Natural and Synthetic Image Classification

Project 1 | ECEN 662 - Estimation & Detection Theory

Team: Vikas Varma, Kai He, Chongzhi Zhao

Abstract:

Discriminating computer generated images from natural photographs is an important problem in image forensics. In this project, a binary classifier is developed by exploring the statistical differences between different types of images to classify them into two possible classes: Natural and Synthetic images.

Difference between photographs and graphics:

If we think about the features differentiating photographs and graphics, we can conclude that; graphics are typically generated using a limited number of colors and usually contain only a few monochromatic patches. Moreover, highly saturated colors are more likely to be used. Sharp edges are also a typical feature characterizing synthetic images.

On the contorary, very often a photograph depicts real life objects and subjects. These have textures, smooth angles and a variety of under saturated colors. In addition, the acquisition of a photographic through a camera makes it prone to noise which is absent in the synthetic images.

Fig : Natural vs Synthetic images: Depicting the texture, edge, noise and color differences.

Therefore, the difference between these two types of images can be analyzed from a statistical stand point and a hypothesis model can be developed to classify images accordingly.

Outline of the model developed:

In this project we approach the classification problem through the difference in color and brightness intensities between natural and synthetic images. Usually, natural images tend to have a wide spectrum of colors and their shades. Due to the directionality of a light source, the gradient of illumination is smooth for photographs which is usually not the case with computer generated images which use continuous patches of a fewer set of colors. This difference can be clearly visualized in the figure depicted in the previous section.

This difference of color pattern and spatial correlation of pixels in natural and synthetic images is observed by analyzing the gray histograms extracted from raw images. Later, statistical hypothesis are developed for the smoothness of the histogram for both the classes.

A binary classifier is modelled using a likelihood threshold rule on the observed distributions under each hypothesis.

Gray Histogram

The gray histogram is the gray scale version of the color histogram. It represents the distribution of colors in an image, derived by counting the number of pixels of each of all gray intensity. The analysis of this histogram can give an estimation of the distribution of area having the same color/brightness.

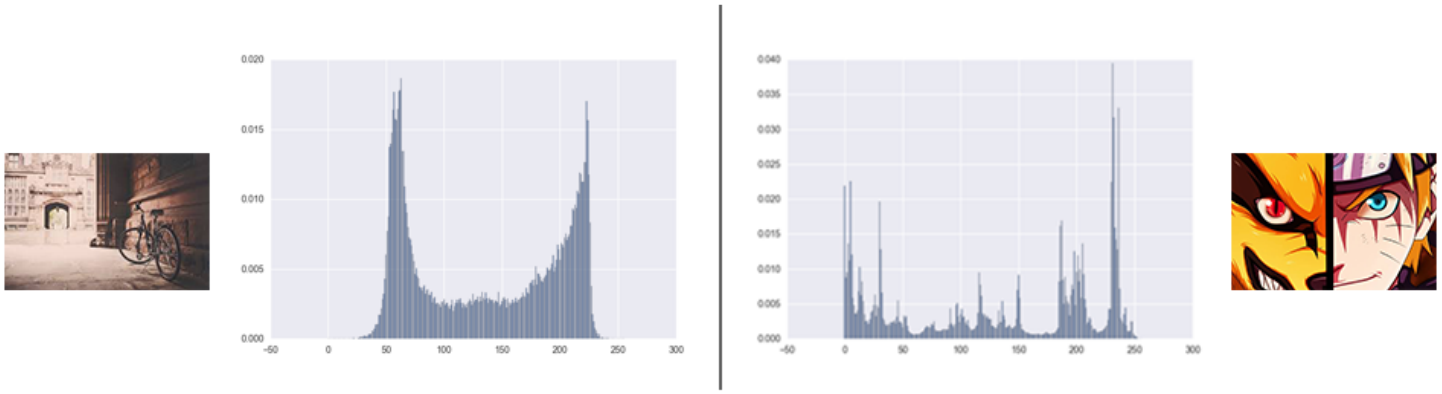


Fig: (left) Histogram of a Natural Image;
(right) Histogram of a Synthetic Image.

In figure above it is possible to see the typical difference between the histogram of a natural image as compared to that of a synthetic one. The natural histogram shows an overall smooth trend with peaks (if any) that are not really sharp. This depicts that the picture has uniform brightness. In the synthetic histogram, there are some high and narrow peaks as the picture contains various areas having uniform brightness.

