Superball Image Analysis

Spectral MD, Inc.

April 5, 2016

Purpose

Identify the Red Superballs in a multispectral image of superballs of multiple colors. Unlike a standard digital image, with three colors (RGB), multispectral images contain 4+ colors taken at various wavelengths in the ultraviolet, visible, and/or infrared spectrum. Multispectral image analysis involves using the various color information captured at each pixel to identify important targets in the image.

In this exercise, you will be given a reference library that contains approx. 2000 pixels representative of the Red Superballs in this image (Figure 1), and approx. 2000 pixels that represent all the rest of the pixels in the same image. The Red Superball pixels are labeled “Red”, and the other pixels are labeled “Background”.

The imager (camera) used to capture the data from these Superballs is a CCD camera (Apogee Alta U260) with Resolution of 512 x 512 pixels. It collects 31 images that represent wavelengths between the range of 400nm - 700nm (figure 2). The image format is PNG (16bit).

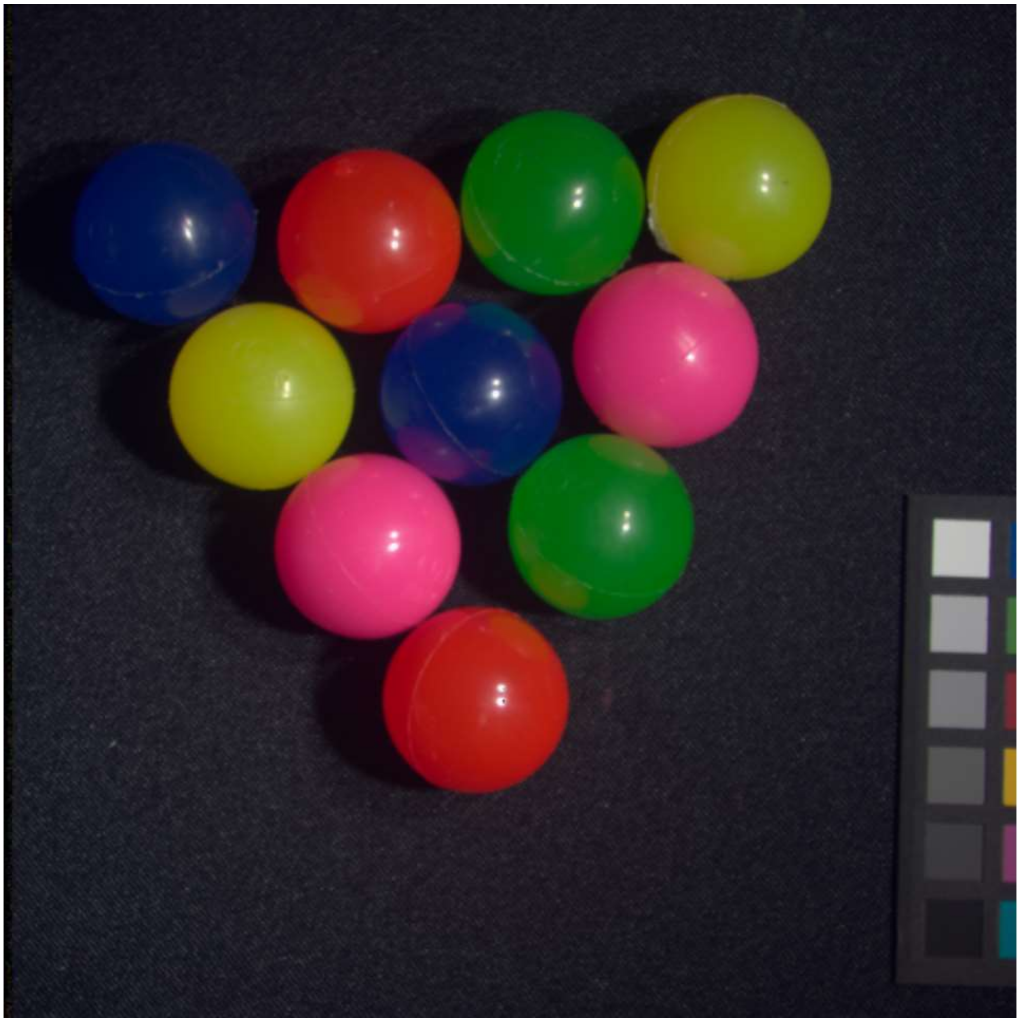

Figure 1. A color representation of the multispectral image target showing the location of the various colored superballs

