Ajanthan Mathialagan
214949861
April 10th 2020

MATH3333
Project Report

Abstract

Image authentication is considerably a new stream of research. A formidable dataset for experimentation and evaluation of new methods is necessary for further progress in image authentication research. We will be prioritizing the Columbia Photographic Images and Photorealistic Computer Graphics dataset which is accessible to the image authentication research community. This dataset is obtained from http://www.ee.columbia.edu/trustfoto where it is a subset of data which is composed of 4 image sets namely, Photorealistic Computer Graphics Set, Personal Photographic Image Set, Google Image Set, and the Recaptured Computer Graphics Set. This report will consist of a method to extract information from the images using the required pixel Intensity.

Introduction

Image authentication can be used distinctively for many purposes for instances looking at commercial and government organizations such as museums, libraries, etc. Many companies invest into new software and technology for image digitization, digital libraries for archiving and retrieval purposes. For instance, digital watermarking has been a relevant research area for many years. Many algorithm's throughout the years have been shown for image authentication and the detection of images which have been tampered with or not. This has been proposed with different authentication methods such as a required signal embedded in an image or even some contextual features in an image that can be extracted before the image can be confirmed. Lately many have come across a new process called passive-blind image authentication. Passive-blind image authentication is when you do not need to know any information and detail of the image beforehand, namely for the authentication for the content of the image and the detection of tampering of the image. An example of this is by looking at the classifications of photographic images and photorealistic computer graphics. Accordingly, there are many image datasets which are available to the research communities to work on using various image authentication techniques. My report will be discussing the design factor and implementation factor of one of these data sets namely, I'll be looking at photographic images which consists of 800 Columbia images.

Data

Description of the Data

The data set I will be using is designed and implemented by the Columbia Photographic Images. This dataset is used in work for the classification of Photographic Images. The Columbia Photographic image set contains 2 specific parts for example 800 of images are from the authors' personal collections (Personal Columbia) and 400 images from the personal collection of Philip Greenspun (Personal Greenspun). Greenspun's collection is included in the data set because it is to increase the diversity of the Personal Columbia set following the image content, the camera models and the photographer styles. The Personal Greenspun set are mainly images that have been taken from travels with content such as indoor, outdoor, people, objects, building and many more. Whereas the Personal Columbia set is acquired by the authors using the professional single-len-reflex Canon 10D and Nikon D70. It has content diversity in terms of indoor or outdoor scenes, natural or artificial objects, and lighting conditions of day time, dusk or night time.

PhotoMeta Data Set

Name: the name of the image (jpg)

Category: Whether the image is artificial, natural, outdoor-night, indoor-dark, outdoor-dawn-dusk, outdoor-rain-snow, indoor-light, outdoor-day

Camera:Whether the pictures were taken on either canon 10D or nikon D70

Location: Whether the photo is taken in boston or new york

Photographer: whether the photographer is either Jessie, Martin or Tian-Tsong

```
> summary(photoMetaData)
              name                    category         camera        location
 CRW_4786_JFR.jpg:  1   outdoor-day       :277   canon 10D:436   boston  :262
 CRW_4787_JFR.jpg:  1   artificial        :142   nikon D70:364   new york:538
 CRW_4788_JFR.jpg:  1   natural           :111
 CRW_4789_JFR.jpg:  1   indoor-light      : 74
 CRW_4790_JFR.jpg:  1   indoor-dark       : 68
 CRW_4791_JFR.jpg:  1   outdoor-rain-snow: 63
 (Other)          :794   (Other)           : 65
    photographer
 jessie    :102
 martin    :178
 tian-tsong:520
```

Methodology

The method I will be using consists of dividing the images into a matrix containing its pixel intensities based on RGB scale which will be my dependent variables where Red will be Vector1, G will be Vector2 and B will be Vector3. My response variable will be the categories which are artificial, outdoor-day, and so on.

