ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20190404155145

Original scientific paper

Hiding Data and Detecting Hidden Data in Raw Video Components Using SIFT Points

Savas CITLAK*, Ozkan KILIC

Abstract: Steganography is a science of hiding data in a medium whereas steganalysis is composed of attacks to find the hidden data in a cover medium. Since hiding data in a text file would disturb the coherence of the text or make it suspicious, systematically changing pixels of a visual is a more common method. This process is performed on pixels that are spatially (and/or temporally, for video components) distant from each other so that a viewer's eye can be deceived. Online media are subject to modification such as compression, resolution change, visual modifications, and such which makes Scale Invariant Feature Transform (SIFT) points appropriate candidates for steganography. The current paper has two aims: the first is to propose a method that uses the SIFT points of a video for steganography. The second aim is to use Convolutional Neural Networks (CNN) as a steganalysis tool to detect the suspicious pixels of a video. The results indicate that the proposed steganography method is effective because it yields higher peak signal-to-noise ratio (PSNR = 95.41 dB) compared to other techniques described in cybersecurity literature, and CNN cannot detect hidden data with much success due to its 52% accuracy rate.

Keywords: CNN; LSB; PSNR; SIFT; Steganalysis; Steganography

1 INTRODUCTION

Steganography, which is a Greek word meaning covering writing [1], is an important sub-discipline of data-hiding methods, which involves the process of hiding data in a medium. These media components may be a picture, an audio, a video, a web page, and such. This technique is usually employed by illegal groups who want to disseminate information online in an untraceable way. Therefore, it is important to investigate possible more sophisticated methods and their discoverability in the context of cybersecurity and cryptography. The most important difference between steganography and cryptography is that the former is able to detect whether there is meaningful data in the target object. The purpose of steganography is to keep the confidential and hidden so that neither can be discerned by third parties. In short steganography is described as "the art of hiding data" [2].

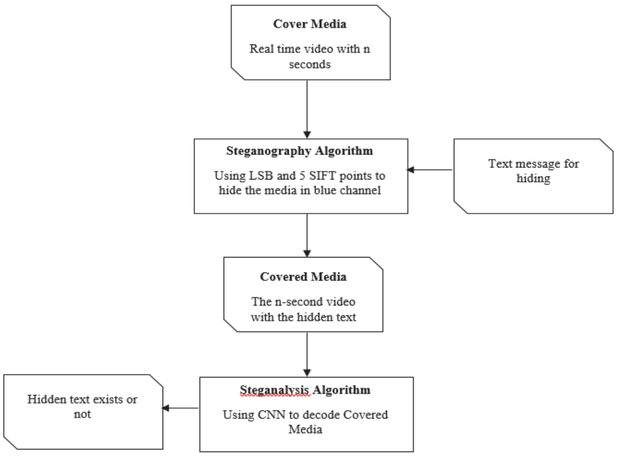
Cover Media
Real time video with n seconds
Steganography Algorithm
Using LSB and 5 SIFT points to hide the media in blue channel
Text message for hiding
Covered Media
The n-second video with the hidden text
Steganalysis Algorithm
Using CNN to decode Covered Media
Hidden text exists or not

Figure 1 Pipeline of the current study

Similarly, steganalysis is a method of attack to uncover confidential data in a cover medium. If a hidden data is found, to obtain or, at least, alter this information. Most steganalysis applications are based on mathematical/statistical analysis [3]. However, in recent years, deep-learning-based steganalysis studies have become popular [4]. Such methods are designed to operate over the data scattered in the spatial environment (picture), over time scattered data (sound) and over the data scattered over both time and spatial environment (video).

The current study proposes a real-time method that uses SIFT (Scale-Invariant Feature Transform) [10] points of a video to hide information. It further investigates whether a CNN can detect a frame with hidden data embedded with this technique. Fig. 1 explains the pipeline of the study.