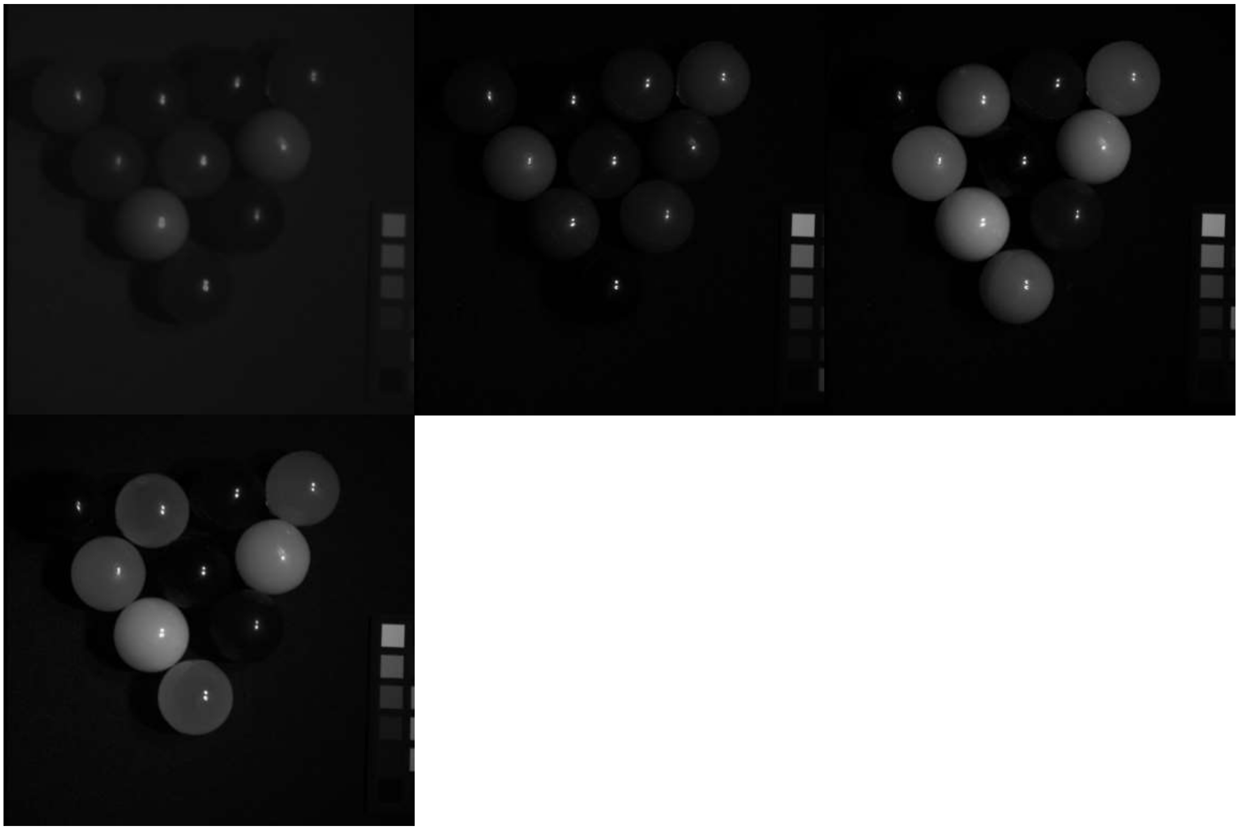

figure 2. Four of the 31 images captured by the multispectral camera. We can observe that at these four various wavelengths, the different colored Superballs have different reflectance spectra.

