II. TRABALHOS RELACIONADOS

Em pesquisas realizadas, a classificação e reconhecimento de imagens para folhas e outras partes das plantas, tem se tornado amplo, e de interesse de muitos pesquisadores, busca-se contribuir com o conhecimento automatizado e inteligente sobre reconhecimento.

Pesquisas como a realizada pelo A. Sabu e K. Sreekumar [1] nos mostram como podemos realizar todo o processo de classificação de imagens, identificando como realizar extração de características para os atributos, características dos seguintes tipos: textura, forma, cores entre outras características, que podem ser utilizadas para compor a base de atributos para classificação desejada. Na mesma pesquisa de revisão são identificados pré-processamentos que poderão ser utilizados para realizar padronização de escala, orientação de imagens, o que ajuda a manter alinhamento e tornar feições invariantes, utilizamos recursos para realizar a remoção do background, revisão de classificadores, descrevendo-os e comparando-os.

Em pesquisa de Pacifico, Luciano DS, Valmir Macario e Joao FL Oliveira [2] realizam a classificação de plantas utilizando os algoritmos citados em [1] e incluem uma rede neural que melhora a classificação significativamente, utilizam o dataset da UCI para realizar os testes necessários para que possam melhorar a classificação das plantas.

R. Janani e A. Gopal em [5] realizam classificação de imagens onde utilizam para isso também uma rede neural artificial buscando atender e melhorar assim sua classificação, com extração de características sobre forma, texturas, cores e realizando combinações entre elas para que possa identificar qual a combinação com melhor eficiência, conseguindo assim realizar uma boa classificação.

### III. METODOLOGIA

Para iniciar o projeto, seguimos com a coleta de dados, realização de pesquisas e download de imagens reais, após banco de dados pronto, seguimos para o pré-processamento das imagens, com a separação das folhas, obtendo assim as informações desejadas para a classificação.

Do OpenCV utilizamos as funções `cvtColor` para obtenção dos níveis de cinza das imagens, `Threshold` e `findContours` para os contornos sobre a folha, e por fim utilização da máscara dos contornos com `bitwise` sobre a imagem original [16].

A biblioteca de aprendizagem de máquina *scikit-learn* [11] é o principal framework utilizado neste trabalho, considerando o problema de classificação através de aprendizado supervisionado, o *scikit-learn* contribui de forma a execução de algoritmo e funções já implementada, seu código open-source simples e de fácil mineração de dados é acessível e reutilizável.

Para a validação cruzada utilizamos o *k-Fold Cross Validation* para  $k = 10$  com 5 repetições totalizando assim 50 bases para testes de classificação para cada algoritmo, a função *RepeatedKFold* foi utilizada para realizar esta ação. Para execução da classificação com a validação cruzada foi utilizado a função *cross\_val\_score* do *scikit-learn* [11], função que realiza todo o treinamento dos dados inseridos e suas respectivas predições, com retorno das classificações é possível obter as seguintes informações de scoring: acurácia, desvio padrão e mediana, realizamos a comparação das execuções dos classificadores com as variações entre os algoritmos, buscando obter a melhor média de acurácia entre as classificações.

Utilizamos a função *cross\_validate* para realizar a verificação do tempo de treinamento de cada algoritmo, com scores de treinamento, teste e classificação. Os resultados obtidos foram utilizados como métricas de avaliação.

Por fim realizamos o teste de hipóteses para obter os dados estatísticos sobre a classificação, utilizamos a função *multipletests* [14], ela realiza vários testes retornando à informação sobre rejeição dos testes, corrigindo os valores p. O *multipletests* realiza o teste de *Holm-Sidak*, que faz cada teste em nível de significância limitado para garantir que a taxa falso-positiva da família não exceda o valor especificado [15].

### IV. BASE DE DADOS

Para realização das bases de dados foram utilizadas imagens coletadas da internet, seguindo as pesquisas de acordo com os nomes científicos de cada erva medicinal, informações sobre ervas que na atualidade são de utilização da área de medicina.

A busca resultou na coleta de várias imagens, ao qual foram pesquisadas com ênfase nas exibições das folhas resultando num total de 540 imagens exemplificada na Figura 1. Cada imagem destaca as folhas com suas respectivas características originais, visando a identificação de toda informação descrita na imagem.