The method used to build the histogram array is simple. For each pixel in the image, the value of the array at the index corresponding to the pixel brightness (0 to 255) is incremented by 1. The histogram is then normalized by dividing the value of all bins by the number of pixels. Give below is the function to compute the gray histogram of an image.

In [ ]:
```
## Psuedo Code:
def getgrayhist(image):
    for all i,j: # Iterating over each pixel.
        intensity = image[i,j]
        gray_histogram[intensity]++
    return gray_histogram
```

Smoothness of the histogram:

Having computed the histogram of the image, we now need a measure on its smoothness. A smoother histogram indicates that the image is natural and less smoother histogram is infered to pertain to a synthetic image.

To evaluate the smoothness, we compute the absolute difference between the adjacent elements of the gray histogram, i.e., we find out the difference between the fraction of pixels having adjacent intensities. Therefore, a smaller absolute difference indicates an overall smoother histogram and a larger value indicates otherwise.

The table below shows the histogram smoothness measures for three natural and synthetic images used as training data in developing the model.

Natural	Synthetic
0.0835	0.951
0.0754	1.254
0.225	0.691

In [ ]:
```
# Psuedo Code:
def getsmoothness(hist):
    for each intensity in gray_histogram
        smoothness += abs(gray_histogram[intensity]-gray_histogram[int
ensity+1])
    return smoothness
```

Training the model:

We collected a bunch of images from the web, with an emphasis to cover a variety of synthetic and natural images. The smoothness feature for each image in the training data set is simultaneously calculated and stored in a CSV file. It is important to note that this is a computationally intense operation and if our dataset consists of thousands of images, it is adviced to resize/compress the image before feature extraction. Hence, we have compressed the images to a size of 256X256 to overcome the computational overhead in processing large images.

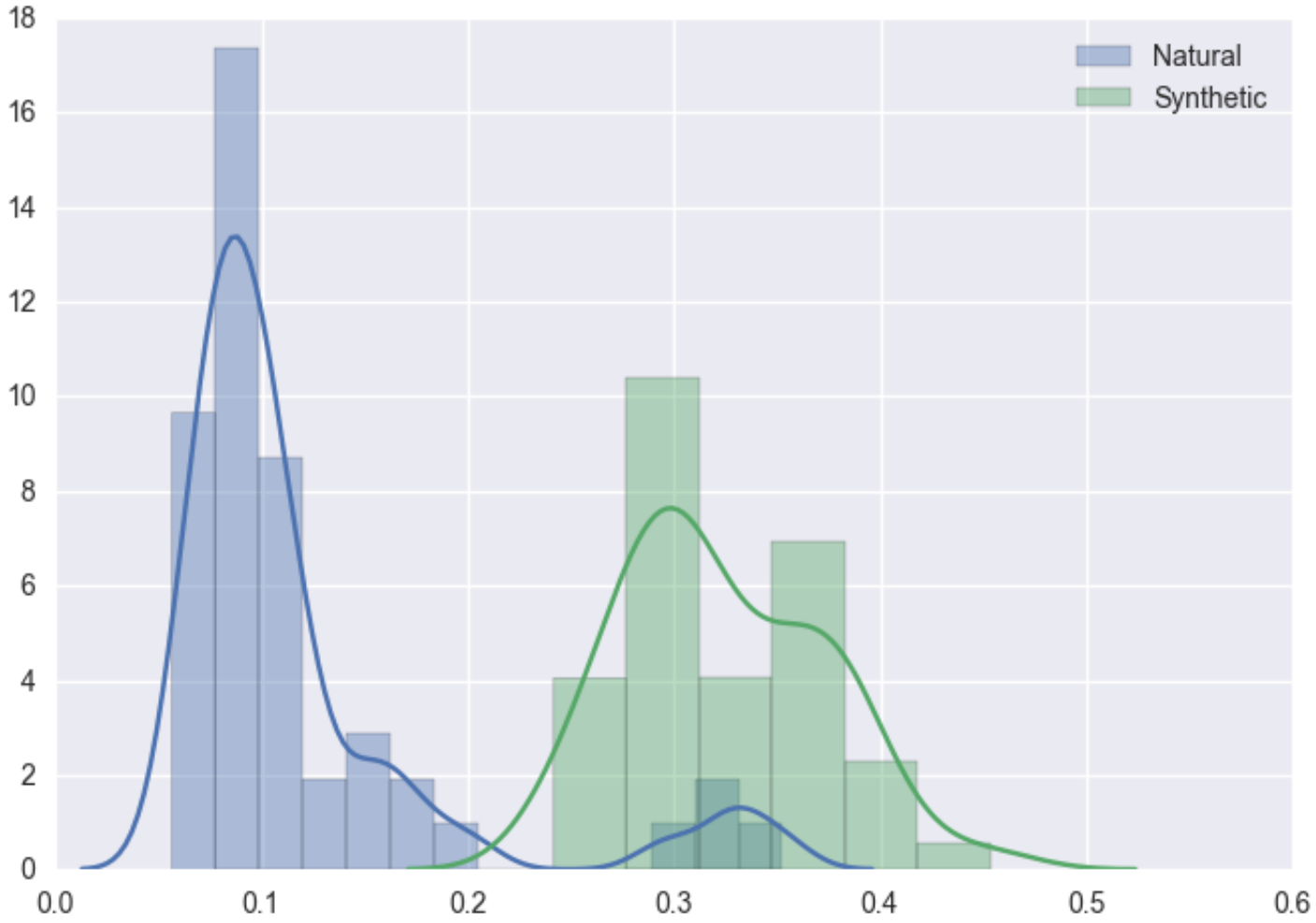
In [29]:
```
# Pseudo Code:
for scene in scene_dataset:
    gray_histogram = getgrayhist(scene)
    smoothness = getsmoothness(gray_histogram)
    smoothness_list.append(smoothness)
    % repeat for cgi dataset %

    smoothness_dictionary = {'Natural': scene_smoothness_list,
                             'Synthetic': synthetic_smoothness_list}
    df = pandas.Dataframe(smoothness_dictionary)
    df.to_csv('HistSmoothResult.csv')
```

