Content-Based Steganography approach for Invisible Watermarking using CNN

Sakthi Gunasekar*, Dr. Subarna Chatterjee*,
* M. S. RAMAIAH UNIVERSITY OF APPLIED SCIENCES, Bengaluru - 560054

***Abstract*—In today's age of the internet, devices and data are interconnected with each other at unfathomable levels. But with this, there comes a feeling that nothing is privy anymore. This is where steganography comes into play. Steganography is the art of hiding data behind other ordinary and often publicly available "host data". In this work, we explore text-hiding behind images. For an added layer of security, we explore the options of an alternative text.Here, we selected the dancing men text to encode our hidden text. We first generate a dancing men image of the secret text which is embedded into the Least Significant Bits (LSB) of the Host Image. On the output end, we extract the embedded watermark. This watermark is then processed and passed on through a neural network for character recognition and is then collated together to assemble the deciphered text. On measuring the Similarity scores between the reconstructed text and the original text using the Cosine and the Lehvenstein distance metrics, we were able to obtain consistently high results. We also found that the processing time required for reconstruction is a function of the text rather than the host image. To check the effect of adding the text on the host image, we used the metrics SSIM and PSNR and obtained SSIM scores very close to 1 in all cases. This proved that text can be hidden in the host images without causing significant changes to the host image.**

***Index Terms*—Steganography, Content-Based Watermarking, Neural Network, LSB, SSIM, PSNR**

I. Introduction

Steganography is the art of concealing information in such a way that it is impossible to identify concealed messages. Steganography literally translates to ”covered writing” in Greek, as it is derived from the two Greek words 'Steganos' (meaning covered) and 'Graph' (meaning to write). It uses a variety of covert communication methods to keep the concealed message's existence hidden.

People have buried information using a variety of ways and variations throughout history. Steganography's beginnings may be traced all the way back to 440 BC. Whether it was the ancient Greeks who wrote texts on wax-covered tablets, or the Germans developing and implementing microdot technology, or the camouflaging of secret messages in innocent-sounding letters through invisible inks during World War II, or document texts concealing a hidden message through the use of null cyphers (unencrypted messages), we can see that Steganography has a long history. In today's world, thanks to recent developments in computer power and technology, Steganography has climbed to the forefront of today's security techniques.

In this paper, we propose a hybrid steganography approach with a two-step image and text encryption process. We were able to identify a few research gaps in the literature that we have listed over here:

1) There is always a gap at the receiver end security protocol as most of the approaches and implementations followed one step encryption, which means that it is likely to predict the information with ease.
2) Some of the implementations followed a dual scheme but still, extraction of the text had reasonably less accuracy.
3) The literature did have good accuracy in the text extraction, in those the author had followed the direct approach where the "secret text" is in human-readable form.

To solve the above-stated problems, we propose a solution that involves:

1) Dual step embedding where the "secret text" is being converted into a Non-Human readable format Dancing man font.
2) A content-based watermark embedding pipeline for greater lesser host data loss and faster run-time.
3) Algorithm with Good text extraction / Good Object Classification accuracy.

II. Related Works

Steganography has risen to the forefront of today's security techniques thanks to recent advances in computer power and technology. Jammi et al (2010) [1] were one of the first studies published in the early 2010s to document and examine ancient as well as present Steganographic methods, as well as to draw a clear separation between Steganography and Cryptography.

In Ibrahim et al (2011), [2] the authors proposed a two-layer steganography algorithm and created the SIS (Steganography Imaging System). They discovered that the stego image had no discernible distortion using the proposed technique (as seen by the naked eyes). They also found that the stego image had a greater PSNR value based on the PSNR value of each image, demonstrating the efficacy of the algorithm they devised.

