Mini-project

Imperial College London

Department of Life Sciences

Comparison of Plant Seedlings Images Classification Models

Author:
Xiaosheng Luo (CID: 01627437)

Date: March 8, 2019

1 Abstract

Image classification is a basic task in the field of computer vision, with the development of deep learning, the accuracy of computer vision recognition is constanly improving, showing the bright future in the practical applications, especially in the file of precision agriculture. Many studies are working to develop a crop detection technology that combines unmanned aerial vehicle (UAV) and spectroscopy. And a good image recognition model is the basis of this technology, but also the problem to be solved by this mini project. Here, I build several baseline models (KNN, SVM, Random Forest, XGBoost and Resnet18), to evaluate the performance of these algorithms and also improve my understanding of them.

Keywords: Image Clssification, Plant Seedlings, Machine learning, Model Comparison, ROC Curve

2 Introduction

Precision agriculture is the direction of agriculture development, aim to improve the performance of crops and the environment quality by applying advanced application techniques and principles to manage spatial and temporal changes associated with all aspects of agriculture production.[**?**] In the actual planting process, it is necessary to carry out individualized management for different crops as well as the problem of weeds for different varieties. In the effort of achieving that, remote sensing combines with effective recognition algorithms have been commonly considered as an effective technique.

Previous studies have made extensive research on the development of algorithms and models for field crop identification. Firstly, leave shape is one of the important basis for plant classification. Therefore, the identification of weeds and specific crops can be accomplished by analyzing the blade shape to derive certain edge features and shape features. For instance, by collecting visible light images of farmland, after binarization, extracted and analyzed characteristic parameters such as blade area, long axis, short axis, and centroid position, it can reach 73% recognition rate for tomatoes and 68.8% for weeds[**?**]. Similarly, the leaf parameter k-NearestNeighbor (KNN) classifier for wheat and weeds can be achieved to 82% and 79%, and the Bayesian classifier can reach to 81% and 75%[**?**]. Secondly, due to the different tissue structure of the leaves, it provides the possibility to distinguish different crops by using spectral features. By collecting and analyzing the 435-1000nm spectral data of weeds and crops, KNN classifer can achieve 97% accuracy for weeds and Multi-layer neural network can go upto 80.1% for crops[**?**].

In the last decade, with the accumulation of data and the enhancement of computing power, deep learning has ushered in a big outbreak. In 2012, in order to prove the potential of deep learning, the Hinton research group participated in the ImageNet image recognition competition for the first time, and won the championship by constructing the Convolutional neural network(CNN) AlexNet, and crushed the classification performance of the second place (Support Vector Machines, SVM). It is also because of this competition that CNN has attracted the attention of many other researchers. In the subsequent games, other well-known neural networks appeared, such as VGGnet, Inception network and ResNets.

Here, in order to better understand the performance of traditional machine learning models and neural networks in image classification, and to better apply the effective model in the precision agriculture crop detection, I build and compare the performance of KNN, SVM, ensemble method like Random Forest and XGBoost, and CNN ResNets18, train from scratch and transfer learning, based on the image dataset download from the kaggle playground, the plant seedlings classification dataset.

3 Materials & Methods

3.1 Dataset

The dataset used to evaluate the performance of the image classification model comes from Kaggle competition playground "Plant Seedlings Classification". The dataset comprises 12 plant species, each image has a filename that is its unique id. Here is the composition of the dataset:

Black-grass	263 images
Charlock	390 images
Cleavers	287 images
Common Chickweed	611 images
Common wheat	221 images
Fat Hen	475 images
Loose Silky-bent	654 images
Maize	221 images
Scentless Mayweed	516 images
Shepherds Purse	231 images
Small-flowered Cranesbill	496 images
Sugar beet	385 images
Total	4750 images

