IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 27, 2020, accepted January 7, 2021, date of publication January 25, 2021, date of current version February 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053998

Image Steganography: A Review of the Recent Advances

NANDHINI SUBRAMANIAN[1], (Member, IEEE), OMAR ELHARROUSS[1], SOMAYA AL-MAADEED[1], (Senior Member, IEEE), AND AHMED BOURIDANE[2], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[2]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

Corresponding author: Nandhini Subramanian (ns1808900@qu.edu.qa)

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP11S-0113-180276.
Open Access funding provided by the Qatar National Library.

ABSTRACT Image Steganography is the process of hiding information which can be text, image or video inside a cover image. The secret information is hidden in a way that it not visible to the human eyes. Deep learning technology, which has emerged as a powerful tool in various applications including image steganography, has received increased attention recently. The main goal of this paper is to explore and discuss various deep learning methods available in image steganography field. Deep learning techniques used for image steganography can be broadly divided into three categories - traditional methods, Convolutional Neural Network-based and General Adversarial Network-based methods. Along with the methodology, an elaborate summary on the datasets used, experimental set-ups considered and the evaluation metrics commonly used are described in this paper. A table summarizing all the details are also provided for easy reference. This paper aims to help the fellow researchers by compiling the current trends, challenges and some future direction in this field.

INDEX TERMS Image steganography, GAN steganography, CNN steganography, information hiding, image data hiding.

I. INTRODUCTION

Technology has blitz scaled over the past years leading to a wide usage of multimedia for transferring data, especially Internet of Things (IoT). Usually, the transfer happens over insecure network channels. In particular, the internet has gained accelerated popularity for exchanging digital media and individuals, private companies, institutions, governments use these multimedia data transfer methods for exchanging data. Though there are numerous advantages attached with it, one prominent disadvantage is the privacy and security of the data. The availability of numerous readily available tools capable of exploiting the privacy, data integrity and security of the data being transmitted has made the possibility of malicious threats, eavesdropping and other subversive activities. The prominent solution is data encryption where the data is converted into a cipher text domain using encryption key. At the receiving end, the cipher text is converted into plain text using a decryption key. Using data encryption the original data is not visible, however, cipher text is visible in a scrambled form to human eyes leading to suspicion and further scrutiny. A new research topic, steganography, has gained acceptance in this context to hide the data that is not perceptible to human eyes.

Information hiding techniques have been available for a long time but their importance has been increasing recently. The main reason is the increase in the data traffic through the internet and social media networks. Though the objectives of cryptography and steganography are similar, there is a subtle difference. Cryptography makes the data unbreakable and unreadable but the cipher text is visible to human eyes. Steganography, which is used to hide the information in plain sight, allows the use of wide variety of the secret information forms like image, text, audio, video and files. Digital watermarking is another method where confidential information is embedded to claim ownership. Cryptography is the popular method used for information hiding, but, steganography is gaining popularity in recent times.

Steganography can be defined as the process of hiding a secret small multimedia data inside another but much larger

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

multimedia data such as image, text, file or video [1]. Image steganography is a technique to hide an image inside another image. In image steganography, the cover image is manipulated in such a way that the hidden data is not visible thus making it not suspicious as in the case of cryptography. Inversely, Steganalysis is used to detect the presence of any secret message covered in the image and to extract the hidden data [2]. Steganalysis helps in classifying if the image is either a stego image or a normal image. Apart from classifying the image, further investigation is carried out to detect the location and the content of the secret image inside the cover image.

With the availability of massive amounts of data, deep learning (DL) has become the trend and is extensively used for many applications. Deep learning is a useful tool in various applications like image classification, automatic speech recognition, image recognition, natural language processing, recommendation systems, processing of medical images [3]. Though research on steganography is quite recent, it has benefited from DL methods including convolutional Neural Networks (CNNs) Generative Adversarial Networks (GANs) based methods and their deployment in both steganography and steganalysis.

The main goal of this paper is to review the available methodologies, present trends and discuss the challenges that are currently available in the studies. Along with these studies, the datasets that are publicly available and commonly used, the evaluation metrics considered are also discussed. Finally, a comparison on the performance among the methods and a possible discussion identifying the gaps in the present studies, pros and cons of the methods are elaborated.

