Fire Detection Image Processing

Chen, Shih-Chieh (Jack)

December 27, 2019

Introduction

Overview

This project looked at various images. Some of which contains fires while others do not. The purpose of this project is to create a model to classify them into images with fire (refered to as fire images) and images without (refered to as no fire images). A few key steps were done to create this model. The images were stored as jpeg or png images. They were stored in two folders with one folder storing fire images and another folder storing no fire images. Thus, the pixel information had to be extracted and then randomly split into training and testing sets. Afterwards, a random forest model was created using on the randomForest package.

Data Characteristics

The data was obtained from Kaggle's database, which can be found here https://www.kaggle.com/atulyakumar98/test-dataset. Unfortunately, I couldn't upload the image onto github due to memory limitations. However, I did upload processed test and train data sets onto github, found here https://github.com/chenjshihchieh/Fire-Detection. Looking at the files, we have 651 different images. We can take a look at the structure of the image with str() and some addition information with image_info, from the magick package.

```r
str(image)
```

```
## Class 'magick-image' <externalptr>
```

```r
image_info(image)
```

```
## # A tibble: 1 x 7
##   format width height colorspace matte filesize density
##   <chr>  <int>  <int> <chr>      <lgl>    <int> <chr>
## 1 JPEG    1500   1087 sRGB       FALSE  1173423 100x100
```

We can see that the images are of class magick-image. We see that this particular image has a width of 1500 pixels, height of 1087 pixels and colourspace of sRGB. The colour space tells us that the image is composed of the colours red, green, and blue. We can better see this when we convert the image to a matrix.

```r
numeric_image <- as.numeric(image[[1]])
dim(numeric_image)
```

```
## [1] 1087 1500    3
```

The dimensions x and y gives us the height and width. Dimension z gives us the number of colour channels, red, green, and blue for a total of 3. In other words, there are three 1087x1500 matrices. Each matrix represents a colour space with the values in each matrix representing the brightness of the respective colour on a pixel. The image we see is a combination of the 3 matrices.

```
## [1] "The above image was resized to better fit the page"
```