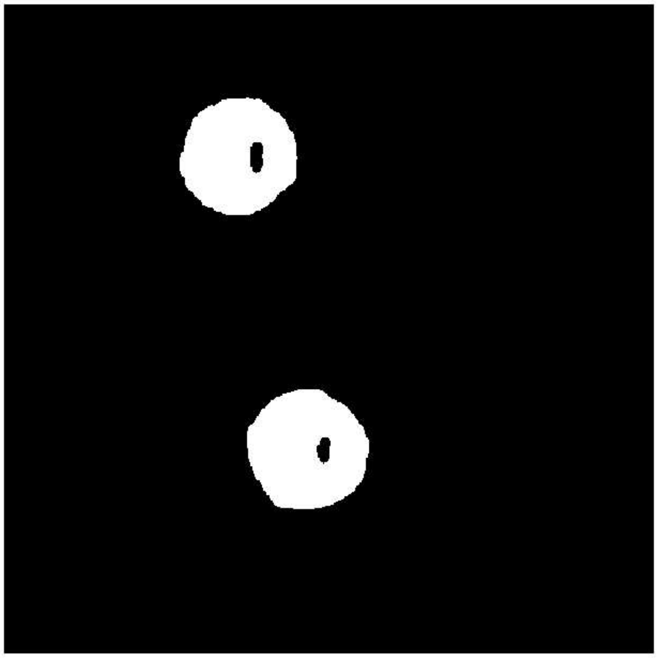

Figure 3. The mask (512 * 512 pixels) used to label the raw imageing pixels as “Red” and “Background”.

The following are a list of questions for you to work through. Please use one of following programming languages: R; Python; or Matlab. Present your assignment in the form of a report, with images, that also includes your code such that the analysis could be reproduced by another person.

You DO NOT NEED TO ANSWER ALL OF THESE QUESTIONS in your submission. You only need to answer as many as you can in a reasonable amount of time (less than 3 hours of work is reasonable, but you can work as long as you wish). Additionally, your FINAL ACCURACY IS NOT AS IMPORTANT AS YOUR METHODOLOGY. Your methods and presentation of results are more important than classification accuracy, and there is no need to spend hours improving the accuracy of your algorithms.

Files included:

`training.csv` - a data frame of 4022 rows and 33 variables. Variable Key: **X** - pixel number; **V_** - the value of the pixel for image 1 - 31 (i.e., wavelength 1 - 31); **Class** - the class of the pixel as either “Red” or “Background”.

`test.csv` - a data frame of 1978 rows and 33 variables. Variable Key: **X** - pixel number; **V_** - the value of the pixel for image 1 - 31 (i.e., wavelength 1 - 31); **Class** - the class of the pixel as either “Red” or “Background”.

`superballs_ms_1.png` - the first of 31 Raw Data images taken by the multispectral imager. **_1** indicates the image is from the first of 31 images taken at different wavelengths. There are 31 total images in the Raw Data, one from each of the 31 wavelengths.

`superballs_RGB.png` - NOT PART OF THE RAW DATA, just a reference color image

`Red_Mask.png` - the Mask used to label pixels as “Red” or “Bacground”

Question 1

For this problem. Load the reference library (`training.csv`) and explore the distributions of data for each group in the `Class` variable. Produce at least one graph.

Question 2

In this question, train a predictive algorithm to predict the variable `Class` . Provide the training (in-bag) accuracy of the algorithm. You can use any appropriate accuracy metric such as Sensitivity & Specificity, Kappa, etc. **Hint**: this can be performed by subsetting the training data into training and test sets. This is often done using bootstrapping or by simply splitting the data with a 60:40 ratio.

Question 3

In this section, apply your predictive algorithm to the testing dataset (`test.csv`). Provide the testing (out-of-bag) accuracy of the algorithm. You can use any appropriate accuracy metric such as Sensitivity & Specificity, Kappa, confusion-matrix, etc.