Table 1: Dataset structure

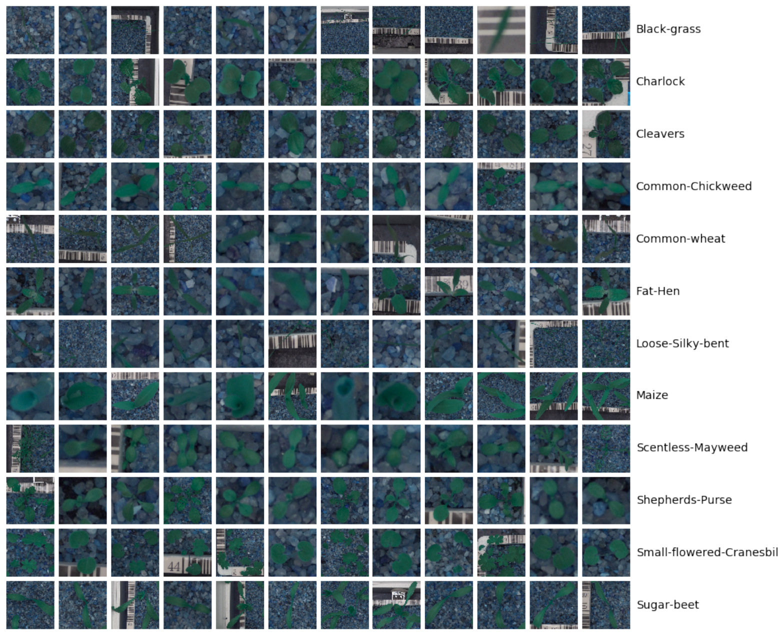

Figure 1: Plant Seedlings Dataset Samples

3.2 Data manipulation

Prior to the use of the image data for model fitting, manipulation was required to extract the features that was relevant to my analysis. Most of the data feature processing is based on python module openCV[**?**] and sklearn[**?**].

1) Due to the different image sizes, we resized to a fixed size of 128 X 128.

2) Convert images to different color spaces, RGB(Red Green Blue), HSV(Hue Saturation Value), a color model based on the physiological characteristics of a person's observation of color (the human visual system is more sensitive to brightness than the color value), and HLS(Hue Saturation Lightness).

3) Extract the histogram of the image to reduce the feature dimension

4) Based on the image features, we found that the pixel distribution is bimodal, that is, there is a big variance between the plant and the background. Through the threshold segmentation method, the image histogram is extracted after the mask.

5) During the fitting process, we separate 20% of the dataset as test set for evaluation.

3.3 Model evaluation metric

Normally, for binary classification problem, the predition results will be as follows:

Label \ Predition	+1	-1
+1	True Positive(TP)	False Negetive(FN)
-1	True Positive(FP)	True Negetive(TN)

Table 2: Analysis of the results of the binary classification problem

The indicator for classification model evaluation usually can be accuracy, precision, recall, f1-score, ROC curve, AUC, etc. Their calculation formula is as follows:

1) Accuracy = (TP+TN)/All Samples, accuracy is the most common and basic evaluation metric. However, in the case of unbalanced positive and negative cases, especially when we are more interested in the minority class, it will have the “accuracy paradox”.

2) Precision = TP/(TP+FP)

3) Recall = TP/(TP+FN)

4) F1-score = 2 * Precision * Recall/(Precision + Recall), F1-score is a metric that takes into account precision and recall. In the case of multi-class classification, macro-average is better than micro-average, because macro-average treats each class equally, so its value is mainly affected by rare class.

5) True Positive Rate(TPR) = TP/(TP+FN) = TP/actual positives, False Postive Rate(FPR)=FP/(FP+TN) = FP/actual negatives, the ROC curve is composed of points (TPR, FPR) that set different thresholds, and AUC is the area of the ROC. The bigger the AUC, the better.

In this project, we comprehensively use these two types of classification model metrics to examine the image classification performance of different models.

3.4 Model fitting

KNN: Calculate the distance of each feature corresponding to the new data and the data in the sample set, extract the classification labels of the k most similar data, and select the classification with the most occurrences among the most similar data as the classification of the new data. One of the advantages of KNN is that the model is easy to understand and usually does not require too much adjustment to get good performance. Trying this algorithm is a good benchmark before considering the use of more advanced techniques. It is usually very fast to build a KNN model, but if the training set is large (the number of features is large or the number of samples is large), the prediction speed may be slow. When using the KNN, it is important to preprocess the data. Although the KNN algorithm is easy to understand, it is often not used in practice because it is slow to predict and cannot process data sets with many features.

SVM: The mathematical principle behind the support vector machine is a bit complicated, and the main idea is to find the hyperplane with the largest classification interval.The SVM is a very powerful model that performs well on low-dimensional data and high-dimensional data, while the disadvantage of SVM is that it takes great care to preprocess data and tuning. This is why many applications today use tree-based models, such as random forests or gradient boosts, which require little pre-processing.

Ensemble methods like random forests and xgboost are based on dicision tree model, they use the methods of bagging and boosting respectively. Their advantage is that they can get a good training result with almost no data pre-processing and reduce over-fitting through multiple weak learning models.