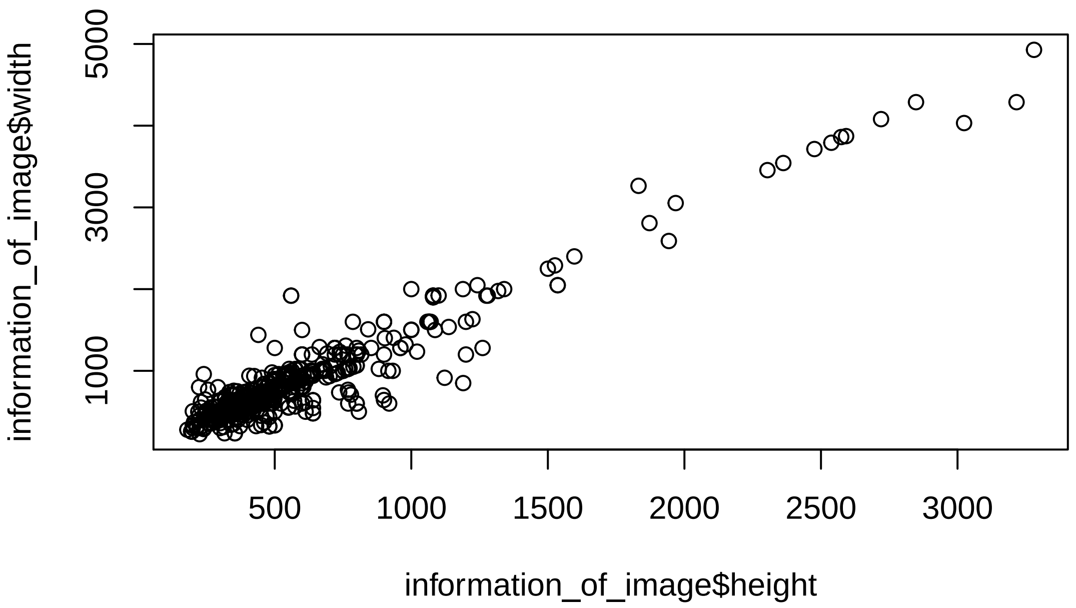
Analysis

Data Wrangling

Looking at the data, we see that the sizes of the images are quite different.

```
##     format              width            height         colorspace       
##  Length:651         Min.   : 225.0   Min.   : 180.0   Length:651        
##  Class :character   1st Qu.: 559.0   1st Qu.: 360.5   Class :character  
##  Mode  :character   Median : 664.0   Median : 461.0   Mode  :character  
##                     Mean   : 839.2   Mean   : 561.0                     
##                     3rd Qu.: 907.5   3rd Qu.: 600.0                     
##                     Max.   :4928.0   Max.   :3280.0                     
##    matte           filesize          density         
##  Mode :logical   Min.   :   3883   Length:651        
##  FALSE:645       1st Qu.:  48720   Class :character  
##  TRUE :6         Median :  91979   Mode  :character  
##                  Mean   : 215552                     
##                  3rd Qu.: 174095                     
##                  Max.   :7242687
```

If we plot the height and width, we can see that the ratio of some of the images are also quite differ-



ent. To address these differences, instead of using the value of every individual pixel, the image is divided into 100 equally sized regions. For each region, the average pixel value will be calculated for each colour space (red, blue, and green). This means that there will be 3 (colour space) X 100 (regions) predictors. The larger sized images have also been scaled down due to concerns with memory size and computation time. If the longest side of an image was over 640 pixels, the image was scaled down so the longest side has 640 pixels. The width to height ratio of the image was kept. In summary, the process was:
1. Resizing the images so no image has a side that is more than 640 pixels long 2. Randomly divided the image into training and testing sets 3. Divide the images into regions and extracted averaged pixel values for each colour space in each region