Question 4

Calculate the correlation between each feature (variable) in the reference library (`training.csv`). Present these results in any form such as a list, table, or graph.

Question 4

In this question we will go back to the RAW image data. Before we apply the predictive algorithm to the image, let’s take a look at the pixel data in the image. Provide an exploratory graph (similar to question 1) that includes each pixel in the Raw image. You may find that this will require you to do some data wrangling involving a conversion of the images to a data table where each row is a pixel and each column is a wavelength. In this case, the pixels are not labeled and we are interested in a graph of the distributions of pixel values for each of the 31 wavelengths.

Question 5

Now that you have wrangled the image data into a suitable form for prediction, apply your predictive algorithm to each pixel in the image. Graph the results of the predictive algorithm in the form of an image.

Question 6

Also provided in the folder is a `Red_Mask.png` file. this mask depicts the location of the red Superballs in the image. Use this mask to label each pixel in the original Raw image as either “Red” or “Background”. Use this information to compare the results of your predictive algorithm from **Question 6** to the actual status of the pixels in the image. You can use any appropriate accuracy metric such as Sensitivity & Specificity, Kappa, confusion-matrix, etc.

Question 7

List the importance of each feature in your classifier from best to worst.

Question 8

Compare the accuracy of three classifier types, such as KNN, LDA, Decision Trees, chose any three you wish.

Question 9

Compare the accuracy of any one classifier with and without a pre-processing method of your choice, such as normalizing, centering, PCA, or other data transformations.

Question 10

Using the Raw Image data, develop additional new features, using any methods you wish, train a classifier and test the accuracy. You could explore methods to generate new features using computer vision toolboxes such as openCV, etc.

Appendix - Example Results using Linear Discriminant Analysis (LDA)

The following are results we generated by training an LDA classifier with the pixels in the `training.csv` file and then applying this algorithm to classify the raw data. You may also use this same approach in your submission.