SIFT is a computer-based visual algorithm used to determine and identify key point properties in the image [10]. Objects may be subject to changes in subsequent frames, since images are limited to two dimensions while real objects are three-dimensional. These changes that may be exposed in object tracking are in the form of "Dimensional Change", "Angular Change", "Spatial Variation", "Noise in the Environment" and "Brightness Changes in the Environment". The SIFT algorithm is not affected by the size of the image, the amount of light, the change of camera angle, contrast or noise [10]. The authors believe that SIFT keypoints are useful for steganography because videos streamed online are usually modified to enhance their speed; and thus, these points will generally stay intact in case of any modifications to guarantee persistence of the hidden message. Since it is composed of a complex set of operations, only a deep learning technique which can automatically extract complex features should be able to capture the existence of the hidden data.

1.1 Steganography

Steganography aims to hide confidential data in a different medium. Audio, static images, video images, text, and such are used to hide the data. Its idea is to make sure that only the person to whom information is sent and who is in possession of the key can obtain the confidential data [5]. The medium that will hold the confidential data is called "cover media"; after the encoding, it is called "covered media". The confidential data to be encoded in the cover media can be a text file or an image.

The most important criteria in steganography are the non-detectability and impalpability of hidden data. The concept of impalpability means undetectable by human senses; unpredictability refers to be immune to mathematical analysis. Similar to cryptology, where encrypting is not accepted as confidential and the whole responsibility of communication security belongs to crypto keys, steganography employs a data-hiding method that is not accepted as confidential. Each mystery leads to a potential point of failure, and privacy is the main cause of fragility [6].

"Stego" media is the name given to the message to be hidden. This message could be a plain text, chipper text, other images, or anything that can be digitized in bits. As a result of the embedding process, the cover medium and the message itself constitute the covered media.

Data-hiding science is divided into three parts "Algorithm Domain", "Data Environment" and "Perception". Generally subdivisions of "Algorithm Domain" namely "Spatial Domain" and "Frequency Domain" method are being used, [7]. Several steganography methods have been developed to hide information on image files. These can be classified under 3 titles [8, 9]. These are: "Adding the Least Significant Bits", "Masking and Filtering" and "Algorithms and Transformations".

The current study proposes a method that falls in under the Spatial Domain of Algorithm Domain because it uses the blue channel of the best 5 SIFT points to hide text data. Using SIFT algorithm ensures that it is mathematically hard to discover as given in the CNN-based steganalysis method in the following section.

1.2 Steganalysis

Based on mathematical and statistical methods, steganalysis is generally applied to images, sound and video to look for hidden data. It is generally assumed that the attacker (steganalyst) knows the steganographic system. If the steganalyst does not know the system used, the job gets highly complicated. Steganalysis methods are divided into three categories with respect to their aims: "Passive Steganalysis" which identifies only the existence of the secret message; "Active Steganalysis" which aims to find some or all of the secret message; and "Distorting Steganalysis", a top level of the active steganalysis which aims to detect and destroy the hidden message and/or to replace it with a fake message. The data to be hidden could be encrypted before being embedded in the carrier. Although encrypting confidential data is ineffective in passive steganalysis, it provides solutions to active steganalysis methods.

There is no general steganalysis method to reveal the hidden data by means of steganographic methods. However, most steganalysis applications are based on mathematical/statistical models [3]. The purpose of steganalysis is to develop a large-scale system which can be applied to all data hiding methods rather than to a single method. Steganalysis methods are also used to measure the durability of a steganographic system. For each steganographic method, a separate steganalysis method generally needs to be developed. One method of steganalysis that yields reliable results for one method may not be good for another. Steganalysis methods are organized into three categories according to their type of attack. These types are "Sensory Attacks", "Structural (Signature/Pattern) Attacks" and "Statistical Attack". Examples of statistical attacks are "Neural Networks", "Clustering Algorithms", "Artificial Intelligence", "Machine Learning" and "Deep Learning". The current study employs a deep-learning-based steganalysis attack on the proposed steganography algorithm.

2 METHOD

For the steganography part, data-hiding transactions were performed in the current study using non-repetitive, best-quality SIFT keypoints on momentarily received real-time raw videos. The blue channels of top 5 SIFT points' LSBs (Least Significant Bit) are used to hide the data as given in Algorithm 1.

Algorithm 1

```
Input:
D        // the stego file
 n// size of D in bits
V// the cover media
width// width of V
height// height of V
captureTime// video capture time in msec
framePerSecond// fps of the capture video in units

Output: D'// covered media
  for each (frame f ⊆ V and b_i⊆D)
     topSIFT_1-5 = FindSIFTs(f, width, height) // Find top 5 SIFT points
for each (s ⊆ topSIFT_1-5)
D'= D' ∪ ChangeLSB(s,f,b_i) //Encode next bit from D into LSB of SIFT of f and add it to D'
return D'
```

Table 1 Basic information of repetitive key SIFT keypoints

kp.column	kp.row	kp.octave	kp.angle	kp.response	kp.size
1854.4204	2059.1489	1114879	273.7077	0.113873	2.2969
1854.4204	2059.1489	1114879	97.1726	0.113873	2.2969

In an unpublished PhD dissertation [15], the time complexity of the original SIFT algorithm [10] is given as $\Theta(\alpha\beta N2)$, where $N2$ is the size of a frame, α is the fraction of local extrema in a frame, and β is the fraction of local extrema that turn out to be SIFT descriptor. Both of these fractions are between 0 and 1 and depend on the visual. For the proposed Algorithm 1, assuming that number encoding bits will be less than or equal to the total number of frames, the time complexity is $\Theta\left(n\cdot\alpha\cdot\beta\cdot\text{width}\cdot\text{height}\right)$. Changing the LSB ensures that the encoding is not visible to human eye.

An example application to a ".bmp file" which has a width of 2231 pixels, a height of 3361 pixels and a bit depth of 24 is given in Fig. 2. Fig. 2a shows all SIFT keypoints. Fig. 2b contains enriched partial SIFT keypoints. In Fig. 2c, there are two SIFT keypoints whose coordinates are the same but repetitive in different spatial domain at different angles as given in Tab. 1.

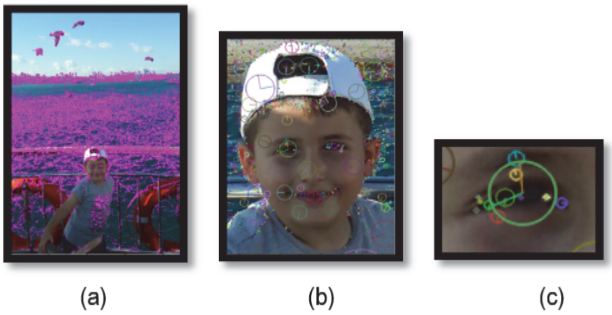

Figure 2 Display of SIFT keypoints

In Fig. 3, four non-repeating keypoints from the top 5 SIFT keypoints are shown in red circles.

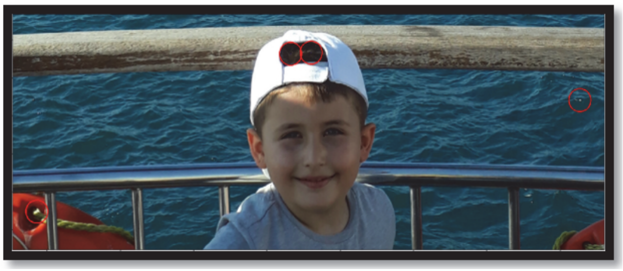

Figure 3 Non-repeating top 4 SIFT keypoints

The x and y points of the repetitive SIFT keypoints correspond to the same coordinates in the spatial domain. The only difference between these points on the same coordinates is the differing angle of the keypoints (kp.angle). This shows that, as seen in Fig. 3, although there are 5 top-quality SIFT keypoints from the real-time raw video components for each frame, 4 non-repetitive SIFT keypoints can be used in the spatial domain.

