*Machine Learning Based Peanut Maturity Classification from Hyperspectral Image*

Sree Nirmillo Biswash Tushar

Department of Electrical and Computer Engineering, Texas State University,

San Marcos, TX-78666, United States of America

***Abstract-* The maturity of the seed is key information to ensure quality of the crops and better economic returns. But the assessment of the maturity of the peanut requires exocarp removal which suffers from observer’s color assessment skill and experience. Moreover, it requires a great amount of time for a large number of peanut and often ends up in a blasting peanut pod. In order to find an optimal solution, researchers tried to apply digital image processing method. Although this method does not cause pod blasting and does not require personal inspection, the method demands exocarp removal. Recently, a research group has shown why traditional RGB image fails to classify peanut maturity and they have introduced a hyper-spectral unmixing based classifier to solve the problem. In this project, I have worked the same problem with logistic regression, random forest and support vector machine to compare their performance with the previous classifier.**

Keywords ***– Machine Learning, Hyperspectral image, Peanut Maturity***

1. INTRODUCTION

Peanut ([*Arachis*](https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/arachis) hypogaea)  is consumed all over the work due to its nutrition value and deliciousness. To be specific, peanut is source of protein, fat, carbohydrates, vitamin, and minerals and it prevents cardiovascular disease, cancer, diabetics, and obesity [1]. Therefore, peanut is a vastly cultivated crop in the United States. In 2021, the total production of peanut was around 6.63 billion pounds from 1.63 million acres of land in US [2]. To maximize the peanut production, maturity level of the peanut pod is one of the most important things to know. If the pod is immature while harvesting, it will not provide optimum seed quality, grade, and flavor. On the contrary, over-mature peanut pod can break the pegs where pods are attached [AZ,3].

Visual inspection has been a popular method among agricultural community for maturity classification for a long time. The most recognized method for peanut maturity level classification is MPB (Maturity Profile Board) which maps the color of the mesocarp to the five different maturity categories and the shades within the color into different subcategories [2]. But this method needs exocarp removal and it often causes pod blasting. Since, immature pods are

fragile compared to mature pods, most immature pod blows apart. Focusing on not destroying too many immature pods consumes a considerable amount of time. Moreover, color categorization using MPB requires human visual inspection, which is very subjective to human visual condition, lighting condition, observer skill in discriminating color that leads to the possibility of large error. This is very time consuming when one has to classify large number of pods. Authors in [6] also introduced a mesocarp color and pod size based nearest neighbor classifier for pod maturity classification which requires pod blasting but free from visual inspection. Thereby, recent advancement of vision technology has inspired researchers to automate the peanut maturity assessment which can speed up the process with optimum accuracy.

The idea of non-destructive peanut maturity classification is initiated by authors of [5], who have shown a positive correlation between tannin and maturity level. Interestingly, hyperspectral reflectance also varies with different chemical composition. Therefore, HSI image can be used as a peanut pod tannin distinguisher and hence their maturity level. Using all the information, authors in [4] established a linear unmixing model and fully constrained least square algorithm to classify mature and immature pods. They have made the classification based on the multispectral reflectance of the pericarp which does not require exocarp removal. They have also shown that visible part of the spectrum is indistinguishable for mature and immature pod.

But linear hyperspectral unmixing is a computationally costly process where spectrum of every pixel needs to participate. However, when all pixels of mature and immature peanut are averaged, the resulting average spectrum shows clear distinction between mature and immature peanut [AZ]. Moreover, average spectral feature based peanut maturity classification with a few key wavelength features can leverage a lot of computational time. Earlier, standard machine learning algorithm (SVM and random forest) has shown promising result for pod maturity classification of snap bean based on spectral and biophysical features [Am].

Therefore, the scope of this paper is to explore traditional machine learning (logistic regression, support vector machine and random forest) classifier to classify peanut pod maturity by selecting few key features and demonstrate the tradeoff between the number features and performance of the model. In this work, multiple peanuts have been identified and classified simultaneously which speed up the process for identifying optimum matured peanut pod. It is shown in this paper that it is feasible to automate the optimum matured peanut identification task by intelligent feature selection and classification.

1. METHODOLOGY
2. Data Preprocessing

All the data preprocessing steps are shown in figure-1 as a part of the methodology. Spectral Python library (SPy) [7] has been used to read hyperspectral image data. Since each image file contains 15 images total, firstly, the peanut pixels have been segmented from the background. After segmentation, calibration panel and non-uniform background have been removed by cropping the segmentation mask beyond calibration panels’ location. Then, all the pixels in the image have been identified and labeled according to which peanut they belong to. Lastly, peanut spectrum has been calculated by averaging the spectrum of all pixels of a peanut and used for maturity classification.