Fig. 1. Exemplo das imagens do dataset (Fonte: Google Imagens)

### V. PRÉ-PROCESSAMENTO DAS IMAGENS

#### A. Separação das folhas

No processamento das imagens realizamos a separação das folhas, com destaque apenas as características específicas escolhidas para a classificação, são elas textura e cor, como pode ser observado na imagem a seguir (Fig. 2) este o resultado após o processamento de separação das folhas.



Fig. 2. Exemplo das imagens com a separação das folhas (Fonte: Google Imagens)

#### B. Remoção do Background

O framework OpenCV [16] com funções que realizam observações sobre imagens, nos permite detectar contornos e identificar como utilizar essa informação para extrair apenas os dados importantes para a classificação da imagem, este recurso realiza a utilização da imagem binária como máscara sob a imagem real da base de dados, deixando apenas a folha em destaque.

#### C. Extração de Características

##### 1) Características de textura

Características deste tipo nos fornece informações sobre cor e/ou intensidade que ocorre na imagem, conforme demonstrado em [4] utilizamos o GLCM (Grey Level Co-occurrence Matrix) para realizar a extração de características de textura, contraste, dissimilaridade, homogeneidade, energia e correlação. E por fim para textura foi calculado também a entropia de Shannon [8].

Cada uma das características é calculada através das funções determinadas em Mryka Hall-Beyer [4]:

- Contraste:

$$\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$$

- Dissimilaridade: (1)

$$\sum_{i,j=0}^{levels-1} P_{i,j} |i - j| \quad (2)$$

- Homogeneidade:

$$\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1 + (i - j)^2} \quad (3)$$

- Segundo Momento Angular (ASM):

$$\sum_{i,j=0}^{levels-1} P_{i,j}^2 \quad (4)$$

- Energia:

$$\sqrt{ASM} \quad (5)$$

- Correlação:

$$\sum_{i,j=0}^{levels-1} P_{i,j} \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (6)$$

- Entropia

$$\sum_i p_i \log_b p_i \quad (7)$$

onde  $p_i$  é a frequência ou probabilidade dos pixels de valor  $i$  aparecer na imagem [8].

## 2) Características de cores

Para a extração de informações de cores, utilizamos o cálculo de média (*mean*) e desvio padrão (*stddev*), as informações são realizadas de forma separadas em cada canal das cores das imagens das folhas, para o desvio padrão a realização do cálculo se faz apenas na diagonal da matriz de covariância normalizada [1].

$$N = \sum_{I, mask(I) \neq 0} 1 \quad (8)$$

$$mean_c = \frac{\sum_{I, mask(I) \neq 0} src(I)_c}{N} \quad (9)$$

$$stddev_c = \sqrt{\frac{\sum_{I, mask(I) \neq 0} (src(I)_c - mean_c)^2}{N}} \quad (10)$$

Onde N é a utilização da máscara sobre a imagem calculada.

## 3) Criação da Base de Características

Após o cálculo das informações características das imagens, todas são inseridas a um DataFrame com a informação de classe de cada uma das imagens, podendo assim ser classificada, com um total de 13 atributos, onde são coletadas da extração de características. (1 a 7 – Textura, 8 a 13 – Cores)

São eles:

1. Contraste
2. Dissimilaridade
3. Homogeneidade
4. Energia
5. Correlação
6. Segundo Momento Angular
7. Entropia
8. Média (R)
9. Média (G)
10. Média (B)
11. Desvio Padrão (R)
12. Desvio Padrão (G)
13. Desvio Padrão (B)
14. Classe

## VI. CLASSIFICADORES

Esta seção apresenta os classificadores que utilizamos nos experimentos, foram utilizados classificadores de aprendizado supervisionado, são eles: Árvore Decisão (Decision Tree Classifier - DT), Naive Bayes Classifier (NB), K-Vizinhos Próximos (K-Nearest Neighbors - K-NN), Máquina de Vetores de Suporte (Support Virtual Machine – SVM) e Regressão Logística (Logistic Regression).

Árvore Decisão é um método de classificação e regressão não paramétrico de inferência indutiva, que atua na divisão das informações em conjuntos de elementos chamados nós, onde possuem nó raiz, nós de decisão e nó de término. O seu funcionamento se realiza com a movimentação do nó raiz até o nó folha de acordo com as regras armazenadas para uma tomada de decisão.

Naive Bayes é o método baseado no teorema de Bayes com suposição ingênua de independência condicional entre recursos [9], utilizado o método Gaussian Naive Bayes que corresponde a seguinte função (11), ao qual é realizado o cálculo para a probabilidade gaussiana estimada por máxima verossimilhança.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (11)$$

K-NN é o método de classificação utilizado por proximidade dos vizinhos, que possui o tipo de aprendizado baseado por instância, utilizando a aprendizagem e armazenado dos dados a partir da proximidade majoritária dos vizinhos para cada ponto, onde  $k$  é um valor inteiro escolhido pelo usuário para informar a quantidade de vizinhos mais próximo, utilize funções de distância para verificar a proximidade, comumente utiliza-se a distância euclidiana.

No próprio KNN existente algumas variações, utilizamos uma delas chamada KNN Ponderado onde podem ser utilizados pesos uniformes, de distância, de escolha do usuário, visando assim ponderar a classificação realizada.

Realizamos os testes com três variações (valores 3, 5 e 7) para o  $k$  do KNN e para o WKNN, incluindo no WKNN a informação de pesos uniforme e distância.

SVM é a técnica baseada no plano de decisão que analisa os dados e reconhece os padrões para realizar a classificação, encontrando o hiperplano que diferencia muito bem as duas classes [10]. O SVM possui uma versatilidade onde pode ser escolhido diferentes funções de kernel para realizar na decisão da função, na nossa classificação foram utilizadas as versões de kernel *linear* e *rbf*, normalmente os núcleos lineares e polinomiais são menos demorados e fornecem menos precisão do que os núcleos *rbf* ou gaussiano.

Regressão Logística é um modelo linear de classificação que descreve resultados possíveis de um teste, com isso é calculada a probabilidade de sucesso da classificação, normalmente o valor categórico da variável é do tipo binária. A regressão analisa dados distribuídos de forma binomial (12), onde a forma dos números de ensaios de Bernoulli  $n_i$  são conhecidos como probabilidades de êxito e  $p_i$  são valores desconhecidos [13].

$$Y_i \sim B(p_i, n_i), \text{ for } i = 1, \dots, m \quad (12)$$

Posteriormente com os valores obtidos para cada ensaio, ao qual pode ser informada a probabilidade final, através de um vetor que possui a seguinte informação:

$$p_i = E\left(\frac{Y_i}{n_i} \middle| X_i\right) \quad (13)$$

E com isso são calculados os logits de probabilidades binomiais desconhecidos, através de uma função linear dos  $X_i$ .

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i} \quad (14)$$

Os valores de  $\beta$  são estimados através do método de máxima verossimilhança.

A regressão logística visto como um problema de otimização, penalizada na classe binária L2 minimiza a seguinte função de custo [11].

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^n \log\left(\exp(-y_i(X_i^T w + c)) + 1\right)$$

Onde,  $y_i$  possui valores no conjunto  $\{-1, 1\}$ .

## VII. ANÁLISE EXPERIMENTAL

Para cada teste conseguimos realizar a classificação obtendo êxito na maioria delas, tivemos um índice de classificação variando entre 60% e 95% (Fig. 4) a depender do classificador, cada classificador foi executado por 5 vezes com 10 folds de validação cruzadas, tivemos um total de 13 classificadores, sendo alguns deles com variações como o KNN e WKNN que possuem o número  $k$  variando entre 3, 5 e 7 e o WKNN com pesos de *distance* e *uniform* o qual possuem pontos de ponderação pelo inverso da distância, para o KNN peso é padrão para qualquer vizinho, para o SVM tivemos a variação do kernel sendo eles *linear* e *rbf*, resultando assim num total de 650 testes.

Cada resultado do teste é salvo e verificada a média de cada validação cruzada para análise de testes de hipóteses, ao chegar no final identificamos os seguintes resultados obtidos pela classificação.