Distribution of the feature:

We observe from the training features that the distribution of our smoothness measure can be approximated fit to the curve plotted in the figure below. For natural images, this parameter is centered around a mean of 0.09 (appx.) while for a synthetic image, it is centered around 0.32 (appx.). These observations parallel our assumptions that a smaller smoothness indicates a natural image as the distribution for the natural image histogram smoothness is narrower and has a small variance in comparison to that of the synthetic images whose density is spread out. These distribution therefore characterize our hypothesis testing.

In [ ]:
```
# Psuedo Code:
training_data = pd.read_csv('HistSmoothResult.csv')
plot.distribution(natural,synthetic)
```



Decision Rule:

From the distribution plots above we can clearly see that a maximul likelihood threshold on the smoothness can be converted into a threshold on its absolute value and choosing a threshold close to 0.22 should give a good performance overall.

If we consider equal priors for both the classes, then the decision rule can be modelled as choosing H0 (Natural image) if the smoothness value is below threshold and choosing H1 (Synthetic Image) if the smoothness value lies above.

$$\varphi = \begin{Bmatrix} 0, if smoothness < 0.22 \\ 1, if smoothness > 0.22 \end{Bmatrix}$$

In [10]:
```
# Psuedo Code:
def executedecisionrule(test_path):
    thr = 0.22
    for image in test_directory:
        res = 1 if sm>(thr) else 0
        res_list.append(res)
    return res_list
```

While testing the decision rule for ten randonly chose images from the test dataset, we observed the following classification results:

Image	Result
Natural1.jpeg	Natural
Natural2.jpeg	Natural
Natural3.jpeg	Natural
Natural4.jpeg	Natural
Natural5.jpeg	Natural
Synthetic1.jpg	Synthetic
Synthetic2.jpg	Synthetic
Synthetic3.jpeg	Natural
Synthetic4.png	Synthetic
Synthetic5.jpeg	Synthetic

Performance:

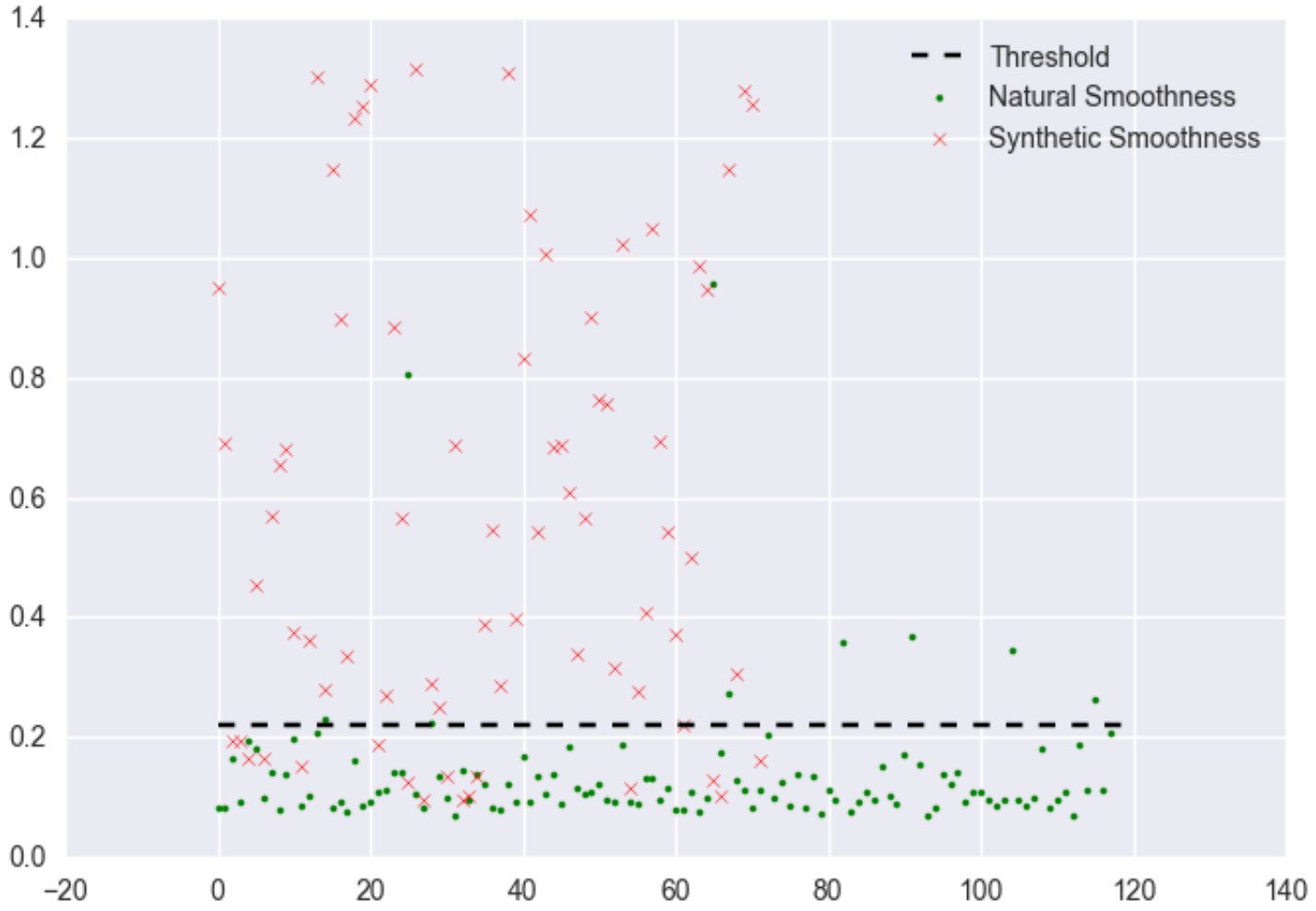
The decision rule is observed to perform well for the training dataset used. The rule correctly identified 109/118 natural images tested and 56/72 sythetic images in from the training data.

```
Probability of false alarm = 0.076  |  Probability of Detection = 0.78
```

The lower performance of the classifier can be attributed to the fact that the model has been trained from a limited number of training samples and also because of model limitation of using only a single distinguishable feature. The performance of the decision rule can be improved by considering more features and concluding the hypothesis based on weighted individual parametric decision.

Scatter plot for gray histogram smoothness of test images:

Below is the scatter plot of the smoothness value for all the test images used in evaluatin the performance of the decision rule:


Threshold
Natural Smoothness
Synthetic Smoothness
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−20
0
20
40
60
80
100
120
140

In [ ]: