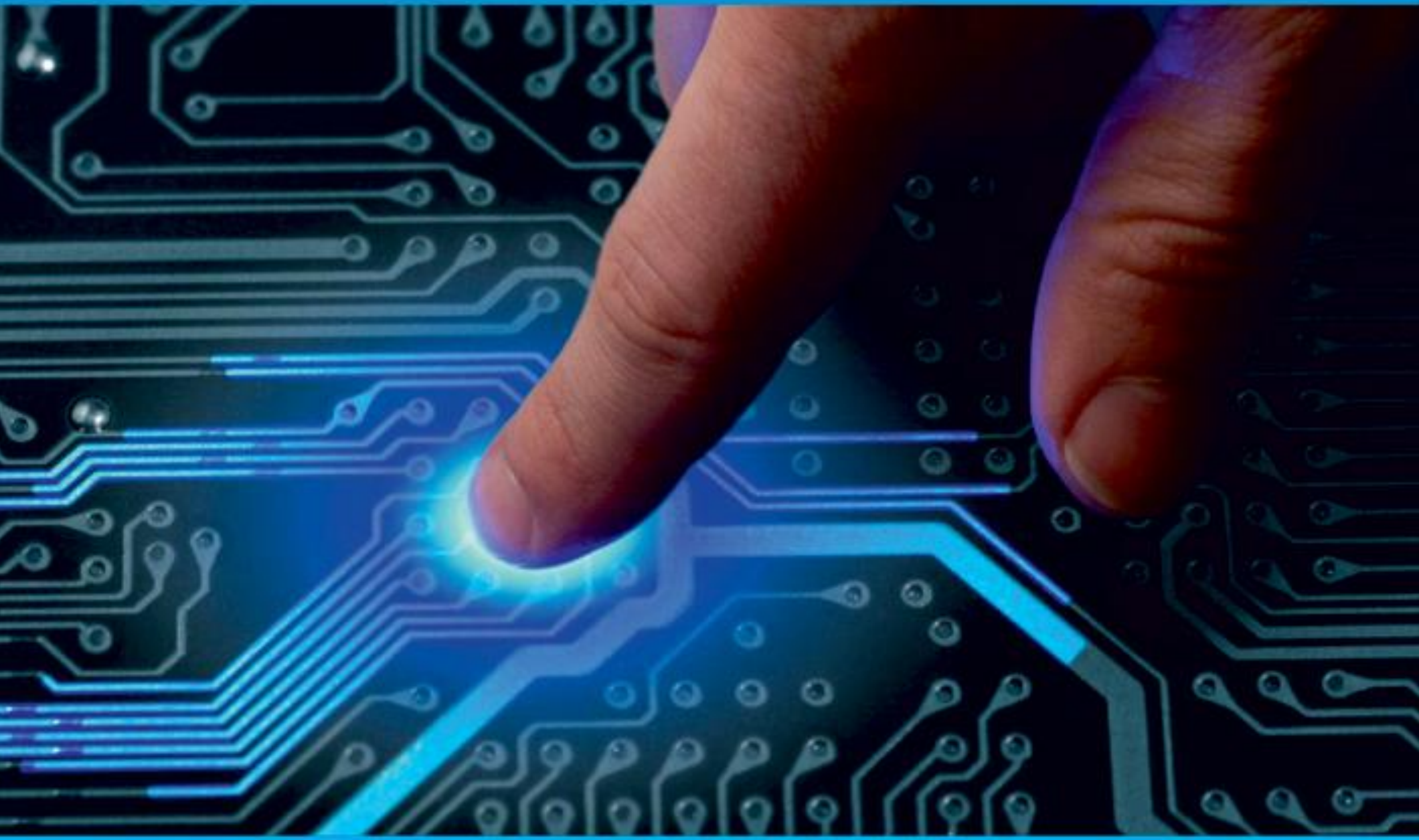
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 4, April 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104054 |