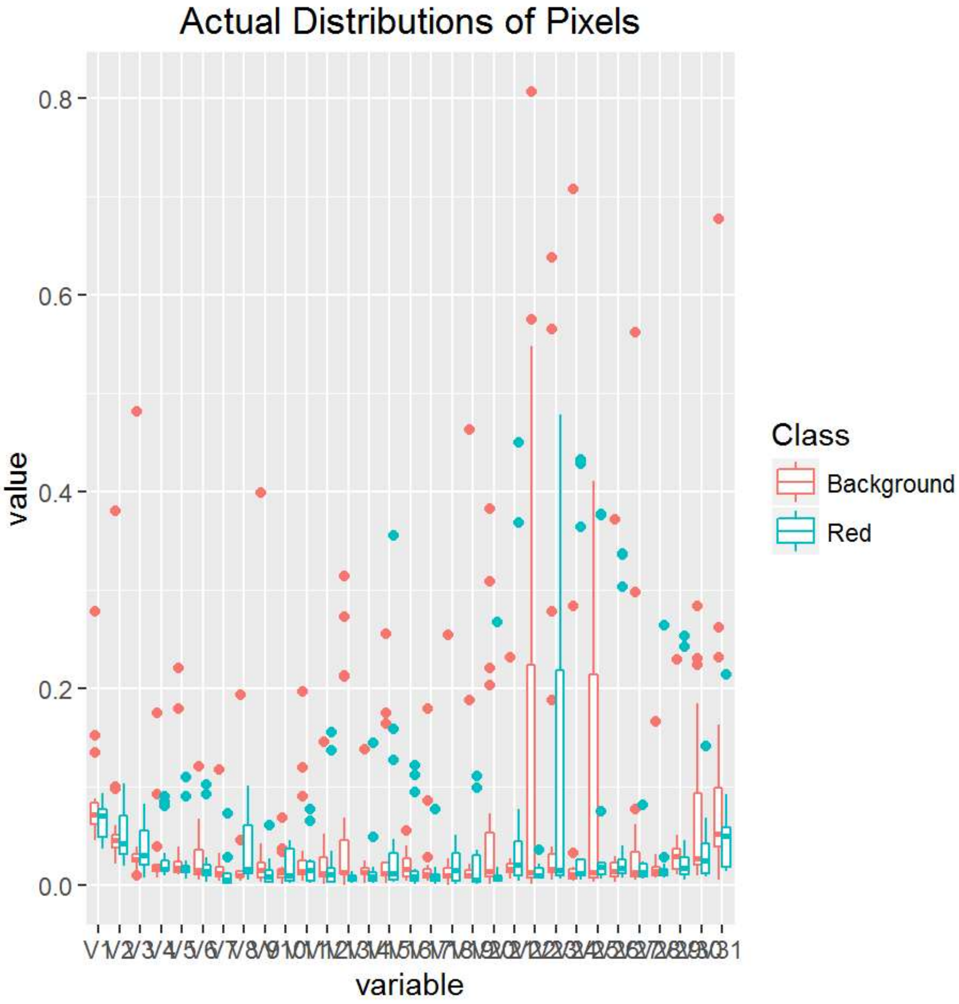

Figure 4. Example plot of the distributions of data from the two pixel classes in the reference library `training.csv`.

Table 1. Training accuracy (in-bag) accuracy of the classifier.

```
## Loading required package: MASS
## Confusion Matrix and Statistics
## 
##             Reference
## Prediction   Background Red
##   Background        819 170
##   Red               328 661
##                                           
##                Accuracy : 0.7482          
##                  95% CI : (0.7285, 0.7672)
##     No Information Rate : 0.5799          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.4965          
##  Mcnemar's Test P-Value : 1.988e-12       
##                                           
##             Sensitivity : 0.7140          
##             Specificity : 0.7954          
##          Pos Pred Value : 0.8281          
##          Neg Pred Value : 0.6684          
##              Prevalence : 0.5799          
##          Detection Rate : 0.4141          
##    Detection Prevalence : 0.5000          
##       Balanced Accuracy : 0.7547          
##                                           
##        'Positive' Class : Background      
## 
```

