Exploring Machine Learning Classifiers in Classifying Shape and Texture Features of Plant Leaf

Jefry Christiawan
Computer Science Department
Faculty of Science & Engineering
University of Nottingham Ningbo
Ningbo, China
sgyjc4@nottingham.edu.cn

***Abstract*—In the era of digitalization, the utilization of computer in plant leaf identification is important. Computers are always considered as a powerful tool and better than humans, that's why the use of computers can solve various real-life problems. In terms of plant identification, computers need to be involved to solve botanist limitations in recognizing plants. In this case, we want to focus on identifying the plant based on its leaf because it's simpler and easier to start with even for the botanist. In this experiment, we first utilize previous hand-crafted feature extraction that manage to achieve the best performances which is Multiple Triangle Descriptor (MTD) for extracting shape features and Local Binary Pattern Histogram Fourier (LBP-HF) for extracting texture features. We then combine both features with weight 0.5. In this experiment, we use Swedish Leaf dataset and CVIP100 Leaf dataset. We train their features into 5 different machine learning models, which are K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Gaussian Naïve Bayes, Logistic Regression, and Decision Tree. The best performance in our experiment was achieved by the logistic regression model.**

Keywords—MTD, LBP-HF, KNN, SVM, Logistic Regression, Naïve Bayes, Decision Tree

I. Introduction

Plants are one of the most essential elements in our life. They offer a lot of benefits in various aspects of our life. For instance, they can become medicine, food, or even as an environment protector. However, in order to actually get those benefits, a correct plant identification is one of the important actions. Nowadays, biologist is facing a lot of struggle in correctly identifying plants [1]. Firstly, it's because learning plant species require them to see living specimen, which is quite difficult since different leaf characteristics live in different environments. Secondly, variability of plant species become the most challenging part of the identification and require a lot of experiences.

Following that limitations, many researchers are now starting to explore the use of computers to help with the plant identification process. Most research focused on identifying plants through their leaves. The reason is because leaves are the easiest part of the plant to find and collect and they often represent the characteristics of the plant. In comparison, other part of plants, like flowers are more difficult to work with. Firstly, because it takes a longer time for plant to grow flowers and usually it only grows at spring season. Secondly, the features of the flower is more complicated because the shape is quite vary and come in different color and texture. In plant leaf identification, color, shape, and texture are the most commonly used features. These features can be utilized individually or combined. However, when it comes to color, it is rarely used individually as most leaves shared similar color, like green or red. Because of that, the utilization is usually alongside with shape and textures. Unlike shape and textures, the color features only sometimes affect the identification in small cases.

Looking specifically at shape features, some researchers have tried to create an algorithm to extract it. The main objective in capturing shape features is by extracting the contour and calculating the geometry part of the leaf. For example, Yang et al. [2] used Multiple Triangle Descriptor which draw triangle for every three contours to describe the leaf shape, then apply fourier transform to make the shape more robust. Other implementations that build on the same concept include Improved Multiple Triangle Descriptor (IMTD) [3], High-level Triangle Shape Descriptor (HTSD) [4] are built to enhance the level of understanding of shape features. Besides using triangle, another research also try to use deep learning architecture [5] to learn the shape features. Another more traditional descriptor is proposed by Marko et al. [6] that apply Hu moments that are more focus on contour-based and extracting fixed 7 features. A various algorithms to describe shape allows researcher to keep doing experiment to see what's best in order to improve the leaf-identification performance.

Although shape is quite precise in identification, relying too much on one feature is not as precise as combine two features. The problem with only using shape features is some leaves may have the same shape. Because of that, some researchers try to explore another feature, which is texture features. The extraction of textures feature is based on pixel intensities of the image. Some algorithms that exist are Local Binary Patterns (LBP) [6], Laplacian of Gaussian (LoG) filter [7], deep learning architecture [5] to learn texture features, etc. The usage of texture features usually go hand by hand with shape features. The main reason is because with shape only, many researchers are able to get accurate identification. Because of that, texture only acts as supportive features in most cases.