A Modified Siamese CNN Based Method for Image Steganalysis

Prasanth P Nair, Mohan S

Lecturer in Computer Engineering, Government Polytechnic College, Neyyattinkara, Thiruvananthapuram, India

Lecturer in Computer Engineering, Government Polytechnic College, Nedumangad, Thiruvananthapuram, India

ABSTRACT: To detect if an image contains some hidden data, image steganalysis is used. Current development in machine learning has helped in creating more advancements in the research of image steganalysis. Especially the usage of Convolutional Neural Networks has shown great results. The issue of arbitrary size image steganalysis explored based on the assumption that, natural image noise is similar in between different image regions. Therefore, a Siamese CNN-based architecture is used to create the network. Which consist of two symmetrical subnets with first convolutional layer using the TLU activation function. The whole architecture has three phases: pre-processing, feature extraction, classification.

KEYWORDS: Information security, Image steganalysis, Steganography, Deep learning.

I. Introduction

Image steganography is the technique used to hide data inside an image. There are different kind of image steganography such as HUGO, WOW, LSB. These steganography methods are used usually in applications such as sending undetectable messages through images. But it is also misused by terrorist and other organizations to communicate covertly, to resolve this issue image steganalysis is introduced. The image is analysed to find out whether it contains hidden data. There are billions of images getting transferred through the internet in a millisecond which explains the scale of this problem. Research in steganalysis is being done on recent years. At first the researchers used normal methods such as image quality matrices to analyse images. When machine learning created a revolution, different industries started using it. Image steganalysis based on machine learning started from there. Which gave greater results in the long run. SRM [1] models and SPAM [2] models are then created where hand crafted features are designed to make the best model. When CNN [3] was applied in image steganalysis, better results were produced. Different examples of CNN based steganalysis are SRnet, YeNet, SiaStegNet. Some of these models created new ways to solve the image steganalysis problem.

In this project, a different approach to image steganalysis is introduced by using a different activation function which is called TLU. Previous classical image steganalysis approaches have focused on the statistical properties of hand-crafted features between cover and stego images. Methods based on traditional machine learning are limited by their manuallydefined features. Most of them uses high pass filter to use the noise for classification. A recent paper has shown that even without using the high pass filter an image can be classified as stego or cover. When discussing about arbitrary sized image steganalysis, the past five years has shown that deep learning has emerged as a great way of steganalyzing arbitrary sized images. There are lots of methods that does image steganalysis comparatively better, but arbitrary sized image steganalysis is done by very few of them. [4]. However, because resizing the image decreases the signal-to-noise ratio between the cover image and the steganographic signal, there are currently only a few state-of-the-art approaches that can steganalyzearbitrary size images [5].

The system is created which is end-to-end, highly discriminative neural network for differentiating steganographic images from original images, one which provides a more satisfying performance in the evaluation of images of various sizes. This network adopts a Siamese, CNN-based architecture, which is used to capture relationships between image sub regions by using two supervisory signals simultaneously.

II. Problem Statement

The existing image steganalysis methods are based on CNN and machine learning. When deep learning introduced into Steganalysis the accuracy got improved. Current best models for image steganalysis uses deep learning to solve image steganalysis. CNN working was similar compared to the whole architecture of the handcrafted feature set models which helped in developing new CNN models. The main problem which was not explored in the previous research was arbitrary sized image steganalysis. It is considered as one of the objectives of the project, also improving the accuracy while doing arbitrary sized steganalysis.

III. Noise Residual

Since the steganography by cover modification makes only small changes to the pixels, working only on the noise [1] component (noise residual) of images will make the modelbetter. The noise component of an image is called noise residual

IV. SRM (Spatial Rich Model)Kernel

This specific Kernel is used to extract the noise component. This is a high pass filter which is basically used in the first convolutional layer to take the noise part from the image. The image processing layer filters the input image with the fixed high-pass KV filter kernel of size 5x5 to obtain a noise residual.

$$K_{kv} = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix}$$

Fig. 1. High-pass filter SRM kernal

V. CNN

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommended systems and natural language processing (NLP).

