Final report for an implementation on the paper "understanding Benford's law and its vulnerability in image forensics"

XIAOYU WANG
University of Winnipeg, Manitoba, Canada

ABSTRACT
In this report, we try to follow the idea of the paper "Understanding Benford's law and its vulnerability in image forensics" [1]. There're two major concerns of this implementation. First, apply Benford's law on natural and synthetic image forensics. Second, apply Benford's law on detecting JPEG compression. The vulnerabilities of applying Benford's law into the two topics mentioned above will be discussed later with a special case of so called "histogram manipulation attack" [1].

1. INTRODUCTION

1.1 MOTIVATION

Digital images are widely used nowadays for their properties of easy to produce, transmit and display. While, the integrity of digital images is compromised by their "easy" properties since they are also easy to be modified. It's quite easy for people who are even not professional to digital image processing to accomplish image modification with a variety of software, such as Photoshop. The aim of image forensics is to tell the modified images from the original ones so that guarantee the integrity of digital images. In this project, we will apply a statistical law (Benford's law) into image forensics, which is quite an innovative point of view comparing to traditional methods.

1.2 BACKGROUND

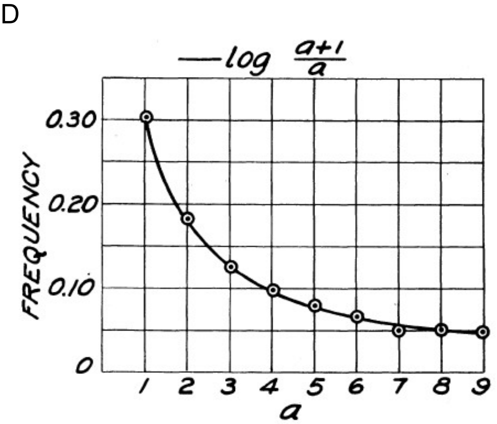

Figure 1: the plot of fist digit distribution by Benford [3]

The phenomenon that the 10 digits (from 0 to 9) do not occur in the same frequency is first discovered by Newcomb [2] in 1881. On observing the logarithmic tables, he found that for the first significant digit (FSD), 1 tends to occur more frequently than 9. What is more, for a series of natural numbers, the frequency on the FSD is a decreasing function from digit 1 to digit 9.

In 1938, this phenomenon is formalized by Benford [3] using a larger sample of data. In his paper, a probability distribution function is given as:

$$P(d) = \log_{10}\left(1 + \frac{1}{d}\right)$$

where d is the FSD from 1 to 9.

For example, for d = 1, the probability P(1) = $\log_{10}(1+1)$ = 0.301 and for d =2, the probability P(2) = log10(1+1/2) =0.176.
This principle is called “Benford’s law” or “first digit law”. This is a statistic law and can be used into various applications such as fraud detection for accountings. [4]
There’re some fundamental properties for benford’s law, as mentioned in the paper of Perez [7]:

1: A random variable X follows benford’s law if the random variable Y = $\log_{10}(X)$ mod 1 is uniform from [0, 1), X is called strong benford and Y is called the benford domain.
2: If a random variable X is strong benford, then Z = aX is also strong benford for an arbitrary a.
3: If a random variable X is strong benford, then Z = XY is also strong benford for a random variable Y that is independent of X.

From the properties above we can conclude that the condition in which Benford’s law can be applied is that the probability of the random variable should be uniform on a log10 scale. What is more, this law is scale-invariant and applies to independent data.

1.3 SCOPE&LIMITATION

To the best of our knowledge, Jolin [5] was the first one considering to apply Benford’s law into image processing and discovered the gradient of pixel values were follows Benford’s law. Acebo and Sbert [6] pointed out that under certain constraints, the intensity of digital image agrees with Benford’s law. Perez [7] presented a generalization of Benford’s law and proved it applies to the discrete cosine transform (DCT) domain. Fu and Shi [8] showed a method using Benford’s law to detect JPEG compression. Jingwei and Byung [1] put forward some vulnerabilities of applying Benford’s law into image forensics.
As an implementation of [1], this project is trying to cover the main stream of this paper. More than that, we also try to reach for papers related pretty much with [1], for example [6] and [8]. There are still some important issues that we are not able to cover. First, since Benford’s law is a statistics law, it takes time to do the statistics, for instance, calculating the pixel values of thousands of images to decide parameters to a model. Our time is limited so that we have to rely on some of the existing models in the paper I just mentioned. Second, some technologies in those papers are challenging to us currently, such as the radiosity and rendering, so we have to give up some ideas that we intended to have a try. Third, not all resources we need are available, for instance, we tried hard but found no way to get “raw images” where there is no interpolation, no gamma correction and etc.