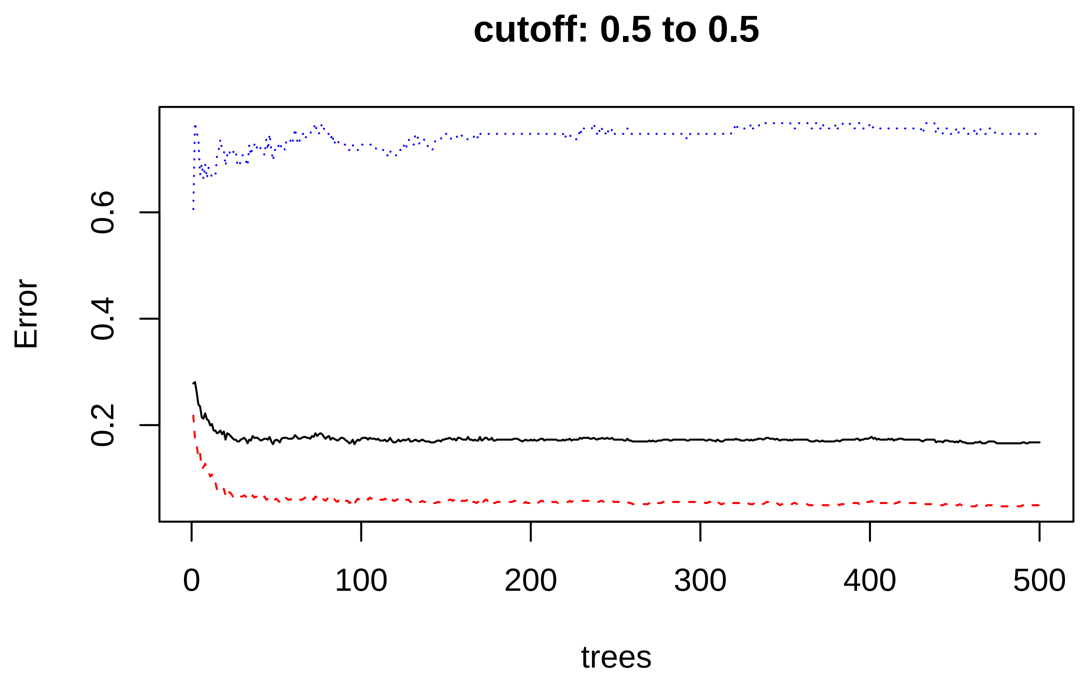
Model Creation

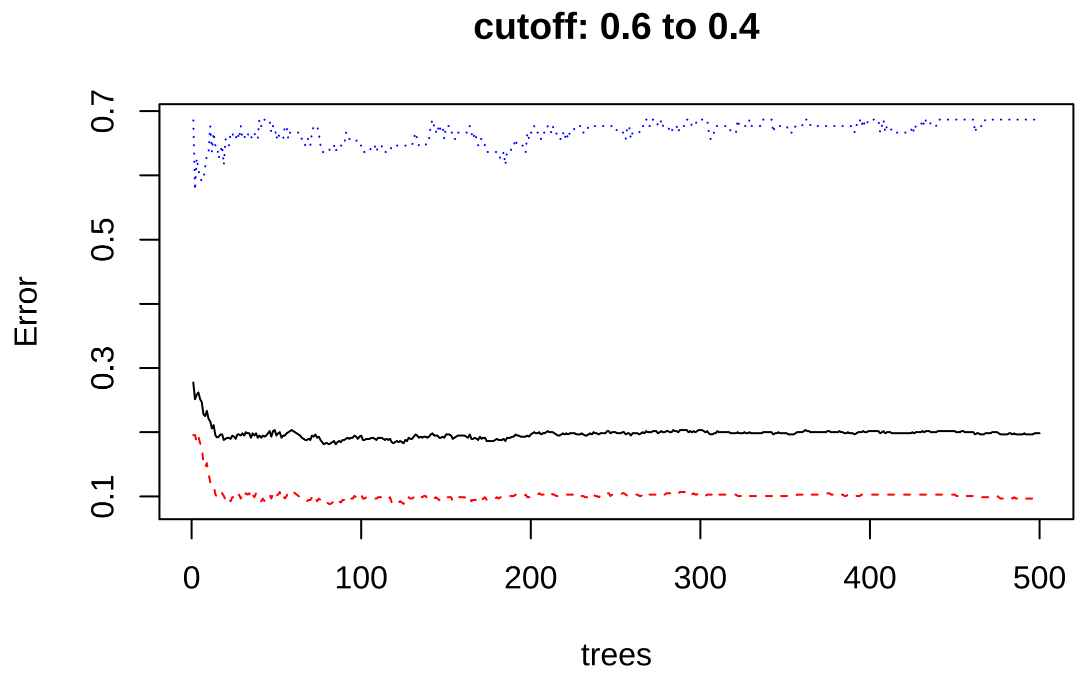
The randomForest package was used to develop a random forest model to classify the images. The training set was used to create and test the model. After optimizing certain parameters on the training set, the test set was used to determine performance. Looking at the training set, it seemed that there is a much larger proportion of no fire images than fire images.

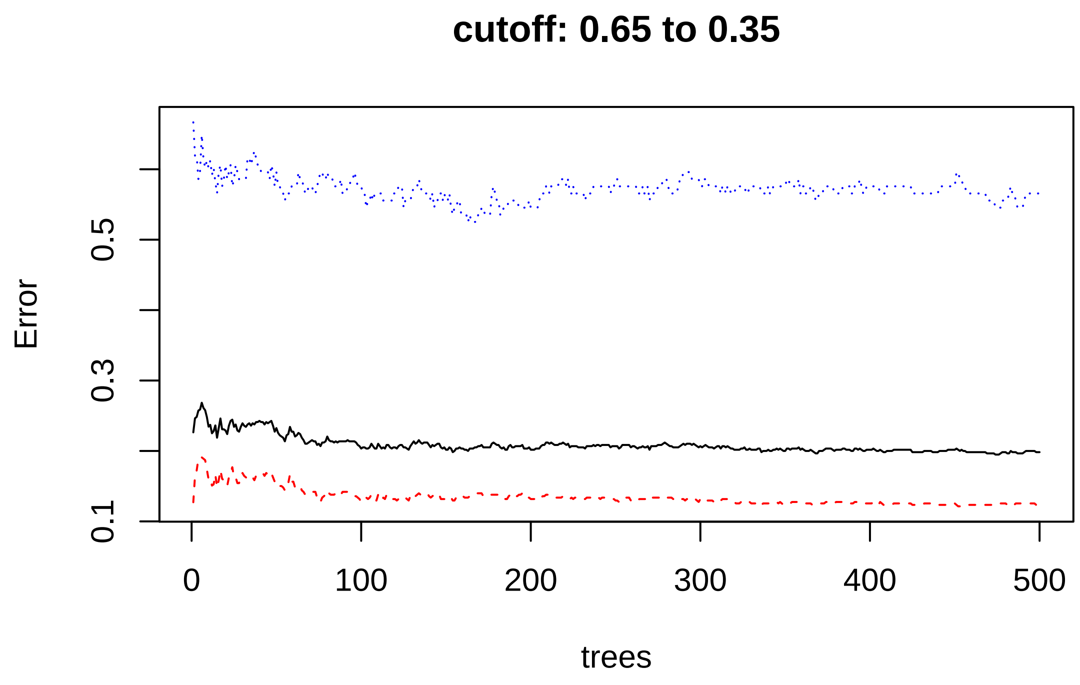
```
## [1] "Images with fire:99"
```

```
## [1] "Total number of images:585"
```