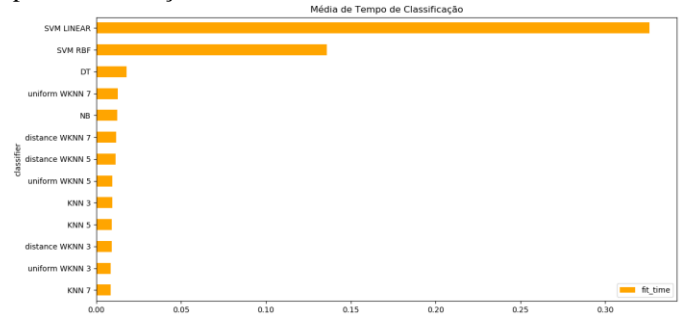


Fig. 3. Média de Tempo de Classificação

Os algoritmos de classificação que mais se destacaram foram o DT e o WKNN, com resultados acima de 90% de acurácia e desvio padrão de  $\pm 0.03$  (Fig. 3).

Por fim realizamos os testes de hipótese com o multitests citado anteriormente, nos mostrando a classificação da acurácia e se o teste é rejeitado, obtendo o seguinte ranking sobre os classificadores e seus respectivos valores de  $p$ .

Em todos os resultados obtidos, tivemos falso para rejeição da validação cruzada obtida, e valores  $p < 0.01$ , podendo assim validar a classificação realizada.

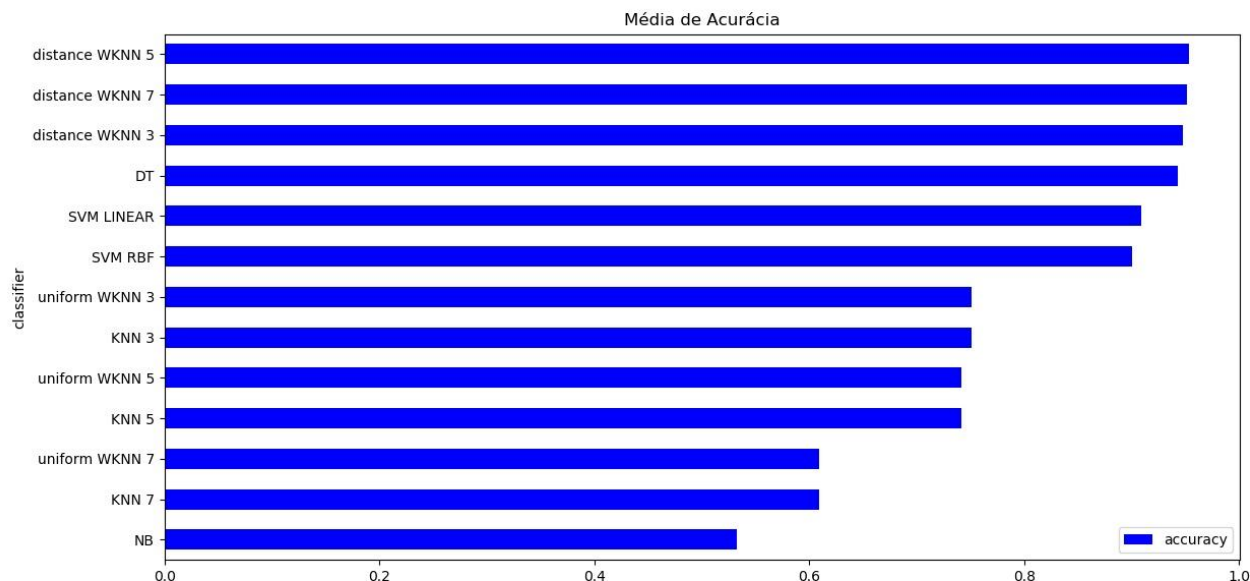


Fig. 4. Média de Acurácia

TABELA I. MÉDIAS OBTIDAS

Classificador	Médias		
	Acurácia	Desvio Padrão	Mediana
NB	53,40%	0,05586	53,70%
KNN 7	60,70%	0,05461	61,11%
U WKNN 7	60,70%	0,05461	61,11%
KNN 3	74,91%	0,06009	75,70%
U WKNN 3	74,91%	0,06009	75,70%
KNN 5	75,25%	0,05447	75,70%
U WKNN 5	75,25%	0,05447	75,70%
SVM LNR	91,25%	0,04536	92,52%
SVM RBF	91,25%	0,04251	91,60%
DT	94,14%	0,03728	94,44%
D WKNN 3	94,36%	0,03334	94,44%
D WKNN 7	94,70%	<b>0,03162</b>	94,44%
D WKNN 5	<b>94,77%</b>	0,03506	<b>94,44%</b>