2 BENFORD’S LAW ON REAL AND SYNTHETIC IMAGES

2.1 GENERAL IDEA

There is one thing that I need to claim first. In our target paper [1], there is an example about cartoon image detection using the DCT coefficients. [1] pointed that out as one of the vulnerabilities of applying Benford’s law on image forensics. It can be easily learnt that this “vulnerability” is against [6]. While the point is, there is not even a word about

DCT in [6] and the fact that DCT coefficients of an image obey a generalized Benford's law was claimed to be discovered in [7], which is later published than [6]. That is to say, [1] totally get confused with [6], which talked about pixel intensities rather than DCT coefficients. To our comprehension of [6], there is no way to use Benford's law on pixel intensity to tell cartoon images apart from non-cartoon images. Further, to the best of our knowledge about DCT coefficients, there is no way to do this on DCT coefficients either. So we drop the wrong concept here and the following discussion will be around concepts and methods in [6].
Acebo and Sbert mentioned in [6] that there is still "a great variety of natural and artificial images" whose intensity level will follow Benford's law.
For synthetic images, [6] categorizes them into physically realistic group and physically non-realistic group. The conclusion drawn is that Benford's law works well for the physically realistic group while not applicable for the physically non-realistic group. Several examples are given in [6] and I will present some as follows.

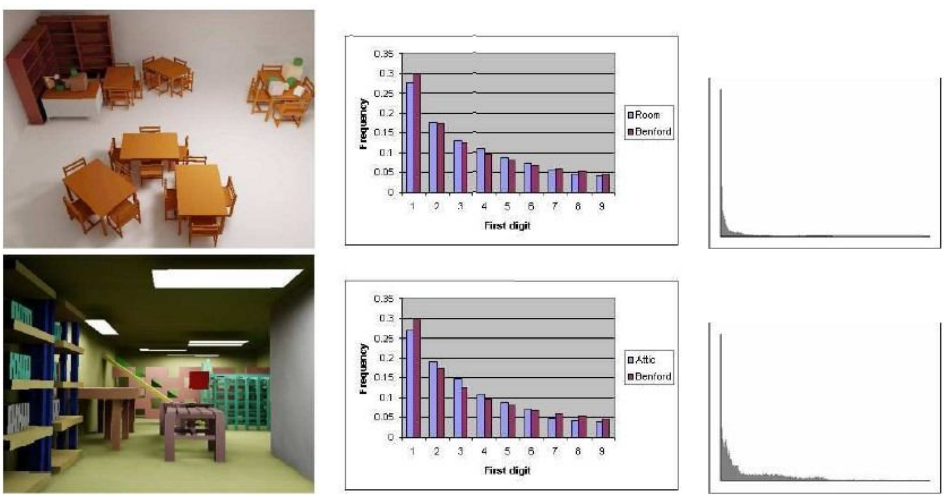

Figure 2: physically realistic group

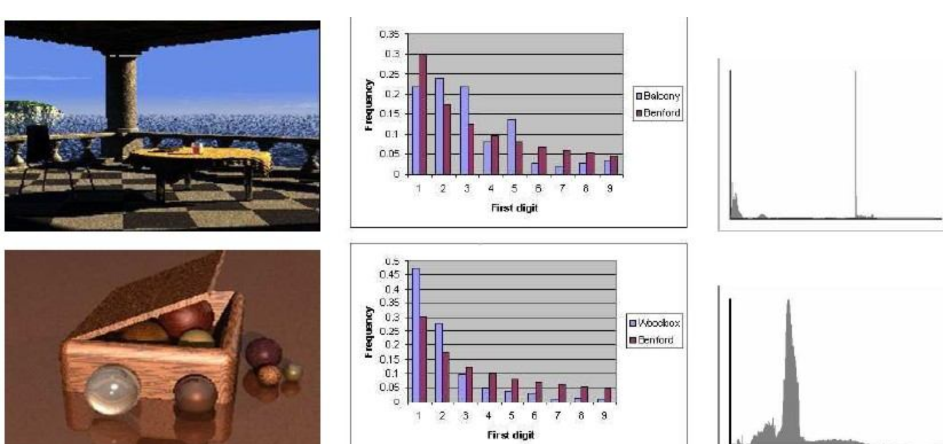

Figure 3: physically non-realistic group

For real images, the problem remains open. What we mean "real images" are actually pictures of real images. The situation is that when a digital image is presenting on the screen of our computer, a lot of processing methods such as gamma correction and interpolation was applied on it, meaning it is not as original and real as we expected. On condition that we can access "raw images", still there are various kinds of noises getting involved when the real image is sensed by sensors of a photographing machine. Probably these are the reasons why this problem remains open: some obey the law while others not. In [6] and some other papers, the concept of X^2 divergence was put forward as standards to see the quality of fitness to Benford's law:

$$X^2 = \sum_{i=1}^{9} \frac{\left(f(d) - \log_{10}\left(1 + \frac{1}{d}\right)\right)^2}{\log_{10}\left(1 + \frac{1}{d}\right)}$$

where d is the FSD from 1 to 9 and f(d) is the frequency of FSD d by intensity in the target image.

