Leaf Classification Project

ECE 228 FINAL PROJECT

Presented by Group 86

Jiacheng Hu
PID:A91013815
jih135@ucsd.edu

Yitao Liu
PID: A53316312
yil099@ucsd.edu

Jia Liu
PID:A92082252
jil528@ucsd.edu

Abstract

Leaf classification is a good topic in the field of biology. It is sometimes hard to tell the difference between leaves in different species by eyes. Nowadays, with the development of machine learning, leaf classification now can be done by computer itself. In this paper, several models are implemented with the same dataset of leaf features. After comparing and analyzing their results, the group members find the good and bad models for leaf classification.

1. Introduction

The goal of this project is to classify the species of plant by recognizing the features of the leaf using Machine Learning techniques.

In academic definition, a leaf is an organ of a vascular plant and is the principal lateral appendage of the stem, usually borne above ground and specialized for photosynthesis [1]. However, why is leaf important to be classified? First of all, the fact that there are approximately 391,000 species of vascular plants existing on earth, with about 2,000 use plant species discovered or described every year, makes it important, also convenient, to classify different species of plants [2]. The other reason why leaf is important is the fact that leaf to plant is like a "fingerprint" to humans. Leaf contains unique identity information of plants, e.g. features of shape and margin, which is critical to be utilized to identify its species.

Then why automatic classification is important? As mentioned above that there are so many plant species existing and new species discovered each year, it is difficult to classify each species of plant and this possibly causes the problem of identification duplication. Therefore, automatic classification of plant species is necessary and can further help in many ways including: tracking and preserving species population, helping plant-based medicinal research and managing plant food supply [3].

The input to our algorithm is images of leaves. We then use a few different models including Naive Bayes, SVM, Logistic Regression, KNN, Linear Discriminant Analysis and CNN to output a predicted species of plant.

2. Related work

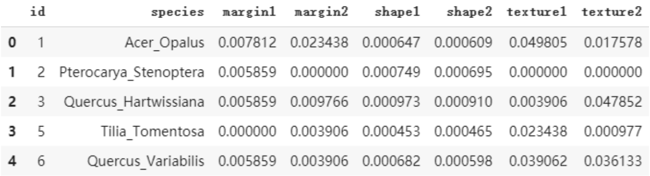
Researchers tried to solve this problem over the last few years. The first reference we used is called "Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features" [4] written by Charles Mallah. This paper introduces a new data set of sixteen samples each of one hundred plant species and describes a method designed to work in conditions of small training set size. They processed each of three features in separate ways: they used histogram accumulation for margin and texture features and normalised description of contour for shape feature. For each feature using the K-NN algorithm, they generated a separate posterior probability vector and then combined posterior estimates to give the final classification. The best result they got is 96% mean accuracy when combining all three features. The result of their approach is very impressive considering how small their data set is. That is also what motivates us to use a relatively small data set for our project [4].

Another related work we referred is called "Plant discrimination by Support Vector Machine Classifier based on spectral reflectance" [5] written by Saman Akbarzadeh. They used SVM classifier which is proposed to classify broad leaf and narrow leaf plants. The strength of this model is high speed, while still achieving 97% accuracy which is improved using raw reflected intensities and kernel tricks [5].

There are some other references we used for our project but above two are our favorite because of their clear demonstration and high accuracy. Some of the other references used different approaches, for example naive bayes [6], which are attached in the reference section.

3. Dataset and Features

The dataset is obtained from kaggle website named "leaf-classification".The dataset contains approximately 1,584 images of leaf specimens (There are 99 species with 16 samples each, 10 samples among them are labeled, and another 6 samples are used to evaluate the model), and these images have been converted to binary black leaves against white backgrounds.Three sets of features are also provided per image: a shape contiguous descriptor, an interior texture histogram, and a fine-scale margin histogram. For each feature, a 64-attribute vector is given per leaf sample. [7]



	id	species	margin1	margin2	shape1	shape2	texture1	texture2
0	1	Acer_Opalus	0.007812	0.023438	0.000647	0.000609	0.049805	0.017578
1	2	Pterocarya_Stenoptera	0.005859	0.000000	0.000749	0.000695	0.000000	0.000000
2	3	Quercus_Hartwissiana	0.005859	0.009766	0.000973	0.000910	0.003906	0.047852
3	5	Tilia_Tomentosa	0.000000	0.003906	0.000453	0.000465	0.023438	0.000977
4	6	Quercus_Variabilis	0.005859	0.003906	0.000682	0.000598	0.039062	0.036133

Figure 1. Sample data structure

Figure 1 here shows five samples as examples. Actually, there are 64 margins, 64 shapes and 64 textures for each sample. They are all features extracted from images. As we can’t show them completely here. We select two data points of each three features here(2x3 out of 64x3).