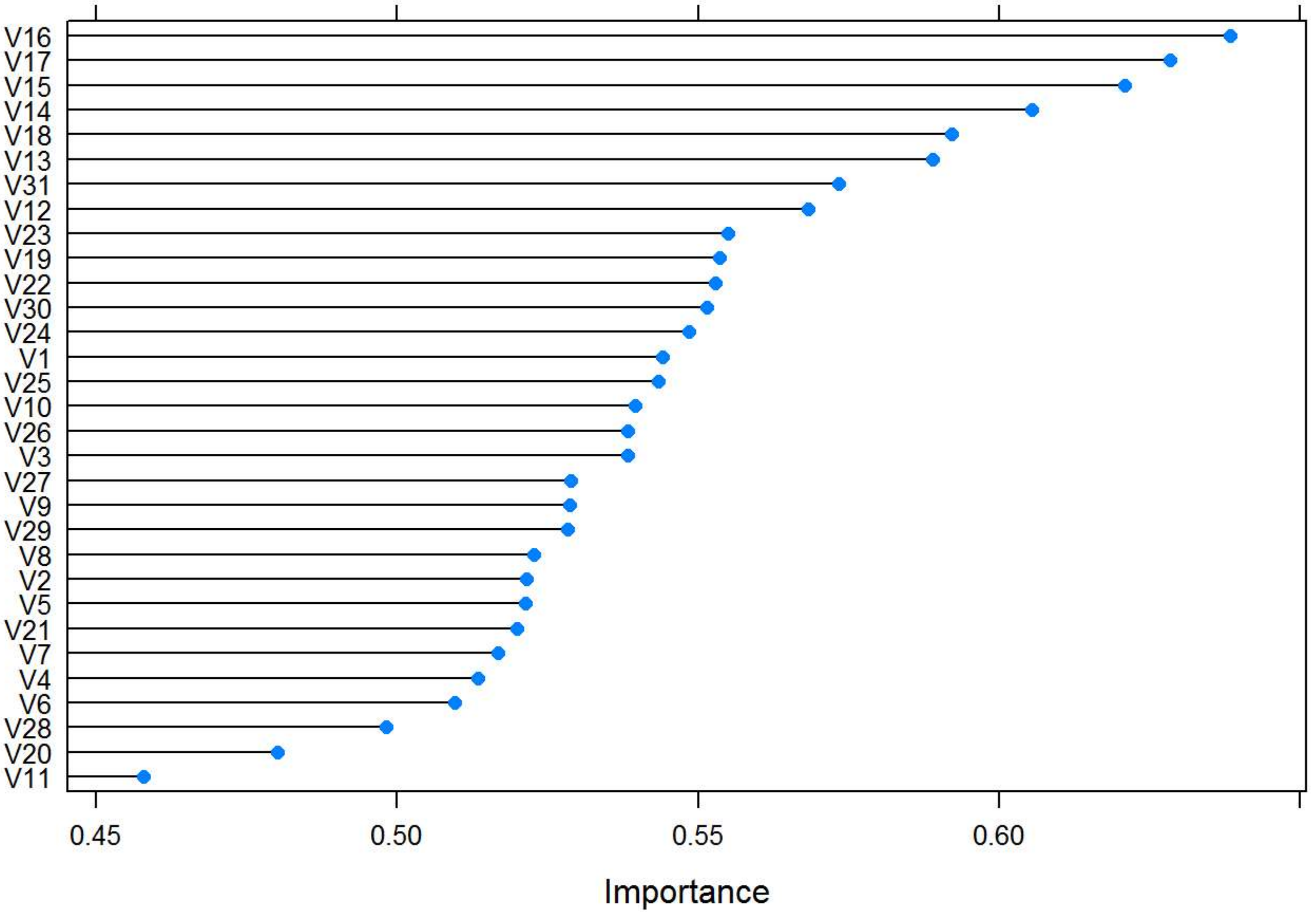

Figure 5. Importance of each variable used in the LDA algorithm.