2.2 OUR APPROACH

2.2.1 INPLEMENTATION METHOLOGY

Following the idea in [6], we aim to get the divergence from a digital image. Here follows our steps:
Step 1: read in an image and get the first digit for intensity of every pixel. In case of a colour image, we get the 3 intensities of its RGB components.
Step 2: do the statistics on the result above thus getting the frequency of FSD f(d).
Step 3: calculate the divergence X^2.
Though intensities of digital image is integers from 0 to 255, there could be situations that we have to convert these integers into fractions and even negatives. So a universal description of the algorithm to get the FSD of an intensity value adopted all through this project is:
Step 1: get the absolute value of this intensity.
Step 2: get the maximum integer that is less than this intensity value.
Step 3: divide the integer from the second step by 10.
Step 4: if the result from the third step is less than 1, this result is FSD of this intensity. If not, repeat from step 3.

2.2.2 RESULTS AND ANALYSIS

For synthetic images, the result we get is not as good as examples in [6]. As shown in figure 4, this image of "radiosity factory" is a typical one that disagrees with Benford's law. From the bar graph we can see the FSD distribution of this image is more oscillation like with the valley at FSD 4 than logarithm like. This example is not exceptional; we met a lot of similar cases like this in the test work. On the contrary, we are doubtful that those successful examples in [6] are somehow intended.
Let's have a review on figure 2. We noticed that the histograms of successful examples are kind of special in shape. They are logarithm like. Perhaps this is the best answer to why they obey Benford's law. If the histogram of an image is logarithm shaped (especially logarithm base 10), the random variable $Y = \log_{10}(X) \bmod 1$, where X represents the variable for intensity values, is uniform from [0, 1). By property 1 we mentioned in section 1.1 we know that in such conditions, X obeys Benford's law and it

- 5: 5

is called a strong benford. While, here comes the question: how can we make sure the histogram of a synthetic image with a shape of logarithm? By the experience of watching lots of histograms, we guess the only way is to intentionally select intensity values to produce the image. In truth, it's possible since synthetic simply means man-made. But the point is that there is no guarantee.

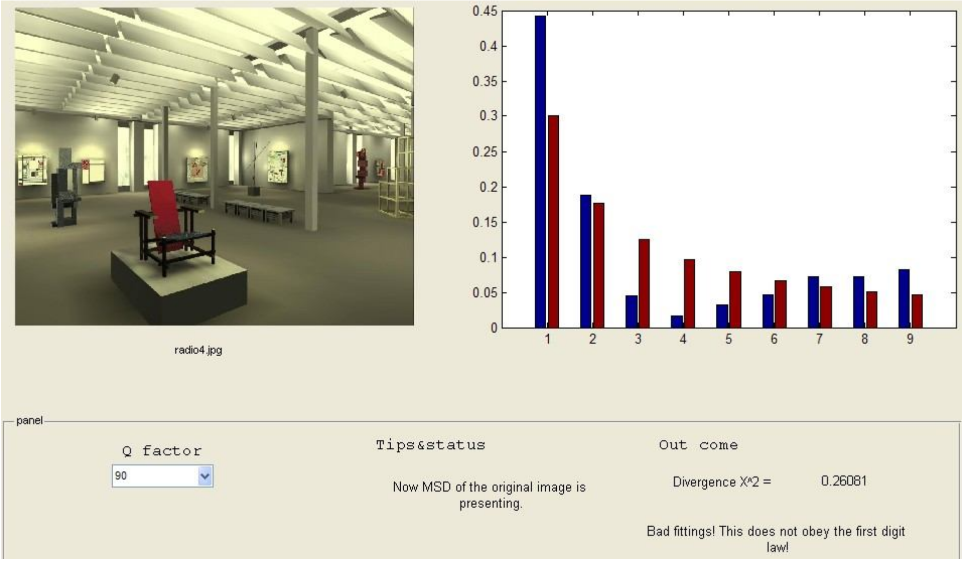

Figure 4: a synthetic example "radiosity factory"
blue bar is the FSD distribution of this image and red bar is Benford's distribution

For real images, our result is also negative. The only fact we can conclude is that images of natural scenery tend to agree more with Benford's law. An example is shown below.