As the features of leaves have been given by the dataset, we do not have to extract them from images by ourselves, which saves us a lot of work. There are three kinds of features which are important to recognize the species of leaves. They are margin, shape and texture. Each of them has a 64-attribute vector. As the results shown in [4], using all three features combined, we will get the best result. Therefore, we will use all three given features to train and test our model.

4. Methods

4.1 Naive Bayes

Naive Bayes method is a classification method based on Bayes' theorem and independent assumption of characteristic conditions. For a given training data set, first learn the joint probability distribution of input/output based on independent assumptions of feature conditions; then based on this model, for the given input x, use Bayes' theorem to find the output y with the largest posterior probability. If you note the conditional independence assumption (a strict condition), the Naive Bayes classifier will converge faster than discriminant models, such as logistic regression, so you only need less training data. Even if the NB conditional independence assumption is not established, the NB classifier still performs very well in practice. As the dataset we used here is actually small, we try Naive Bayes method first.

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$

[8]

4.2 Support Vector Machine

SVM can be well applied to high-dimensional data, avoiding the problem of dimensional disaster. And even if the data is linearly inseparable in the original feature space, as long as a suitable kernel function is given, it will run well. As the number of samples is greater than the number of features here, we use RBF kernel in our project.

$$K(\mathbf{x},\mathbf{y}) = e^{-||\mathbf{x}-\mathbf{y}||^2/2\sigma^2}$$

[9]

This is the gaussian radial basis function we used.

4.3 Logistic Regression

Logistic regression is a discriminant model, and it is accompanied by many methods of model regularization, and you don't have to worry about whether your features are related as you are using Naive Bayes.As all sample points have contributions when optimizing parameters, we do not use kernel function in logistic regression. It has good performance in handling two classification problems. However, in our project, we need to classify 99 species of leaves. So logistic regression may not perform well here.

4.4 K-Nearest Neighbours

K-NN algorithm is a non-parametric method that can be used in pattern recognition for classification and regression. Here we use it for classification. The input is k closest training samples in the feature space and output is a class membership [10]. To use the K-NN algorithm, we first computed the Euclidean distance by the following equation, where m=64x3 in our case.

$$\|x - z\|_2 = \sqrt{\sum_{i=1}^{m}(x_i - z_i)^2}.$$

Then ordered the labeled samples by increasing distance and determined value of k by cross validation. Finally we could classify each leaf using the result of K-NN [10].

4.5 Linear Discriminant Analysis

LDA is a method to ”find a linear combination of features that characterizes or separates two or more classes of objects or events” [11] in pattern recognition and machine learning. LDA can be derived from simple bayes rule which is shown in Naive Bayes. In our multivariate case, the conditional probability is modeled as following where d is dimensions of features.

$$P(x|y=k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_k)^t\Sigma_k^{-1}(x-\mu_k)\right)$$

For all class k we assume they have the same covariance, so we can reduce the log posterior to $\log P(y=k|x) = -\frac{1}{2}(x-\mu_k)^t\Sigma^{-1}(x-\mu_k) + \log P(y=k) + Cst.$

4.6 Convolutional Neural Network

CNN is so far the most popular used model to analyze images. CNN consists of fully connected layers, convolutional layers and polling layers, where convolutional layers are essential to work. For regularization part, L2 regularization is use which would add term to loss as following [12]

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq y_i}\max(0, f(x_i;W)_j - f(x_i;W)_{y_i} + 1) + \boxed{\lambda R(W)}$$

We chose 0.3 for dropout rate to avoid overfit.

5. Results and Discussion

Classifier	Accuracy(%)	Loss
Naive Bayes	55.05	15.5
LogisticRegression	65.65	4.16
SVC	84.34	4.64
KNN	91.91	2.79
Linear Discriminant	97.47	0.51
CNN	98.99	0.04

Table 1: results for all classifiers

The group first split the original data into training and validation datasets in 8:2 proportion. Then the group trained each model with training data.

After that, the group used the train model to classify the validation data. Accuracy and loss were from classification results. Accuracy measures the true rate of the trained model working on validation data. Log loss measures how the result of the trained model varies from the validation set.

For the model implementation, the models other than CNN were based on imported python libraries. For CNN, the layers were designed by the group.In addition, the result figures were drawn by using python plotting libraries. [14][15][16][17][18][19][20][21]

From observation on this table, Naive Bayes and logistic regression models do not have good accuracy. On the other hand, KNN, linear discriminant and CNN all have over 90% accuracy in classification.