The main objective of this research is to utilize previous hand-crafted techniques in extracting features and then apply machine learning algorithms for classification processes. In this experiment, we will apply Multiple Triangle Descriptor as a shape descriptor and Local Binary Pattern Histogram Fourier as a texture descriptor. The description and explanation of both algorithms in detail can be found in Yang et al. [2] research. We believe that in term of image identification and classification task, deep learning is the best approach. However, the computationally intensive always becomes main consideration especially when the data is really large. It usually consumes run-time and a lot of efforts for training and fine-tuning. Due to less accuracy and precision of machine learning compared to deep learning, many researchers are not considered it anymore. That's why, in this research we want to focus on looking at the performance of 5 different machine learning algorithms, which are Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes, Logistic Regression, and Decision Tree. Besides that, we will also make comparison with previous research to see if the machine learning algorithms can perform better or not in different leaf datasets.

II. Related Works

A. Hand-Crafted Features

In recent years, many algorithms have been developed in order to extract the best features to describe leaf. Some researchers focus on shape-only features and texture-only features. However, nowadays researchers find that combination of shape and texture features is the best way to describe leaf. That's why many researches nowadays focus on combination on both of them.

Starting from shape-only features, researchers have developed several ways to apply on shape descriptor in leaf identification process. For instance, Yang et al. [8] in 2016 proposed Multiscale Triangular Centroid Distance (MTCD). MTCD worked by looking at the contour point and then draw different triangle. After that, the calculation of dissimilarity measurement was using a simple cosine distance. By utilizing a simple way to describe shape, they managed to gain high accuracy and efficiency. Another new approach done on research in 2018 [9] using Local Area Integral Invariants (LAIIs). This method was extracted on 5 different scales. This features has abilities to generalize the data well and also support scale and rotation invariance. Besides that, the efficiency of computation was better compared to deep learning architecture, such as Convolution Neural Network (CNN). In 2019, Multiscale Fourier Descriptor (MFD) [10] is introduced. This descriptor actually is the improvement of MTCD. Rather than only consider the centroid of triangle, MFD considers areas, angles, ratios of two side lengths, and center distance of triangle. Also, for dissimilarity measurement, MFD apply logarithmic distance. This improvement allows increasing performance in retrieving and distinguishing shape. Although representation of shape performs well with triangle shape descriptor method, another method that proposed by Xin Chen & Bin Wang [11] also give us almost perfect accuracy in leaf classification (99%). The method that they proposed was histogram of gaussian convolution vectors (HoGCV). HoGCV focused on solving the problem that many leaf have similar characteristics in terms of shape. Their method also have already supported scale and rotation invariance with the ability to picture local geometry in multiple scales. Other recent variations of triangle approach in describing shape features is High Level Triangle Shape Descriptor (HTSD) [4]. The difference with both previous similar features is now these features are able to describe not only the contour point but also the salient point of the leaf. The combination of both contour and salient points are encoded with fisher vector. The final calculation of dissimilarity measurement is by utilizing simple Euclidean distance. Improvement of the triangle shape descriptor gave higher recognition performance.

Some researchers also conducted research along the texture-based features. However, the utilization is not as popular as shape-based features. In 2010, the process of texture analysis is done by joint distribution of different scales in Gabor Filter [12]. Then, use Jeffrey-divergence measure to calculate the difference between textures. By applying that, they managed to reach approximately 80 in the performance of two leaf datasets. Another method was proposed in 2018, which is GIST Texture features [13]. The researchers try to utilize Principal Component Analysis (PCA) and explore three different classifiers. However, those methods only perform best on Flavia Dataset. Shadi et al. [7] in 2020 try to utilize Gabor Filter again for the texture features but they used semi-supervised spherical K-means Clustering to handle most unlabeled data. Overall, the texture features itself have lower performance compared to shape-based features. That's why the implementation is not quite popular among researchers.

Nowadays, researchers focus on finding the best combination of features, like color, texture, shape, etc. to describe the leaf. In 2014, the fusion of shape and texture features are introduced using a set of curvelet transform coefficient and Gabor Filter with Gray Level Co-occurrence Matrix respectively [14]. With Neuro-Fuzzy Controller (NFC) as a classifier, they managed to achieve 97.6 % of accuracy. Marko et al. [6] proposed the combination of Hu Moments and Local Binary Pattern (LBP). After extracting the features, they applied PCA to reduce the dimensionality and Support Vector Machine as Classifier. The main strength of this research is its simplicity and high accuracy which is approximately 94 % in classification on Flavia dataset. Another research by Yang et.al [2] combined Multiple Triangle Descriptor (MTD) as a shape descriptor and Local Binary Pattern Histogram Fourier (LBP-HF) as a texture descriptor. Basically, MTD is the improvement of MTCD and MFD and LBP-HF is considered as the variation of LBP where LBP-HF can deal with rotation change. The combination of both features is using weighted distance where they consider shape and texture features are equally important. With simple K-Nearest Neighbor classifier, they managed to achieve the best performance compared to other previous methods. Recently, different approaches to feature combinations are being proposed. For instance, the utilization of texture features and combined with multispectral features is introduced [15]. They used 5 different machine learning algorithms as the classifier and managed to reach approximately 99% of accuracy.