The remaining paper is organized as follows. Section II summarizes the working principle of the methods grouped into three categories - Traditional methods, CNN-Based methods and GAN-Based methods. Datasets used commonly are elaborated in section III along with evaluation in section IV. A table with the comparisons of the results from the different methods are provided with the experimental set-ups generally used in section IV. Finally, the challenges faced, a brief discussion, and conclusion are added in section V, VI and section VII respectively.

II. SUMMARY OF THE METHODS

After reviewing all the frameworks available, the methodologies are primarily grouped into three categories, namely, traditional image steganography methods, CNN-based image steganography methods and GAN-based image steganography methods. Traditional methods are frameworks which use methods that are not related to machine learning or deep learning algorithms. Many traditional methods are based on the LSB technique. CNN-based methods are based on deep convolutional neural networks for embedding and extracting the secret messages and GAN-based methods use some of the GAN variants. Figure 1 gives an overview of a steganography and steganalysis architecture. As shown in figure 1, inputs are cover image and the secret information which can be either text or image. DL model can be either a CNN-based or GAN-based. While the steganography block generates the stego image, the steganalysis model takes the stego image as input to detect and perhaps extract the secret information. In some methods the probability score of the input image being normal or stego image is given as the output.

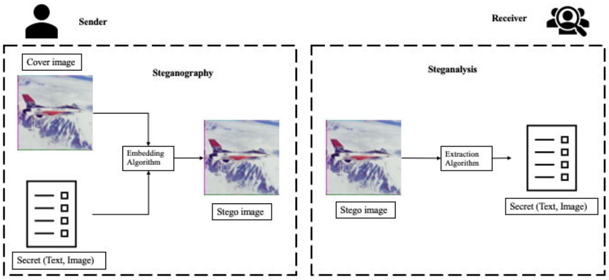


FIGURE 1. **General working principle of steganography and steganalysis. The inputs are the cover image and the secret information and an embedding algorithm is used to generate the container stego image. The extraction algorithm takes the stego image as input to extract the ingrained secret information.**

Text data, color or grayscale images are usually used as secret media. Two factors - the nature of the secret media, and the technique used are considered in classifying the works available to different categories as can be seen in the figure 2. From the analysis on figure 2, it is observed that text is the most commonly used secret information and GAN-based methods are the preferred mode for secret communication for text hiding.

A. TRADITIONAL-BASED STEGANOGRAPHY METHODS

Conventionally, Least Significant Bits (LSB) substitution method is employed to perform image steganography. Images are usually of higher pixel quality, out of which not all the pixels are used. LSB methods works under the assumption that modifying a few pixel values would not show any visible changes. The secret information is converted into a binary form. The cover image is scanned to determine the least significant bits in the noisy area. The binary bits from the secret image are then substituted in the LSBs of the cover image. The substitution method has to be performed cautiously as overloading the cover image may lead to visible changes leaking the presence of the secret information [4] and [5].

With the LSB method as the baseline, a number of related methods have been proposed. For example, a slight variation in converting the secret message into binary codes is undertaken in [6]. A Huffman encoding method is used to encode the secret message into the binary bits. The encoded bits are then embedded in the cover image using the LSB method. In [7], another version of the LSB method is used for RGB images. The cover image is in 3 channels and they are bit sliced. The secret message is embedded in all the three planes in the 2:2:4 ratio for R, G and B planes. Not only spatial domain, quantum images are also used [8] and [9]. The frequency domain is exploited in quantum image domain and the pixels which are considered to be affecting the color are