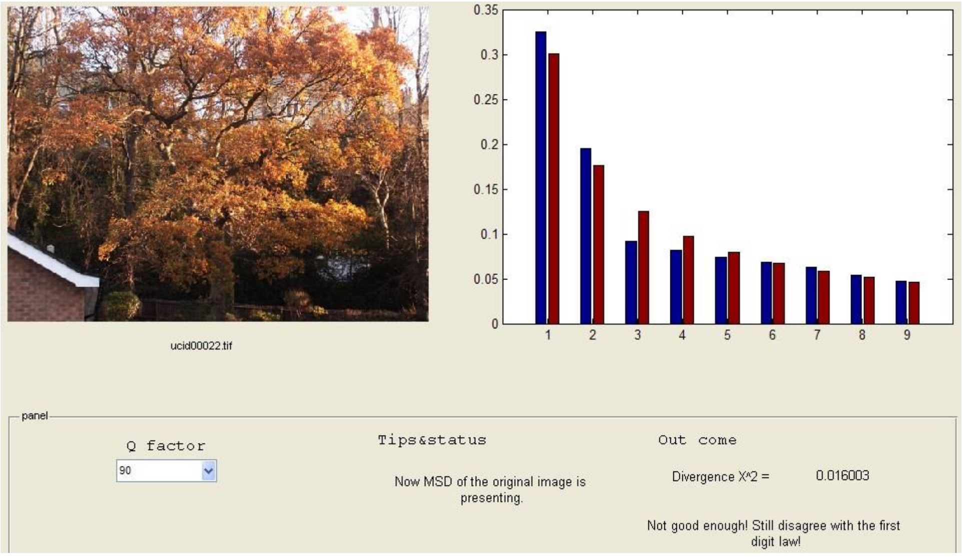

Figure 5: an example for natural scene obeying Benford's law

But we don't think Benford's law can be potential in something like natural scenery detection because there is a major adversary that will affect FSD distribution from time to time: repeating patterns. When a pattern repeats, it means some specific intensity values repeat. For natural scenes, there is always such a pattern that you can never avoid: the blue sky. An example is shown as follows.

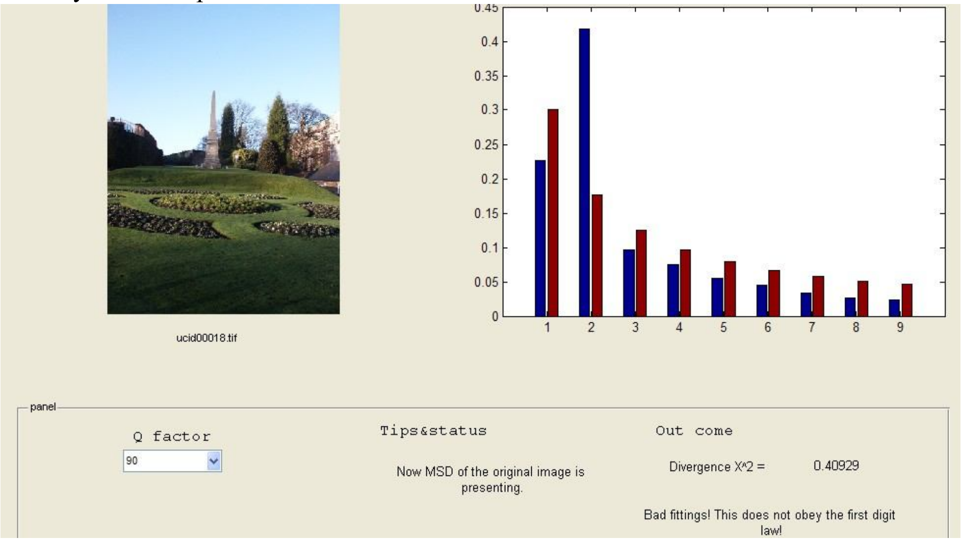

Figure 6: an example for natural scene with a blue-sky pattern

From the bar graph we can see that there is an obvious disagreement for MSD 2. This is for the sake of the blue sky pattern in the top half of the image. We can see this much more clearly using the 9-color representation of the image.

A 9-color representation is to re-print the image according to FSD of intensity levels. For example, if the intensity of a pixel is 178, we will print this pixel with red, since FSD of 178 is 1. If the intensity is 16, we print this pixel with red too, since FSD of 16 is also 1. If the intensity is 233, we will print this pixel with green, since the FSD of 233 is 2. If the FSD is 3, we print it with blue and etc. Every FSD from 1 to 9 is given a specific color, as we can see below.

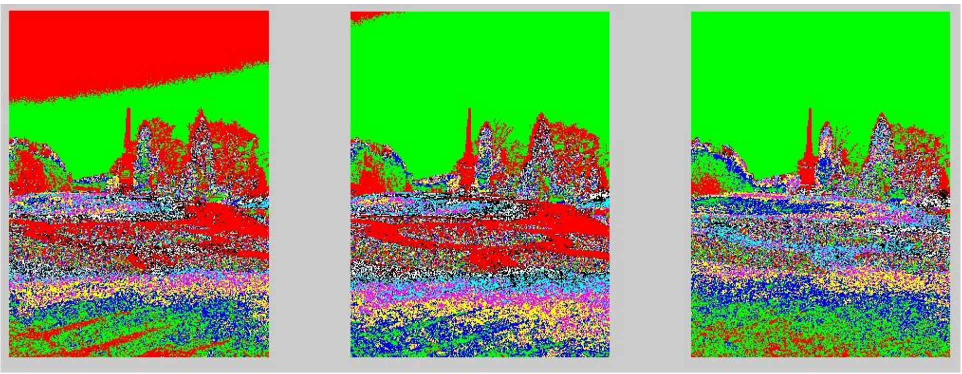

Figure 7: 9-color representation of figure 6
from left to right corresponding to R, G, B component

9-color representations can help us build the concept how repeating patterns can affect the intensity distributions. Recall from figure 6 that there is a strong disagreement with FSD 2, here in figure 7 FSD 2 is represented by the color of green.

There is one thing to emphasize. The target images of my trials are not “raw images” described in [6]. Please refer to section 1.3 for detailed reasons.

From the trials over real and synthetic images, our attitude towards pixel intensity values themselves obeying the Benford's law is negative. The reason behind the fact is that the range for intensities of 8 bit images is from 0 to 255, which is not only too narrow but also biased. In this period, the probability of FSD 1 and 2 are much higher than others. To get wider range, one way is to store an image with more bits. We may come back to this idea when image with higher depth (more than 8 bits) become widely used in the future. Another way to break the unpleasant range is to make image transformations, which we will discuss in the coming section.

3 BENFORD'S LAW ON JPEG COMPRESSION

3.1 GENERAL IDEA

Both [7] and [8] mention that the discrete cosine transformation (DCT) coefficients of digital images obey Benford's law. The definition of DCT is as follows: [9]

$$D(u,v) = \sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y) * \alpha(u) * \alpha(v) * \cos\frac{(2x+1)*u\pi}{2M} * \cos\frac{(2y+1)*v\pi}{2N}$$

$$\text{where } \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1,2 \ldots M-1 \end{cases}$$

The DCT coefficients refer to the value of DCT transform of an image. While the JPEG coefficients refer to value of the DCT coefficients after quantization, as we know DCT is one step of JPEG compression and quantization is another. [8]
Moreover, the distribution of their JPEG coefficients is also logarithm like and can be described with a generalized Benford's formula:

$$P(d) = N * \log\left(1 + \frac{1}{s + d^q}\right)$$

where d is FSD from 1 to 9 and N, s, q are generalization parameters.

Statistical experiments for more than 1000 digital images are conducted in [8] and models for different Q-factors are gained consequently. A graph from [8] shows the result.

Q-factor	Model Parameters			Goodness-of-fit (SSE)
	N	q	s	
100	1.456	1.47	0.0372	7.104e-06
90	1.255	1.563	-0.3784	5.255e-07
80	1.324	1.653	-0.3739	3.06838e-06
70	1.412	1.732	-0.337	5.36171e-06
60	1.501	1.813	-0.3025	6.11167e-06
50	1.579	1.882	-0.2725	6.05446e-06

Figure 8: resulting factors for N, s and q

The fact that JPEG coefficients obey Benford's law is good indeed. While, what really cheerful is that double JPEG compression will break this abeyance. [8] mentioned three potential usage of such a property: JPEG compression detection, JPEG Q factor

estimation and JPEG double compression detection. The three methods are quite similar with each other and only the first one is concerned in our target paper [1]. So, in our project we also focus on JPEG compression detection.
Here comes the method:
Step 1: compress the target image with JPEG factor 100.
Step 2: get the FSD of the JPEG coefficients of new image
Step 3: do the statistics and see whether it obeys the generalized Benford's law
If it obeys, this image has never been JPEG compressed before, or it was once JPEG compressed.
The reason is that if a JPEG compressed image goes through step 1, it is double compressed so that its JPEG coefficients will violate the law. On the contrary, if a non-compressed image goes through step 1, it is just JPEG compressed for once and its JPEG coefficients still hold generalized Benford's distribution.

3.2 OUR APPROACH

3.2.1 INPLEMENTAION METHOLOGY

In this project, we basically follow the method mentioned in the section 3.1 without significant change. The generalized model we adopted is that from [8], shown in figure 8. Here come our reasons: [8] uses plenty of samples (more than 1000) to do the statistics and [8] also clearly pointed out what image database was used to do this statistics. So that in our project we are able to use the same image database, the uncompressed colour image database (UCID) [10].
There are still some details to mention.
First, since DCT is processed with 8 by 8 sub images, so before calculation, we have to divide the image into 8 by 8 blocks. For images whose length or height can't divide exactly with 8, we need to pad the image with zeros to make sure both length and height can divide exactly.
Second, through DCT and quantization, a lot of zeros occur. Zero is not defined as a FSD in Benford's theory so that we should treat this as a special case. In this project, we develop a method called zero correction to make up.

$$\mathrm{Pz(d)} = \mathrm{P(d)} * (1 - \mathrm{f(0)})$$

where d is FSD from 1 to 9,
P(d) is the original model
Pz(d) is the corrected model
f(0) is the frequency missing from the FSD statistics (the value less than 1)

In such a way, we are able to exclude the zeros so that the generalized Benford's law is not affected.
Third, from experiments of detecting JPEG compression we found some improper aspects of using X^2 divergence (recall from section 2.1) as the standard to judge how 2 arrays fit with each other. We can imagine the situation that FSD 1 and 9 are with a same offset by 10%, their contributions to X^2 vary.
Then FSD 1 contributes $(0.3 * 0.1)^2/0.3 = 0.0003$ to X^2,
while FSD 9 contributes $(0.046 * 0.1)^2/0.046 = 0.000046$ to X^2.
Such cases occur pretty often: you can easily find out that one distribution is much worse fitting the generalized Benford's distribution than another while the X^2 divergence of the two are almost same value. This does not make sense. The idea of using X^2 divergence as the standard of quality of fitting came from [6] and they did not provide any supporting

of making this decision. This standard may not affect the result of [6] much while it does affect the result of JPEG compression detection.
In this project, we changed the standard into:

$$X^2 = \sum_{i=1}^{9} \frac{\left(f(d) - \log_{10}\left(1 + \frac{1}{d}\right)\right)^2}{\log_{10}\left(1 + \frac{1}{d}\right)^2}$$

where d is the FSD from 1 to 9 and f(d) is the frequency of FSD d by intensity in the target image.

In this way, we can guarantee that for a same percentage change in value, 2 FSD contribute in the same amount for X^2. The reason the square operation is used here is to enlarge the fitting differences in order to make sure the X^2 value for JPEG compressed and uncompressed images can be well separated. This change turns out to be a great success and the result will follow later.

3.2.2 RESULTS AND ANALYSIS

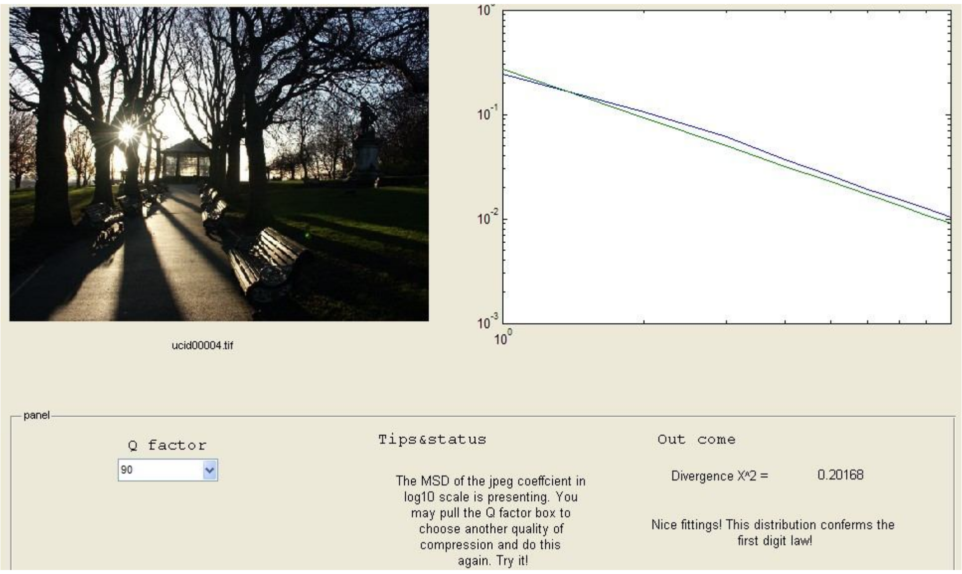

Figure 9: JPEG coefficients of an uncompressed image and generalized Benford's law
the green line belongs to the Benford's

Figure 9 shows how JPEG coefficients agree with generalized Benford's law. The image in figure 10 is a JPEG compressed version of figure 9. Though the result is not as good as uncompressed images, it is still acceptable and later we will show that comparing to double compressed images, this sort of accordance is pleasant enough.
There may be some misleading on the fact that the image in figure 9 is color image while that in figure 10 is gray level image. The truth is that when we calculate the JPEG coefficients of an image, the image is first converted into a gray level image. This is because JPEG compression for color images and gray level images is different, they have different quantization tables. And the most important reason for this is the model in [8] which we adopt in this project is based on gray level images.

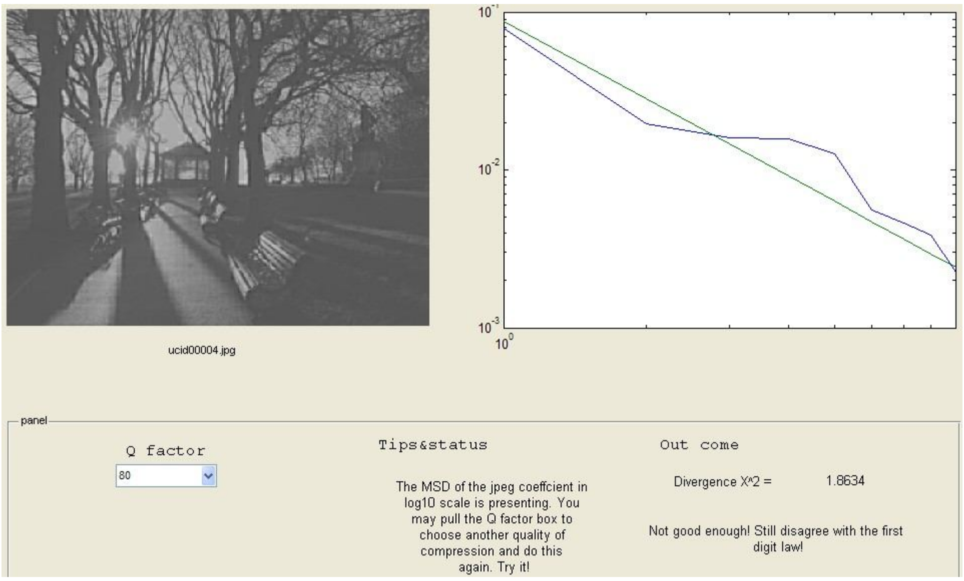

Figure 10: JPEG coefficients of a single JPEG compressed image and generalized Benford's law the green line belongs to the Benford's

As for the result of JPEG coefficients, almost all the uncompressed images and most of the single JPEG compressed images do quite well. But there are exceptions in the trial that for some JPEG images on some Q factor 90, the result is totally a mass. A probable reason is that the generalized Benford's model on Q factor 90 in [8] is not precise enough under our working condition. As mentioned in section 1.3, we do have limitations on the power of statistics to get a proper model and such exceptions remain unsolved.
Recall the X^2 standard we brought about in section 3.2.1, we are going to tell whether an image is JPEG compressed only by this value. 60 images are randomly chosen from the UCID database, reserve one copy and compress them into JPEG images. 50 TIF and JPG pairs are selected as the training group while the 10 pairs left are the testing group. The result of the training group is show:

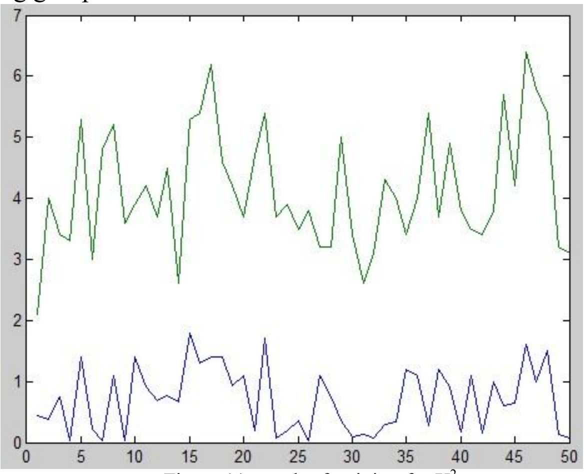

Figure 11: result of training for X^2
blue curve stands for TIF group (uncompressed) and green stands for JPEG group

There is pretty good separation of the compressed and uncompressed group. We choose the border value in this way:

$$\text{BorderValue} = \frac{(\text{Min}(\text{JPEG\_group}) - \text{Max}(\text{TIF\_group}))}{2}$$

The border value from figure 11 is 1.95. In figure 12 we can see that the green curve (representing JPEG compressed group) is over 1.95 and the blue curve (representing the uncompressed) is under 1.95 for all the values. This means our test on JPEG and TIF images is 100% correct: neither a JPEG image is recognized as uncompressed nor a TIF image is recognized as compressed. By the way, the result in [8] also claims a 100% correctness while using another standard.

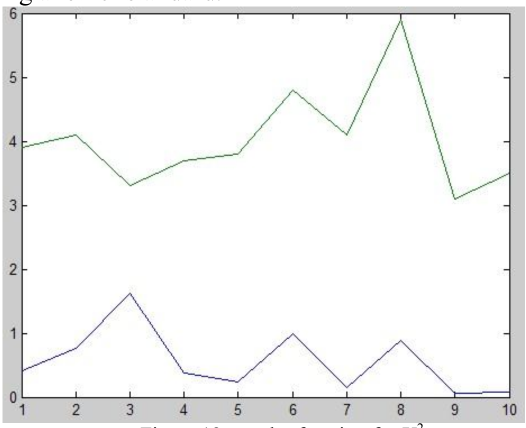

Figure 12: result of testing for X^2
blue curve stands for TIF group (uncompressed) and green stands for JPEG group

3.2.3 VULNERABILITY

In the target paper [1], a compensation model is put forward. The model can be seen in figure 13. Such model can distort the distribution of FSD with "histogram manipulation attack" [1].

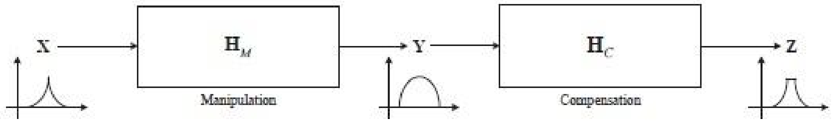

Figure 13: compensation block model [1]

Though the distribution of FSD in Y may not follow Benford's law, for example it represents the coefficients of a double compressed image, there're compensation methods that will guarantee the distribution of z still conform Benford's law. In this case, the compensation method can be histogram equalization, shown in figure 14.