B. Deep Learning Architecture

The advancement of technology brings deep learning in popularity among several various tasks. In terms of image identification and classification, Convolutional Neural Network is the most powerful method and mostly used in various image tasks. Meet, et al. [5] introduced dual-path CNN. Basically this method focuses on building two pathway of CNN. One path for learning shape features and the others for texture features. After that, merge both of the pathways. The result of this architecture managed to achieve 99% of accuracy in average. New utilization of CNN was applied by Sue et al. [16] They proposed plant identification based on multi-organ. Based on the way botanist identify plants which is by exploring leaves, flowers, and other organ to specifically recognize plants, the researchers want to build an architecture that mimicking botanist way of plant recognition. In this case, they use Hybrid Generic-Organ Convolutional Neural Network (HGO-CNN) and together with Recurrent Neural Network (RNN) for learning the structure of the plant. Jiachun et al. [17] proposed simpler way by just using CNN directly with ten layers and with augmented data. However, the

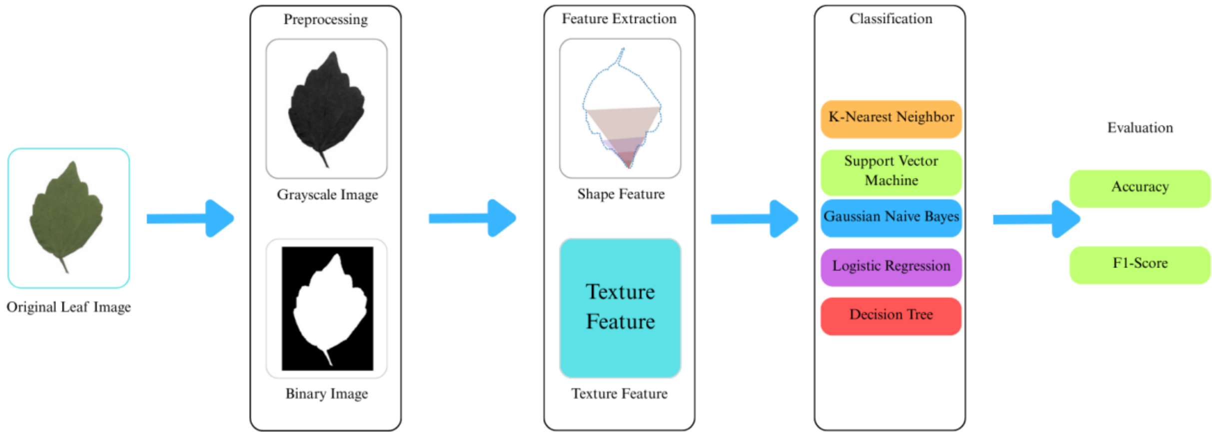

Fig. 1. The pipeline of our approach for classifying leaf images

performance is not as better as hand-crafted feature which only 87.92% in overall accuracy. Ali et al. [18] simulated the botanist process in three steps of identifying plant species. They built three models for each process. The main purpose of the model was actually to increase the confident in identifying plant. So, suppose the first model is not really sure, then it will be learned further by second model until the third model. It's really similar like the way botanist learn and managed to achieve the performance of approximately 99% of accuracy. Another way to extract features is by combining hand-crafted features and convolutional features. A research conducted by Hao et al. [3] improved the MTD features and combine with convolutional features. They also make comparisons with other previous methods and shows that their method was achieving the best performance in classification and feature extraction.