The reason for the bad performance of Naive Bayes is that the leave features implemented in it are dependent. And Naive Bayes model is not good at correlated features. For logistic regression, it works well in Dichotomy. As for the task here, 99 species should be classified. This can be a reason for the bad performance of logistic regression models.

As for the models with high accuracy, like explained and shown in the last session, they work well either for high dimensional dataset or image classification.

The group members choose each one model separately from the bad ones and the good ones to analyze. Naive Bayes and KNN are chosen as examples for comparison.

For naive bayes, figure 2 below is its confusion matrix. The points away from the diagonal are wrong predictions. As a bad model with only 55.05% accuracy, its confusion matrix contains lots of wrong results.

Figure 3 below is a sample wrong classification result of Naive Bayes model. The upper one (species 23) is the wrong result and the lower one (species 69) is what the result should be. The trained naive bayes model cannot tell the difference between such distinct leaf species. This demonstrates its low accuracy and ridiculously high log loss in leaf classification.

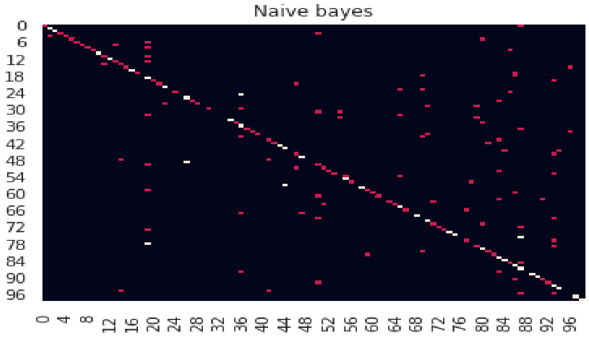


Figure 2: Confusion matrix for naive bayes model

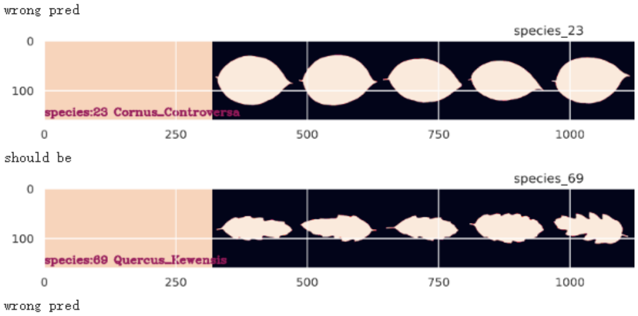


Figure 3: Sample wrong classification of naive bayes

For the KNN model, figure 4 below is its confusion matrix. As a model with high accuracy as 91.91%, its confusion matrix contains much less wrong results than Naive Bayes'.

Figure 5 shows a sample wrong classification result of KNN model. As mentioned before, the upper one (species 19) is the wrong classification result and the lower one (species 47) is the right answer. As observed, the wrong classification answer is close to the correct species. The confusion matrix and the sample wrong classification demonstrates this model's high accuracy and low loss.

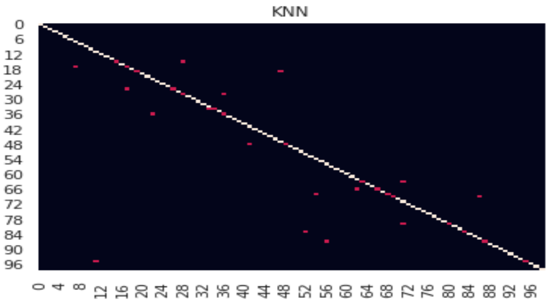


Figure 4: Confusion matrix for KNN model

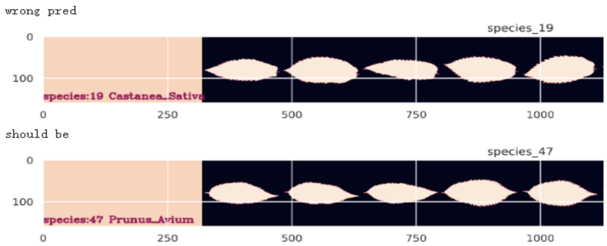


Figure 5: Sample wrong classification of KNN

The best model the team inserted was CNN. Different from other models implemented before, the layers of CNN were customized by the group members. Because the dataset the group used was the extracted image features from leaf images, as a result, the group members skipped the extraction procedure. They directly implemented dense layers for classification with input as extracted image features. And within the dense layers, they implemented dropout layers to prevent the overfitting problem to improve the validation of this CNN model. As figure 6 below, the accuracy and loss parts show the model's high performance on validation and train data. The model’s accuracy is 98.99% with 0.04 loss.