Tab. 2 contains the feature information of the top 5 key SIFT keys in the cover media before the data was hidden of the ".bmp" file shown in Fig. 2a. For the ranking of the most powerful keypoints, "kp.response" value is taken. As seen in Tab. 2, the x and y points of the kp1 and kp2 keypoints correspond to the same coordinates as the whole number in the spatial domain. The only difference between these points which overlap in the same coordinates, are the angles as seen in "kp.angle" column. Although we have 5 high quality SIFT keypoints for real-time raw video components for each frame, 4 non-repetitive SIFT keypoints can be used in the spatial domain. Therefore, 4-bit data can be stored due to the repetition of keypoints (only angles are different) of the frame which has 5-bit data embedded capacity.

Table 2 Best quality 5 SIFT keypoints before applying the Steganography Method

kp.column	kp.row	kp.octave	kp.angle	kp.response	kp.size
1854.4204	2059.1489	1114879	273.7077	0.113873	2.2969
1854.4204	2059.1489	1114879	97.1726	0.113873	2.2969
1065.8787	1937.9920	7995907	92.3812	0.108107	40.4287
373.0611	2352.0317	15728897	50.7502	0.099566	8.9292
1121.7276	1934.7036	11534851	91.2286	0.097974	42.4523

Data have been hidden in RGB (Red-Green-Blue) channels for the top quality 5 SIFT keypoints of the ".bmp" format image file located in Fig. 2a by means of LSB method. Tab. 3 shows the changes in the keypoints that occur when the top-quality 5 SIFT keypoints of covered media are acquired again after the data-hiding transaction. The differences between Tab. 3 and Tab. 2 are indicated in bold. After the data-hiding process, there was little structural change in the quality order of SIFT keypoints. Therefore, it has been observed that the text stored in the cover media as a result of steganography application (encode process) is identical to the text obtained from the covered media (decode process).

Table 3 Best quality 5 SIFT keypoints after applying the Steganography Method

kp.column	kp.row	kp.octave	kp.angle	kp.response	kp.size
1854.4125	2059.1491	787199	273.7452	0.114242	2.2876
1854. 4125	2059. 1491	787199	97.1486	0.114242	2.2876
1065.8789	1937.9920	7995907	92.3812	0.108106	40.4288
373.0611	2352.0317	15728897	50.7502	0.099566	8.9292
1121.7275	1934.7036	11534851	91.2286	0.097974	42.4524

Tab. 4 shows the variation of the best quality of the 5 SIFT keypoints RGB values before and after data-hiding. This invariance before and after the data hiding process allows the hidden text to be recovered from the covered media. As shown below, it is clear that the RGB changes at the points in the spatial domain (kp_3, kp_4 and kp_5) are +1 for each, while the RGB changes of the repeating (kp_1, kp_2) points in the spatial domain are +2.

Table 4 Variations of RGB channels for before and after steganography process (LSB)

col	row	origina BGR	stegoBGR	blue	green	red
1854	2059	[246 248 236]	[248 250 238]	[248 248 236]	[246 250 236]	[246 248 238]
1854	2059	[246 248 236]	[248 250 238]	[248 248 236]	[246 250 236]	[246 248 238]
1065	1937	[22 23 33]	[23 24 34]	[23 23 33]	[22 24 33]	[22 23 34]
373	2352	[203 254 252]	[204 255 253]	[204 254 252]	[203 255 252]	[203 254 253]
1121	1934	[14 20 27]	[15 21 28]	[15 20 27]	[14 21 27]	[14 20 28]

This is repeated for the required number of frames until the entire text is hidden in the video. In order to retrieve the hidden text back, the process is reversed. In other words, top 5 SIFT points of each frames are retrieved from the covered media. Then, the blue channel values of the points are converted to text. Since the order of these points in each frame is identical to the order of letters in the text, the hidden text is successfully recovered.