VI. TLU (TruncatedLinear Unit)

In contrast to CV (Computer vision) tasks or some high SNR applications, where the ReLU function can well adapt to the distributions of the object signals, the activation function adopted in steganalysis should take into account the structure of the embedding signals, especially in the first few convolutional layers. In image steganography, the embedding signals are usually in the range of -1 to 1, a new activation function known as truncated linear unit (TLU), which is slightly modified from ReLU is used. TLU defined as given below.

$$f(x) = \begin{cases} -T, & x < -T \\ x, & -T \leq x \leq T \\ T, & x > T \end{cases} \tag{1}$$

where T is defined by experiments. It can be seen that it is a modified version of Relu.

VII. LOSS FUNCTIONS

Neural networks are trained using an optimization process that requires a loss function to calculate the model error. In this project, two loss functions are used cross entropy loss and contrastive loss. Cross entropy loss is used for classification and contrastive loss is used for checking the similarity between two regions of the image. Cross entropy loss is defined as

$$L_{\text{CLS}}(p,y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{if } y = 0 \end{cases} \qquad (2)$$

VIII. OVERVIEW

Deep learning is a relatively powerful technique. The superiority of CNN-based steganalysis techniques over all previous methods have been proven repeatedly. However, when variations are made to the content, size or lighting of the image, or to the shooting equipment, the cover images may look very different. One of the main challenges of steganalysis is to develop robust supervisory signals in order to overcome these alterations.

The image sub regions relation is identified according to the classification. Which contains three phases pre-processing, feature extraction, and fusion/classification [7]. First, two subareas of the input image - ‘subi’ and ‘subj’ – separately enter the two parallel subnets. The subnets share structures, parameters, and weights. Each subnet consists of two phases: preprocessing, and feature extraction. At the front of each subnet, the preprocessing phase is used to produce image noise residuals, which are highly related to the steganographic signal. Next, the feature extraction phase is used to extract the feature vector of each sub-area noise residual. These two pieces of evidence - fsubi and fsubj - are imported into a symbiotic relationship within the original image. Finally, feature vectors of the two subnets are learned under the direction of two supervisory signals in the fusion/classification phase. The first is a classification signal, which classifies the fusion of feature vectors of two subnets with probabilistic values between the stego and cover; this is achieved by means of a two-class classifier consisting of a fully-connected layer, culminating in a Softmax layer with cross-entropy loss. The second is a similarity signal, which encourages feature vectors extracted from different image sub-regions of a cover image to become similar; this is achieved via contrastive loss, which is based on the Euclidean distance.

IX. SIAMESE NETWORK

The siamese network is used in the architecture [7], which is an important aspect of this project. Two symmetrical subnets with same weights and biases are called siamese network. Because of the assumption that two different regions of an image contain similar noise content these subnets are used to analyse different regions of the image. The weights of the first layers are initialized with learnable SRM kernels [1] which are used to extract the noise residuals. SRM kernel is important because if the raw pixels are used the whole system will fail. It is not a normal classification problem here the changes in the two images can’t be seen with human eyes. Therefore, taking raw pixels will result in bad outputs thus the noise residuals should be extracted and used for classification.

The subnets have two parts preprocessing part and feature extraction part. In preprocessing part the noise residuals is extracted and in feature extraction part the feature maps are created which led to the binary classification of the image given.

X. ARCHITECTURE

The whole architecture can be seen in Figure 2. The image from the input dataset is taken and then it is divided into two equal parts. These two sub areas of images are given as input to the two subnets from which the network is trained for classification of the image. Two loss function is used and a fusion function is also used.

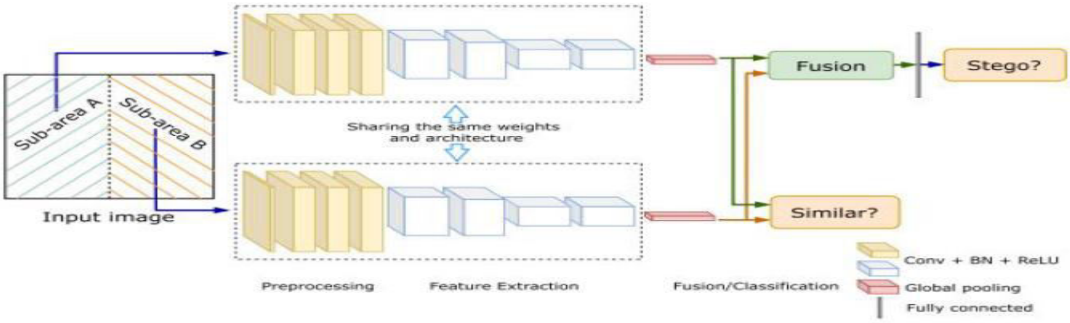


Fig. 2. Architecture

XI. SUBNETS

These are two deep convolutional network which is used to extract the features of the image to find out whether the image is stego(staganographic image) or cover(original image). These subnets have two phases which are:

- Preprocessing
- Feature extraction

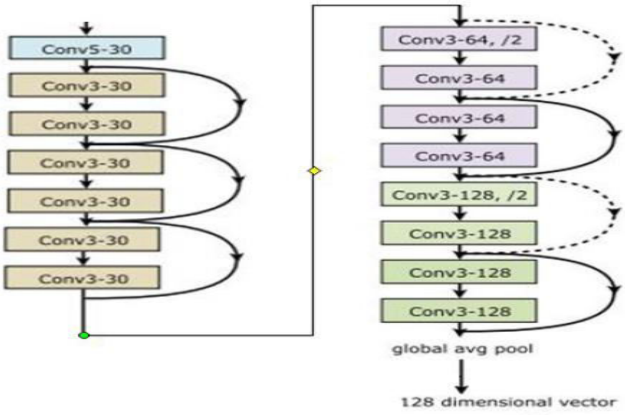


Fig. 3. Subnet structure

In the Figure 3 left part is the preprocessing part and right part is where feature extraction takes place.

A. Preprocessing

Raw image pixels are used in image classification tasks. Even though differentiating original images from steganographic images is a classification task, this specific classification is difficult because normal image signal has high

SNR (Signal to Noise Ratio) therefore training raw image data is not a good option. The embedded signals have very low SNR therefore training the noise residuals is far better than training raw pixels. To train the noise residual in the first layers of subnets, a high pass filter is used. This high pass filter is based on SRM (Spacial Rich Models). Which is basically a custom filter to extract the noise residuals from the image.

TLU is used in the first few layers of the convolutions. When Relu is used information loss was high which is due to ignoring the negative values. In TLU according to the definition it takes the values in a range which gives us more information to work on.

$$f(x)=\begin{cases} -T, & x<-T \\ x, & -T\leq x\leq T \\ T, & x>T \end{cases} \tag{3}$$

where T is a parameter determined by experiments.

B. Feature extraction

After the preprocessing in this phase two residual building blocks are alternatively applied to the feature extracted during the preprocessing phase. This part of the subnet is inspired from the ResNet. Then in the end a global average pooling is applied to get the 128-dimensional vector which will be the output of the whole subnet. Which is basically image representation of the corresponding side.

XII. Classification

The two supervisory signals are then used simultaneously.

A. Contrasive loss

It is applied between the two 128 dimensinal vectors to find out the similarities between the corresponding two sub regions. It is based on Euclidean distance.

$$\begin{gathered} L_{\mathrm{SML}}\left(\mathbf{f}_{\mathrm{sub}_i}, \mathbf{f}_{\mathrm{sub}_j}, y\right)=(1-y)\tfrac{1}{2}\left\|\mathbf{f}_{\mathrm{subl}}-\mathbf{f}_{\mathrm{sub}}\right\|_2^2 \\ +(y)\tfrac{1}{2}\left[\max\left(0, m-\left\|\mathbf{f}_{\mathrm{sub}_1}-\mathbf{f}_{\mathrm{sub}_j}\right\|_2\right)\right]^2 \end{gathered} \tag{4}$$

where m is the margin, y is the binary target stating whether the image is stego or cover.

B. Fusion

In this, the two 128-dimensional vector goes through a fusion function. This function four non-linear moments – the maximum, minimum, mean and variance - of fsubi and fsubj

1) Element wise maximum of fsubi,fsubj .
2) Element wise minimum of fsubi,fsubj .
3) Element wise Mean of fsubi,fsubj .
4) Element wise Variance of fsubi,fsubj .

After four element-wise statistical moments of two subnet outputs are calculated and concatenated. The resulting 512- dimensional vector, which captures information from the subareas and their relationships, is fed into a two-class classifier (a fully-connected layer, which culminates in a Softmax layer with cross-entropy loss). A dropout layer is also containedupstream from the classifier in order to prevent overfitting (dropout ratio set to 0.5).

C. Steganalysis via Deep Residual Network

This paper has investigated a category of very deep convolutional neural network modelthe deep residual networkforsteganalysis. Because of its large depth and new residual learning method, the deep residual network is naturally suitable for discriminating cover images and stego images. Extensive experiments on several challengingsteganographic algorithms validate that the deep residual network achieves significantly better performances than the classical rich model method and other CNN based methods. The DRN model network contains three sub-networks, i.e. the high-pass filtering (HPF) subnetwork, the deep residual learning sub-network and the classificationsub-network. These sub-networks have their ownroles in processing the information in the overall model. The limitation of this method was that the detector was essentiallytrained to recognize,when presented with batches of unlabelled cover stego pairs, which one of them is cover and which isstego,which is a significantly easier task.

XIII. Backpropagation

Using the two supervisory signal and loss functions the network is trained and weights and biases are optimized. This is shown in Figure 5.3. These two supervised signals are

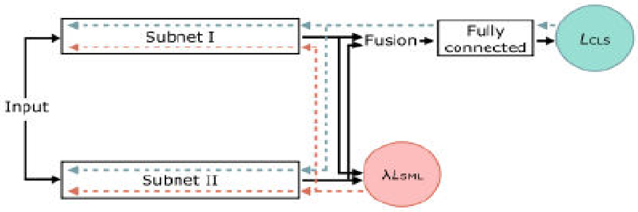


Fig. 4. Backpropogation

weighted by hyperparameter in the loss function used for our network:

$$L = L_{CLS}(p, y) + \lambda \cdot L_{SML}\ (\mathbf{f}_{sub_i}, \mathbf{f}_{sub_j}, y) \qquad (5)$$

XIV. Implementation Details

Adamax optimizer is used with _1 = 0.9 and _2 = 0.999 in order to minimize contrastive loss and binary cross-entropyloss. The balancing hyperparameter is set to 0.1, and the margin m of contrastive loss is set to 1. The initial learning rate is 0.001 and is divided by 10 whenever the error plateaus. L2 regularization is used to prevent overfitting of the model. The weight decay is set to 0.0001. The network is implemented using Pytorch version 1.4.0, and the reported runtime results are obtained using high end Nvidia GPU. The batch size is set to 32, and so requires about 6GB of GPU memory for training 256 × 256 images. The dataset is trained on Bossbase 1.01 dataset, a process which takes about 15 hours to complete. The experimental result is taken from the final results instance, it is evident that our network is computationally efficient as a 256 × 256 image can be judged within 0.01s.