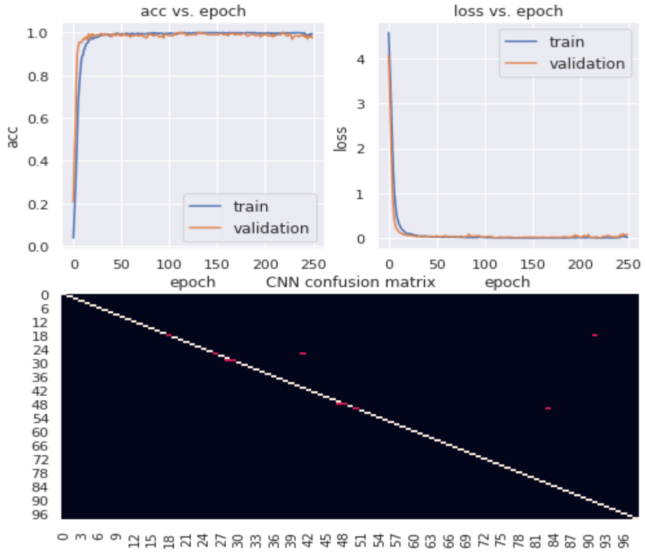


Figure 6: results of CNN model

6.1 Conclusion

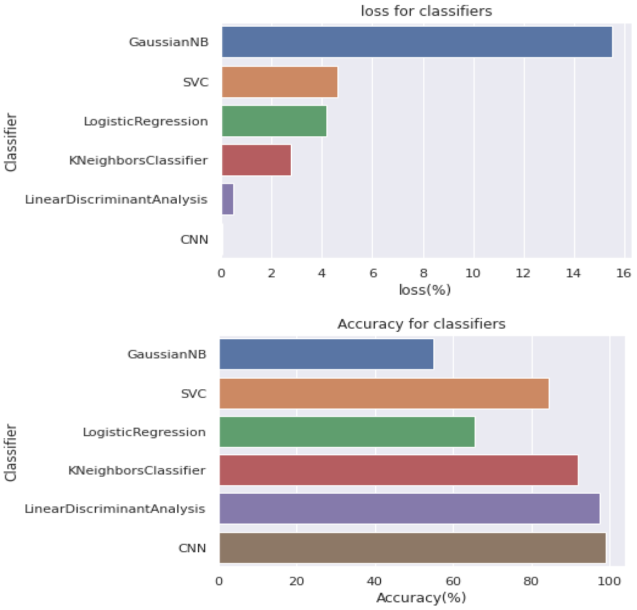


Figure 7: loss and accuracy bar chart for classifiers

As figure 7 shows, Naive Bayes and Logistic Regression have poor performance on classification of leaves. It may be because the Naive Bayes model assumes that the features are independent of each other. When the correlation between features is large, the classification effect is not good. And logistic Regression is good at handling two classification problems. It has poor performance on the Multi-classification problem here. The other four classifiers all perform quite well. Among them CNN is the best. It has the highest accuracy and the lowest loss. CNN has good performance in image recognition.

6.2 Future Work

Although we have tried so many models, we did not design different pre-processing methods for different models. If we have more time, we may try this way in the future, which can compare different models more objectively.

7. References

[1] “Leaf.” *Wikipedia*, Wikimedia Foundation, 27 May 2020, en.wikipedia.org/wiki/Leaf.

[2] “How Many Plant Species Are There in the World? Scientists Now Have an Answer.” *Mongabay Environmental News*, 12 May 2016, news.mongabay.com/2016/05/many-plants-world-scientists-may-now-answer/.

[3] “WenjinTao/Leaf-Classification --Kaggle.” *GitHub*, github.com/WenjinTao/Leaf-Classification--Kaggle/blob/master/Leaf_Classification_using_Machine_Learning.ipynb.

[4] Mallah, Charles, et al. “Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features.” Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Applications, 2013, doi:10.2316/p.2013.798-098.

[5] Akbarzadeh, Saman, et al. “Plant Discrimination by Support Vector Machine Classifier Based on Spectral Reflectance.” Computers and Electronics in Agriculture, vol. 148, 2018, pp. 250–258., doi:10.1016/j.compag.2018.03.026.

[6] Padao, Francis Rey F., and Elmer A. Maravillas. “Using Naïve Bayesian Method for Plant Leaf Classification Based on Shape and Texture Features.” 2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 2015, doi:10.1109/hnicem.2015.7393179.

[7] “Leaf Classification.” Kaggle, www.kaggle.com/c/leaf-classification/data.