4 Results

4.1 Data manipulation

As we can see from the histogram below, it has obvious bottom-breaking crests, which we can explain from the images, the background of the images is mostly stone and completely different from green plants, which makes it possible to mask out the plants according to different image digital number threshold (Figure3).

Finally, we select the digital number data of the three color spaces, the data after calculating the histogram, and the data after masking and histogram dimensionality reduction. According to the statistics of 4750 images, we can see

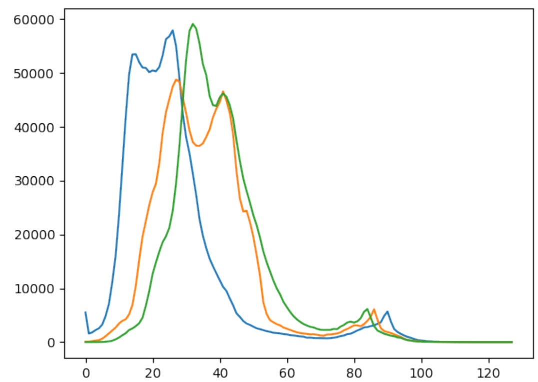

Figure 2: RGB Channels Histogram Distribution

that raw pixel images will occupies 228.00MB memory, while after calculating the histogram, it takes only 14.25MB, which is 16 times smaller than raw pixels data.

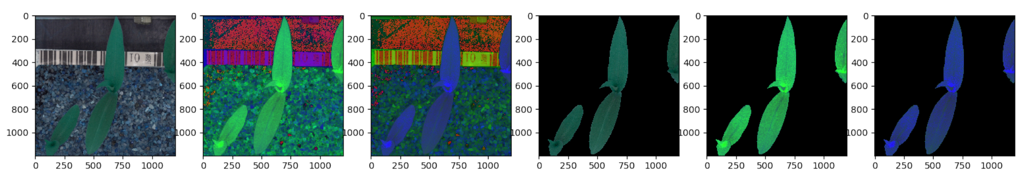

Figure 3: Image after color space conversion and masking
Images from left to right are BGR, HSV HLS color spaces.

4.2 KNN model comparison

KNN is the one of the simplest and most efficient algorithm for classification model. Generally, KNN has two important hyperparameter, the number of nearest neighbors and the metrics for distances between data points, here I search for the hyperparameter K based on HSV color space after masking and histogram calculation for 1 to 20 (Figure4).

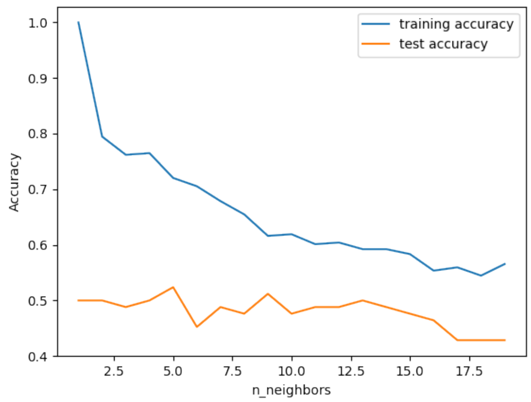

Figure 4: RGB Channels HistogramDistribution

Result shows that when considering only one single neighbor, the predition on the training set are perfect, and as the number of neighbors increase, the model becomes simpler and the accuracy decreases. The test set accuracy for single neighbor is lower than when using more neighbors, which means that the model of a single neighbor is too complex, lead to overfitting, so the best number should be when the two urves are relatively close, and her I chose K=9 to further explore different data processing methods.

Model / ACC/Time	RGB	HSV	HSL	histoRGB	histoHSV	histoHSL	maskRGB	maskHSV	maskHSL
Training set(%)	41.68	49.61	51.53	70.58	76.79	77.11	74.55	78.39	78.32
Test set(%)	26.42	38.53	38.53	58.84	67.16	70.84	68.21	72.42	71.37
Timeconsuming(s)	19’47	19’57	19’55	10	6	6	2	2	2

Table 3: KNN model comparison

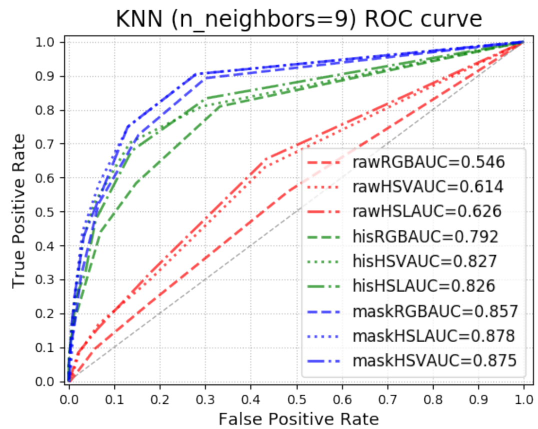

Figure 5: KNN models ROC curve

So by combining the three color spaces with the three processing methods, we have fitted the KNN algorithm to the 9 different image processing data. Using ROC curve and the area under the curve (AUC) as evaluation metric, we can find that:

1) KNN model fitting performance on histogram data is generally better than raw image pixel data, and spend expected less time.