For the second part of the study, steganalysis, deep learning-based CNN detectors, are employed to detect the existence of hidden data. For this purpose, two sets of training data are prepared: one set with 2010 images with hidden data embedded by the proposed Steganography, and 1020 clean original images. A Python deep learning library, Keras, was used to create a sequential model [13]. Keras provides easy and rapid prototyping and can support convolutional networks, repetitive networks and hybrid networks consisting of both [12]. Tab. 5 represents the model used for steganalysis.

Table 5 Parameters for the Steganalysis Model

	layer (type)	output shape	param
1	conv2d_1 (Conv2D)	(None, 62, 62, 32)	896
2	max_pooling2d_1 (MaxPooling2)	(None, 31, 31, 32)	0
3	conv2d_2 (Conv2D)	(None, 29, 29, 32)	9248
4	max_pooling2d_2 (MaxPooling2)	(None, 14, 14, 32)	0
5	conv2d_3 (Conv2D)	(None, 12, 12, 32)	9248
6	max_pooling2d_3 (MaxPooling2)	(None, 6, 6, 32)	0
7	conv2d_4 (Conv2D)	(None, 4, 4, 32)	9248
8	max_pooling2d_4 (MaxPooling2)	(None, 2, 2, 32)	0
9	flatten_1 (Flatten)	(None, 128)	0
10	dense_1 (Dense)	(None, 128)	16152
11	dense_2 (Dense)	(None, 1)	129
12	Train/validation data/batch_size	2040 / 510 / 32	
13	Optimizer	Adam	
14	Activation	Relu and Sigmoid	
15	Avg Epoch Time	750 s	
16	Avg Step Time	750 / 2040 = 367 ms	
	Total Params:	45.281	
	Trainable params:	45.281	
	Non-trainable params:	0	

The model has 4 convolutional [14], 4 maxpooling, 1 flatten and 2 dense layers and a total of 45,281 parameters to be trained. The results are given in the following section.

3 RESULTS

In a similar study in cyber security literature [11], texts with different contents and sizes were stored in images. To compare the cover media before data-hiding with the covered media after data-hiding, a full reference quality image metric, PSNR (Peak Signal Noise Ratio), was suggested to measure the method's success rate.

In this study, data-hiding operations are not performed on pixels that have a predetermined static pattern according to some mathematical operations. Instead, videos are taken in real time to detect the highest quality SIFT keypoints of each frame dynamically. Therefore, the initial state of the snapshot and the cover media is absent. The proposed approach makes it difficult for another party to perform steganalysis. The current approach was applied to the highest quality 5 SIFT keypoints for each frame in the digital images taken in real time. The performance tests were performed with 300 frames per 10 second instant images (30 f/s). As seen in Tab. 6, the loss of instant frame was found to be approximately 82%.

Table 6 Instant frame loss using SIFT method

Video Duration / sn	normal frame number	Real-time frame number	frame loss rate / %
1	30	3	90
5	150	25	83.3
10	300	61	79.66
20	600	112	81.33
30	900	174	80.66
40	1200	183	84.75
50	1500	302	79.86
100	3000	616	79.46
average			82.38

Tab. 7 shows the individual SIFT keypoints, which are hidden by the frame-based data. 35 SIFT keypoints need to be obtained for the 7 frames discussed, but 23 SIFT keypoints were obtained because of the repeated points in the spatial domain (different angles of the same keypoint).

Table 7 Embedding data on individual keypoints in a frame

frame no	kp 1	kp 2	kp 3	kp 4	kp 5	total keypoints
frm 1	(446,498)	(435,610)	(433,632)	repetitive	repetitive	3
frm 2	(449,498)	(437,610)	(436,632)	(287,159)	repetitive	4
frm 3	(449,498)	(437,610)	(287,158)	repetitive	repetitive	3
frm 4	(449,498)	(437,610)	(436,632)	(287,158)	repetitive	4
frm 5	(450,497)	(288,158)	repetitive	repetitive	repetitive	2
frm 6	(450,497)	(439,610)	(437,631)	repetitive	repetitive	3
frm 7	(454,490)	(442,602)	(271,130)	(269,133)	repetitive	4
…						23 pieces (7 × 5)

PSNR performance values and repeated loss rates of keypoints are shown at Tab. 8.

