applied sciences

MDPI

Article

A False Negative Study of the Steganalysis Tool Stegdetect

Benjamin Aziz [1], **Jeyong Jung** [2,*], **Julak Lee** [3] **and Yong-Tae Chun** [4]

[1] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK; benjamin.aziz@port.ac.uk
[2] Department of Police Science, College of Social Sciences, University of Ulsan, Ulsan 44610, Korea
[3] Department of Industrial Security, Chung-Ang University, Seoul 06974, Korea; julaklee71@cau.ac.kr
[4] Department of Security Management, Kyonggi University, Suwon 16227, Korea; chunyongtae@kyonggi.ac.kr
* Correspondence: pancon@ulsan.ac.kr

Received: 7 October 2020; Accepted: 14 November 2020; Published: 19 November 2020

Abstract: In this study, we evaluated one of the modern automated steganalysis tools, Stegdetect, to study its false negative rates when analysing a bulk of images. In so doing, we used JPHide method to embed a randomly generated messages into 2000 JPEG images. The aim of this study is to help digital forensics analysts during their investigations by means of providing an idea of the false negative rates of Stegdetect. This study found that (1) the false negative rates depended largely on the tool's sensitivity values, (2) the tool had a high false negative rate between the sensitivity values from 0.1 to 3.4 and (3) the best sensitivity value for detection of JPHide method was 6.2. It is therefore recommended that when analysing a huge bulk of images forensic analysts need to take into consideration sensitivity values to reduce the false negative rates of Stegdetect.

Keywords: steganograph; steganalysis; stegdetect; digital forensics

1. Introduction

In recent times, the rapid growth in computer technology has become core in our lives. The technological advancement such as Cloud computing, Internet of Things, and social media platforms has brought about efficiency, effectiveness, and convenience to both individual and organisational users. However, there is a downside to all this. There are more and more tools on the market today, and the tools created by advanced technology will become more and more difficult to control. Enterprises must increase investment and introduce new defense solutions to deal with them. This all has provided a new type of risk and threats. Due to an increasing reliance upon devices those users are exposed to various Cyber security risks [1]. In particular, individuals as well as organisations which essentially value information secrecy and privacy were greatly concerned about how to secure their data. Information hiding has become a pivotal characteristic of digital society. Against this backdrop, several methods such as steganography and cryptography with complex algorithms have been developed to secure information privacy [2]. Cryptography is intended to conceal the content of messages via data encryption or scrambling, but it cannot hide their existence [3]. In contrast, steganography hides the very existence of secret information while being communicated in cover media files [2–5]. If successful, it attracts no suspicion at all of the presence of such secret information from the point of view of an external observer. This is the main reason why steganography in recent times has received much attention amongst security research. In addtion, steganography has found other uses, such as copyright and e-document forging prevention [6], secret image transfers over Clouds [7] and protection of private health records [8], amongst others.

The problem of detecting hidden content was first formulated in a clear manner by Simmons [9], who modelled the problem as two prisoners attempting to communicated in a covert manner secret messages related to the plan of escape from the prison, whilst the warden would inspect every message communicated. If suspecting that hidden content was included in a message, the warden would then destroy the message and send the two prisoners into solitary confinement. This is known as the prisoners problem. In fact, there are a lot of real life applications of steganography in politics, diplomacy, and the military [3].

In hiding information using a steganographic procedure, one needs both an embedding algorithm, which takes as input a cover media file in which the secret data message will be embedded resulting in a stego-file. On the other end, one needs a detection algorithm that identifies the stego-file with an affirmation of the existence of the secret message and an extraction algorithm to extract the secret message from the stego-file. This method used in extracting and detecting steganographic activities in any stego-file is called steganalysis. However, similar to any other numerical analysis, steganalysis can have false results, which can be divided into false positive, where an image is clean but it is flagged by the analysis tool as being loaded with a secret message, and false negative, where an image is loaded with a secret message but it is flagged by the analysis tool as being clean.

Previously, in [10], we presented a false positive rate study of the well-known image analysis tool, StegDetect [11]. In this paper, we continue the false rates study of StegDetect by investigating the rate of false negative cases that the tool exhibits. Understanding the rate of false negatives is equally as important as it demonstrates the rate at which the tool fails in detecting hidden content, something that has implications for security and privacy.

The rest of the paper is organised as follows. In Section 2, we discuss related work in current literature. In Section 3, we give an overview of the methodology we used to conduct our research, and describe the datasets used for the analysis. In Section 4, we present the results of the analysis. Finally, in Section 5, we discuss some of the limitations of our experiments and in Section 6, we conclude the paper giving directions for future work.

2. Related Work

In terms of information hiding, steganography and watermarking are interconnected [12]. Although they share some technical traits, the largest difference is their purpose of use. The former is aimed at engaging in secret communication while the latter is for verifying the identity and authenticity of the owner. Ref. [12,13] argue that imperceptibility, robustness, and payload capacity are parameters of steganography. Compared to this, watermarking concerns the most whether it is robust in order to avoid watermarks being removed or replaced. These parameters can be referred to distinguish it from watermarking and cryptography as well as to compare various types of steganograpy techniques.

There are two groups of people who use steganographic techniques. A steganographer uses analysis tools to reassure whether a steganographic process has been successful, and thus the message is undetectable or unreadable [14]. On the opposite side, a stegoanalyst attempts to detect and read stego-messages. In either way, steganalysis involves two stages: (1) identifying the existence of steganographic messages and (2) reading the embedded message [15].

Various digital steganography methods have been developed in recent years. One commonality is that all methods is based on the fundamental concept that secret messages are embedded in a cover medium to create an output, a stego-file. There are a wide range of steganograpy techniques depending on a type of a cover medium (e.g., text, image, video and audio).

It has been an ongoing debate whether steganography is used by terrorists or criminals [16]. scanned a couple of million images and identified 20,000 suspicious images using 'Stegdetect' [11]. Although no hidden messages were identified in the research, we cannot categorically conclude that stegnography was not misused by malicious actors. Before making the conclusion, available tools should be examined whether they are reliable or not. Therefore it is of importance to check their reliability. However, there have been few research on this.

Detection of steganographic messages does not necessarily have to reveal the hidden content, but merely detecting their presence can carry significant implications in that this can draw unwanted attention from opposite parties. As such, the precision of the detection algorithm is one of its important attributes. This presents a crucial implication to digital forensic analysts. Ref. [17] defined digital forensic as the approved method used to preserve, collect, validate, identify, analyse, interpret evidence obtained for a digital investigation. In the digital communication era, any sort of criminal investigations are bound to involve digital devices. To establish facts in a court, digital data stored on devices such as computers and smartphones have to be investigated by a forensics analyst.

As malicious actors are equipped with state-of-the-art technologies, forensic analysts have tried to keep pace with them. According to [18], in digital crime there are different methods used by an analyst during their investigation. These methods throughout the investigation must be done in a forensically sound manner. Ref. [19] noted that an investigation is successful and acceptable if the evidence obtained from the original source is not altered in any way. Morever, to raise criminal arrests and convictions, forensic analysts need to ponder over how to reduce the false negative ratio of a tool. If the false negative ratio is high, this indicates that there is a high possibility that a stego-file is not detected, failing to weed out criminals. In this respect, this study aims to investigate the false negative rates of a steganalysis tool, Stegdetect, in order to examine whether this is a reliable tool for digital forensic analysts. This paper complements an earlier study on the false positive rates for Stegdetect carried out in [10].

There are many tools currently in the market for exposing hidden content in images. StegExpose [20,21] is an open-source Java-based steganalysis tool. The tool was designed primarily as a bulk analyser for lossless images, it works by classifying images as clean or stego images based on whether or not a pre-defined analysis threshold has been exceeded. In its standard operation, StegExpose is aimed at detecting different methods (nonlinear adaptive encoding, equidistribution and pseudorandom distribution) Least Significant Bit (LSB) originating from many tools including SilentEye [22], OpenPuff [23] and OpenStego [24]. StegSecret [25], like StegExpose, is also an open-source Java-based tool. It features a very simple graphical user interface, and is aimed at high performance bulk analysis, however this comes at the cost of lack of customisation; the tool is less configurable compared to StegExpose.

Other approaches have been combined in recent times to enhance steganalysis techniques. For example, the authors in [26] used convolutional neural networks to identify noise in different regions in an image and the relationship between noise in different sub-regions, and then classify images according to whether features of the image indicate the presence of hidden content or not. In [27], the authors use ensemble learning methods to construct effective steganalysis of images, which searches for colour correlativity between pixels and colour channels. Convolutional neural networks are used again in [28] as a method to embed hidden content in cover images. Most of these machine learning approaches suffer from lack of transparency as to how general the method can effectively either embed or analyse hidden content.

3. Methodology

The study has selected one of the automated steganalysis tools, Stegdetect [11] developed by Niels Provos. The purpose of the tool is to identify steganographic content by analysing JPEG images. It is able to detect several steganographic methods (F5 (header analysis), JPHide, invisble secret, outguest and camouflage) [29]. In analysing JPEG images it expresses the level of detection accuracy by appending stars (*, **, ***) to whichever steganographic method is detected. One star means the level of confidence in the detection of the specific steganographic method is low, two star means the level of confidence in the identification of steganographic method is quite good, and three star shows a high level of confidence in it. In this paper, we have used Stegdetect Windows version 0.4 which has an easy to use graphical interface. The tool's detection rate was based on the sensitivity value which is between 0.1 and 10.0. However, we have considered sensitivities of (0.1, 0.3, 0.5, 0.7, 10.0). Ref. [10] indicated that the sensitivity values affect the tool's false-negative ratio.

To achieve the purpose of the paper, we looked for a popular steganographic method that embeds data in JPEG image which is detectable by Stegdetect. JPHide [30] has both Windows and Linux version developed by A. Latham in 1999. In this paper we have chosen the Window version 0.5 with a user-friendly interface. Jphide uses least significant bit of the discrete cosine transform coefficient to hide data into any image with JPEG format. Meanwhile, according to [30], 5% insertion rate of data into an image will be very difficult to identify in the absence of the original image. Detection of the Jphide method is independent of the size of the message embedded into the image. This below shows the process we used in generating stego images.

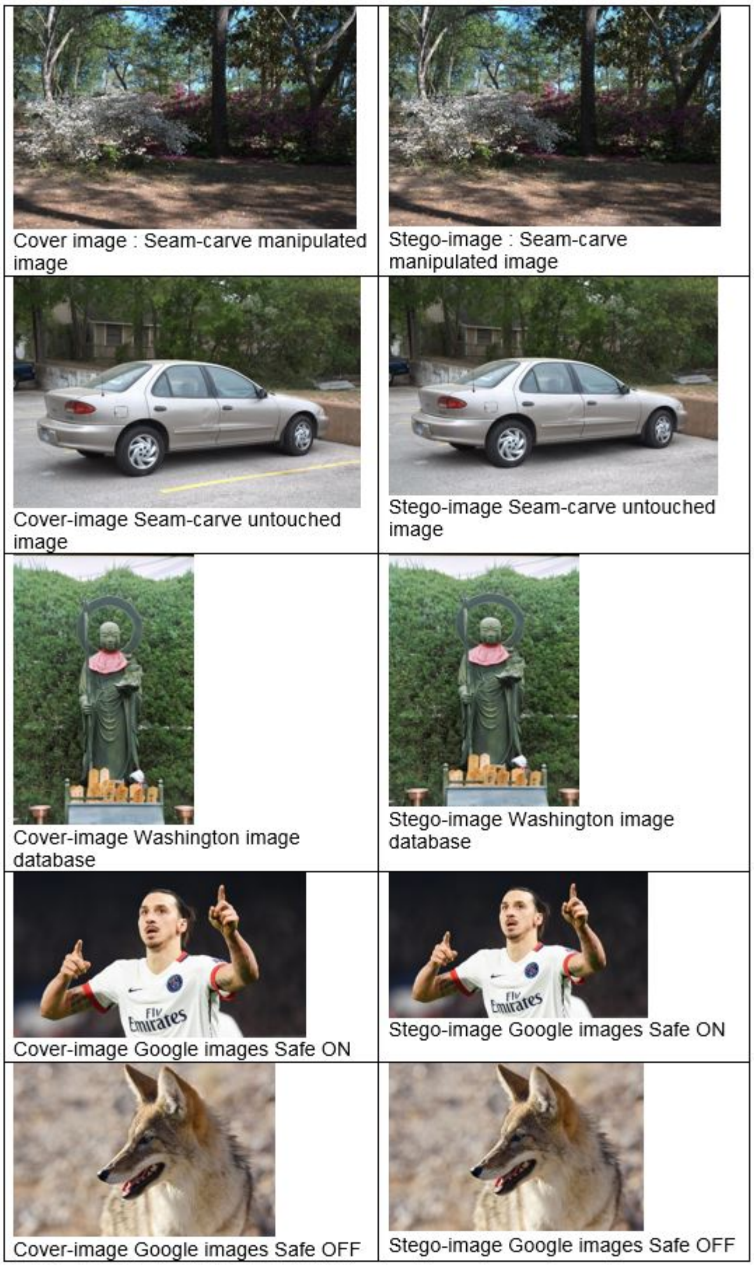
The Image Dataset

To help us study the false negative using Stegdetect to analyse steganography content automatically, the tool require images that contain embedded data. This research is based upon hiding bits of messages into 2000 JPEG images files using the embedding tool, JPHide. We searched and selected images from Sam Houston State University, University of Washington and Google image databases. Unfortunately, with our initial google images, there was a problem with the size of the images which affected the stego-object, which made statically modified after embedding obvious. To resolve the issue the following parameters were set for the downloading from google.

- Size of image: 2 MP (1600 × 1200);
- Colour of image: Any;
- Type of image: Any;
- Time: Any;
- Image file type: JPG files;
- Usage rights—not filtered by license.

However, we also activated both the search ON/OFF for the downloading of 300 images from Google to get the effect of this parameter on the outcome of the analysis. In addition to this, we also downloaded 700 clean JPEG images from University of Washington (Department of Computer science and Engineering) and 1000 images from Sam Houston State University image, 500 untouched and 500 manipulated with 75 bot quality.

Figure 1 shows samples of images hidden with messages using jphide. An automated utility, Stegdetect, which analyses bulk images with a hidden message with JPHide has been chosen to study its false negatives. For this purpose, we obtained JPHide version 0.5 as well as the Windows version of Stegdetect. We regulated the sensitivity value of Stegdetect against 2000 stego-object (obtained from different image databases such as google, Sam Houston State University and University of Washington). It was installed on a Windows 7 enterprise core i5 with 8 GB RAM.



Cover image : Seam-carve manipulated image

Stego-image : Seam-carve manipulated image

Cover-image Seam-carve untouched image

Stego-image Seam-carve untouched image

Cover-image Washington image database

Stego-image Washington image database

Cover-image Google images Safe ON

Stego-image Google images Safe ON

Cover-image Google images Safe OFF

Stego-image Google images Safe OFF

Figure 1. Sample results of the jphide method.

4. Results

All the results were analysed and interpreted in different phases deepening on the image dataset. Phase one analysed a total of 500 images manipulated by seam-carve from SAM Houston university image database, both at 75 quality before embedding using jphide with randomly generated bits. The table below gives a summary of the overall detection during the analysis.

We noted that detection of jphide method in the images was based on the changes in the sensitivity values. However, other algorithms detected by the tool are the circumstances in which stegdetect during the analysis identified other steganographic methods which during the embedding process we did not use. Table 1 shows that the highest ratio of detection with sensitivity results is 67.13% of the manipulated images by seam-carve which considering the level of the ratio is very high. Meanwhile, detection results for jphide were very low.

Table 1. Rate of sensitivity results from 500 images manipulated by seam-carve.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1–10.0	67.13%	2.00%	11.80%	14.71%	4.07%	0.29%

All results for false negative, jphide and other algorithm keep changing with change in sensitivity as shown in Table 2. The beginning of the analysis with low sensitivity value (0.1) the false negative ratio was very high (98%). However, a systematic drop was realised in the false negative ratio between sensitivity values 0.1 and 0.7, furthermore, the false negative ratio with sensitivity values 5.2–10.0 had a drastic drop as shown in Table 2 below. Here it becomes clear that the tool became more effective in detecting steganographic method used in embedding the secret messages.

Table 2. Results from manipulated images by seam-carve based on sensitivity values.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1	98.00%	2.00%	0.00%	0.00%	0.00%	0.00%
0.3	97.80%	2.00%	0.00%	0.00%	0.00%	0.20%
0.5	97.80%	2.00%	0.00%	0.00%	0.00%	0.20%
0.7	96.40%	2.00%	1.40%	0.00%	0.00%	0.20%
1.5	89.40%	2.00%	6.20%	2.00%	0.00%	0.40%
3.4	87.20%	2.00%	2.20%	0.00%	8.20%	0.40%
5.2	17.60%	2.00%	69.60%	2.20%	8.20%	0.40%
7.3	11.40%	2.00%	17.40%	59.00%	9.30%	0.40%
10.0	8.60%	2.00%	9.40%	69.20%	10.40%	0.40%

As shown in Figure 2 above, between sensitivity values 0.1 and 0.5 there were no changes in the results for jphide. Meanwhile, detection of jphide increased substantially between 0.7 and 10.0 with their related confidence levels (*, **, ***). Between 0.1 and 0.5 jphide (*) was stable until it got to the range 0.7–3.4 when there was fluctuation in the detection ratio, it then had a sharp increased with 5.2 sensitivity, after which it experienced another sharp decrease between (7.3 and 10.0). For jphide (**) between 1.5 and 10.0 there was a constant increase except with sensitivity of 3.4 which experience some drop. However, jphide (***) maintained the increasing of its ratio.

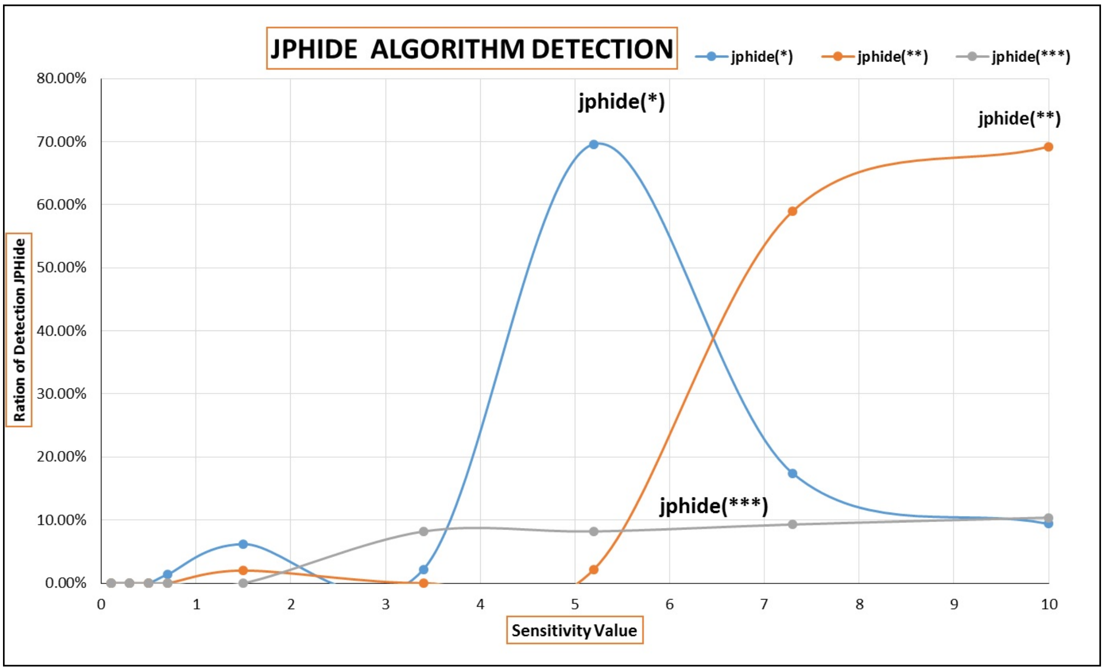


Figure 2. Changes in the jphide rate with different sensitivities for seam carve manipulated images.

As per the analysis above, the level of confidence in detection by stegdetect is directly proportional to the sensitivity values. Meaning, the higher the sensitivity value the higher the confidence in detecting jphide. Furthermore, the high increase of confidence in detecting jphide was between (3.4 and 10.0). During the analysis, stegdetect detect other steganographic methods in the images other than jphide which we used. Figure 3 below shows that 0.2% of the detection was for other algorithms between 0.3 and 0.7 sensitivity which stegdetect claims was used in embedding secret messages in those images. Meanwhile, the percentage of other algorithm detected increased to 0.4% between (1.5 and 10.0). Finally, the images from the database were already manipulated before jphide method was used to embed the messages. It is therefore possible that the images were manipulated using any of the algorithms detected during the analysis.

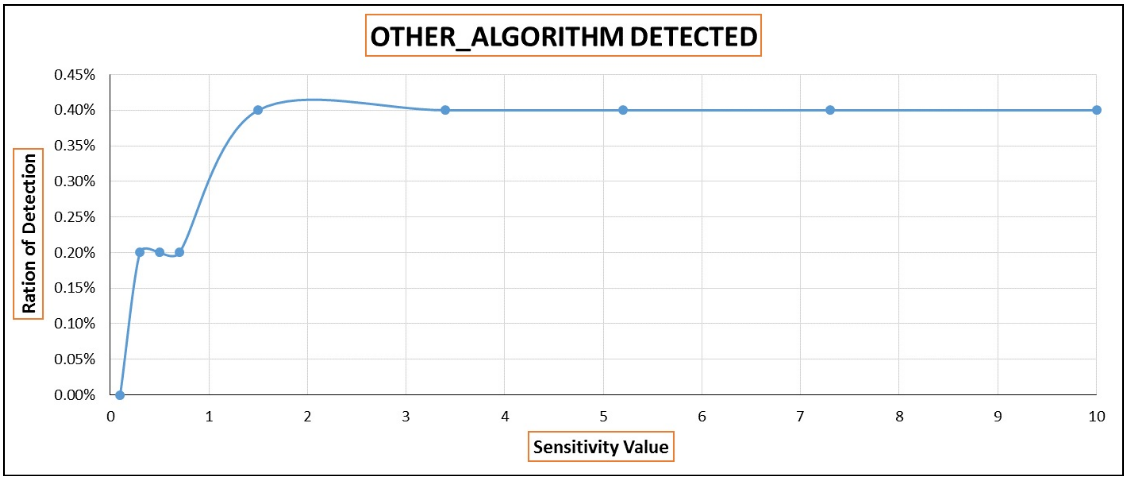


Figure 3. Changes in other algorithms detected with different sensitivities.

Phase two of the analysis was focused on 500 Seam-carve untouched (clean) images from SAM Houston university image database which were embedded with a secret message using jphide. Compared to the detection results of the manipoulated images, there was slight incease in the detection for the false negative ratio, skipped (false positive likely) and jphide (*) while other algorithms and jphide (**, ***) experience a slight decreased with different sensitivity as shown in Table 3 below.

Table 3. Rate of sensitivity results from 500 seam-carve untouched images.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1–10.0	67.78%	2.40%	11.91%	14.09%	3.62%	0.20%

As show in Table 3 above, 67.78% of the overall detection was false negative which is very high. However, with an increase in sensitivity, the detection ratio for false negative, jphide and other algorithm all changed. Furthermore, as shown in Table 4 below, there was a significant increase in the confidence detection of steganographic method jphide with changes in sensitivity values. We observe slight changes in the detection between the manipulated and the untouched Seam-carving images. Detection of jphide in the untouched images embedded with bits of messages started with 0.5 sensitivity while detection for jphide in the manipulated images started with 0.7 sensitivity, after which there was a continuous increase in the confidence in detection of jphide method.

Table 4. Results of 500 images from seam carve untouched images with different sensitivity values.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1	97.60%	2.40%	0.00%	0.00%	0.00%	0.00%
0.3	97.60%	2.40%	0.00%	0.00%	0.00%	0.00%
0.5	97.40%	2.40%	0.20%	0.00%	0.00%	0.00%
0.7	96.20%	2.40%	1.40%	0.00%	0.00%	0.00%
1.5	90.80%	2.40%	4.40%	2.00%	0.20%	0.20%
3.4	87.20%	2.40%	3.40%	0.20%	6.40%	0.40%
5.2	20.60%	2.40%	66.60%	3.40%	6.60%	0.40%
7.3	12.60%	2.40%	20.20%	55.00%	9.40%	0.40%
10.0	10.00%	2.40%	11.00%	66.20%	10.00%	0.40%

The false negative results for untouched seam-carving images at the beginning were high 97.60% as shown in Figure 4 with 0.1 sensitivity value, this result is not different from the manipulated images, however there was slight decrease between 0.1 and 3.4, then there was massive fall in the false negative between 5.2 and 10.0 with increase in sensitivity value.

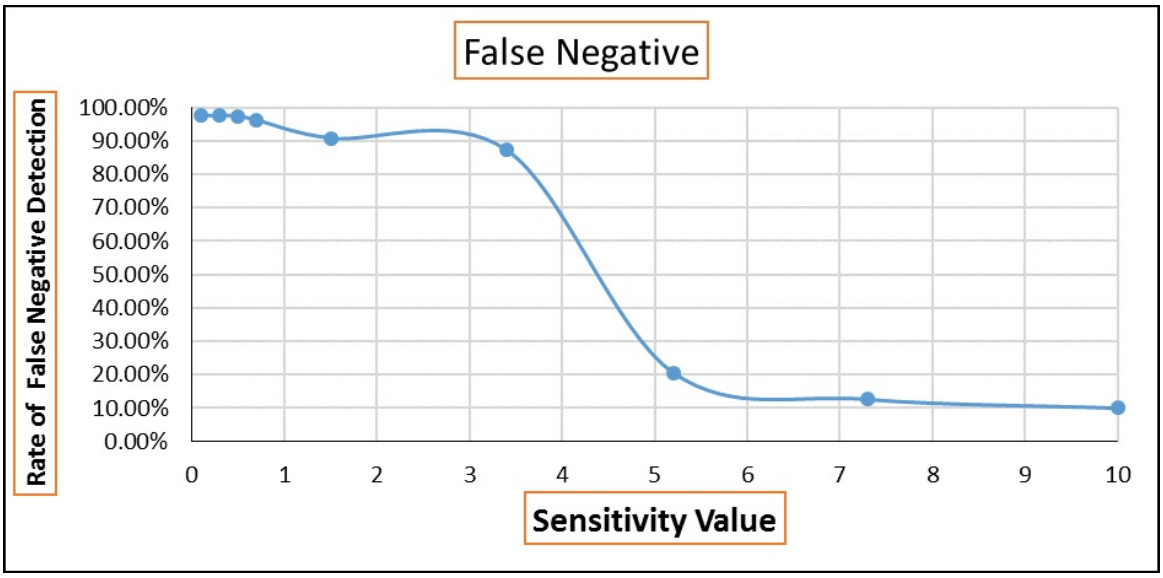


Figure 4. Overall false negative rate seam-carving untouched images with different sensitivity values.

The detection results for jphide (*, **, ***) between 0.5 and 3.4 was very marginal until the sensitivity was increased to 5.2 when jphide (*) had sharp increase meanwhile, with continuous increase in the sensitivity value between 7.3 and 10.0 the detection of jphide (*) experience a continuous decline, at the same time between 5.2 and 10.0 the level of confidence in detecting jphide (**) had a continuous increase while jphide (***) maintained its steady increase as shown in Figure 5 below.

Figure 6 shows that there was no effect of the sensitivity between 0.1 and 0.7 on the results for other algorithm detected, then between 1.5 and 10.0 there was a minor increase in the detection of other algorithms by the tool. However, between 3.4 and 10.0 the tool (stegdetect) maintain a constant detection ratio for other algorithms.

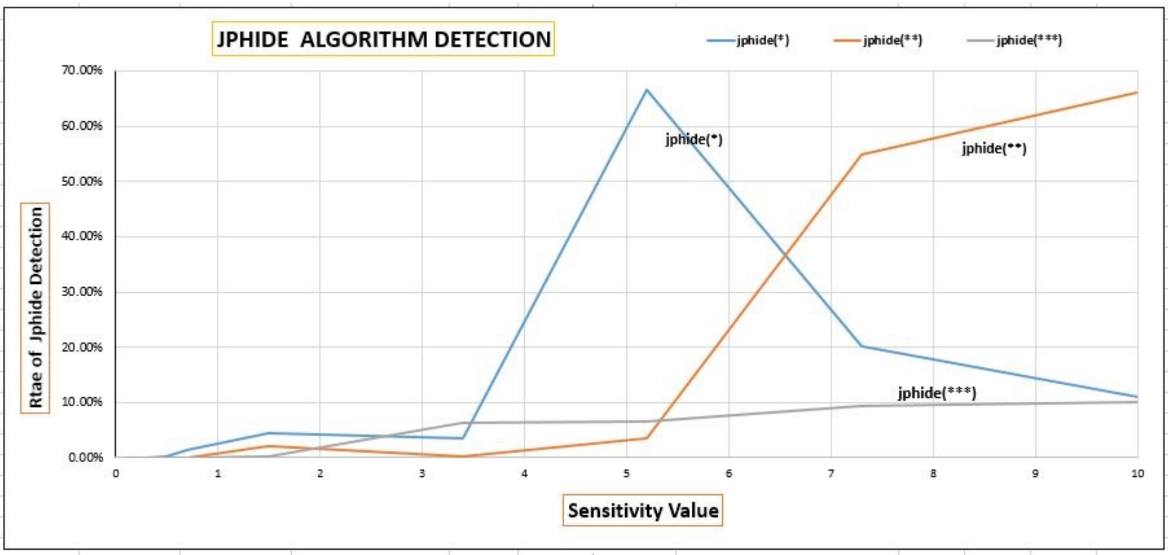


Figure 5. Changes in the jphide rate with different sensitivities for seam carve untouched images.

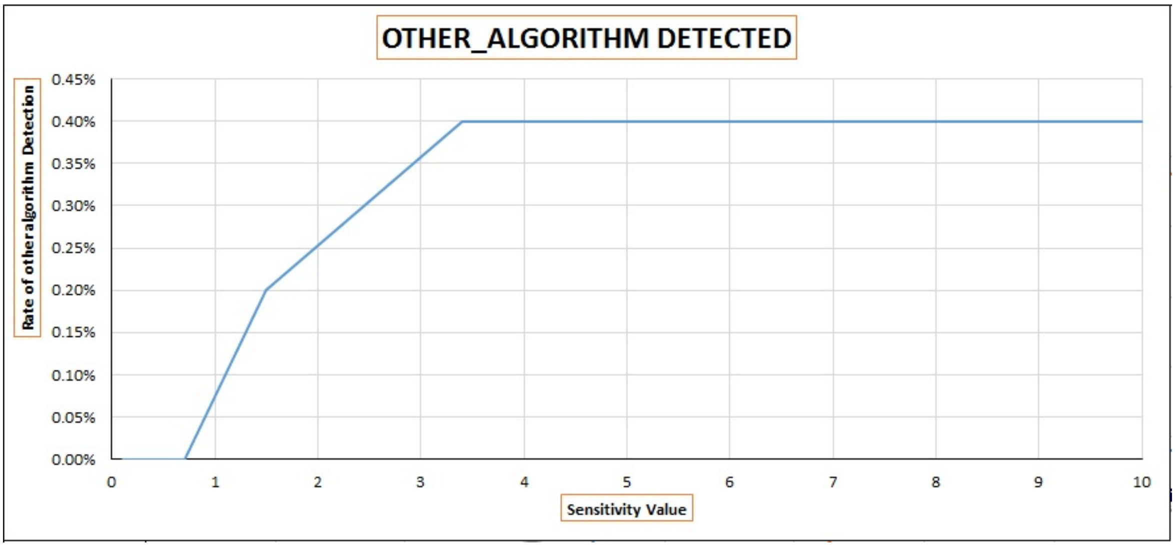


Figure 6. Changes in other algorithms detected with different sensitivities.

Phase three of the experiment analysis 700 images from the Department of Computer and Engineering, university of Washington image database. Each image was embedded with a different generated bits of a message using jphide. During the analysis of the 700 stego-images, 3.71% resulted in error between 0.1 and 10.0 sensitivity which compared to the volume of the images involved is quite small. In the case of the error images, stegdetect could not analysis because of the following stated reason. 1. Bogus DQT index 6, 2. Invalid JPEG file structure: SOS before SOF, and the last 3. Quantization Table 0 × 00 and 0 × 01 was not defined. The error rate can be seen in Table 5 below. It wealth noting that all the images analysed were subject to frequency counts. In other words, the analysis of any detection (false negative or jphide) was added to find the highest detection ratio (i.e., a number of times a specific detection occur). After which they were quantified as shown in Table 5 below.

Table 5. Results of 700 images from Washington University image database with different sensitivity values.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1	78.29%	18.00%	0.00%	0.00%	0.00%	3.71%
0.3	77.14%	18.00%	1.00%	0.14%	0.00%	3.71%
0.5	77.00%	18.00%	0.57%	0.57%	0.14%	3.71%
0.7	75.86%	18.00%	1.29%	0.43%	0.71%	3.71%
1.5	71.43%	18.00%	4.14%	1.43%	1.29%	3.71%
3.4	43.43%	18.00%	27.86%	3.00%	4.00%	3.71%
5.2	20.71%	18.00%	23.43%	27.14%	7.00%	3.71%
7.3	18.00%	18.00%	8.43%	24.43%	27.43%	3.71%
10.0	17.71%	17.57%	3.29%	22.86%	34.86%	3.71%

The false negative result between 0.1 and 1.5 sensitivity was 78.29% which is a bit high, then when the sensitivity was change between 3.4 and 10.0 there was a sharp drop and a continuous decline until it reaches 17.71%. Moreover, comparing the false negative results of the previous seam-carving images (both manipulated and untouched images) we realised that with the previous experiment between 0.1 and 3.4 they had a significantly higher false negative ratio which was 80% to 98% before it had a sharp decline. Though the images from Washington University seem to have had a low false negative ratio compared to the seam-carving images, they all seem to have had a sharp decrease at some point, then when the sensitivity was set to 5.2 it maintained a slow but steady decrease as shown in Figure 7 below.

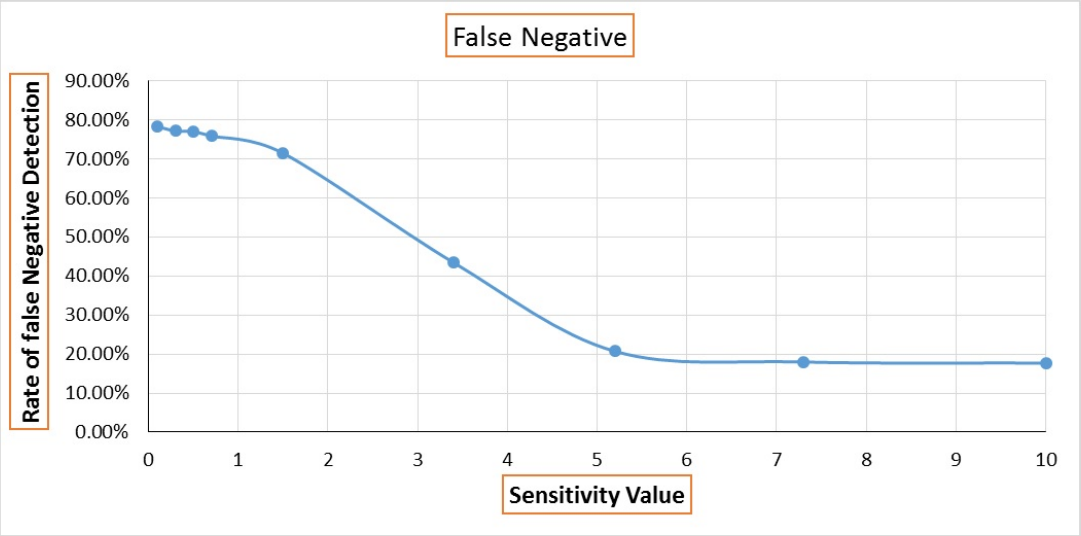


Figure 7. Overall false negative rate of Washington university image database with different sensitivity values.

The detection results of jphide (*, **, ***) started between sensitivity values (0.3 and 1.5), then there was a significant increase in the detection between (3.4 and 10.0). The detection for jphide (*) was consistently increasing until 3.4–5.4 sensitivity when there was a height jump, meanwhile, between 7.3 and 10.0 sensitivity the detection for jphide (*) started to decrease and jphide (**) also had similar result like in the case of jphide (*) where it experience a stable increase then a slight decrease with 0.7 sensitivity before it started to increase in detection again between 1.5 and 10.0 sensitivity. Finally, jphide (***) maintain a continuous steady increase in detection between 0.5 and 5.2 then a height jump in the detection between 7.3 and 10.0 as shown in Figure 8 graph below.

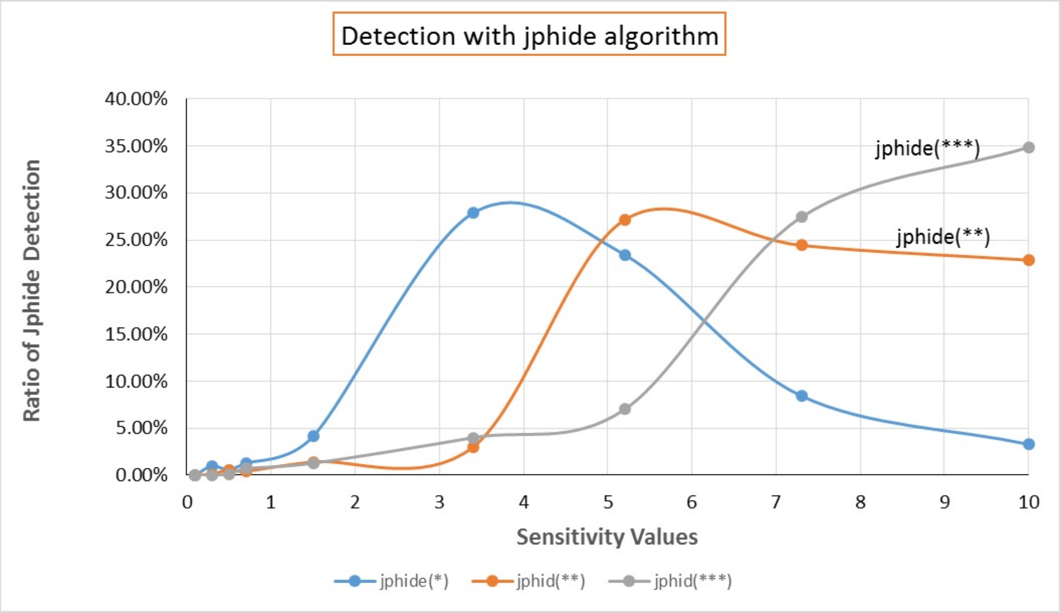


Figure 8. Changes in the jphide rate with different sensitivities for Washington university image database.

Phase four analysis 300 image from google (SAFE ON/OFF), the results for skipped false negative likely, and errors were changed with different sensitivity, other algorithms detection was constant between 0.7 and 10.0. The detection results for false negative was still between (0.1 and 3.4). However, with (5.2–10.0) sensitivity just like the previous experiment, there was a significant fall in the false negative ratio as shown in Figure 9 graph below.

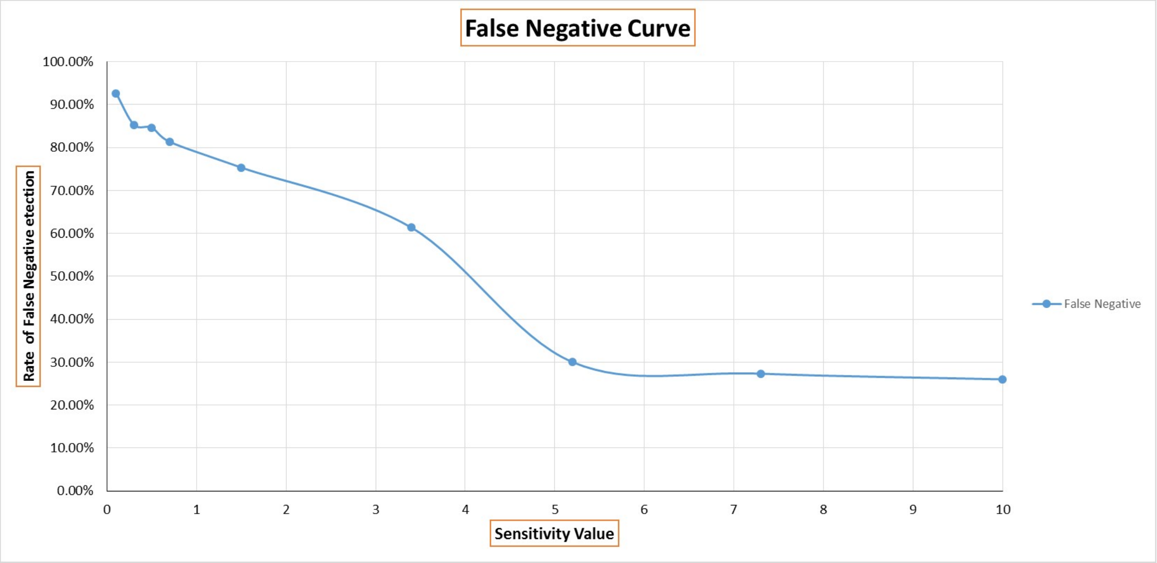


Figure 9. Overall false negative rate of google image database (SAFE ON) with different sensitivity values.

Again comparing the results with the other experiments conducted earlier the confidence level in jphide detection ratio keep change with changes in the sensitivity value as shown in Figure 10 below. For this set of images jphide (*) had similar results we acquired from the images from seam carve and Washington university image databases, respectively. For all those experiment there was sharp increase in detection ratio and then another sharp decline in detection for jphide (*) with different sensitivity values. However, jphide (** and ***) had a different results from all the other experiments performed, for this experiment we realised a continuous increment in the detection ratio for both jphide (** and ***) with increasing sensitivity value as shown in Figure 10 below.

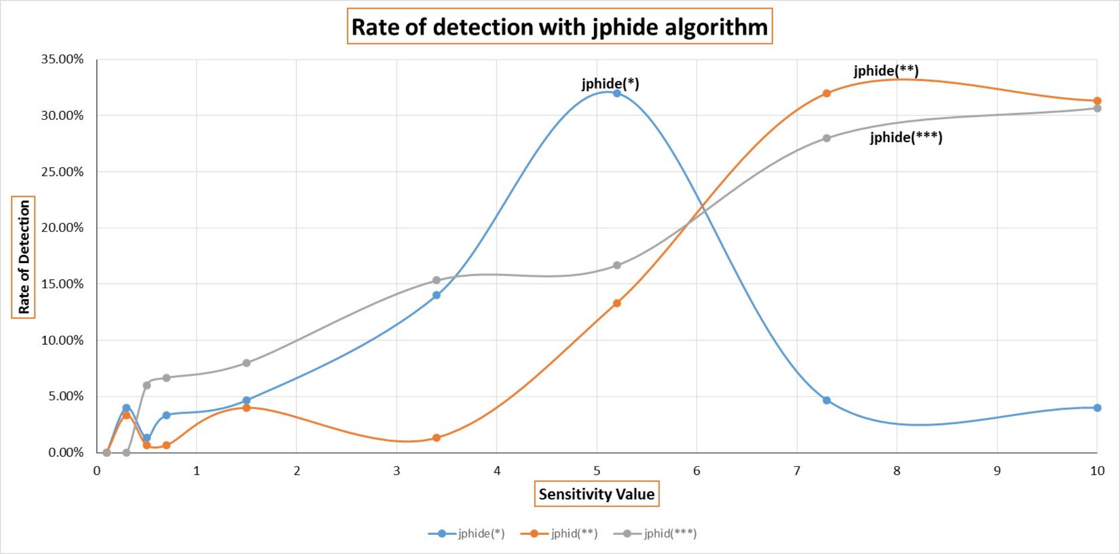


Figure 10. Changes in the jphide rate with different sensitivities for google image database (SAFE ON).

We realised that there were different results especially for the jphide and false negative from all previous experiments. For instance, between (0.5 and 10.0) sensitivity there was continuous and significantly higher confidence in detecting jphide (***) from the previous experiments. However, Google safe (OFF) as shown in Table 6 below gives slightly different results considering the confidence in detecting jphide (***).

Table 6. Results of 150 images from google image database (SAFE OFF) with different sensitivity values.

Sensitivity Value	False Negative	Likely False Positive (Skipped)	jphide(*)	jphide(**)	jphide(***)	OTHER _ALG
0.1	90.67%	4.67%	0.00%	0.00%	0.00%	0.67%
0.3	90.67%	4.67%	0.00%	0.00%	0.00%	0.67%
0.5	89.33%	4.67%	1.33%	0.00%	0.00%	0.67%
0.7	87.33%	4.67%	3.33%	0.00%	0.00%	0.67%
1.5	84.67%	4.67%	2.67%	2.00%	1.33%	0.67%
3.4	74.67%	4.67%	9.33%	1.33%	5.33%	0.67%
5.2	33.33%	4.67%	41.33%	9.33%	6.67%	0.67%
7.3	30.00%	4.67%	7.33%	40.00%	13.33%	0.67%
10.0	29.33%	4.67%	4.00%	41.33%	16.00%	0.67%

The highest was again at the beginning of the experiment was the false negative ratio 90.67%, which is much different from the previous experiment, and had a further drop with increasing sensitivity. Figure 11 shows that the curve is not different from the previous experiment.

The detection results for jphide (***) from google safe (OFF) is different from the results from the safe (NO) results. With the safe (off) detection of jphide (***) started and continuous to increase between (1.5 and 10), but detection for jphide (***) in safe (ON) started between (0.5 and 10.0), and jphide (*) continuous to increase in detection between 0.5 and 5.2 before the detection started to fall has sensitivity increase between 7.3 and 10.0. Finally, jphide (**) results at 1.5–5.2 sensitivity there was a steady increase before a quick and continues increase between 7.3 and 10.0. The two image groups were compared to show how the properties of images can affect the detection of Jphide method in images. Figure 12 gives a graphical representation of the jphide results.

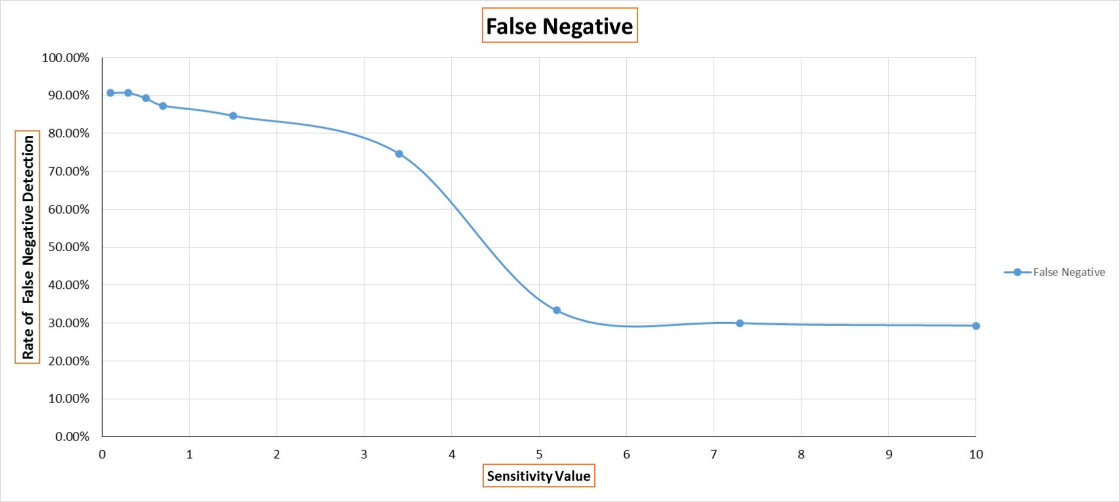


Figure 11. Overall false negative rate of google image database (SAFE OFF) with different sensitivity values.

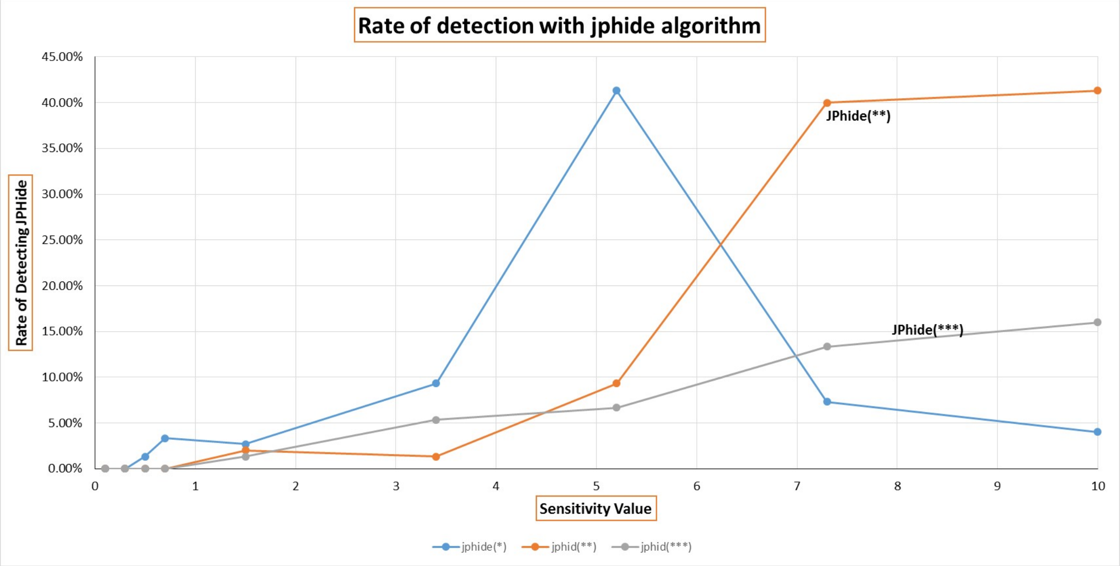


Figure 12. Changes in the jphide rate with different sensitivities for google image database (SAFE OFF).

The final phase, analysis the overall false negative ratio of the tool, this is to help forensic analyst during an investigation by providing accurate statistics of stegdetect false negative ratio, because in the court of law the forensic analyst must prove beyond every reasonable doubt that the results of the tool can be relied upon as evidence. This analysis was done using the results from all the different image databases, note that all the images had different properties, because there were some that had been manipulated with dotted at a quality of 75 and there were those that were untouched. The overall false negative results for all the different images it is very high between (0.1 and 3.4) but had a quick fall between (5.2 and 10.0), and as the false negative results drop the confidence in detecting jphide (*, **, ***) increases, this is an important information for the analyst investigating images from different sources. Especially noting that false negative ratio of the tool and how the higher the sensitivity between (5.2 and 10.0) influences the results of bulk images under investigation. Table 7 shows all the false negative values for the different datasets used.

Figure 13 below present the overall false negative ratio which was very high, but there is very important information about the graph the forensic analyst need to know. We set our acceptable false negative ratio to be 21%, which intersect with the mean of all the false negative at some point on the sensitivity. All the different image at 5.2 sensitivity had a quick fall in the false negative ratio but with a continuous increase in the sensitivity gave a stable and slow decline in the false negative ratio.

Note, with our acceptable 21% false negative its correspondent sensitivity is 6.2. This will inform the analyst on the kind of sensitivity they can use depending on their acceptable false negative ratio during an investigation. During the analysis, the following observations were noted,

I. Between (0.1 and 5.0) the tool seem not to be very sensitivity in detecting steganographic method in images.

II. Between (6.2 and 10.0) the analyst is likely to get a more accurate and a more reliable, which give a low false negative result. In this case, there is a likelihood that the tool runs slow because its become very sensitive in detecting steganographic methods in JPEG images.

Table 7. Overall false negative rates of ALL the different image databases with different sensitivity values.

Sensitivity Value	Seam-carve False Negative	Untouched Images False Negative	WU Images False Negative	Google (SAFE ON) False Negative	Google (SAFE OFF) False Negative
0.1	98.00%	97.60%	78.29%	92.67%	90.67%
0.3	97.80%	97.60%	77.14%	85.33%	90.67%
0.5	97.80%	97.40%	77.00%	84.67%	89.33%
0.7	96.40%	96.20%	75.86%	81.33%	87.33%
1.5	89.40%	90.80%	71.43%	75.33%	84.67%
3.4	87.20%	87.20%	43.43%	61.33%	74.67%
5.2	17.60%	20.60%	20.71%	30.00%	33.33%
7.3	11.40%	12.60%	18.00%	27.33%	30.00%
10.0	8.60%	10.00%	17.71%	26.00%	29.33%

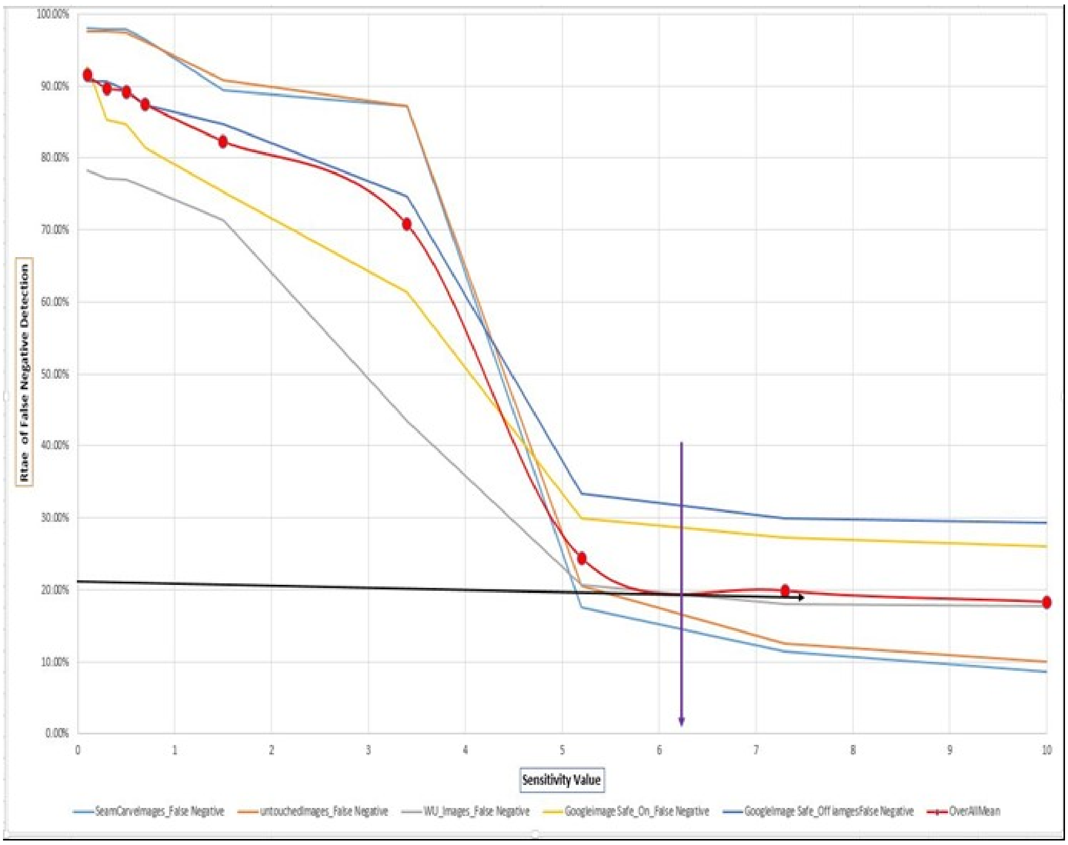


Figure 13. Overall false negative ratio from all different image databases.

5. Limitations of the Experiment

The research we conducted in this paper was constrained by a number of limitations. We attempt below to summarise these:

- In terms of the algorithms behind StegDetect, these are not known as they are not published by the developers of the tool, therefore, our understanding of the behaviour of the tool remains at the level of black box testing. We consider this to a certain extent sufficient from the perspective of the end users community. If such algorithms were made public, we would be able to use the results of this work as well as the previous work [10] to make recommendations on how to improve the algorithms.
- The initial plan was to collect a large sample size of images, but the research started to run into problems when collecting images from google images database. In steganography process, to get a good quality stego cover, there are some qualities that the cover medium needs to meet. First is capacity, which refers to the amount of hidden data it can contain. Secondly is security, which makes it unable for any intruder access. Lastly is its robustness, the ability or the amount of distortion its can withstand. However, the initial images from google after embedding the secret message had a notable modification of the stego cover.
- Furthermore, we wanted to compare the detection ratio of the different methods stegdetect claims to detect, so we used jsteg and F5, but could not give any informative results to analysis as shown in the graph below. Reddy (2007) noted that is difficult for stegdetect to detect F5 method.

6. Conclusions

The main purpose of steganography is to hide secret data during communication to avoid intruders from discovering the hidden message within the stego image without the right permission. Meanwhile, [29] stated that steganalysis is not as straight forward as steganography, this is a disadvantage to the forensic analyst who will be trying to detect hidden data in stego images. However, in steganalysis, only a few can automatically analyse a bulk of stego images at the same. To check the accuracy of a steganalysis tool which will help forensic analyst, our research exam the false negative rate of Stegdetect one of the popular steganalysis tools in the market. In our experimental results, we observed that when the sensitivity values were sets between (0.3 and 0.7) for all the various image databases jphide started to be detected. It could be concluded that the different sensitivity value range affects the detection rate for this method (jphide). The main purpose of the study was about the false negative rate of the tool, we concluded that the tool has a high false negative rate, especially between (0.1 and 3.4) sensitivity. We recommend that the best sensitivity value for detection of jphide method should be 6.2. This detection sensitivity value is very important for the forensic analyst. Because the false negative ratio had a deep sharp fall from this point onwards. However, we recommended that forensic analyst using stegdetect need to take into consideration the sensitivity values with the high false negative value when analysing a huge bulk of images. Moreover, based on our analysis of the tool, we observed and proposed a reference point of the sensitivity value with its related quantified false negative rate based on the mean of all the various image databases. Overall, the mean proposed can act as a baseline which will help the forensic analyst in making a much better decision during their investigation proceedings. However, based on the mean of all the false negatives of the tool, it is also argued that it has a high probability of false negative ratio between 0 and 10% even if the sensitive value is set beyond our recommended.

In conclusion, the fight between steganalysis methods and steganographic methods will ever continue. As more sophisticated steganographic algorithms are developed every day, a more powerful and sophisticated universals algorithms will also be required in detecting these steganography methods. This will be a more challenging but exciting research area in the near future. Currently, most steganalysis tools are very good in detecting specific steganographic methods. Example, Stegdetect which is an automated steganalysis tool is very good and effective in detecting content hidden in JPEG image formats than any other image format like Tiff, PNG and Gif. However, it is also more effective in detecting specific steganographic methods such as jphide, F5, invisible secret, jsteg and outguess than any other steganographic method. In this view, a future research should be conducted to consider a universal steganalysis tool. With current advancement in technologies for secure communication and its issues of privacy for individual users, a further research need to be considered to find the effect steganalysis tools will have on security protocols.

Additionally, we also plan to conduct a more complete comparative study of the available steganalysis tools, in order to obtain a more general understanding of what can be achieved and the degree of accuracy.

Author Contributions: B.A. contributed through the outlining of the idea underlying the paper and the implementation of the analysis of the tool. J.J. contributed through applying the methodology and writing—the original draft. J.L. made a contribution through writing—reviewing and editing and supervising. Y.-T.C. contributed through administrating the project and supporting resources. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the 2020 Research Fund of University of Ulsan.

Acknowledgments: The authors acknowledge the comments and feedback they received from the anonymous reviewers, all of which helped to improve the presentation of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moradoff, N. Biometrics: Proliferation and constraints to emerging and new technologies. *Secur. J.* **2010**, *23*, 276–298. [CrossRef]
2. Li, Y.; Xiong, C.; Han, X.; Xiang, R.; He, F.; Du, H. Image steganography using cosine transform with large-scale multimedia applications. *Multimed. Tools Appl.* **2018**, *81*, 161.
3. Anderson, R.J.; Petitcolas, F.A.P. On the Limits of Steganography. *IEEE J. Sel. Areas Commun.* **1998**, *16*, 474–481. [CrossRef]
4. Ghebleh, M.; Kanso, A. A robust chaotic algorithm for digital image steganography. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1898–1907. [CrossRef]
5. Chaeikar, S.S.; Zamani, M.; Manaf, A.B.A.; Zeki, A.M. PSW statistical LSB image steganalysis. *Multimed. Tools Appl.* **2018**, *77*, 805–835. [CrossRef]
6. Channalli, S.; Jadhav, A. Steganography an art of hiding data. *arXiv* **2009**, arXiv:0912.2319.
7. Abd El-Latif, A.A.; Abd-El-Atty, B.; Elseuofi, S.; Khalifa, H.S.; Alghamdi, A.S.; Polat, K.; Amin, M. Secret images transfer in cloud system based on investigating quantum walks in steganography approaches. *Phys. A Stat. Mech. Appl.* **2020**, *541*, 123687. [CrossRef]
8. Hashim, M.M.; Rhaif, S.H.; Abdulrazzaq, A.A.; Ali, A.H.; Taha, M.S. Based on IoT Healthcare Application for Medical Data Authentication: Towards A New Secure Framework Using Steganography. In Proceedings of the IOP Conference Series: Materials Science and Engineering, 3rd International Conference on Sustainable Engineering Techniques (ICSET 2020), Baghdad, Iraq, 15 April 2020; Volume 881, p. 012120.
9. Simmons, G.J. The prisoners' problem and the subliminal channel. In *Advances in Cryptology*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 51–67.
10. Khalind, O.S.; Hernandez-Castro, J.C.; Aziz, B. A study on the false positive rate of Stegdetect. *Digit. Investig.* **2013**, *9*, 235–245. [CrossRef]
11. Provos, N. StegDetect. 2020. Available online: https://github.com/abeluck/stegdetect (accessed on 17 November 2020).
12. Cheddad, A.; Condell, J.; Curran, K.; Mc Kevitt, P. Digital image steganography: Survey and analysis of current methods. *Signal Process.* **2010**, *90*, 727–752. [CrossRef]
13. Zhang, L.; Gao, Y.; Xia, Y.; Dai, Q.; Li, X. A fine-grained image categorization system by cellet-encoded spatial pyramid modeling. *Multimed. Tools Appl.* **2015**, *62*, 564–571. [CrossRef]
14. Bailey, K.; Curran, K. An evaluation of image based steganography methods. *Multimed. Tools Appl.* **2006**, *30*, 55–88. [CrossRef]
15. Zöllner, J.; Federrath, H.; Klimant, H.; Pfitzmann, A.; Piotraschke, R.; Westfeld, A.; Wicke, G.; Wolf, G. Modeling the Security of Steganographic Systems. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 344–354.
16. Provos, N. Scanning USENET for Steganography. 2001. Available online: http://niels.xtdnet.nl/stego/usenet.php (accessed on 17 November 2020).

17. Agarwal, A.; Gupta, M.; Gupta, S.; Gupta, S. Systematic digital forensic investigation model. *Int. J. Comput. Sci. Secur.* **2011**, *5*, 118–131.
18. Dezfoli, F.N.; Dehghantanha, A.; Mahmoud, R.; Sani, N.F.B.M.; Daryabar, F. Digital forensic trends and future. *Int. J. Cyber Secur. Digit. Forensics* **2013**, *2*, 48–76.
19. Casey, E. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*; Academic Press: Cambridge, MA, USA, 2011.
20. Boehm, B. Stegexpose-A tool for detecting LSB steganography. *arXiv* **2014**, arXiv:1410.6656.
21. Boehm, B. StegExpose. 2020. Available online: https://github.com/b3dk7/StegExpose (accessed on 17 November 2020).
22. SilentEye. SilentEye. 2020. Available online: https://achorein.github.io/silenteye/ (accessed on 17 November 2020).
23. OpenPuff. OpenPuff. 2018. Available online: https://embeddedsw.net/doc/OpenPuff_Help_EN.pdf (accessed on 17 November 2020).
24. OpenStego. OpenStego. 2020. Available online: https://www.openstego.com/ (accessed on 17 November 2020).
25. StegSecret. StegSecret. 2020. Available online: http://stegsecret.sourceforge.net/ (accessed on 17 November 2020).
26. You, W.; Zhang, H.; Zhao, X. A Siamese CNN for Image Steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 291–306. [CrossRef]
27. Shojae Chaeikar, S.; Ahmadi, A. Ensemble SW image steganalysis: A low dimension method for LSBR detection. *Signal Process. Image Commun.* **2019**, *70*, 233–245, [CrossRef]
28. Kim, J.; Park, H.; Park, J. CNN-based image steganalysis using additional data embedding. *Multimed. Tools Appl.* **2020**, *79*, 1355–1372. [CrossRef]
29. Ibrahim, A. Steganalysis in Computer Forensics. In Proceedings of the 5th Australian Digital Forensics Conference, Perth, Australia, 3–4 December 2007; p. 10.
30. Latham, A. Steganography: JPHIDE and JPSEEK. 1999. Available online: https://github.com/h3xx/jphs (accessed on 17 November 2020).

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.