Response variables

```
> Artificial<- as.numeric(photoMetaData$category == "Artificial")
> Natural <-as.numeric(photoMetaData$category == "natural")
> Outday <-as.numeric(photoMetaData$category == "outdoor-day")
> Outnight <-as.numeric(photoMetaData$category == "outdoor-night")
> Indoord <-as.numeric(photoMetaData$category == "indoor-dark")
> Indoorl <-as.numeric(photoMetaData$category == "indoor-light")
> Outddd <-as.numeric(photoMetaData$category == "outdoor-dawn-dusk")
> Outdrs<-as.numeric(photoMetaData$category == "outdoor-rain-snow")
```

Dependant Variables

(first few values of the predictors)

```
X <- matrix(NA, ncol=3, nrow=n)
for (j in 1:n) {
    images <- readJPEG(paste0("C:\\Users\\Ajanthan\\Desktop\\math3333\\columbiaIm
ages\\",photoMetaData$name[j]))
    X[j,] <- apply(images,3,median)
}
```

```
> head(X)
          [,1]      [,2]      [,3]
[1,] 0.4274510 0.4000000 0.3764706
[2,] 0.5254902 0.4470588 0.4352941
[3,] 0.5294118 0.4745098 0.4235294
[4,] 0.5490196 0.4980392 0.4470588
[5,] 0.4784314 0.4745098 0.4235294
[6,] 0.3568627 0.4352941 0.3215686
```

Models

Since my response variable consists of 8 categories I'll be running a generalized linear model analysis based on each category listed below, since there are 8 models we have,

Artificial Image

```
> out1 <- glm(Artificial ~ X, family=binomial, subset=trainFlag)
Warning message:
glm.fit: algorithm did not converge
> summary(out1)

Call:
glm(formula = Artificial ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.657e+01  4.447e+04  -0.001        1
X1           1.558e-13  2.401e+05   0.000        1
X2          -2.302e-13  4.485e+05   0.000        1
X3           1.683e-13  2.868e+05   0.000        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.000e+00  on 382  degrees of freedom
Residual deviance: 2.222e-09  on 379  degrees of freedom
AIC: 8

Number of Fisher Scoring iterations: 25
```

Natural

```
> out2 <- glm(Natural ~ X, family=binomial, subset=trainFlag)
> summary(out2)

Call:
glm(formula = Natural ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7099  -0.5135  -0.3783  -0.2621   3.1463

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.537      0.407  -3.777 0.000159 ***
X1           -14.628      2.995  -4.884 1.04e-06 ***
X2            21.803      4.322   5.045 4.53e-07 ***
X3           -11.111      2.587  -4.295 1.75e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 285.18  on 382  degrees of freedom
Residual deviance: 245.25  on 379  degrees of freedom
AIC: 253.25

Number of Fisher Scoring iterations: 6
```

Outdoor-day

```
> out3 <- glm(Outday ~ X, family=binomial, subset=trainFlag)
> summary(out3)

Call:
glm(formula = Outday ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0981  -0.8333  -0.5281   0.9937   2.4611

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.8742     0.3281  -5.713 1.11e-08 ***
X1           -6.3431     2.0397  -3.110  0.00187 **
X2            3.0172     3.3266   0.907  0.36441
X3            8.3905     2.1695   3.868  0.00011 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 497.11  on 382  degrees of freedom
Residual deviance: 400.69  on 379  degrees of freedom
AIC: 408.69

Number of Fisher Scoring iterations: 5
```

Outdoor-night

```
> out4 <- glm(Outnight ~ X, family=binomial, subset=trainFlag)
> summary(out4)

Call:
glm(formula = Outnight ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.15307  -0.28634  -0.16492  -0.07494   2.94935

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.1981     0.5675  -2.111   0.0348 *
X1            3.9335     2.5600   1.536   0.1244
X2           -7.0888     6.1051  -1.161   0.2456
X3           -8.5554     6.1396  -1.393   0.1635
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 126.60  on 382  degrees of freedom
Residual deviance: 100.76  on 379  degrees of freedom
AIC: 108.76

Number of Fisher Scoring iterations: 8
```

Indoor-dark

```
> out5 <- glm(Indoord ~ X, family=binomial, subset=trainFlag)
> summary(out5)

Call:
glm(formula = Indoord ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9176  -0.4666  -0.3142  -0.1965   2.6318

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.6686     0.4223  -1.583    0.113
X1            1.0911     2.1816   0.500    0.617
X2           -4.7414     4.5388  -1.045    0.296
X3           -4.0601     3.7749  -1.076    0.282

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 220.12  on 382  degrees of freedom
Residual deviance: 194.23  on 379  degrees of freedom
AIC: 202.23

Number of Fisher Scoring iterations: 6
```