In the case where a sufficient number of captured video frames is available, that is, if there is no capacity problem in the carrier medium, the content of the hidden data is the same as the content of the data. The SIFT-based steganography model had a PSNR of 95.41 dB which is better than the previous studies. For example, an LSB steganography algorithm based on quantum circuits reports its best PSNR value as 51.14 [16] and 77.45 for another [20]. Another method employing pseudo random number generators to change LSB for data hiding reports 59.73 peak PSNR value [17]. Hajduk and his friends hide a QR code in various images and achieve 71.44 [18]. Similarly, Zhou and his friends propose another LSB-based steganography technique for colored images and get 56.513 PSNR [19]. Besides its higher PSNR value, the proposed method in this paper also provides an easier way to retrieve the hidden text. In other words, since the quality and order of SIFT points do not change after the data hiding process, the original text can be retrieved from the covered media. Tab. 9 summarizes the overall characteristics of this steganography algorithm.

Table 8 Average singular loss rate of keypoints (for five different texts)

text number	Embedded bit number	Frame number-Data hidden	Kp number (× 5)	unique kp number	singular kp loss rate / %	Average PSNR value-for the whole frames
1	784 bit	223	1115	785	29.59	90.57 dB
2	664 bit	172	860	667	22.44	100.82 dB
3	296 bit	65	325	300	7.69	98.18 dB
4	768 bit	194	970	768	20.82	99.77 dB
5	1064 bit	270	1350	1064	21.18	98.05 dB
avarage					20.34	97.41 dB

Table 9 Summary of the SIFT-based steganography application

Parameter/Characteristics	Value/Explanation
Carrier Type	Real-time raw video components
Data-hiding method	LSB - Blue Channel from RGB channels
Pixel locations to hide	Top quality 5 SIFT keypoints per frame to space plane non-repetitive points
Stego text content character count	269 bytes
Stego text content (encode / decode)	*Andrew Tanenbaum has a BA in MIT from the University of California at Berkeley. Steganography is the technique of hiding confidential information within any media. Steganalysis is process to detect of presence of steganography. scitlak25-20190130.*
Number of bits to hide (1 byte = 8 bits)	269 × 8 = 2152
Number of frames per second	30 f / s
Video capture time	360 s
Total number of frames received	360 × 30 = 10800
Number of SIFT process frames	1021
Number of frames with SIFT that does not hide data	408
Number of frames with data hidden with SIFT	613
Number of captured keypoints in SIFT (5 pcs)	613 × 5 = 3065
Captured number of unique SIFT keypoints	2156
Number of pixels hidden in data	2152
Steganography overall performance rate (PSNR)	95.41 dB