Diagram

Description automatically generated

Fig. 1 Training and Testing Procedures

1. *Image segmentation*

First of all, reflectance values of Red, Green and Blue wavelength from HSI image were merged together to form an RGB image for segmentation purpose (figure 2-a). For red, green and blue channel, 450 nm, 550 nm and 650 nm wavelength were considered respectively. Ostu thresholding has shown good peanut segmentation performance for all of the images. An example of ostu thresholded image is shown in figure-2 (b) where peanut pixels are labeled as white and backgrounds are labeled as black. In the segmentation mask, some peanut’s region has been labeled as background. Implementation of the morphological dilation operation with kernel size (12,4) has shown improvement in filtering those false background pixels in peanut region but it enlarges the peanut mask at the edges (figure-2c). Segmentation mask found after all of those mentioned operation has been used for the peanut indentification and feature matrix creation.

A picture containing arranged

Description automatically generated

(a)

A picture containing text

Description automatically generated A picture containing metalware

Description automatically generated

(b) (c)

Fig. 2 Peanut RGB image(a), Ostu Thresholded Image (b) and (b) Morphologically filtered image of TUFRunner 511 replication 2

1. *Indentify each peanut from the image:*

The identification of each peanut in the segmentation mask is a tricky task. The pixel position (row and column of that pixel) of a peanut is far away from other peanuts. Therefore, peanut position is given as an input to the k-means clustering was algorithm to group peanut pixels based on the distance between the pixel position. Figure-3 is showing the masks of all identified peanut by k-means clustering. The convention of reading peanut number is given in the caption. From figure-3, it is clear that kmeans algorithm does not label all the peanut squentially. In figure-3, first peanut in the 3rd row has been labeled as 3. This issue has been handled by re-ordering the label of all the peanut pixel considering each peanuts position. Here, peanuts at the top row has been numbered as 1-5 (left to right), second row 6-10, 3rd row 11-15. The process is semi-automated since it requires manually inputing the number of peanuts but still it can help to speed up the mature peanut sorting process which has not been introduced by [AZ].

Bubble chart

Description automatically generated

Fig.*3*: *Each* TUFRunner 511 replication 2 *identified; Reading convention: Identified 1-5 are shown as red symbol in the center, 6-10 as green, 11-15 blue; in all color group (such as in red 1-5) are divided as square, X, circle, upward triangle, downward triangle sequentially. For an example, first one in the bottom row is red circle, so it is in 1-5 range since red and 3 since circle.*

1. *Create Feature Matrix for classification*

Section 2 only returns position of each peanut in the image. For spectral feature creation, average reflectange at all the pixel positions of a peanut at certain wavelength is calculated first and the average reflectance for all the wavelengths gives the spectrum of that peanut. Then, every spectrum is normalized by the sum of all reflectance value which ensures equal total reflectance. Figure-4a shows the spectrum for 2016 and 2017 data for mature and immature peanut. Figure 4-b shows the spectrum for mature and mature peanut for Georgia-OG replication 3 which resembles the spectrum found by [AZ].

*B. Create a Dataframe:*

Then, a dataframe has been created where Features are the spectrum of each peanut witheach peanut’s label. The final dataframe is of shape (number peanuts, number of wavelength).

*C. Classifier selection , Classification and Analysis:*

Lastly, a suitable classifier is chosen for maturity classification and hyperparameters are tuned to find out best model. Detailed analysis of classifier selection and performance of the classifier will be discussed in the Section IV.

*D. Evaluation Metrics:*

The performance of all the models is compared with [AZ] using the same evaluation metric used in [AZ]. The evaluation metrics were – confusion matrix, accuracy, precision, recall, specificity, and balanced accuracy.

1. DATASET

The dataset used by [AZ] were collected from the field experiment of the North Florida Research and Education Centre. In 2016 and 2017, total five and seven cultivators, were used respectively for experiment. The name of the cultivators used in 2016 are TUFRunner 511, FloRun 157, Georgia-06G and TUFRunner 257, FloRun 331. In 2017, two more cultivars named UF 08036 and FloRun 107 were added to the exiting 5 cultivars. Each cultivar had 3 replications and every replication had 15 samples. Therefore, 225 (5×3×15) and 315 (7×3×15) peanut samples were collected from 2016 and 2017 respectively. The details about sample collection will be found in [AZ].

All hyperspectral images were taken using pushbroom line-scan hyperspectral imaging system. Then, all images were denoised using 4th order Savitzky eGolay filter. The size of the image was 1376×1000 for all 467 spectral channels in 400nm-1000nm range with 1.4 nm spectral resolution. The provided dataset has two files for each 15 peanuts - one header file with an extension hdr and one data file with an extension hyp. Reflectance values were quantized by 32 bit and therefore, the image contains very high-resolution reflectance value.

Chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

1. RESULTS

The result section two subsection – 1) Results when 2016 dataset was used for training and the model was tested on 2017 dataset and 2) Results when 2017 dataset was used for training and vice versa. For Each result section two different feature selection technique has been used and the shrieked feature matrix has been used to train 3 different machine learning model.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

1. Training on 2016 dataset and testing on 2017 dataset

Feature selection has been a key factor for hyperspectral image classification. Proper feature selection can reduce the computational complexity by a large margin. For hyperspectral images, some wavelengths show strong contrast between the classes and sometimes, only a few wavelengths hold enough information for good classification. In this work, two methods have been adopted for feature selection - taking N features having largest difference between mature and immature class and ANVOA F-test.

1. N strongest features:

In this approach, only N features having greatest difference between mature peanut reflectance and immature peanut reflectance were taken. An interesting observation was that taking features from both sides of the spectrum where mature peanut and immature peanut intersect improves performance of the machine learning models. It is intuitive since feature at the left side of the intersection hold large reflectance for immature peanuts whereas right-side feature holds large reflectance for mature peanuts. Therefore, taking feature from both sides help to balance weight and thus may have contributed to improving models’ performance. Three machine learning models (logistic regression, support vector machine and random forest) has been implemented to classify maturity of the peanut after proper feature selection and feature extraction by principal component analysis. Like as [AZ], only features after 650 nm have been considered for feature selection. In [AZ], authors have mentioned that selection few key features may improve model’s performance. Therefore, number of features have been selected through N strongest feature selection first, then orthogonal features have been extracted through PCA. It was observed that all the features are orthogonal and therefore, taking only 5-10 PCA components provide good results. The number of features has been varied from 6 to 60. The reason for choosing this range is based on experiment.

Figure 1 shows the accuracy and balanced accuracy with respect to the number of features. It is evident from figure 1 that adding more features does not necessarily improve model’s performance. The reason is that adding some features creates confusion to the model. For logistic regression model, increasing the number of features from 6 to 48 boosts the training accuracy by around 2% and testing accuracy by around 3% (figure 1-a). There are a lot of fluctuations in the testing results with more than 5% variation when increasing the number of features from 12 to 15 which shows that the model is sensitive with selecting wavelengths for feature selection. But adding more features after 15, slowly improve the performance of the model. The best training accuracy has been found for just 5 PCA component from 24 wavelength features.

All logistic regression models’ class weights are balanced by multiplying the loss function by the proportion of mature and immature class present in the dataset. The hyperparameter of the best logistic regression model is given in the appendix and best model in the stushar7.github. Hyperparameter tuning is done by a creating a hyperparameter grid in scikit learn and selected the best model by 10-fold cross validation. The result of the best logistic regression model has been compared with the result found in [AZ] in table 1 (in the logistic regression column outside the bracket). Our logistic regression model performs better in testing precision, testing specificity and training recall (marked as green in Table-1a). Also, the models give competitive performance for all the rest of the metrics with a handful number of features. Even in classifying peanut of different color, the model’s performance is very similar to the original work (Table-1c). The only major drawback is that model performs very bad in predicting Georgia-OG cultivar maturity. While trying to improve the prediction accuracy of the Georgia-OG cultivar, all the metric (except recall) decreases by a

Where C is the inverse regularization parameter,

Support vector machine underperforms than Logistic regression. Increasing number of features does not improve the performance rather reduces the performance in model for feature size 30 to 48. The model SVM model was for 48 features:

SVC(C=30,gamma=0.001,kernel='sigmoid', probability=True, random\_state=1)

Random Forest performs better than all models. But the main demerit of random forest is that it requires many trees for obtaining better result. Since, our aim was to reduce computational cost, we tried to reduce the number of tress and then tried to get optimum results. After obtaining the graph, the result was optimized for the best results. The best Random Forest model was:

RandomForestClassifier(max\_depth=6, min\_samples\_leaf=2,min\_samples\_split=5, n\_estimators=10, random\_state=1))

Lastly, the best logistic regreesion, SVM and Random Forest model was compared with the [AZ]. It is observed that is some cases our model performs better than [AZ] with a handful number of features. Lastly, the confusion matrix is shown for all 3 models and compared with [AZ].

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Accuracy | 0.924 | 0.952 | 0.884 | 0.946 | 0.871 | 0.902 | 0.933 | 0.933 |
| Precision | 0.939 | 0.938 | 0.887 | 0.983 | 0.855 | 0.879 | 0.915 | 0.946 |
| Recall | 0.893 | 0.945 | 0.901 | 0.925 | 0.918 | 0.968 | 0.967 | 0.941 |
| Specificity | 0.951 | 0.957 | 0.864 | 0.977 | 0.816 | 0.805 | 0.893 | 0.922 |
| Balanced Accuracy | 0.922 | 0.951 | 0.882 | 0.951 | 0.868 | 0.886 | 0.930 | 0.931 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Species | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| TUFRunner 511 | 91.1% | 93.3% | 84.4% | 93.3% | 77.8% | 93.3% | 86.7% | 97.8% |
| FloRun 157 | 91.1% | 97.8% | 91.1% | 95.6% | 91.1% | 91.1% | 93.3% | 97.8% |
| Georgia-06G | 95.6% | 95.6% | 97.8% | 97.8% | 97.8% | 93.3% | 97.8% | 95.6% |
| TUFRunner 297 | 88.9% | 95.6% | 75.6% | 100% | 82.2% | 88.9% | 93.3% | 95.6% |
| FloRun 331 | 97.8% | 93.3% | 93.3% | 95.6% | 86.7% | 86.7% | 95.6% | 93.3% |
| UF 08036 |  | 97.6% |  | 84.4% |  | 82.2% |  | 84.4% |
| FloRun 107 |  | 93.3% |  | 95.6% |  | 95.6% |  | 88.9% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Color | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Black | 98.2% | 100% | 96.4% | 100% | 98.2% | 100% | 100% | 100% |
| Brown | 92.4% | 92.3% | 84.8% | 86.5% | 86.4% | 94.2% | 93.9% | 89.4% |
| Yellow | 71.9% | 88.6% | 71.9% | 93.2% | 59.4% | 63.6% | 78.1% | 90.9% |
| Orange | 97.2% | 97.6% | 93% | 100% | 91.5% | 89.3% | 94.4% | 92.9% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Accuracy | 0.924 | 0.952 | 0.893 | 0.965 | 0.893 | 0.921 | 0.942 | 0.924 |
| Precision | 0.939 | 0.938 | 0.866 | 0.978 | 0.866 | 0.886 | 0.958 | 0.941 |
| Recall | 0.893 | 0.945 | 0.951 | 0.963 | 0.951 | 0.995 | 0.934 | 0.930 |
| Specificity | 0.951 | 0.957 | 0.825 | 0.969 | 0.825 | 0.813 | 0.951 | 0.914 |
| Balanced Accuracy | 0.922 | 0.951 | 0.888 | 0.966 | 0.888 | 0.904 | 0.943 | 0.922 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Species | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| TUFRunner 511 | 91.1% | 93.3% | 77.8% | 97.8% | 80% | 91.1% | 86.7% | 95.6% |
| FloRun 157 | 91.1% | 97.8% | 91.1% | 97.8% | 91.1% | 93.3% | 97.8% | 97.8% |
| Georgia-06G | 95.6% | 95.6% | 97.8% | 100% | 97.8% | 95.6% | 100% | 95.6% |
| TUFRunner 297 | 88.9% | 95.6% | 88.9% | 100% | 88.9% | 91.1% | 91.1% | 91.1% |
| FloRun 331 | 97.8% | 93.3% | 91.1% | 97.8% | 88.9% | 93.3% | 95.6% | 88.9% |
| UF 08036 |  | 97.6% |  | 86.7% |  | 88.9% |  | 86.7% |
| FloRun 107 |  | 93.3% |  | 95.6% |  | 91.1% |  | 91.1% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Color | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Black | 98.2% | 100% | 98.2% | 100% | 98.2%(98.2%) | 98.8%(89.2%) | 98.2% | 100% |
| Brown | 92.4% | 92.3% | 92.4% | 93.3% | 92.4%(98.5%) | 93.3%(81.7%) | 89.4% | 87.5% |
| Yellow | 71.9% | 88.6% | 56.3% | 93.2% | 59.4%(90.6%) | 77.3%(75%) | 84.4% | 81.8% |
| Orange | 97.2% | 97.6% | 94.4% | 98.8% | 94.4%(98.6%) | 94%(91.7%) | 100% | 96.4% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Accuracy | 0.952 | 0.871 | 0.962 | 0.88 | 0.990 | 0.849 | 0.978 | 0.871 |