Indoor-light

```
> out6 <- glm(Indoorl ~ X, family=binomial, subset=trainFlag)
> summary(out6)

Call:
glm(formula = Indoorl ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9787  -0.3687  -0.2365  -0.1547   2.7412

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.2005     0.6507  -7.992 1.32e-15 ***
X1           11.2409     2.1412   5.250 1.52e-07 ***
X2            5.5716     4.0180   1.387 0.165544
X3          -11.6488     3.1052  -3.751 0.000176 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 264.89  on 382  degrees of freedom
Residual deviance: 174.10  on 379  degrees of freedom
AIC: 182.1

Number of Fisher Scoring iterations: 6
```

Outdoor-dawn-dusk

```
> out7 <- glm(Outddd ~ X, family=binomial, subset=trainFlag)
> summary(out7)

Call:
glm(formula = Outddd ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9983  -0.3359  -0.2087  -0.1275   2.9284

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.9038     0.5630  -1.605    0.108
X1           -8.6037     5.3117  -1.620    0.105
X2           -7.5008     8.6431  -0.868    0.385
X3            8.2794     5.0886   1.627    0.104

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 139.14  on 382  degrees of freedom
Residual deviance: 119.80  on 379  degrees of freedom
AIC: 127.8

Number of Fisher Scoring iterations: 7
```

Outdoor-rain-snow

```
> out8 <- glm(Outdrs ~ X, family=binomial, subset=trainFlag)
> summary(out8)

Call:
glm(formula = Outdrs ~ X, family = binomial, subset = trainFlag)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7907  -0.4423  -0.3618  -0.2732   2.8138

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.3016     0.4507  -2.888  0.00388 **
X1           -2.3506     3.0355  -0.774  0.43872
X2           -7.0116     5.6176  -1.248  0.21198
X3            5.8837     3.5944   1.637  0.10165
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 205.43  on 382  degrees of freedom
Residual deviance: 195.50  on 379  degrees of freedom
AIC: 203.5

Number of Fisher Scoring iterations: 6
```

Analysis of model using wald tests

Wald test is a method we can use to see that the explanatory variables are significant in the model. In this case we are looking at the explanatory variables Red, Green and Blue if they have any significance in identifying the image based on their respective category.

```
> wald(out1)
 numDF denDF      F.value p.value
     4   379 5.328327e-07       1
            Estimate  Std.Error DF  t-value   p-value Lower 0.95 Upper 0.95
(Intercept) -26.56607  44465.5  379 -0.000597 0.99952  -87456.54  87403.4
X1            0.00000 240074.6  379  0.000000 1.00000 -472045.01 472045.0
X2            0.00000 448512.5  379  0.000000 1.00000 -881884.50 881884.5
X3            0.00000 286772.4  379  0.000000 1.00000 -563864.26 563864.3
> wald(out2)
 numDF denDF F.value p.value
     4   379 33.8615 <.00001
            Estimate   Std.Error DF  t-value   p-value Lower 0.95 Upper 0.95
(Intercept)  -1.537166 0.406985  379 -3.776958 0.00018  -2.337397 -0.736934
X1          -14.627930 2.994926  379 -4.884238 <.00001 -20.516682 -8.739178
X2           21.802856 4.321536  379  5.045164 <.00001  13.305667 30.300046
X3          -11.110561 2.586952  379 -4.294846 0.00002 -16.197137 -6.023985
> summary(photoMetaData)
              name                   category          camera        location
 CRW_4786_JFR.jpg:  1   outdoor-day      :277   canon 10D:436   boston  :262
 CRW_4787_JFR.jpg:  1   artificial       :142   nikon D70:364   new york:538
 CRW_4788_JFR.jpg:  1   natural          :111
 CRW_4789_JFR.jpg:  1   indoor-light     : 74
 CRW_4790_JFR.jpg:  1   indoor-dark      : 68
 CRW_4791_JFR.jpg:  1   outdoor-rain-snow: 63
 (Other)          :794   (Other)          : 65
    photographer
 jessie    :102
 martin    :178
```

The following output shows the wald test of the models based on the categories Artificial and Natural. As you can see the p-values for the Artificial model is extremely high all of them are basically 1 this implies that the colour scheme RGB are not significant predictors in identifying the image to be Artificial. Whereas we can see that in the data set there are 142/800 images that are Artificial this is worrisome because the model would not be accurate in detecting these images. On the other hand if we look at the Natural model we can see that the colour scheme RGB are significant predictors to the model as it will be accurate in detecting Natural images. We can evaluate the other models as well using the same methodology