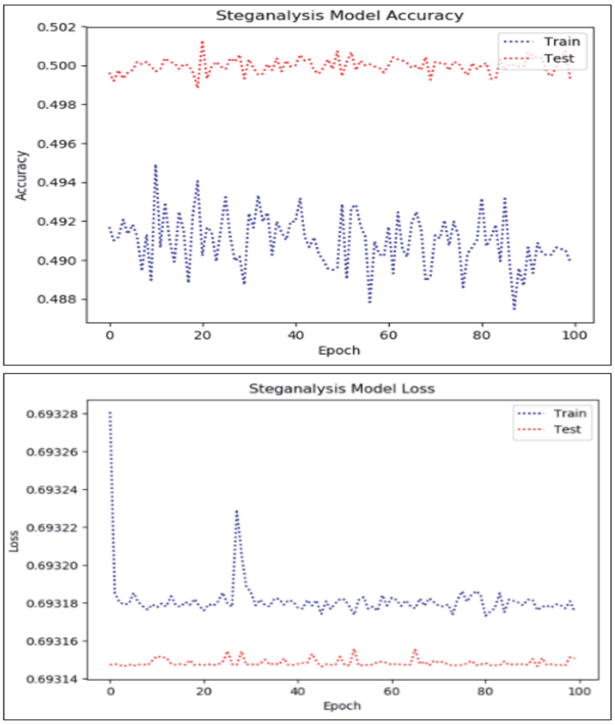

Figure 4 CNN-based Steganalysis model's accuracy and loss (100 epochs)

In order to test the existence of data hidden by the proposed method, the CNN is employed in multiple tests. In the "accuracy" graph in Fig. 4, the accuracy value increases to 0.52 when epoch time is 100. It was observed that the accuracy value varied between 0.49 and 0.52 in the tests. This result is considered normal because both steganalysis attacks based on pixel neighboring matrix and steganalysis attacks with CNN detectors are not successful at all. In other words, the proposed method cannot be attacked by a CNN.

As shown in Tab. 10, a total of 2040 "640 × 480" pixels ".bmp" files were used for training. 1020 images were "cover" files whereas 1020 of them were "covered/stego" files. Similarly, a total of 510 images were used for testing. 255 of them were "cover" while 255 were "covered/stego" files. 10, 20, 50 and 100 values were used as epoch number. The total number of images reported to have no data hidden by the CNN can be seen in the "cover? " column. The "stego?" column shows the CNN's answer for the number of covered files. The reason for the inconsistency in the values in the "stego accuracy" column is validation accuracy varied 0.49 and 0.52.

The consecutive tests showed that the CNN ran no better than tossing a coin for the proposed steganography method.

Table 10 Results for Steganalysis tests

test no	epoch time	train data	validation data	batch size	stego test data	cover?	stego?	stego accuracy / %
1	10	2040	510	32	255	33	222	87.05
2	20	2040	510	32	255	145	110	43.13
3	50	2040	510	32	255	147	108	42.35
4	100	2040	510	32	255	91	164	64.31

4 CONCLUSION

In this study, steganography and steganalysis studies were performed for instantly taken real-time raw videos. Text data was hidden using up to 5 non-repeating SIFT keypoints within each frame. Data-hiding was applied to the related pixel's blue channel with the LSB method in the spatial domain.

It was seen that there was little structural change between the SIFT keypoints before and after the data-hiding. As a result of data-hiding, it has been observed that the hidden message embedded in the cover media (encode process) is identical to the message extracted from the covered media (decode process). This technique allows invariant data to be hidden in the streaming media due to SIFT.

The success rates of data-hiding were measured with PSNR values. The steganography application was more successful than it was in the other studies in the literature [16-20]. In addition, the steganalysis was implemented using deep learning-based CNN detectors. The accuracy was found to vary between 0.49 and 0.52 which makes the proposed method hard to detect. The reason why these steganalysis attacks were not very successful is that the SIFT-based steganographic performance value (PSNR) is high and because there are few structural differences between the "cover file" and the "covered file" to be captured.

Applications

Applications are developed with the Python programming language. For computer vision algorithms, Python 2.7 "opencv" library is used. The Keras 2.2.4 library based on CNN was used on Python 3.6. deep learning class (cnnKerasSteganalysis), as processor"Intel (R) Core (TM) i5-4210U CPU @ 1.70 GHz 2.4 GHz", as memory "8.00 GB RAM" and as operating system "x64-based 64-bit operating system" is used. Mean epoch time was calculated as 750 seconds.

5 REFERENCES

[1] Murray, A.H., Burchfiled, R.W., (et al.), (1933). The Oxford English Dictionary: Being a Corrected e-issue. Oxford, England: Clarendon Pres.