Philjon et al. (2011) [3], by merging Cryptography and Steganography, developed a perfect security solution and invented a new methodology called Metamorphic Cryptography. The message was converted into a cipher image with a key, then it was hidden in another image with Steganography by transforming it into an intermediate text, before being changed back into an image. Gupta et al. (2012) [4] introduced the concept of steganography using a new Algorithm called "Enhanced LSB Algorithm", which had negligent distortion as compared to the Least Significant Bit (LSB) Algorithm. The amount of distortion that was detrimental to the human eye was also reduced. Soon, Dagadita et al. (2013) [5] created an application that hid and recovered data using the LSB steganography approach. They proposed a parallel and serial implementation for the implementation and used three images of size 1.9MP, 24.3MP, and 131.7MP. Aman Singh (2013) [6] presented a survey of the existing literature on Steganographic techniques in the time duration of 2003-2013. He found that most of the time, the LSB (least significant bits) technique was utilized, while the MSB (most significant bits) technique was used extremely rarely.

Coming to the more recent approaches, Gupta et al. (2018) [7] proposed a novel method for text steganography using Natural Language Processing, in which natural language was used as cover as well as the secret message to be sent. This strategy yielded good results because the stego generated was also meaningful text while simultaneously concealing the secret information.

Traditional embedding-based steganography embedded the secret information into the content of an image, leaving a slight trace of the modification that could be detected by increasingly advanced machine learning steganalysis algorithms. To overcome that issue, Hu et al (2018) [**?**] proposed a new method of image steganography that included the popular deep learning algorithm, deep convolutional generative adversarial network (DCGAN). This method demonstrated high and accurate information extractions, but the use of DCGANs to image steganography resulted in a few downsides due to the normal weaknesses of DCGANs.

Tang et al. (2019) [**?**] reported a steganographic approach that used a unique operation termed adversarial embedding (ADV-EMB) to hide a stego message while deceiving a CNN-based steganalyzer without adding unexpected objects observable by other classifiers. The procedure produced adversarial stego pictures with the fewest possible modifiable parts. The results revealed that adversarial stego pictures outperformed CNN counterparts even when both the stenganographer and the steganalysis iteratively altered their techniques based on new information about the other side.

Li et al. (2020) [**?**] proposed a machine-learning-based text Emotional modulation steganography approach. Their solution was based on a conjunction structure using the Word2Vec word-vector tool, and they used a machine-learning algorithm to extend the emotional lexicon. Combining extended emotional dictionaries with matrix embedding enhanced the steganographic algorithm embedding efficiency, lowered the chance of detection by statistical analysis tools, and improved the privacy and robustness of secret information embedding, according to experiments.

III. Methodology

A. Image Generation

1) Host Image: The host images are inconspicuous images that can be found all over the internet. The only restriction in their selection is that they should not be predominantly black.

2) Secret Text Image: Developed as part of the Sherlock Holmes series, Dancing Men font is a substitution cipher. It consists of a dancing man in various position to signify alphabets with the man holding a flag to signify separation of words. The secret texts were converted into this and an image generated for the same. The algorithm also supports the characters ”.”, ”,”, ”-”, and ”_” but these are taken without any encoding.

B. Content-Based Watermark Embedding System

The secret image is now embedded into the host image. For faster run-times, both the host and secret images were resized and divided into 12 (6x2) tiles which are then passed into a multiprocessing pool for embedding. The function starts with converting the pixel values of the host and secret images to binary. The secret image is encoded into the 4 Least Significant Bits (LSBs) of the host image. For this, we are employing a content-based watermark embedding system. In it:

- A *”0000” value is assigned to the LSBs if the pixel is a foreground (black) pixel*
- The original host image LSB values are fed for background (non-black) pixels

Changing the values for only the foreground (black) pixels has many advantages. The majority of the pixels are not possessed since they consist of background (white) pixels. From **table III-B** and **fig. 1** it can be observed that foreground pixels consist of less than 10% of total pixels.

Secret Image	Foreground Pixel	Total Pixels	Percentage % Foreground Embedded
P2GlobalWarming.jpg	107002	1546155	6.920
P3Interlaken.jpg	110167	1563361	7.047
Sensore.jpg	13392	297216	4.506
skills.jpg	10864	131175	8.282
P5Tamil.jpg	134276	1589668	8.447
P4Europe.jpg	112512	1352832	8.317
Level.jpg	74259	1672620	4.440
P1ChildLabour.jpg	133691	1608345	8.312
Workflow.jpg	12150	184912	6.571
rule.jpg	9362	174760	5.357
States.jpg	50637	826358	6.128

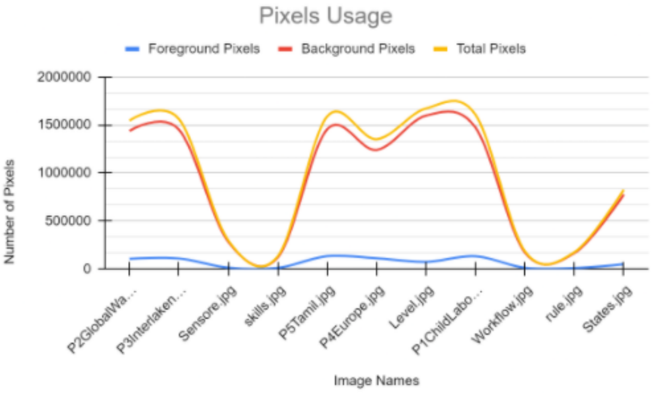


Fig. 1. Percentage % Foreground Pixels Embedded

This reduced number of changed pixels along with changes in only the 4 LSB values of those pixels allows for reduced loss in host images after embedding. This can be observed with reduced error metrics like Mean Squared Error (MSE), Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR) between the host and embedded images.

- $SSIM(x,y) = [l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}]$
- $PSNR = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE})$
- Where $MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]$

There are also no discernible differences between the two images (see fig. 2). Following the embedding process, the encoded image is presented in the GUI.

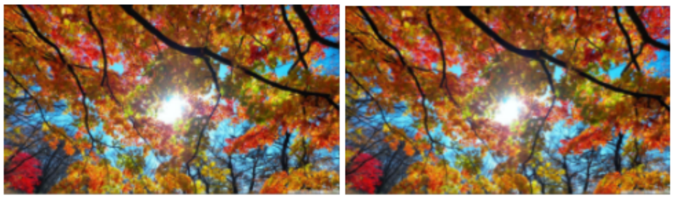


Fig. 2. The host image before (left) and after (right) embedding “P2GlobalWarming.jpg”

C. Watermark Extraction

Similar to the embedding process, the images are divided into 12 tiles and processed. Similar to the encoding process, the pixels in the encoded image are into binary. Then, the foreground pixels with *”0000”* 4LSBs are separated. These pixel locations are set to black by concatenating with another *”0000”* to form a binary 8-bit image. To reduce the MSE of the extracted host image received after watermark extraction, the *”0000”* was replaced wtih *”0101”*

D. Text Deciphering

The extracted secret image is pre-processed to obtain the individual characters present in the image. The rows are identified using OpenCV contours and extracted from the secret image while noting structures as spaces and paragraphs present in it. This is followed by extraction of each individual characters which is then sent to the neural network for character classification.

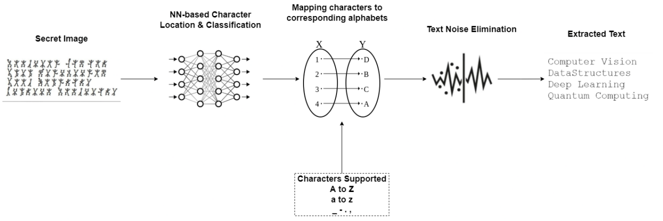


Fig. 3. Text Extraction Workflow

E. CNN Network

Using a Neural Network (NN) for decoding not only avoids the use of manual hand-tuning-based image processing but also helps decrease the runtime since a NN takes significantly less time than the conventional image thresholding-based method which still doesn’t deliver adequate levels of accuracy. The models were trained from scratch on a custom dataset

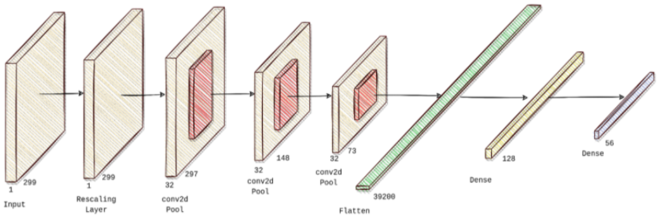


Fig. 4. CNN Architecture

1) Training Dataset: For training, the dataset consists of individual dancing men characters with both non-space and space (with a flag) variations and corresponding English alphabets as the labels. The dataset is split into 80:20 train-test split, whereafter, it consistently obtained an accuracy of 1 on both train and test scenarios.

2) Evaluation Metrics: The similarity between the original and the deciphered text is measured using the Cosine and lehvenshtein distance metrics, which are defined as follows:

- **Cosine Similarity:** It measure the similarity between two vectors of an inner product space $(A \cdot B = ||A||||B|| \cos\theta)$
- **Levenshtein Distance:** It is a string metric for measuring the difference between two sequences.

$$lev(a,b) = \begin{cases} |a| & \textit{if —b—=0,} \\ |b| & \textit{if —a—=0,} \\ lev(tail(a), tail(b)) & \textit{if a[0]=b[0],} \\ 1 + min \begin{cases} lev(tail(a), b) \\ lev(a, tail(b)) \\ lev(tail(a), tail(b)) \end{cases} & \textit{otherwise} \end{cases} \tag{1}$$

F. Interface

The overall pipeline is integrated into a GUI interface with options for embedding and extracting the secret image along with deciphering the said secret-text image.

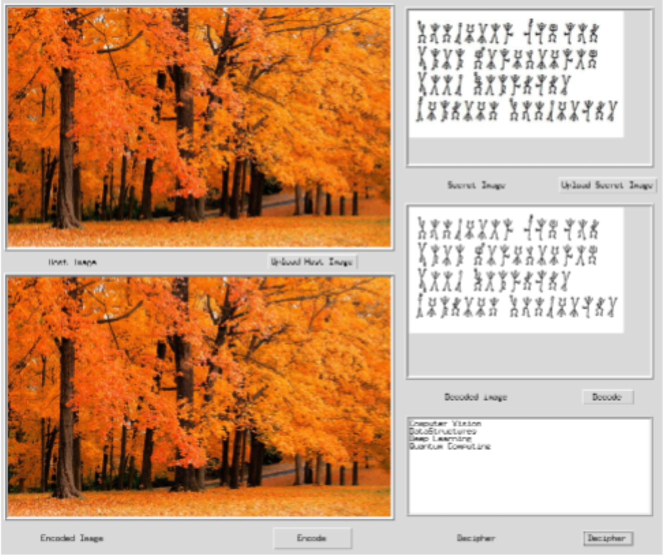
IV. Results & Inferences

Tables 4.1-4.6 show the similarity scores between the original text and the reconstructed text, i.e. how well we are able to recover the text from the host image. The values of the reconstruction similarity scores are different for each host image-text pair. First, we analyze the impact of the nature of the host image in reconstruct ability.

Image	SSIM	PSNR	MSR	Host Image Dimensions (width, height)	Secret Image Dimensions (width, height)	Processing Time (Sec)
GreenlandIceSheet	0.9838	43.3461	9.0281	(2048, 1152)	(1755, 881)	3.0830
Pumpkin1	0.9724	41.2945	14.4795	(2121, 1414)	(1755, 881)	3.3556
RainyLondon	0.9923	46.2722	4.6023	(3840, 2160)	(1755, 881)	7.8550

Epochs	Loss	Accuracy	Val_loss	Val_accuracy
1	4.0440	0.0142	3.8212	0.3750
2	3.7219	0.3939	2.9507	0.4187
3	2.6719	0.5085	1.1263	0.8313
4	0.9414	0.8497	0.1673	0.9625
5	0.1449	0.9625	0.0276	1.0000
6	0.0235	0.9877	0.0080	1.0000
7	0.0077	1.0000	0.0022	1.0000
8	9.1007e-04	1.0000	2.1988e-04	1.0000
9	2.2652e-04	1.0000	1.9118e-04	1.0000

Text	Cosine Score	Levenshtein Score	Original Count	Deciphered Count	Processing Time (Sec)
P2GlobalWarming.txt	0.8237	13	573	574	11.6307
P4Europe.txt	0.9455	5	595	595	12.1228
P1ChildLabour.txt	0.8402	16	715	720	14.6748



```
Computer Vision
DataStructures
Deep Learning
Quantum Computing
```

Fig. 5. CNN Architecture

Table 4.1 shows the scores for the 'Autumn1' host image, Table 4.2 using 'GreenLandIceSheet' and Table 4.3 using 'Pumpkin1'. In each case, the reconstruction scores are suitably high, consistently above 0.8 in each case. An interesting observation is that the scores are approximately uniform for any particular text across all host images, as seen in Tables 4.4 - 4.6. This implies that the constructability of the text is strongly dependent on the nature of the host image, which makes sense intuitively (we would expect a simple blank image to be better compatible with the hidden text extraction filters, while a complex collage of colours might lead to poorer results).

In addition to the constructability, we also analyze the time taken to encrypt and then recover the text from the host images. It is evident that the time required for reconstruction is a function of the text rather than the host image; we can see that the time is almost constant for a particular text across different host images (Tables 4.4 - 4.6), while it varies significantly when the host image is held constant (Table 4.1 - 4.3). Again, this is in line with intuitive expectations; longer texts would take longer to reconstruct.

Next, we check the effect of adding the text on the host

Fig. 6. Figure 5.1: Time Analysis for GlobalWarming text

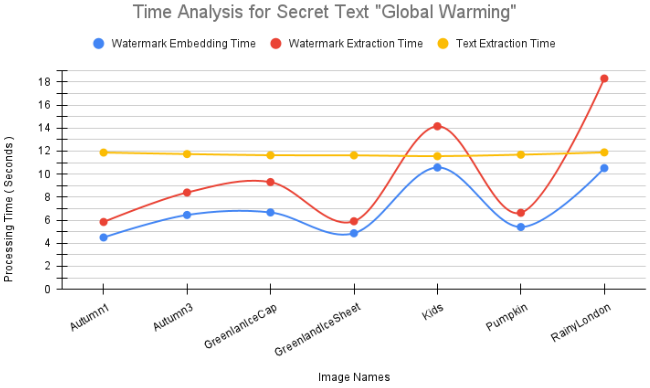
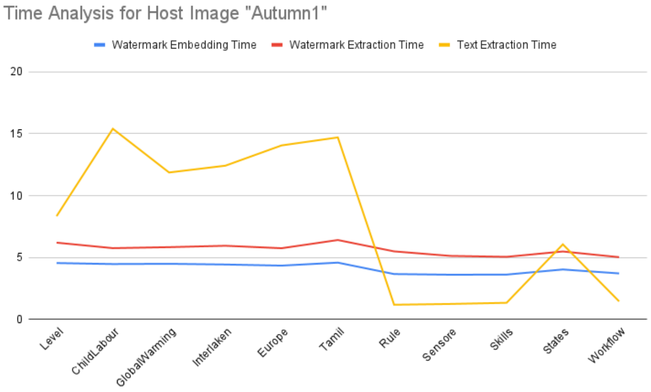


Fig. 7. Figure 5.2: Multiple Execution Time Analysis between for Level text



Text	Cosine Score	Levenshtein Distance	Original Count	Deciphered Count	Processing Time (Sec)
Sensore.txt	1	0	73	73	1.2598
Workflow.txt	0.8018	0	66	66	1.4688
P4Europe.txt	0.9417	5	595	595	14.0587
Level.txt	0.8967	6	396	397	8.3469
P2GlobalWarming.txt	0.843	8	573	574	11.8729
P5Tamil.txt	0.9341	12	714	719	14.7033
P1ChildLabour.txt	0.8448	15	715	720	15.4016
skills.txt	1	0	56	56	1.3542
P3Interlaken.txt	0.9319	11	580	583	12.4209
rule.txt	0.8292	1	49	50	1.1953
States.txt	0.9667	1	282	282	6.0711

TABLE I

Table 5.1 Text Similarity for the host image “Autumn1”

Text	Cosine Score	Levenshtein Distance	Original Count	Deciphered Count	Processing Time (Sec)
P2GlobalWarming.txt	0.8237	13	573	574	11.6307
P4Europe.txt	0.9455	5	595	595	12.1228
P1ChildLabour.txt	0.8402	16	715	720	14.6748
Workflow.txt	0.6682	1	66	66	1.3694
skills.txt	1	0	56	56	1.2707
States.txt	0.9667	1	282	282	5.8388
P5Tamil.txt	0.9326	14	714	718	15.1939
Sensore.txt	0.9286	1	73	73	1.3942
rule.txt	0.8292	1	49	50	1.1282
Level.txt	0.9235	6	396	396	8.0706
P3Interlaken.txt	0.9259	12	580	583	10.5211

TABLE II

Table 5.2 Text Similarity for the host image “GreenLandIceSheet”

Text	Cosine Score	Levenshtein Distance	Original Count	Deciphered Count	Processing Time (Sec)
rule.txt	0.8292	1	49	50	1.0076
States.txt	0.9667	1	282	282	5.8454
P3Interlaken.txt	0.9319	11	580	583	11.8533
Sensore.txt	1	0	73	73	1.6684
Level.txt	0.9074	5	396	396	9.635
P1ChildLabour.txt	0.8495	13	715	720	15.3547
Workflow.txt	0.8018	0	66	66	1.3675
skills.txt	1	0	56	56	1.2815
P2GlobalWarming.txt	0.843	8	573	574	11.6882
P4Europe.txt	0.9455	4	595	595	12.0461
P5Tamil.txt	0.9358	11	714	718	14.6886

TABLE III

Table 5.3 Text Similarity for the host image “Pumpkin1”

image. To do so, we use the metrics SSIM and PSNR. We observe that the values of the SSIM are suitably high, very close to 1 in all cases. This reassures that text can be hidden in the host images without causing significant changes to the same.

V. Conclusion

As proposed, we demonstrated a Content Based Hybrid Steganography approach for Invisible Watermarking, leveraging content-based watermarking and Convolutional Neural Networks (CNNs).Building on previous work, we propose a Content Mask to improve the encryption performance. We demonstrate that our proposed method outperforms previous approaches on a variety of metrics, including SSIM, PSNR and especially run-times. Additionally, a NN-based approach was proposed for extracting text from the embedded watermark image. We then analyze the pipeline against several secret text-host image pairings.

In future, we can incorporate Natural Language Processing tools like sentiment analysis on the secret text into our framework. Applying it on the secret text itself can help the receivers identify the message sentiment/context even before reading the text itself. The embedding/watermarking step can be further improved by state-of-the-art advancements.

Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks . . .”. Instead, try

Host Image	Cosine Score	Levenshtein Distance	Original Count	Deciphered Count	Processing Time (Sec)
GreelanIceCap.png	0.9294	19	714	721	15.0852
Autumn3.png	0.9275	12	714	718	15.014
GreenlandIceSheet.png	0.9326	14	714	718	15.1939
Autumn1.png	0.9341	12	714	719	14.7033
Kids.png	0.9358	11	714	718	14.5449
RainyLondon.png	0.9263	13	714	718	14.7814
Pumpkin1.png	0.9358	11	714	718	14.6886

TABLE IV
Table 5.4 Text Similarity for the host image “P5Tamil.txt”

	SSIM	PSNR	ERR	Host Image Dimensions (IW, IH)	Secret Image Dimensions (IW, IH)	Processing Time (Sec)
Autumn1.png	0.993	41.676	13.261	(1920, 1200)	(583, 225)	3.6295
Kids.png	0.995	48.115	3.0105	(3760, 1728)	(583, 225)	9.8454
GreenlandIceSheet.png	0.998	54.410	0.7065	(2048, 1152)	(583, 225)	3.933
RainyLondon.png	0.997	50.095	1.9083	(3840, 2160)	(583, 225)	12.119
FamilyRunningMAsk.png	0.991	44.375	7.1234	(1080, 1080)	(583, 225)	2.0372
Autumn3.png	0.995	41.703	13.178	(2560, 1440)	(583, 225)	6.071
Pumpkin1.png	0.993	46.273	4.6011	(2121, 1414)	(583, 225)	4.52
dog.png	0.988	44.522	6.8856	(800, 534)	(583, 225)	0.8085
GreelanIceCap.png	0.995	49.366	2.2572	(2560, 1440)	(583, 225)	5.7902

TABLE V
Table 5.5 Image Similarity for the sample text “skills.txt”

Host Image	SSIM	PSNR	ERR	Host Image Dimensions (IW, IH)	Secret Image Dimensions (IW, IH)	Processing Time (Sec)
GreenlandIceShee.png	0.984	43.3461	9.0281	(2048, 1152)	(1755, 881)	4.8666
Pumpkin1.png	0.972	41.2945	14.480	(2121, 1414)	(1755, 881)	5.4067
RainyLondon.png	0.992	46.2722	4.6023	(3840, 2160)	(1755, 881)	10.522
GreelanIceCap.png	0.983	43.7503	8.2257	(2560, 1440)	(1755, 881)	6.6752
Kids_.png	0.980	43.0211	9.7294	(3760, 1728)	(1755, 881)	10.577
Autumn1.png	0.986	39.3411	22.704	(1920, 1200)	(1755, 881)	4.5004
Autumn3.png	0.991	40.0628	19.228	(2560, 1440)	(1755, 881)	6.4531

TABLE VI
Table 5.6 Image Similarity for the sample text “P2GlobalWarming.txt”

Secret Image	Watermark Embedding Time (Seconds)	Watermark Extraction Time (Seconds)	Text Extraction Time (Seconds)
Level.png	4.561	6.2083	8.3469
P1ChildLabour.png	4.4804	5.7648	15.4016
P2GlobalWarming.png	4.5004	5.8452	11.8729
P3Interlaken.png	4.4438	5.9562	12.4209
P4Europe.png	4.3547	5.7589	14.0587
P5Tamil.png	4.5953	6.4202	14.7033
rule.png	3.6733	5.5037	1.1953
Sensore.png	3.6195	5.1392	1.2598
skills.png	3.6295	5.0636	1.3542
States.png	4.0442	5.4953	6.0711
Workflow.png	3.7236	5.0369	1.4688

TABLE VII
Table 5.7 Time Analysis for the Host Image “Autumn_1”

“R. B. G. thanks…”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

[1] ASHOK, Jammi, Ashok Y.RAJU, S.MUNISHANKARAIAH, K.SRINIVAS,. (2010). STEGANOGRAPHY: AN OVERVIEW. International Journal of Engineering Science and Technology. 2.

[2] Ibrahim, Rosziati Kuan, Teoh. (2011). Steganography Algorithm to Hide Secret Message inside an Image.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all

Image	Watermark Embedding Time (Sec)	Watermark Extraction Time (Sec)	Text Extraction Time (Sec)
Autumn1.png	4.5004	5.8452	11.8729
Autumn3.png	6.4531	8.4017	11.7424
GreelanIceCap.png	6.6752	9.3097	11.6379
GreenlandIceSheet.png	4.8666	5.9078	11.6307
Kids.png	10.5767	14.1546	11.5587
Pumpkin1.png	5.4067	6.6411	11.6882
RainyLondon.png	10.522	18.3027	11.8868

TABLE VIII

TABLE 5.8 TIME ANALYSIS FOR THE SECRET TEXT "GLOBALWARMING"

template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.