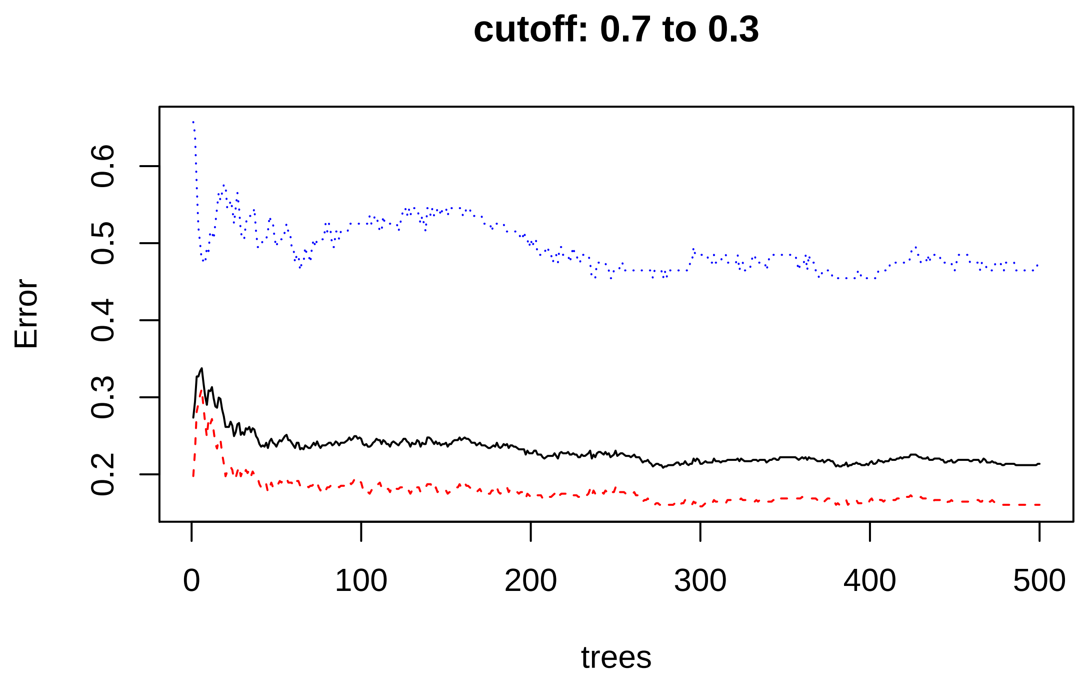
To account for this, I adjusted the class weights so that it represents the ratio of the fire images to the total number of images. Afterwards, I tried various different cutoffs to determine the best cutoff point for the random forest. The error by trees chart was used as reference to determine the best cutoff. The top dotted line represents the error rate of classifying the images as having fire. The bottom dotted line represents the error rate of classifying the images as having no fire. The black line is the average error of the other two error rates.