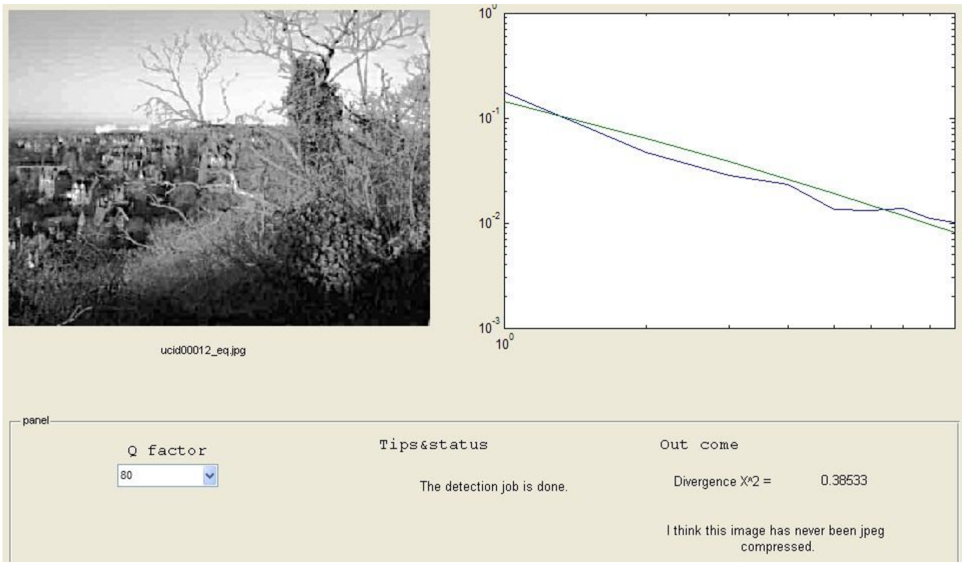

Figure 14: an example for the histogram equalization attack
the green line belongs to the Benford's

This image is got from histogram equalization of one member in the JPEG test group. Recall the fact that we made a 100% correct detection on that group. While, after histogram equalization, our program recognize this JPEG image as uncompressed again. This can be a major concern to applying Benford's law in detecting JPEG compression and the solution still remains open. Considering Benford's law is mainly about histogram statistics, manipulation directly on the histogram is really annoying in this aspect.

4 NOVEL CONTRIBUTIONS

In this project, we review most important works on applying Benford's law to image forensics and bring about the following ideas:

First, in section 2.1, we pointed out the wrong idea to apply Benford's law to tell cartoon image from real image.

Second, in section 2.2.2 we noticed how repeat pattern affects intensity FSD distribution, so that we have the conclusion that it is improper to apply Benford's law on intensity values.

Third, in section 3.2.1 we discovered the weakness of divergence X^2 as fitting quality standard and brought about an improved standard, which leads to the 100% success of the following test.

Fourth, in section 3.2.2 we put forward a concrete method to find a value of the improved X^2 that can tell whether an image was once compressed or is never compressed.

5 CONCLUSIONS

In this project, we have mainly handled two issues. For synthetic and real images, we find intensity values with 8-bit depth are not good target of applying Benford's law. For JPEG detection, we successfully build a system that can recognize the compressed and uncompressed images with high level of correctness. The only problem remained is its vulnerability to so called "histogram equalization attack". In the future, we will mainly focus on this vulnerability and also look for other potential leakage of this method.

REFERENCES

[1] J. Wang, B. Cha, S. Cho and J. Kuo, "Understanding Benford's law and its vulnerability in image forensics," in Proc. IEEE Int'l Conf. Multimedia and Expo, NY, USA, August 2009, pp. 1568-1571
[2] S. Newcomb, "Note on the frequency of the use of digits in natural numbers," Amer. Journal Math., Vol. 4, pp. 39–40, 1881.
[3] F. Benford, "The law of anomalous numbers," Proc. Amer. Phil. Soc., Vol. 78, pp. 551–572, 1938.
[4] C. Durtschi, W. Hillison and C. Pacini, "The effective use of Benford's law to assist in detecting fraud in accounting data," Journal of Forensic Accounting, vol.V, pp. 17-34, 2004.
[5] J. M. Jolion, "Images and benford's law," Journal of Mathematical Imaging and Vision, vol. 14, pp. 73–81, 2001.
[6] E. D. Acebo and M. Sbert, "Benford's law for natural and synthetic images," in Proc. Int'l Workshop Computational Aesthetics in Graphics, Visualization, and Imaging, Girona, Spain, May 2005, pp. 73–81.
[7] F. Perez-Gonzalez, G. Heileman, and C. T. Abdallah, "Benford's law in image processing," in Proc. IEEE Int'l Conf. Image Processing, San Antonio, TX, USA, September 2007, pp. 169–176.
[8] D. Fu, Y. Q. Shi, and W. Su, "A generalized benford's law for jpeg coefficients and its applications in image forensics," in Proc. SPIE Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, January 2007.
[9] R. Gonzalez and R. Woods, "Digital Image Processing 3rd edition," 2008 by Person Education, Inc, pp. 567-569
[10] G. Schaefer and M. Stich (2003) "UCID - An Uncompressed Colour Image Database," Technical Report, School of Computing and Mathematics, Nottingham Trent University, U.K.