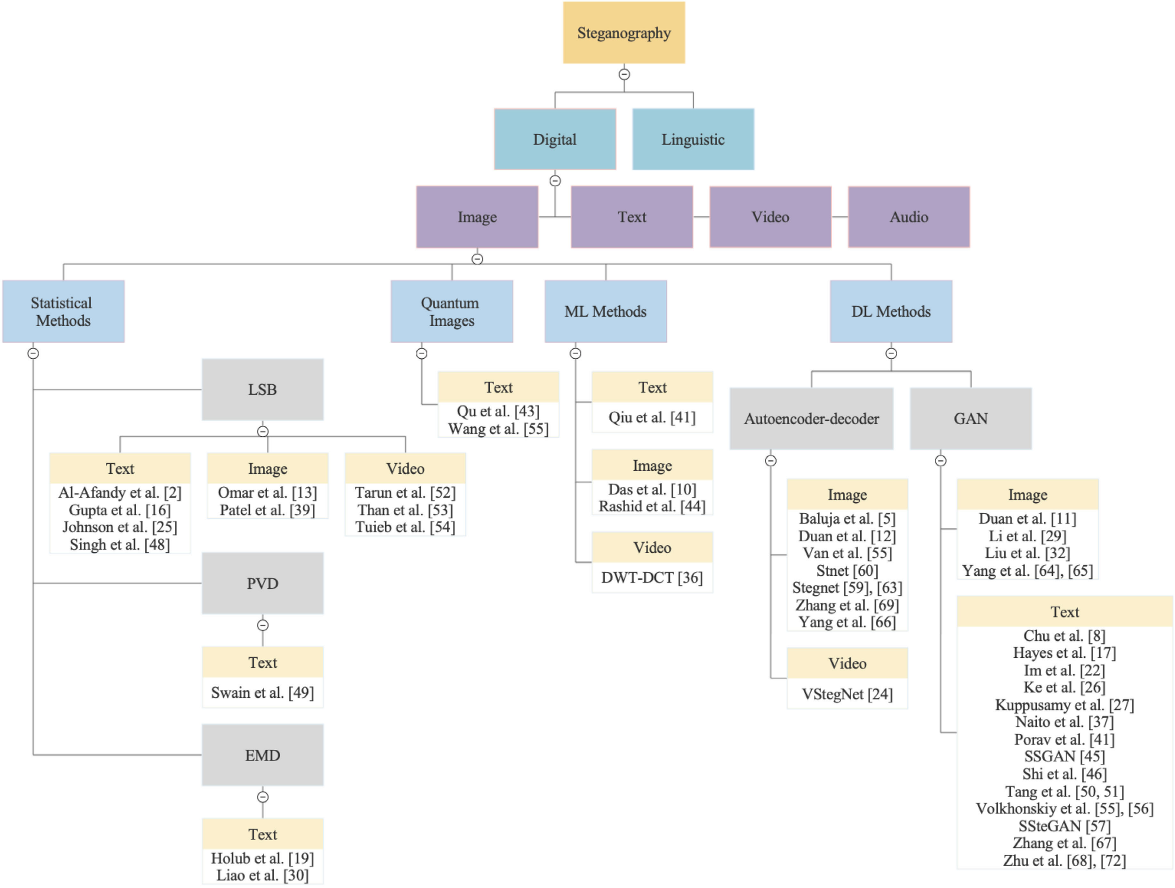


FIGURE 2. **Classification of the existing methods based on the secret media and the method used. First, the methods are classified and under each category, further classification is done based on the secret media.**

used to hide the secret bits. A combination of cryptography and steganography is utilized where the LSB of the cover image is replaced with the most significant bits of the secret image [10]. The pseudo random number generator is used to select the pixels and the key is encrypted using rotation every time. A k-LSB method is proposed where the k least bits are replaced with the secret message [11]. For steganalysis, an entropy filter is used to detect and uncover the secret image [11].

The LSB methods are used in hiding the secret information inside videos also. Videos are sequences of images called the video frames. Each video is dissected into image frames and the binary bits of the secret information is hidden in the LSB of the image frames of the video. A basic form of LSB substitution method [12] and a combination of the huffman encoding and LSB substitution methods is used on videos [13]. Another interesting approach is where along with the image frames of the video, audio is also used to enhance the hiding [14]. Besides the LSB methods, [15], has proposed a combination of Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT) for hiding the secret message inside a cover video. To find the regions of interest, the multiple object tracking (MOT) method is used. The secret data is encoded first and then converted to binary bits before embedding it in the cover video.

Table 6 depicts the detailed review on the traditional image steganography methods. Another classical method used in image steganography field is the Pixel Value Differencing (PVD). PVD works by taking the difference of consecutive pixels to find the locations for hiding the secret bits in such a way the consistency of the cover image is maintained. For every 8 bits, a combination of LSB on the first two bits and PVD on the remaining six bits is designed [18]. In addition, some other techniques used are the coverless steganography where the cover image is not given rather it is generated based on the secret information. The secret information is taken and a relationship management is performed to produce the cover image using the object detection method. Similar coverless steganography is proposed where the Local Binary Patterns (LBP) features of the cover image and the secret images are hashed firstly. Later, the hashes are matched to create the stego image [19]. Similarly, instead of LBP, the edges of the color cover images are obtained. Then, the binary bits of the secret information is hidden in the edges discovered in the cover images [20].

TABLE 1. Summary of the details on the traditional methods.