2) Model performance better on HSV, HLS color space than RGB, but silghtly different between them.

3) Whether masking or not does not have much effect on the results.

4.3 KNN, SVM, Random Forest and XGBoot model comparison

Based on the fitting results of KNN in different image processing methods, I choose the HSV color space mask data as object, compared the fittig results of these several models. Although Gridsearch was used to find the best hyperparameter for the Random Forest, it seems to play a minor role in this task, so for the SVM and the ensemble models, I mainly use sklearn's default parameters in order to build a basline model as soon as possible. And the result shows as below:

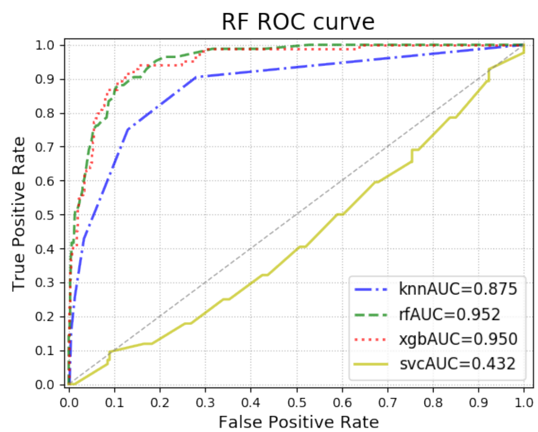

Figure 6: KNN, SVM, Random Forest and XGBoot ROC Curve

Model Metrics	KNN	SVM	Random Forest	XGBoost
Accuracy	72.42	13.16	85.58	87.58
AUC	0.953	0.606	0.985	0.988
Micro F1 score	0.72	0.13	0.86	0.88
Macro F1 score	0.70	0.02	0.85	0.86
Time consuming(s)	2	1'5	2	2'36

Table 4: Four models comparison

As you can see from the results, although the SVM can achieve the same accuracy as the ensemble methods (99%), its generalization ability is much worse, only 13.16% accuracy. As for the other three models, we can say that XGBoost performs the best, followed by the Random Forest, finally the KNN.

4.4 ResNets18 and transfer learning

Based on Pytorch's deep learning framework, the parameters are fine-turn from scratch, or freeze all layers, and add a fully connected layer at the end for parameter optimization. And it turns out that the learning from scratch has the 95.5% accuracy for validation set and 88.9% for training set, while when ConvNet as fixed feature extractor, it only has 57.5% accuracy for training set and 71.1% for validation set.

5 Discussion

Overall, the above results demonstrate that when considering all five proposed models, though they were fitted in a limited time, there is much room for improvement, we can still learn some trade-off during model selection.

Since we are dealing with unstructured data, it is hard to judge and extract the important features. In the process of applying traditional machine learning algorithms, it is very necessary to rely on the prior knowledge of the data for feature engineering. In this example, we can see this very obviously. After histogram calculation, not only can we reduce the data by 16 times but we can also improve the accuracy of 20-30% on the KNN model. However, the performance of the model seems to be lower than I expected after masking. A reasonable explanation is that after the dimensionality reduction by the histogram calculation the noise of the background is not significantly different among different species, which is not enough to be a feature to imporve the performance of the model. And this raises another interesting issue, the machine learning algorithm usually does not care about the spatial relationship of pixels in the process of fitting image data, while convolutional neural networks can obtain some spatial information from the images through the way of weight sharing.

We can also see that most of the fitting results have been overfitting due to some consensus reasons, like too much noise, insuffcent training data, overly complex model etc. The image in the dataset is not uniform, the size is inconsistent and have large variance, sometimes there are more than one plant in a picture. So, it can be better if there is a better way to normalized the data.Besides, ensemble method like Random Forest and XGBoost can theoretically reduce the complexity of the model to avoid overfitting, but it goes up to 99% accuracy for the training set, probably need more effort on hyperparameter turning.

References