III. PROPOSED METHOD

We decide to use hand-crafted features and explore machine learning classifiers to gain best performance through simplicity. In this case, we will utilize Multiscale Triangle Descriptor (MTD) as a shape descriptor and Local Binary Pattern Histogram Fourier (LBP-HF) as a texture descriptor. This feature extraction method was proposed by Yang et al. [2] in 2021. The reason is because both of the feature descriptor achieve the best leaf retrieval with the highest Mean Average Precision (MAP) as a hand-crafted feature. Also, that method achieve quite good performance in classification process. After the feature extraction done, we want to perform classification task by utilizing those combined features and feed it in various machine learning classifiers. We will try to find the best parameters for each machine learning classifier and see which one of the algorithms will achieve the best accuracy and F1-Score. Compared to deep learning, the conventional machine learning classifier is considered more powerful because the computation is very fast and by hyperparameter tuning, we can simply maximize the performance and always reuse it. On the other hand, deep learning is focused on learning the data through many layers and then make decisions. That's why usually it takes a long time and also sometimes the performance is not as good as machine learning.

A. *Pre-Processing*

First, we need to find the contour points of each image. To do that we need to first convert the image into grayscale image. After that apply Otsu thresholding in order to extract binary image. Finally, we find the contour points of each outer part of the leaf. However, we get some problems which the contour points is too much to be used in MTD. That's why we need to resample it since a lot of features or triangle that will be extracted by MTD will increase the complexity. Remember that the contour must be evenly resampled. In this case, we use linear interpolation because it's simpler and effective.

B. *Shape Feature Extraction (MTD)*

In the pre-processing steps, we have already resampled the contour points into N points. Now we want to form the triangle based on the contour points. Let me introduce three important variables: Ts, k, and d(k). First, we calculate the Ts with formula $\log_2(N/2)$. Ts will represent the number of triangles scales we will form. So, for each contour point we will see Ts number of triangles. Next, k represents each scale range from 1 to Ts. For each scale, we will compute triangles. After that, d(k) will represent what point you need to pick in order to form triangles. In this case, the formula of d(k) will be based on logarithmic distance with formula $d(k) = 2^{k-1}$. So, for each contour point P_i, we will form triangles such that the three points used for forming triangles are P_i, $P_{i+d(k)}$, and $P_{i-d(k)}$. In total there will be N x Ts number of triangles that are being formed.

Next, we calculate the signed area of triangles which are defined as $TSA_k(i)$ for each triangle. The result will be split into two values which is degree of bend and the orientation. The way we calculate the $TSA_k(i)$ will go as follows:

$$M = \begin{bmatrix} x_{i-d(k)} & y_{i-d(k)} & 1 \\ x_i & y_i & 1 \\ x_{i+d(k)} & y_{i+d(k)} & 1 \end{bmatrix} \tag{1}$$

$$TSA_k(i) = \frac{1}{2} . det(M) \tag{2}$$

Degree of bend $\alpha_k(i)$ is calculated by taking the absolute value of $TSA_k(i)$ and the sign function $\beta_k(i)$ is set to 1 when $TSA_k(i)$ is positive value, otherwise it's 0. From that, we have already extracted two of three features for MTD. The next feature is the center distance of the triangle. We first need to find the centroid point of x and y which I will define as x_c & y_{c}.

$$\begin{cases} x_c = \frac{x_{i-d(k)} + x_i + x_{i+d(k)}}{3} \\ y_c = \frac{y_{i-d(k)} + y_i + y_{i+d(k)}}{3} \end{cases} \tag{3}$$

Then we calculate the center distance of the triangle for each contour point.

$$\gamma_k(i) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{4}$$

Next, conduct the normalization locally for each contour point for feature α and γ. Then apply discrete fourier transform to eliminate the influence of starting points in the feature performance. Finally, we have done the extraction of shape features that are invariant of scale, rotation, and translation.

C. Texture Feature Extraction (LBP-HF)

The construction of LBP-HF has purpose to solve rotation invariance problem of LBP features. First, compute the LBP values. In this case, we will use uniform LBP values to reduce the computation cost. Now, the only addition that Yang et al. made to build LBP-HF features is by employing Discrete Fourier Transform, then maintain the rotation invariance using the amplitude of Fourier transform. It's important that each texture feature is scale, rotation, and translation invariance so that, the same leaf that are scaled, translated, and rotated can be recognize as the same leaf. If not that case, the classifier model may be fail to classify it.