Method	Dataset	Metrics	Advantages	Disadvantages
[16]	1 RGB image	PSNR and Time	- Less computation time - Robust in both embedding and extracting - Steganalysis and steganography can work without dependency	- Less Secure - Secret information is text
[17]	Lena and Baboon	PSNR and MSE	- Less computation time - Image is secret message - Arbitrary image format is accepted	- Security is less compared to deep learning methods
[10]	Lena	PSNR	- Less computation time - Image is secret message	- Less secure

Other methods of the traditional steganography methods are described below. Medical JPEG images are used and the embedding is implemented by taking the difference in the DCT coefficients of the pixels between two consecutive blocks by considering the pixels at the same positions in the two blocks [21]. A novel method called the Pixel Density Histogram (PHD) is proposed for halftone images [22]. The pixel density of the images are calculated and a pixel density histogram is formed to hide the secret information. The Poisson distribution is utilized to get the burst error and the reconstruction of the images that are compression-resistant is done using the STC decoding [23]. For more clarifications and explanations on the traditional steganography methods, [24] can be referred.

B. CNN-BASED STEGANOGRAPHY METHODS

Image steganography using CNN models is heavily inspired from the encoder-decoder architecture. Two inputs – cover image and the secret image are fed as the input to the encoder to generate the stego image and the stego image is given as input to the decoder to output the embedded secret image. The basic principle is the same except different methods have tried different architectures. The way the input cover image and the secret image are concatenated are also different in different approaches while the variations in the convolutional layer, pooling layer are expected. The number of filters used, strides, filter size, activation function used and loss function vary from method to method. One important point to note here is the size of the cover image and the secret image has to be same, so every pixel of the secret image is distributed in the cover image.

Wu *et al.* [25] and [26] have proposed a encoder-decoder architecture. U-Net based encoder-decoder architecture is used for hiding and a CNN with 6 layers for extraction is proposed by Duan et al. in [27]. The input shape of the U-Net is modified to accept 256 × 256 and 6 channels. The secret and cover images are concatenated to give the input and hence 6 channels. A U-net based Hiding (H-net) and revealing (R-net) network are used by Van *et al.* in [28]. Batch normalization and ReLU activation are used. The cover and the secret images are concatenated before being sent to the network. Two optimization losses using SSIM and MSE are used to reduce the loss and hence improve the performance.

A Separable Convolution with Residual Block (SCR) is used to concatenate the cover image and the secret image [25]. The embedded image is given as the input to the encoder for constructing the stego image which is fed to the decoder to output the decoded secret image. ELU (Exponential Linear Unit) and Batch normalization are used. A new cost function to reduce the effect of noise in the generated container image called the variance loss is proposed [26]. An encoder-decoder architecture was proposed by Rahim *et al.* in [29]. This method differs from the others in the way the inputs are given. The encoder part consists of two parallel architectures each for the cover and the secret image. Features from the cover image and the secret images are extracted through the convolutional layer and concatenated. The concatenated features are used to construct the stego image.

A slightly different approach is proposed by Wang *et al.* in [30] by considering the styling image along with the secret information and cover image. The created stego image is converted into the style image given as an input. The reveal network is used to decode the secret information from the stego image created. Similar to other methods, an auto encoder-decoder architecture with VGG as base is used. An arbitrary image size for the secret information and styling image are taken using the adaptive instance normalization (AdaIN) layer and the output is the size of the cover image. Pixel distribution of the cover image are got by using a pixel CNN by Yang *et al.* in [31]. The secret information is then embedded in the pixel distribution evenly by reduced sampling.

Three networks namely prep-network, hiding network and reveal network are proposed by Baluja *et al.* in [32] based on auto-encoder architectures as a single model. The Prep-network is used to prepare the secret image before feeding it as input to the hiding network, which takes the output of the prep-network and the cover image to produce the container image. The reveal network decodes the secret image from the container image by uncovering the cover image. Two losses are calculated between the cover and the constructed container and between secret and the decoded image. The model is evaluated using the structure similarity index (SSIM). An extension to [32] is proposed by Zhang *et al.* in [33] where the cover image is converted into an YCrCb image format and the secret image is hidden to only the

TABLE 2. Summary of the details on the CNN-based steganography methods.