Analysis and Results

I'll be running different analysis techniques in each model to distinguish their respective outcomes. Namely, I'll be looking at the LASSO method, ROC/ AUC curves and Fisher Linear Discriminant Analysis, to choose a concise model. Since there are 8 models in total i'll be looking at the model which verifies whether the pictures are natural or not. These methods can also be applied to the other models as well. They may differ with each model as we saw from the above on the wald test analysis of the model detecting artificial images and the model detecting natural images .

1. **LASSO method**

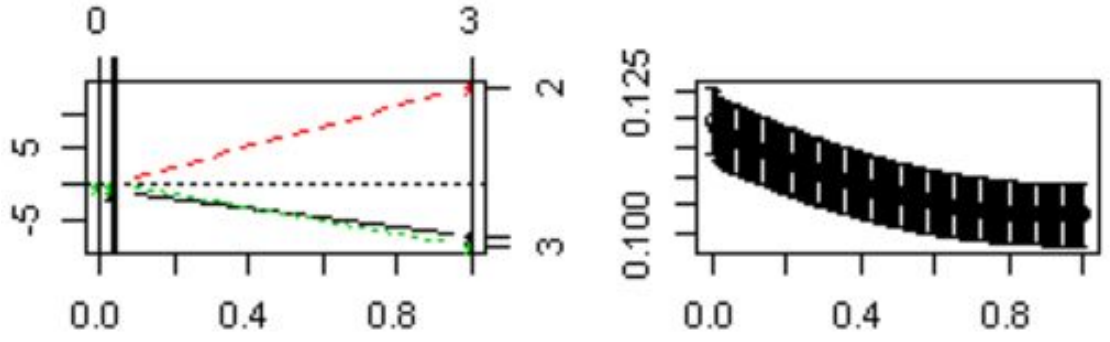
LASSO is a penalty-based variable selection approach that selects variables to be included into the model. This approach is certainly advantageous in regression situations when working in a large data set with a bunch of predictors in our case since we are looking at pixel intensity of the image we are only working with 3 predictors red, green, blue

```
> c <- model.matrix(Artificial~X, subset=trainFlag)
> c <- c[,-1]
> library(lars)

> lasso2 <- lars(x=c,y=Natural, trace=TRUE)
LASSO sequence
Computing X'X .....
LARS Step 1 :    Variable 1     added
LARS Step 2 :    Variable 2     added
LARS Step 3 :    Variable 3     added
Computing residuals, RSS etc .....
> plot(lasso2)

> lasso2

Call:
lars(x = c^2, y = Natural, trace = TRUE)
R-squared: 0.093
Sequence of LASSO moves:
     X1 X3 X2
Var   1  3  2
Step  1  2  3
> coef(lasso2,s=c(0.25,0.50,0.75,1.0), mode= "fraction")
             X1       X2         X3
[1,] -0.6270464 0.712478 -0.4767479
[2,] -1.0123241 1.610619 -1.0128006
[3,] -1.3976018 2.508761 -1.5488534
[4,] -1.7828795 3.406902 -2.0849062
> cv.lars(x=c,y=Natural, K=10)
```



The result acquired from cross-validation which is the average mean square errors and their associated standard error bounds shows that the mean square error increases quite rapidly if the coefficients are too small.
By looking at the graph on the right we see that the mean square curve is smallest at *s* = 1 which is the least squares solution, but we see that it gets flatter for s larger than 0.8. We can also see that the coefficients are not too small as they are negative values which mean we may need to apply more shrinkage in the coefficients, we also see that X2 has the highest value which is 'Green'.