| Precision | 0.938 | 0.836 | 0.983 | 0.874 | 0.989 | 0.810 | 0.974 | 0.855 |
| Recall | 0.945 | 0.893 | 0.952 | 0.910 | 0.995 | 0.943 | 0.989 | 0.918 |
| Specificity | 0.957 | 0.853 | 0.977 | 0.845 | 0.984 | 0.738 | 0.961 | 0.816 |
| Balanced Accuracy | 0.951 | 0.873 | 0.964 | 0.877 | 0.990 | 0.840 | 0.975 | 0.867 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Species | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| TUFRunner 511 | 93.3% | 86.7% | 97.8% | 77.8% | 95.6% | 77.8% | 100% | 77.8% |
| FloRun 157 | 97.8% | 88.9% | 100% | 93.3% | 100% | 86.7% | 95.6% | 91.1% |
| Georgia-06G | 95.6% | 86.7% | 95.6% | 95.6% | 100% | 93.3% | 100% | 93.3% |
| TUFRunner 297 | 95.6% | 84.4% | 100% | 84.4% | 100% | 80% | 100% | 80% |
| FloRun 331 | 93.3% | 88.9% | 95.6% | 88.9% | 100% | 86.7% | 97.8% | 93.3% |
| UF 08036 | 97.8% |  | 86.7% |  | 97.8% |  | 93.3% |  |
| FloRun 107 | 93.3% |  | 97.8% |  | 100% |  | 97.8% |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Color | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Black | 100% | 89.3% | 100% | 98.2% | 100% | 98.2% | 100% | 96.4% |
| Brown | 92.3% | 81.8% | 91.3% | 84.8% | 99% | 90.9% | 98.1% | 87.9 |
| Yellow | 88.6% | 71.8% | 93.2% | 65.6% | 95.5% | 46.9% | 93.2% | 59.4% |
| Orange | 97.6% | 97.2% | 100% | 93% | 100% | 85.9% | 97.6% | 91.5% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Accuracy | 0.952 | 0.871 | 0.971 | 0.884 | 0.930 | 0.876 | 0.965 | 0.853 |
| Precision | 0.938 | 0.836 | 0.973 | 0.863 | 0.899 | 0.862 | 0.954 | 0.894 |
| Recall | 0.945 | 0.893 | 0.979 | 0.934 | 0.995 | 0.918 | 0.989 | 0.828 |
| Specificity | 0.957 | 0.853 | 0.961 | 0.825 | 0.836 | 0.825 | 0.930 | 0.883 |
| Balanced Accuracy | 0.951 | 0.873 | 0.970 | 0.880 | 0.915 | 0.872 | 0.959 | 0.856 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Species | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| TUFRunner 511 | 93.3% | 86.7% | 97.8% | 77.8% | 91.1% | 77.8% | 93.3% | 77.8% |
| FloRun 157 | 97.8% | 88.9% | 97.8% | 91.1% | 95.6% | 91.1% | 97.8% | 91.1% |
| Georgia-06G | 95.6% | 86.7% | 100% | 97.8% | 97.8% | 97.8% | 100% | 97.8% |
| TUFRunner 297 | 95.6% | 84.4% | 100% | 86.7% | 93.3% | 82.2% | 100% | 64.4% |
| FloRun 331 | 93.3% | 88.9% | 95.6% | 88.9% | 93.3% | 88.9% | 95.6% | 95.6% |
| UF 08036 | 97.8% |  | 91.1% |  | 88.9% |  | 91.1% |  |
| FloRun 107 | 93.3% |  | 97.8% |  | 91.1% |  | 97.8% |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Color | Hyperspectral Unmixing [AZ] |  | Logistic Regression |  | Support Vector Machine |  | Random Forest |  |
|  | Train | Test | Train | Test | Train | Test | Train | Test |
| Black | 98.2% | 100% | 100% | 98.2% | 100% | 98.2% | 100% | 98.2% |
| Brown | 92.4% | 91.3% | 96.2% | 89.4% | 99% | 86.4% | 98.1% | 69.7% |
| Yellow | 71.9% | 88.6% | 90.9%( | 56.3% | 70.5% | 56.3% | 86.4% | 71.9% |
| Orange | 97.2% | 97.6% | 98.8% | 94.4% | 90.5% | 94.4% | 96.4% | 95.8% |