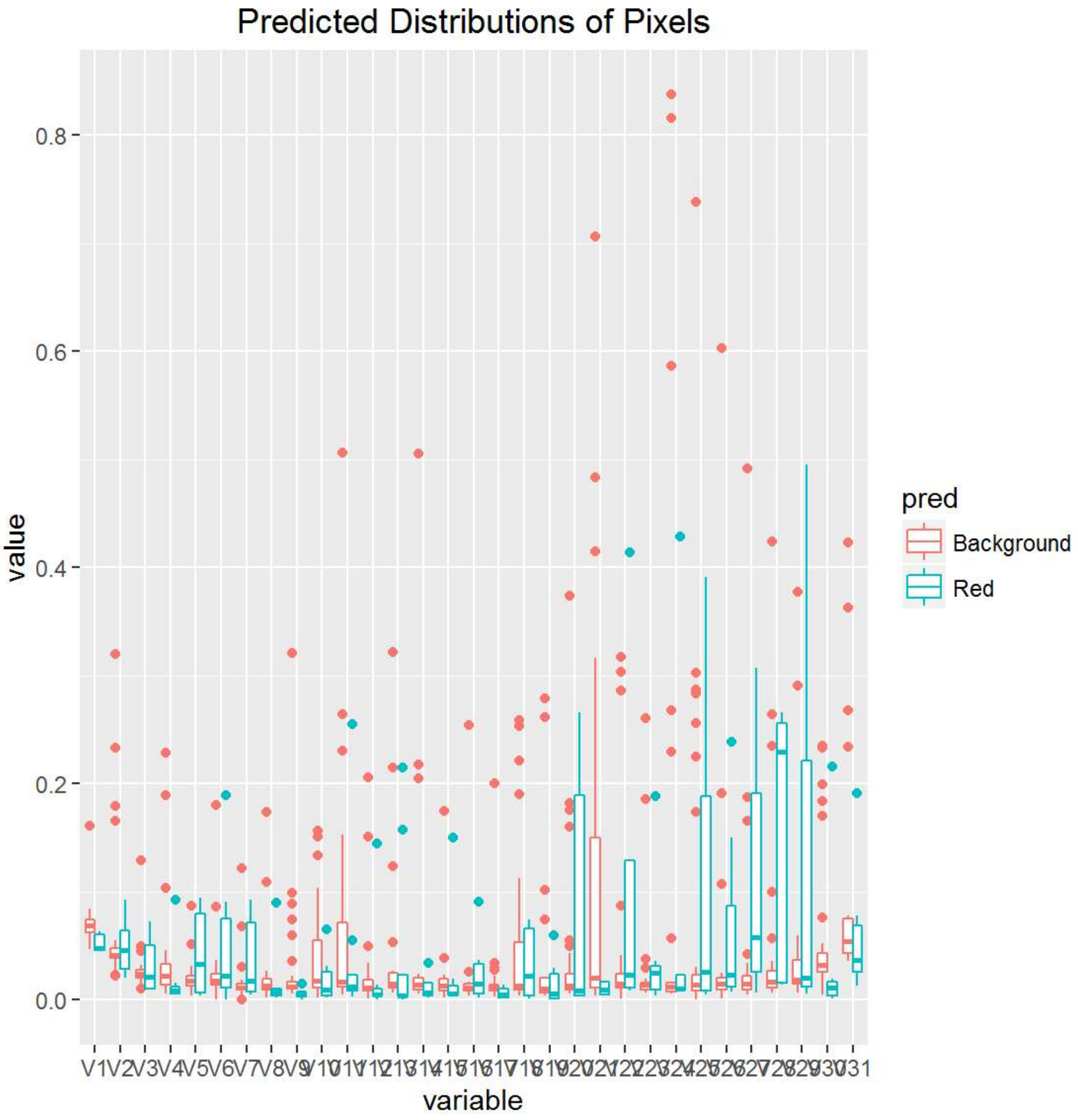

Figure 6. Distribution of pixels in the Raw Image Data based on prediction with the LDA algorithm.