2. Discriminant Analysis

Discriminant analysis is used to predict the probability of belonging to a given class or category from one or multiple predictors. It also will work with continuous and categorical predictors. We will be focusing on LDA linear discriminant analysis on the outcome of the natural images

```
> m1=lda(Natural~X,subset=trainFlag)
>
> m1
Call:
lda(Natural ~ X, subset = trainFlag)

Prior probabilities of groups:
        0         1
0.8772846 0.1227154

Group means:
         X1        X2        X3
0 0.3310341 0.3180789 0.2737979
1 0.2523988 0.2846058 0.2163538

Coefficients of linear discriminants:
          LD1
X1 -12.51429
X2  21.57459
X3 -12.25939
```

```
> eval=n-nt
> rep=5
> errlin=dim(rep)
>
> for (k in 1:rep) {
+         train2=sample(1:n,nt)
+         m1=lda(Natural~.,trainFlag[train2,])
+         predict(m1,trainFlag[-train2,])$class
+         tablin=table(trainFlag$Natural[-train2],predict(m1,trainFlag[-train2,])$c
lass)
+          errlin[k]=(neval-sum(diag(tablin)))/neval
+ }
> merrlin=mean(errlin)
> merrlin
[1] 0.5
```

We evaluate linear discriminant analyses by randomly selecting 600 from 800 estimating the parameters from the training data and classifying the remaining 200 images of the hold out sample. We repeated this 5 times and we achieved a 50% misclassification rate for the linear discriminant analysis, which is certainly high. Thus, the model based on the pixel intensity of the RGB colour scheme isn't too reliable in image authentication for natural images because there is a chance of 50% of the images being misclassified.

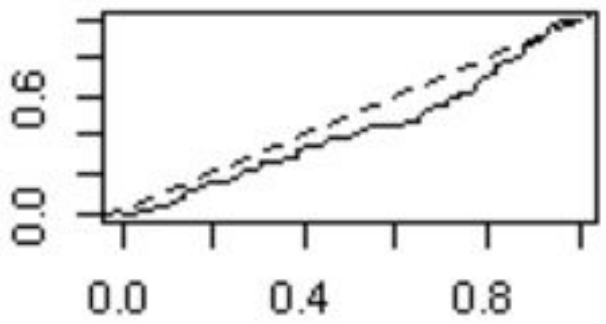
3. AUC/ROC curve

AUC and ROC curve is a type of measurement for classification purposes. In our case we are performing image classification. ROC is represented as a probability curve and AUC is the area under the curve which measures the separability. The higher the AUC the model is considered to be good. In our case the higher the AUC the better in predicting if the picture is a natural image or not a natural image

```
> pred <- 1 / (1 + exp(-1 * cbind(1,X) %*% coef(out2)))
> y[order(pred)]

> y[!trainFlag][order(pred[!trainFlag])]

> mean((as.numeric(pred > 0.5) == y)[trainFlag])
[1] 0.6292428
> mean((as.numeric(pred > 0.5) == y)[!trainFlag])
[1] 0.6330935
```



```
> auc <- function(r) {
+     sum((r$fpr) * diff(c(0,r$tpr)))
+ }
> glmAuc <- auc(r)
> glmAuc
[1] 0.4295519
```

Based on the best logistic model selected, using a random split 80% as training data and left 20% as testing data, the logistic model has an about 62.9% accuracy in predicting Natural images in the training set and about 63.3% accuracy in predicting natural images in the testing set. This shows that the logistic model is consistent in prediction performance because the values in both the testing set and the training set is similar but the value is relatively low as it is in the low 60% quartile. The inferences based on the logistic model isn't too reliable but given new data the model would perform well compared with a random guess with 50% accuracy only. In addition, looking at the AUC on the testing set is 0.42 which is 42% clarity that the model will detect the outcome of whether the image is or not a natural image that is relatively low. So the model prediction performance varies.

Conclusion

Image authentication\classification has a wide area for research, based on the 800 columbia images the model which distinguishes the colour of the image based on the RGB colour scale isn't the best model to use. We can see this through the analysis being inconsistent in deciphering whether the image is natural or not. I could say that there are many other methods that may be better than RGB such as gray-scaling the image and tracing it, adding a grid to each image and finding the median pixel intensity of each cell of the grid, etc. In conclusion, RGB scale is too general in terms of image identification; a model that can break down the image by layers will be more precise in identifying these 800 columbia images with higher accuracy.

Appendix

http://www.ee.columbia.edu/ln/dvmm/downloads/PIM_PRCG_dataset/
http://www.ee.columbia.edu/ln/dvmm/trustfoto/
https://cran.r-project.org/web/packages/jpeg/jpeg.pdf
http://www.ee.columbia.edu/ln/dvmm/publications/05/ng_cgdataset_05.pdf