[8] Gerstoft, Peter. “ECE228 Lecture 4.” 20 Apr. 2020, La Jolla.

[9]Gerstoft, Peter. “ECE228 Lecture 6.” 4 May 2020, La Jolla.

[10] “K-Nearest Neighbors Algorithm.” *Wikipedia*, Wikimedia Foundation, 28 May 2020, en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[11] “Linear Discriminant Analysis.” *Wikipedia*, Wikimedia Foundation, 3 June 2020, en.wikipedia.org/wiki/Linear_discriminant_analysis.

[12] Gerstoft, Peter. “ECE228 Lecture 3.” 13 Apr. 2020, La Jolla.

[13] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[14] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[15] Abadi, Mart&#39;in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In 12th $USENIX$ Symposium on Operating Systems Design and Implementation ($OSDI$ 16) (pp. 265–283).

[16] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[17] Oliphant, T. E. (2006). A guide to NumPy (Vol. 1). Trelgol Publishing USA.

[18] McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

[19] Waskom, M., Botvinnik, Olga, O&#39;Kane, Drew, Hobson, Paul, Lukauskas,Saulius, Gemperline, David C, … Qalieh, Adel. (2017). mwaskom/seaborn: v0.8.1 (September 2017). Zenodo. https://doi.org/10.5281/zenodo.883859

[20] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science &amp; Engineering, 9(3), 90–95.

[21] Bradski, G. (2000). The OpenCV Library. Dr. Dobb&#39;s Journal of Software Tools.

8. Contributions

- Jiacheng Hu

  He worked on data cleansing, data visualization and model implementation.

- Yitao Liu

  He worked on model research and model implementation

- Jia Liu

  He worked on background and literature research, model research and implementation

9. Replies to critical reviews

Critical review from team 5:

1. How do you come up with the structure of the CNN? Could you talk some details about why you use certain number of layer and why you adopt such structure?

The reason why we use CNN model is because our topic is about image classification. And CNN is a good option for this kind of topic.

In our design, we implemented only dense layers and dropout layers. The reason is that the dataset we use is the feature data from the leaf image. You can take a look at part 3 of the report. The dense layers are used for classification and the dropout layers are used to prevent overfitting problems.

2. Could you provide some insights about why these 6 models will have different result? In other words, why CNN is better than other models in leaf classification?

Because CNN can automatically extract features from images and then analyze. You can check part 4 for detailed explanation for all models.

3. I believe that your can talk about more details on your “error_set” function in the slides part because it is intuitive and interesting.

Thanks. It is undoubtful to say that showing and analyzing error sets are interesting and intuitive to figure out why these samples can not be classified correctly. As we showed a few error sets in the presentation, we cannot show them all and analyze in detail due to time limitation in the slide.

Critical review from team 74:

1. the details of other model is unclear, such as KNN and Linear Discriminant.

You can check part 4 of the report for model details.

2. Try more other models.

At this time, we don't expect more models that would be used since six models are enough to get a decent result.

3. I think the normal CNN model should contain convolution layer like con2D layer for the image pixels rather than shape, texture and margin of a image.

The reason why we are not using COV2D is that the dataset we have is already the feature data extracted from images. Thus, we do not need to have extra layers for feature extraction. You can check part 3 for details of our dataset.

Critical review from team 30:

1. I wonder if the number of samples in your dataset is enough for achieving such high accuracy or I misunderstanding the meaning of samples?

I think it’s enough because the number of each feature extracted from a image is 64 for each single leaf image. And I believe that the high accuracy is because of the good extracted features from images which are already given. You can check part 3 of this report for details.

2. Could you explain more clearly about the accuracy and loss in both training data and validation data? I knew your performance is excellent in the bar chart and comparison table,but I have no idea what about the training data and test data?

As I declared in the presentation, I first split the original data into training and validation datasets in 8:2 proportion. Then I train each model with training data.

After that, I use the train model to classify the validation data. Accuracy and loss are from this result. Accuracy measures the true rate of the trained model working on validation data. Log loss measures how the result of the trained model varies from the validation set.

3. As for the CNN model is your best option for this classification problem, could you explain more about how to choose these layers and construct the CNN model?

The reasons for why the CNN model are two reasons. First is that CNN model works well in feature extraction from data and classification of images. The second reason is that the dataset we have is already the feature extractions from the images. You can check details in part 3 of the report. Moreover, the given feature data is perfect. That’s why we can have such good results.

And for the CNN model design, as I mentioned before, the dataset we have is already the extracted features from data. So I skip the image feature extraction step.Then I use dense layers for classification and dropout layers to alleviate overfitting problems.