[2] Rabah, K. (2004). Steganography-The Art of Hiding Data. *Information Technology Journal*, *3*, 245-269. https://doi.org/10.3923/itj.2004.245.269

[3] Pevny, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, *5*(2), 215-224. https://doi.org/10.1109/TIFS.2010.2045842

[4] Qian, Y., Dong, J., Wang, W., & Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. *Proceedings, Media Watermarking, Security, and Forensics*, *9409*. https://doi.org/10.1117/12.2083479

[5] Dereli, Ç. (2010). *A Research on Linguistics Steganography Methods*. Master Thesis, Ege University, Izmir. (in Turkish).

[6] Charles, C. M. (2002). *Homeland Insecurity*. The Atlantic Monthly.

[7] Malik, H., Subbalakshmi, K. P., & Chandramouli, R. (2008). Non parametric steganalysis of QIM data hiding using approximate entropy. *IS&T SPIE: Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, *6819*, 14-12. https://doi.org/10.1117/12.767313

[8] Sahin, A., Buluş, E., (et al.) (2007). *Detection of Confidential Information in Picture by Rqp Steganalysis Method*. Academic Information, Kütahya. (in Turkish).

[9] Fridrich, J., Goljan, M., & Du, R. (2001). Reliable Detection of LSB Steganography in Color and Gray Scale Images. *Proc. of the ACM Workshop on Multimedia and Security*, Ottawa, Canada, 27-30. https://doi.org/10.1145/1232454.1232466

[10] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, *2*, 1150-1157. https://doi.org/10.1109/ICCV.1999.790410

[11] Citlak, S. & Kilic, O. (2018). *Design and Development of Steganography Algorithms for Digital Image Components under Cyber Security*. International Engineering and Technology Management Congress, Istanbul. (in Turkish).

[12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097-1105.

[13] Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, *22*(1), 1237.

[14] Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)]*, 807-814.

[15] Vinukonda, P. (2011). *A Study of the Sace-Invariant Feature Transfom on a parallel pipeline*. Unpublished PhD Dissertation, Louisiana State University.

[16] Jiag, J., Zhao, N., & Wang L. (2016). LSB Based Quantum Image Seganography Algorithm. *Int J Theor Phys*, *55*, 107-123. https://doi.org/10.1007/s10773-015-2640-0

[17] Bhardwaj, R. & Sharmab, V. (2016). Image Steganography Based on Complemented Message and Inverted bit LSB Substitution. *Procedia Computer Science*, *93*, 832-838. https://doi.org/10.1016/j.procs.2016.07.245

[18] Hajduk, V., Broda, M., Kováč, A., & Levický, D. (2016). Image steganography with using QR code and cryptography. *Proceedings of the 26th Conference Radioelektronika*, Košice, Slovak Republic. https://doi.org/10.1109/RADIOELEK.2016.7477370

[19] Zhou, X., Gong, W., Fu, W., & Jin, L. (2016). An Improved Method for LSB Based Color Image steganography Combined with Cryptography. *IEEE ICIS 2016*, June 26-29. https://doi.org/10.1109/ICIS.2016.7550955

[20] Kamdar, N. P., Kamdar, D. G., & N.khandhar, P. (1998). Performance Evaluation of LSB based Steganography for optimization of PSNR and MSE. *Journal of Information, Knowledge and Research in Electronics and Communication Engineering*, *2*(2), 505-509.

Contact information:

Savaş ÇITLAK, PhD Candidate
(Corresponding author)
Computer Engineering Department, Faculty of Engineering, Yildirim Beyazit
University,
Ayvalı Mah. Takdir Cad. 150 Sk. No: 5 Etlik-Keçiören / Ankara / Turkey
E-mail: savascitlak@gmail.com

Özkan KILIÇ, PhD, Asst. Prof.
Computer Engineering Department, Faculty of Engineering, Yildirim Beyazit
University,
Ayvalı Mah. Takdir Cad. 150 Sk. No: 5 Etlik-Keçiören / Ankara / Turkey
E-mail: ozkankilic@gmail.com