cutoff: 0.5 to 0.5
Error
0.2
0.4
0.6
0
100
200
300
400
500
trees

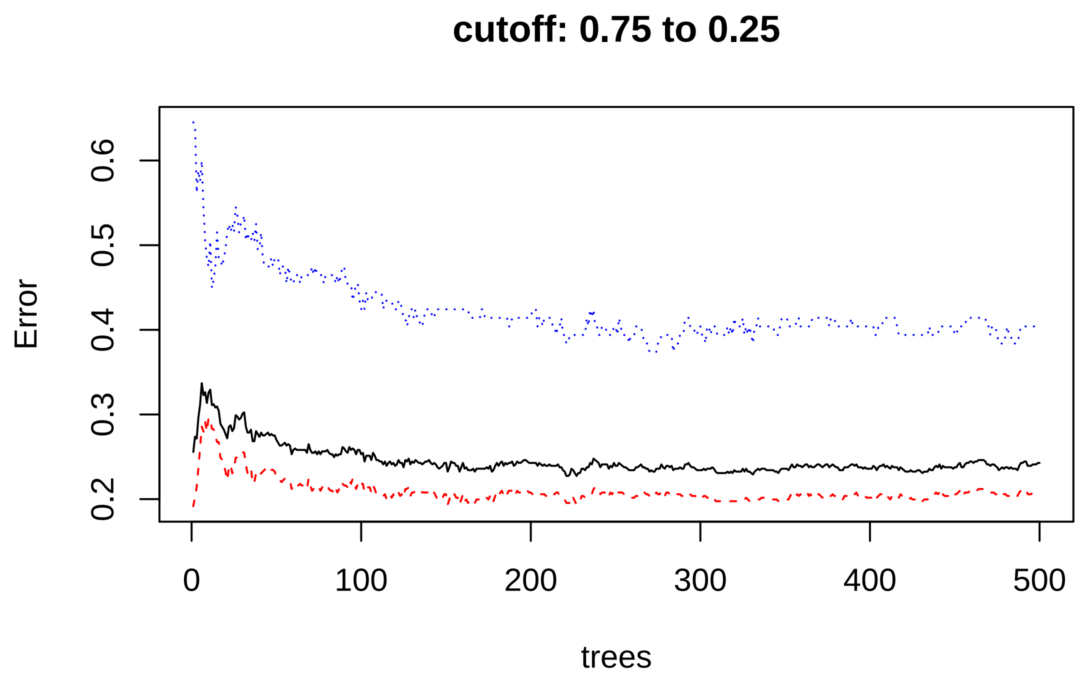

cutoff: 0.6 to 0.4
Error
0.1
0.3
0.5
0.7
0
100
200
300
400
500
trees



cutoff: 0.65 to 0.35
Error
0.1
0.3
0.5
0
100
200
300
400
500
trees



cutoff: 0.7 to 0.3
Error
0.2
0.3
0.4
0.5
0.6
0
100
200
300
400
500
trees



Results

Based on the plots, it was determined that the cutoff of 0.7 to 0.3 gave the best results. It reduced error rates for classifying images as having fire while not sacrificing too much of overall error rate. Using that model to predict our test set, we get the following results.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 47  6
##          1  8  5
##
##                Accuracy : 0.7879
##                  95% CI : (0.6698, 0.8789)
##     No Information Rate : 0.8333
##     P-Value [Acc > NIR] : 0.8744
##
##                   Kappa : 0.2881
##
##  Mcnemar's Test P-Value : 0.7893
##
##             Sensitivity : 0.45455
##             Specificity : 0.85455
##          Pos Pred Value : 0.38462
##          Neg Pred Value : 0.88679
##              Prevalence : 0.16667
##          Detection Rate : 0.07576
##    Detection Prevalence : 0.19697
##       Balanced Accuracy : 0.65455
##
##        'Positive' Class : 1
##
```

We managed to obtained an accuracy of 0.86, however, our sensitivity was 0.45. Looking at our test set, there was 11 fire images with 66 images in total giving us a prevalence rate of 0.1667. Looking at the results, while the accuracy of our model isn't bad, given that the purpose of the model is to detect fires and, hopfuly, alert the user, the model needs much improvement to increase its sensitivity.

Conclusion

While the model achieved an accuracy of 0.86. Given that the prevalence of fire images is 16.67%, we looked towards sensitivity for a more accurate representation of model effectiveness. Our model achieved a sensitivity of 0.45. Given that the purpose of the model is to predict fires, I think that a much higher sensitivity is required. One limiting factor when creating this model was computing power. With a more powerful system, maybe more detailed image could be used or an image could be divided into more regions for analysis. Another limiting factor is that this model only relied on random forest to generate its predictors. Convolutional neural networks could be used and models from previously learned images could have been incorporated. Convolutional neural networking programs with models from previously learned images are readily available for those who know where to look. Even if the classification categories are different, it could still help in our current project. Unfortunately, neural networking was beyond my current abilities and the random forest model was selected.