Method	Architecture	Dataset	Advantages	Disadvantages
[31]	Encoder-decoder	ImageNet	- Image is secret message	- However image size is 64 × 64 which is very small
[27]	U-Net	ImageNet	- Image is secret message - Basic and minimum architecture is used	- However image size is 64 × 64 which is very small - Input images are just concatenated
[28]	CNN	ImageNet and Holiday	- Image is secret message. Basic and minimum architecture is used - New error back propagation function is introduced to speed up training	- However image size is 64 × 64 which is very small - Input images are just concatenated
[29]	Encoder-decoder with VGG base	COCO and wikiart.org	- Domain-knowledge is not required - Highly secure as the generated image is not related to the secret information	- Computation is increased by using additional image
[25] and [26]	Encoder-decoder with SCR	ImageNet	- Highly secure and robust	- Loss used is not optimal - Visible noise can be seen in black or white regions

Y channel as all the semantic and color information are present in the Cr and Cb channels. Also, to reduce the payload by two thirds, the secret image is converted into grayscale image format. The Y channel of the cover image and the grayscale secret image are given as the input to the encoder-decoder network for constructing the stego image. The output stego image is the Y channel which is combined with the Cr and Cb channels of the cover image to produce the stego image in the YCrCb color space. To extract the secret image, the Y channel of the stego image is again given to the revealing network to output the grayscale secret image. Two different variations – basic and residual models are also used in the generative models. A mixed loss function which is more suitable for the steganography using SSIM and its variant multi-scale structure similarity index (MS-SSIM) are also used. For the concealing network, ISGAN architecture is used. Table 2 summarizes the review on CNN-based steganography methods.

Not only image steganography, but also, video steganography is tried using CNN. Usually, 2D convolutional layers are used for images whereas 3D convolutional layers are used for videos. Temporally connected cover and secret video frames are given as the input to autoencoder network based VStegNET [34] to produce the container video. Each frame of the cover image is concatenated with every frame of the secret video to produce the container video. An identical network architecture is used to reveal the hidden secret video.

C. GAN-BASED STEGANOGRAPHY METHODS

General Adversarial Networks are a type of deep CNNs introduced by Goodfellow *et al.* [35] in 2014. A GAN uses the game theory to train a generative model with adversarial process for image generation tasks. Two networks – generator and discriminator networks compete against each other to generate a perfect image in GAN architecture. The generator model is given the data and the output is the close approximation of the given input image. The discriminator networks classifies the images generated as either fake or true. The two networks are trained in such a way that the generator model tries to imitate the input data as close as possible with minimum noise. The discriminator model is trained to effectively find out the fake images. Many variations on GAN have been proposed ever since, making it more powerful and suitable for synthetic image generative tasks.

GANs are known for their good performance in the image generation field. Image steganography can be considered as one such image generation task where two inputs – the cover image and the secret image are given to generate one output – stego image. The existing methods used for image steganography using a GAN architecture can be grouped into five categories - a three network based GAN model, cycle-GAN based architectures, sender-receiver architecture using GAN, coverless model where the cover image is generated randomly instead of being given as input and an Alice, Bob and Eve based model. Details on all the categories and how they are executed are given below.

Generally, a GAN model consists of two main components: the generator and the discriminator. In the context of image steganography, a new network named the steganalyzer is introduced in some of the methods. The main functions of these three components are,

- A generator model, G, to generate stego images from the cover image and the random message.
- A discriminator model, D, to classify the generated image from the generator as either real or fake.
- A steganalyzer, S, to check if the input image has a confidential secret data or not.

The three models, G, D and S are made to compete against each other to produce realistic images close to the cover image given as input. The errors of D and S along with the parameter alpha between [0, 1] are used to produce the realistic image and their quality for steganalysis. One main difference from the GAN here is that G is updated in order to maximize not only the error of D, but to maximize the error of the linear combination of the classifiers D and S.

Volkhonskiy *et al.* [36], [37] has introduced DCGAN [38] based Steganographic GAN (SGAN) which is a simple DCGAN with three modules - G, D and S. Similar to [36], Shi *et al.* [2] and [39] has proposed a three component GAN architecture. The only difference is the architecture used by