D. Machine Learning Classifier

All these features will be trained in machine learning model and assessed the performance with accuracy and F1-Score. Before we train it into machine learning model, I add additional step which is scaling the feature. Based on my experiments, standard scaler is a game changer when we apply it. It makes the model able to perform well. Before I applied it, some of the machine learning model's performances were very bad. The purpose of scaling is to make all values in the scale that we want, so that there is no value that are too high or too small and affecting the model's performance. Although this is quite simple pre-processing, it has a high impact on the performance. These 5 machine learning classifiers are defined as follows:

1) K-Nearest Neighbor (KNN)

This model is the simplest model for classification. The algorithm behind this is just by looking what the k nearest neighbor is classified as. In this case, I use k = 1 as the hyperparameter. The reason is because this hyperparameter provides highest performance result.

2) Support Vector Machine (SVM)

The objective of Support Vector Machine is to find the best separator for the data. In this case, we use Radial Basis Function (RBF) as the kernel. The reason is because our data has too many labels and there is a possibility that the model cannot find a linear separator. So that, we need to map it in higher dimensional space and find linear separator through it using RBF kernel. Then for other parameters like regularization parameters C is set to 10 and gamma γ is set to 0.001.

3) Naïve Bayes

Naïve Bayes is considered as one of the most well-known classifiers. The algorithm works by calculating the probability that a feature belongs to the label. In this case, we use Gaussian Naïve Bayes.

4) Logistic regression

The utilization of logistic regression is using sigmoid function to calculate the probability of the feature belong to the class. In this case, the parameter of logistic regression that we used is max iteration = 1000.

5) Decision Tree

This algorithm works by construction tree to predict the classification. It will keep splitting the node until the criterion is met (e.g., maximum depth). In this case, for the parameter, I just use the default.

IV. Experimental Results

A. Experimental setup

First, we need to combine shape and texture features. We will use weight 0.5 for each feature because we consider the importance of both features is the same. After combined the feature will be trained by machine learning classifier to be able to make classification. Later on, the performance of classifier will be assessed by accuracy and F1-Score. Accuracy is used to see how many leaves are correctly classified. On the other hand, F1-Score will assess the performance when the dataset is imbalanced.

We will use two datasets which are Swedish datasets and CVIP100 datasets. These both datasets are quite popular among leaf researchers to identify the plants. First, we will look at our 5 machine learning models performances. Then, we see which one has the best performance. Then, apply it to another dataset and observe if the same model will give the high performance or not. We will also compare with the previously used method which K-Nearest Neighbor to see if maybe another model can outperforms KNN. In order to make sure, the model is generalized well, we use 20 times random selection and outputting the accuracy and F1-Score by the average of it.

Besides that, we will make comparison with other methods, such as deep learning architecture on the same dataset. Some researchers built layer by layer of deep learning in order to get high performance. However, it takes very long time to really understand the data which means deep learning usually considered as computationally intensive which is in some case, it may be not really compatible. On the other hand, machine learning is more powerful. Suitable algorithms for some data will simply give the best performance and probably in some cases, they produce higher performance than any deep learning architectures.

B. CVIP100 Dataset

CVIP100 Dataset contain 100 leaf categories with each category contains 12 leaf images. Less data for each category and multiclass classification become of the biggest challenge for researchers. The preview for 100 images sample of CVIP100 Dataset can be seen in Fig. 2. The sample is printed for each category and as we can see, the shape, texture and color are various. So, those three features can be utilized in

Fig. 2. CVIP100 Leaf Dataset

TABLE I. CVIP100 Classification Result

Model	Accuracy	F1-Score
K-Nearest Neighbor	93.54%	93.52%
Support Vector Machine	96.18%	96.17%
Gaussian Naïve Bayes	89.75%	89.83%
Logistic Regression	**96.44%**	**96.47%**
Decision Tree	60.42%	59.64%

order to maximize the result. Unfortunately, previous research with the same feature extraction method did not consider this dataset in their experiment.

In this research, we have already tested the performance of five different machine learning algorithms. Referring to table 1, we can see the worst model is performed by decision tree with accuracy 60.42 % and F1-Score 59.64 %. On the other hand, the best performance was achieved by logistic regression algorithms with 96.44% of accuracy and 96.47% of F1-Score. Although, other algorithms are not perform as well as logistic regression, their accuracy and F1-Score was categorized as a good result with the range from approximately 89% to 96%. The most noticeable result in this case is SVM has almost the same performance as logistic regression.