A. Arbitrary sized image steganalysis

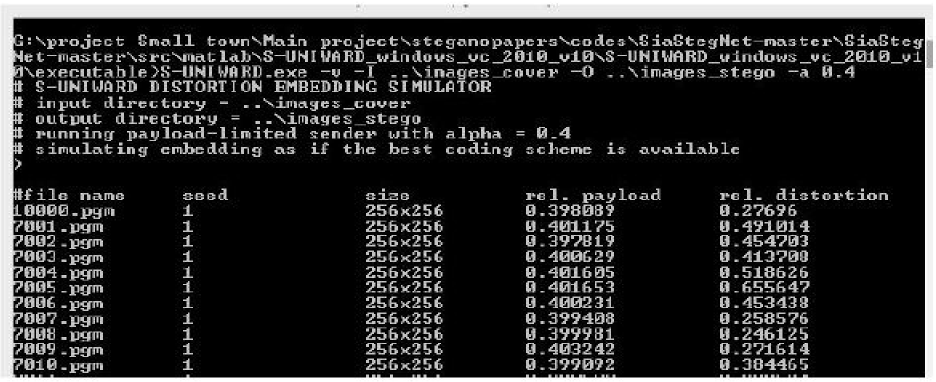
The main idea revolves around the relation between two sub regions of the images therefore arbitrary sized images can also be classified using a part of the image. A crop of an arbitrary sized image is taken using python code.

B. TLU Implementation

The first layer after SRM high pass filter is followed by TLU activation which takes all the values from -3 to 3 in modified model. This method improves the utilisation of data and makes a better model. It is implemented using Hard Tanh function which is found in Pytorch.

C. Creating Datasets

BossBase 1.01 database consist of 10000 images of 512x512 resolution. This project requires a set of cover images (native images) and corresponding stego images (Steganographic images). Therefore, it is required to build stego images from given 10000 images. As the first step the images are converted to 256x256 to train the network faster and for the sake of comparison. To resize the images, Bilinear interpolation algorithm is used (Python PIL library is used for applying the algorithm). After resizing S-UNIWARD distortion embedding simulator is used to create 0.4bpp payload stego images. Which is shown in Figure 7.1 below.



```
G:\project Small town\Main project\steganopapers\codes\SiaStegNet-master\SiaSteg
Net-master\src\matlab\S-UNIWARD_windows_vc_2010_v10\S-UNIWARD_windows_vc_2010_v1
0\executable>S-UNIWARD.exe -v -I ..\images_cover -O ..\images_stego -a 0.4
# S-UNIWARD DISTORTION EMBEDDING SIMULATOR
# input directory - ..\images_cover
# output directory = ..\images_stego
# running payload-limited sender with alpha = 0.4
# simulating embedding as if the best coding scheme is available
>

#file name      seed            size            rel. payload        rel. distortion
10000.pgm       1               256x256         0.398089            0.27696
7001.pgm        1               256x256         0.401175            0.491014
7002.pgm        1               256x256         0.397819            0.454703
7003.pgm        1               256x256         0.400629            0.413708
7004.pgm        1               256x256         0.401605            0.518626
7005.pgm        1               256x256         0.401653            0.655647
7006.pgm        1               256x256         0.400231            0.453438
7007.pgm        1               256x256         0.399408            0.258576
7008.pgm        1               256x256         0.399981            0.246125
7009.pgm        1               256x256         0.403242            0.271614
7010.pgm        1               256x256         0.399072            0.384465
```

Fig. 5. Embedding simulator

XV. Results and Analysis

Different objectives implemented results are evaluated in this section.

XVI. Loss Convergence Curve

The proposed model is trained on 6000 pair of images from the created dataset. The dataset is trained to 500 epochs. The Loss convergence plot in Figure 8.1 shows that the loss value decreases noticeably in the first 100 epochs and then becomes stable around 300th epoch of validation set.

XVII. Comparing with SiaStegNet

The proposed model is compared with existing model(SiaStegNet). The two models are trained on the same 6000 pairs of cover and stego images and accuracy in each epochs are plotted in the given graph in Figure 7. The curve shows that the proposed model is learning better and gives little bit higher accuracy than the base model(SiaStegNet).