## VIII. CONCLUSÃO

Por se tratar de uma base de dados de criação e ser algo novo, conseguimos obter classificações satisfatórias, observando que resultados acima de 60% nos identifica que cada classificação conseguiu ter um índice de acerto maior, deixando como ideia utilizar métodos de reconhecimento de folhas com algoritmos mais inteligentes onde conseguiremos obter resultados ainda melhores sobre cada uma das ervas da base.

## REFERÊNCIAS

- [1] A. Sabu and K. Sreekumar, "Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach," in Inventive Communication and Computational Technologies (ICICCT), 2017 International Conference on. IEEE, 2017, pp. 145–149.
- [2] Pacifico, Luciano DS, Valmir Macario, and Joao FL Oliveira. "Plant Classification Using Artificial Neural Networks." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.
- [3] Cheirsilp, Ronnarit. 3D multimodal image analysis for lung-cancer assessment. The Pennsylvania State University, 2016.
- [4] Hall-Beyer, Mryka. (2017). GLCM Texture: A Tutorial v. 3.0 March 2017. 10.13140/RG.2.2.12424.21767.
- [5] R. Janani and A. Gopal, "Identification of selected medicinal plant leaves using image features and ANN," 2013 International Conference on Advanced Electronic Systems (ICAES), Pilani, 2013, pp. 238-242. doi: 10.1109/ICAES.2013.6659400
- [6] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification", IEEE Trans. On Systems, Man and Cybernetics, 1973, pp.610-621.
- [7] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, A. Bugarín, STAC: a web platform for the comparison of algorithms using statistical tests, in: Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2015.
- [8] SHANNON Entropy. 1. 2017. Disponível em: <[https://en.wiktionary.org/wiki/Shannon\\_entropy](https://en.wiktionary.org/wiki/Shannon_entropy)>. Acesso em: 06 nov. 2018.
- [9] 1.9 Naive Bayes: Gaussian Naive Bayes. 2017. Disponível em: <[https://scikit-learn.org/stable/modules/naive\\_bayes.html#gaussian-naive-bayes](https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes)>. Acesso em: 07 nov. 2018.
- [10] Lorena, Ana Carolina, and André CPLF de Carvalho. "As Máquinas de Vetores Suporte." (2002).
- [11] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.b
- [12] Kadir, Abdul, et al. "Leaf classification using shape, color, and texture features." arXiv preprint arXiv:1401.4447 (2013).
- [13] REGRESSÃO Logística. 2018. Disponível em: <[https://pt.wikipedia.org/wiki/Regress%C3%A3o\\_log%C3%ADstica](https://pt.wikipedia.org/wiki/Regress%C3%A3o_log%C3%ADstica)>. Acesso em: 09 nov. 2018.

- [14] Perktold, Josef, Skipper Seabold, and Jonathan Taylor. "Statsmodels: Statistics in python." (2017).
- [15] OPÇÕES ANOVA: Testes post hoc. Disponível em: <[https://www.ibm.com/support/knowledgecenter/pt-br/SSEP7J\\_10.1.1/com.ibm.swg.ba.cognos.ug\\_cr\\_rptstd.10.1.1.doc/c\\_id\\_obj\\_anova.html](https://www.ibm.com/support/knowledgecenter/pt-br/SSEP7J_10.1.1/com.ibm.swg.ba.cognos.ug_cr_rptstd.10.1.1.doc/c_id_obj_anova.html)>. Acesso em: 06 nov. 2018.
- [16] Mordvintsev, Alexander, and K. Abid. "Opencv-python tutorials documentation." Obtenido de <https://media.readthedocs.org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals.pdf> (2014).