C. Swedish Dataset

Fig. 3. Swedish Leaf Dataset

Unlike CVIP100 Dataset, Swedish Dataset contains more datas in each category. This dataset have 15 categories and for each category contain 75 leaf images. Of course, this dataset will be simpler and easier to identify rather than CVIP100 dataset because the class that need to be classified is not too much. Also, we have sufficient data for each category to make our machine learning model perform better. In previous research with the same feature extraction method, the KNN classifier with 1 nearest neighbor managed to achieve 98.48% of accuracy and 98.54% of F1-Score.

TABLE II. Swedish Leaf Classification Result

Model	Accuracy	F1-Score
K-Nearest Neighbor	96.82%	96.82%
Support Vector Machine	98.64%	98.64%
Gaussian Naïve Bayes	93.30%	93.46%
Logistic Regression	**98.74%**	**98.74%**
Decision Tree	88.57%	88.62%

In this research, we expand into the other machine learning model. However, the best performance is still achieved by logistic regression. Note that, when I do this experiment, I find out that my KNN performance result is different from the previous research. In this experiment, my KNN algorithm performs slightly lower performance with 96.82% accuracy & F1-Score. This may be caused by different experimental setups where the previous researchers took 25 images for each category to train. In our case, we just randomized the data that are needed to train. The reason is because there is no need to maintain the balance data since the performance will be assessed by F1-Score. Although the performance of KNN slightly lower, the performance of logistic regression slightly outperforms the KNN model with 98.74 accuracy & F1-Score.

D. Discussion of Result

The most noticeable part of the five machine learning algorithms is the performance ranking from five of them remains the same with the worst algorithm was performed by decision tree algorithm and the best was performed by logistic regression algorithm. Looking specifically at the decision tree algorithm, the performance is significantly different between two datasets. This might be caused by the dataset. For the dataset with multiple class, which is CVIP100 leaf dataset, decision tree might end up with a lot of branches of tree where it's not only not effective but also has bad performance. Besides the decision tree, the other four models managed to reach the performance up to 90% of accuracy and F1-Score.

Referring to Hao et al. [3] research, we will look at the comparison of previous algorithms. First, let's start by comparing the performance of CVIP100 Dataset. In that research, they provide the results from MTD + LBP-HF feature on CVIP100 dataset and the result is 1% higher than our best result. Looking specifically at the deep learning model, only 1 out of 6 deep learning models which is VGG16 + relu5_2 outnumbered our machine learning model performance. This proof that the simplicity of machine learning model combined with hand-crafted features is still better than deep learning architecture in terms of plant identification. On the other hand, on Swedish leaf dataset, none of the deep learning model outperformed our machine learning model. Only the combination of hand-crafted features and convolutional features that managed to outperform our leaf identification performance in that case. The highest deep learning model performances were achieved by AlexNet + relu5 model and VGG16 + relu5_2 with 98.67% in accuracy and 98.68% in F1-Score. As a result, this shows that logistic regression + MTD & LBP-HF features are quite powerful if we combined with correct machine learning algorithms.

V. Conclusion

We have explored five different machine learning models in leaf classification. First, we extracted the leaf features by combining the shape and texture features with MTD and LBP-HF method. Then, we try to find the best hyperparameter for each of the machine learning models. The best classification result was achieved by logistic regression model. Although, the other 4 models were not achieving the best result, the performance can be still categorized as good performance, with notable exception of decision tree model in CVIP100 leaf dataset. In average, most model achieved the performance up to 90%. In comparison with previous model, we can say that our model performance is one of the top 3 best performance in term of accuracy and F1-Score. The best performance of leaf identification for both dataset is still achieved by IMTD and combined with convolutional features.

Some few limitations in this research are the feature extraction need to be hand-crafted. Although the performance is quite well, we need to find algorithm that suitable for specific case of dataset in terms of extracting feature. The best thing that deep learning can do is learning by itself. The reason is because if the dataset is different, this algorithm may not performed that well. In brief, we can say that some algorithm may not generalize well in other datasets. Moreover, we believed conventional machine learning can still be explored and compared for specific dataset if suitable because in this case we only explore 5 different machine learning models. The machine learning model is still vary especially for the classification tasks.