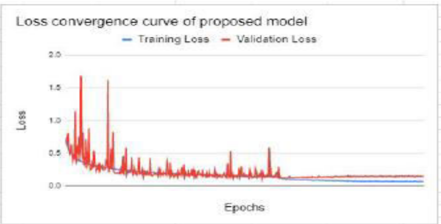


Fig. 6. Loss convergence curve

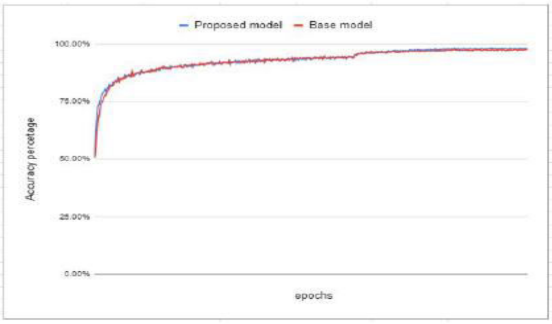


Fig. 7. Proposed model vs SiaStegNet

XVIII. Testing the Model

The model is tested using the 3000 pairs of stego and cover images which is never used to train or validate the model. The result is compared with existing methods which is shown in the table 8.1 given below. The given data show that the proposed model is giving 1% increment in accuracy than the existing methods. Also the number of parameters are 4.7 million in SRNet which is very high comparing to SiaStegNet and Proposed model. Where these two methods have only 0.7 million parameters. Therefore, we can infer that the proposed model has better performance and efficiency. SRNet is the state of the art method but it is based on deep residual network. It uses a lot of computation power and time because of its parameter size. When comparing with SRNet, SiaStegNet gives similar accuracy percentage. The proposed model is better than SiaStegNet but it is similar in accuracy as SRNet. Even though there are different steganography methods such as LSB and WOW which can be used to test on this model, to make it simple S-UNIWARD is used. Also the dataset created using LSB steganography is used in the base model where they got 98% accuracy which shows that LSB is a weak steganographic method therefore to make this balanced S-UNIWARD results are taken.

No	Methods	Parameters	Accuracy(0.4bpp)
1	SRNet	4.7m	94.68%
2	SiaStegNet	0.7m	94.30%
3	Proposed Model	0.7m	95.41%

TABLE I
COMPARING DIFFERENT METHODS TESTED ON S-UNIWARD STEGO DATASET

XIX. Conclusion

Recent developments in digital media steganalysis clearly indicate the immense importance of accurate models that arerelevant for steganalysis. This method gives a new way to solve this steganalysis problem by taking an assumption that two regions of a normal image might possess similarities in the noise ratio. Thus, the relationships between two image sub regions can be employed in an effort to improve steganographic feature distinction. The project proposes a new hybrid network which can work with more information by incorporating TLU activation in the first layer of Siamese network. By using this method performance improvement is achieved. For future work, the arbitrary sized image steganalysis can be improved by stretching the research on noise dense areas of the image.

References

[1] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3):868–882, 2012.
[2] Tom´aˇsPevny, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on information Forensics and Security, 5(2):215–224, 2010.
[3] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In 2016 IEEE internationalconference on image processing (ICIP), pages 2752–2756. IEEE, 2016.
[4] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 14(5):1181– 1193, 2018.
[5] Clement Fuji Tsang and Jessica Fridrich. Steganalyzing images of arbitrary size with cnns. Electronic Imaging, 2018(7):121–1, 2018.
[6] Shunquan Tan and Bin Li. Stacked convolutional auto-encoders for steganalysis of digital images. In Signal and Information Processing Association Annual Summit and Conference(APSIPA), 2014 Asia-Pacific, pages 1–4. IEEE, 2014.
[7] Weike You, Hong Zhang, and Xianfeng Zhao. A siamesecnn for image steganalysis. IEEETransactions on Information Forensics and Security, 16:291–306, 2020.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

📱 9940 572 462 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details