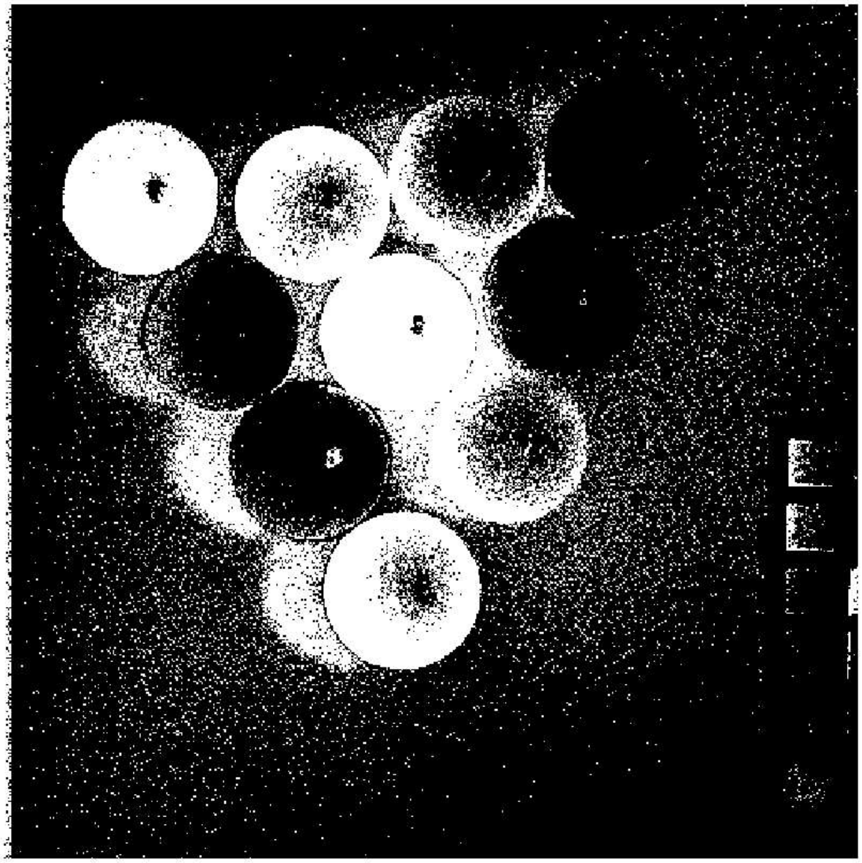

Figure 7. Classified results image indicating which pixels were classified as belonging to the Red superballs in **White**; and which pixels were classified as Background in **Black**.

Table 2. testing (out-of-bag) error of the classifier based on the entire image (using the mask rather than `testing.csv`).

```
## Confusion Matrix and Statistics
## 
##               Reference
## Prediction   Background    Red
##   Background     204526   4328
##   Red             44268   9022
##                                           
##                Accuracy : 0.8146          
##                  95% CI : (0.8131, 0.8161)
##     No Information Rate : 0.9491          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.2061          
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.8221          
##             Specificity : 0.6758          
##          Pos Pred Value : 0.9793          
##          Neg Pred Value : 0.1693          
##              Prevalence : 0.9491          
##          Detection Rate : 0.7802          
##    Detection Prevalence : 0.7967          
##       Balanced Accuracy : 0.7489          
##                                           
##        'Positive' Class : Background      
## 
```

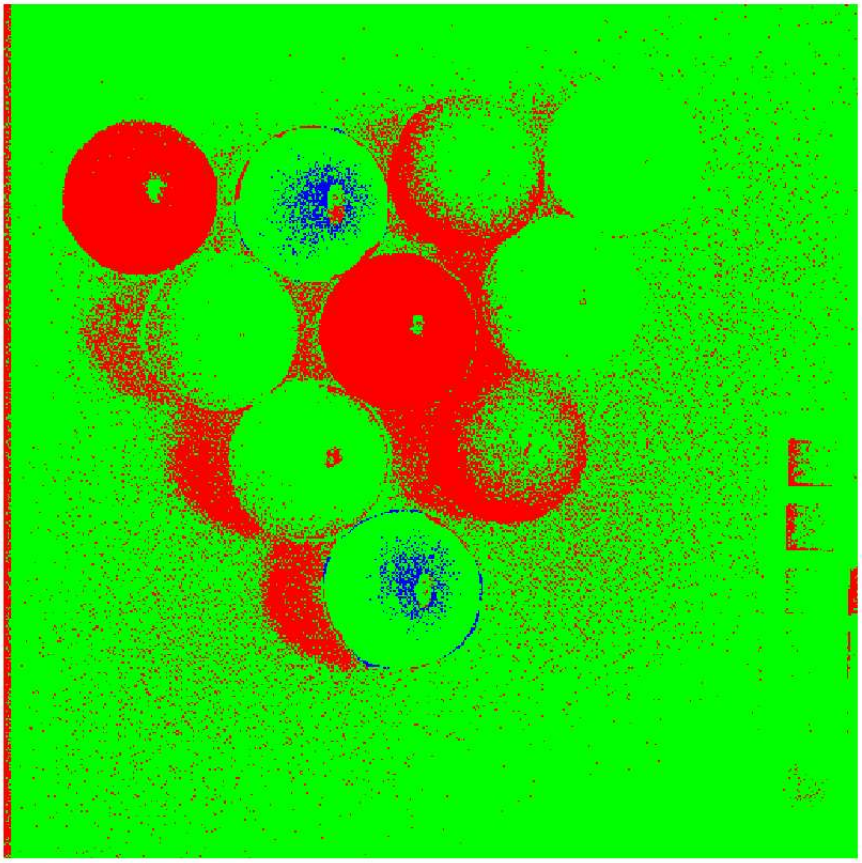

Figure 8. Error image indicating which pixels were correctly classified in **Green**; which pixels were false positives in **Red**; and which pixels are false negatives in **Blue**.