References

[1] B. Kirchoff, "A New Day in Plant Identification," UNC GREENSBORO. Accessed: Apr. 20, 2025. [Online]. Available: https://kirchoff.wp.uncg.edu/plant-identification/#:~:text=Correct%20identification%20affects%20every%20aspect,species%2C%20but%20to%20cultivar%20identification

[2] C. Yang, "Plant leaf recognition by integrating shape and texture features," *Pattern Recognit*, vol. 112, 2021, doi: 10.1016/j.patcog.2020.107809.

[3] H. Wu, L. Fang, Q. Yu, J. Yuan, and C. Yang, "Plant leaf identification based on shape and convolutional features," *Expert Syst Appl*, vol. 219, 2023, doi: 10.1016/j.eswa.2023.119626.

[4] C. Yang, L. Fang, Q. Yu, and H. Wei, "A Learning Robust and Discriminative Shape Descriptor for Plant Species Identification," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 20, no. 1, 2023, doi: 10.1109/TCBB.2022.3148463.

[5] M. P. Shah, S. Singha, and S. P. Awate, "Leaf classification using marginalized shape context and shape+texture dual-path deep convolutional neural network," in *Proceedings - International Conference on Image Processing, ICIP*, 2017. doi: 10.1109/ICIP.2017.8296403.

[6] M. Lukic, E. Tuba, and M. Tuba, "Leaf recognition algorithm using support vector machine with Hu moments and local binary patterns," in *SAMI 2017 - IEEE 15th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 2017. doi: 10.1109/SAMI.2017.7880358.

[7] S. Alamoudi, X. Hong, and H. Wei, "Plant Leaf Recognition Using Texture Features and Semi-Supervised Spherical K-means Clustering," in *Proceedings of the International Joint Conference on Neural Networks*, 2020. doi: 10.1109/IJCNN48605.2020.9207386.

[8] C. Yang, H. Wei, and Q. Yu, "Multiscale triangular centroid distance for shape-based plant leaf recognition," in *Frontiers in Artificial Intelligence and Applications*, 2016. doi: 10.3233/978-1-61499-672-9-269.

[9] C. Hewitt and M. Mahmoud, "Shape-only Features for Plant Leaf Identification," Nov. 2018.

[10] C. Yang and Q. Yu, "Multiscale Fourier descriptor based on triangular features for shape retrieval," *Signal Process Image Commun*, vol. 71, 2019, doi: 10.1016/j.image.2018.11.004.

[11] X. Chen and B. Wang, "Invariant leaf image recognition with histogram of Gaussian convolution vectors," *Comput Electron Agric*, vol. 178, 2020, doi: 10.1016/j.compag.2020.105714.

[12] J. S. Cope, P. Remagnino, S. Barman, and P. Wilkin, "Plant texture classification using gabor co-occurrences," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010. doi: 10.1007/978-3-642-17274-8_65.

[13] F. Mostajer Kheirkhah and H. Asghari, "Plant leaf classification using GIST texture features," *IET Computer Vision*, vol. 13, no. 4, 2019, doi: 10.1049/iet-cvi.2018.5028.

[14] J. Chaki, R. Parekh, and S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers," *Pattern Recognit Lett*, vol. 58, 2015, doi: 10.1016/j.patrec.2015.02.010.

[15] S. Naeem *et al.*, "The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach," *Agronomy*, vol. 11, no. 2, 2021, doi: 10.3390/agronomy11020263.

[16] S. H. Lee, C. S. Chan, and P. Remagnino, "Multi-Organ Plant Classification Based on Convolutional and Recurrent Neural Networks," *IEEE Transactions on Image Processing*, vol. 27, no. 9, 2018, doi: 10.1109/TIP.2018.2836321.

[17] J. Liu, S. Yang, Y. Cheng, and Z. Song, "Plant Leaf Classification Based on Deep Learning," in *Proceedings 2018 Chinese Automation Congress, CAC 2018*, 2018. doi: 10.1109/CAC.2018.8623427.

[18] A. Beikmohammadi, K. Faez, and A. Motallebi, "SWP-LeafNET: A novel multistage approach for plant leaf identification based on deep CNN," *Expert Syst Appl*, vol. 202, 2022, doi: 10.1016/j.eswa.2022.117470.