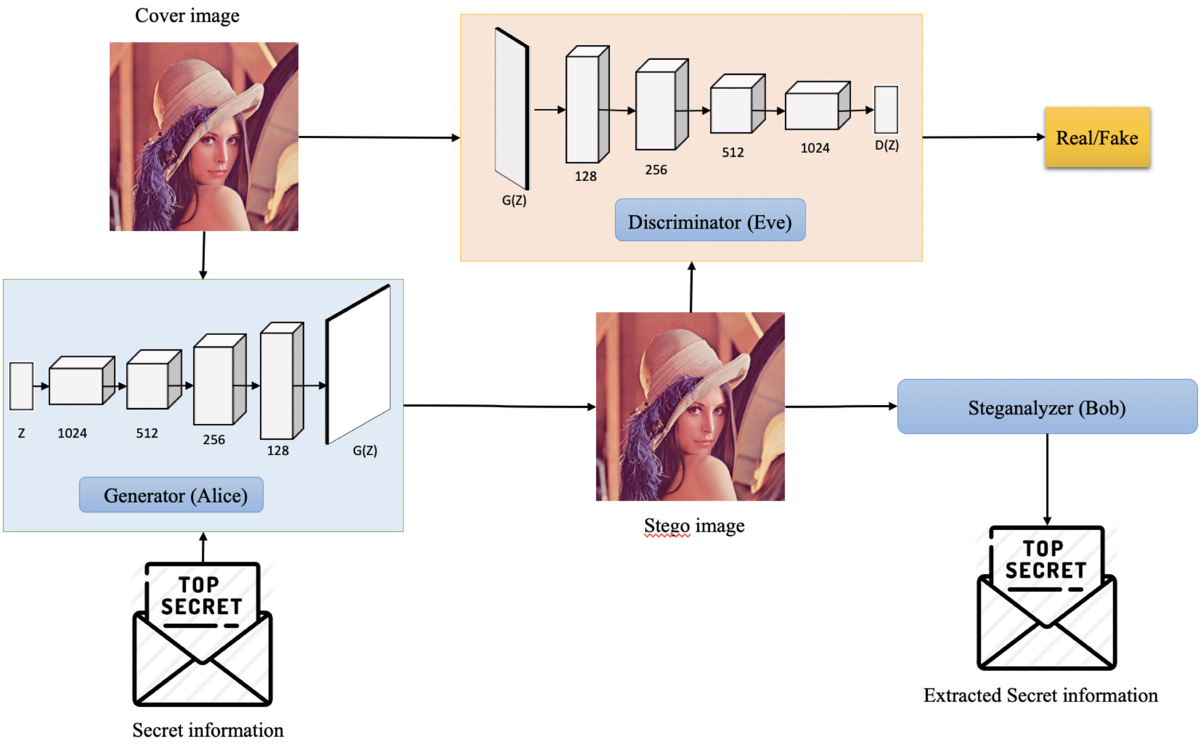


FIGURE 3. General working overflow of SGAN and [40]. The generator takes the cover image and the secret information (usually text) as the input to produce the stego image. The discriminator competes against the generator to classify the generated image as either real or fake. A steganalyzer is added to extract the secret information from the stego image generated.

these methods. A DCGAN is used in [36], and [37] whereas a four fractionally convolutional layer followed by a functional layer with Hyperbolic tangent activation with base WGAN is used in [2] and [39]. Three models compete against each other in such a way that the generator produces the stego images and the discriminator decodes and recovers the secret message while the stegananlyzer eavesdrops on the generator produces the probability. Figure 3 represents the working principle of SGAN.

An embedding simulator is sometimes used in the place of the steganalyzer [41], [42] and [43]. Yang *et al.* [41] and [42] presented a GAN based image steganography with three modules – Generator, Embedding Simulator and Discriminator. The authors in [41] have called the architecture a UT-GAN since they used the U-Net based architecture in the generator network. The generator is used to create the probability map, P, for the input cover image, C and U-Net is considered because of its effective performance in pixel wise segmentation. The U-Net used here also has an expanding and contracting convolutional layers with increasing filter number for convolution and decreasing filter numbers for deconvolution layers. Also, the layers I from the down sampling part is concatenated with the layers L - i from the up sampling part to help in backpropagation process. The probability map, P, from the generator and the random message are given as the input to the embedding simulator to output the modification map, M. The embedding simulator uses a double tanh function [42] and a tanh function [41] because the tanh is differentiable and will preserve the gradient loss during back propagation. The discriminator acts as a steganalyzer with multiple high pass filters for pre-processing to improve the detection accuracy. There are two losses to be considered here – the discriminator loss and the generator loss. The discriminator loss is utilized to distinguish between the cover image and the stego image while the generator loss considers the target payload also. Therefore, the generator loss is made up of two losses – the entropy loss to guarantee the embedding capacity and the adversarial loss to increase the anti-detectability rate.

Automatic Steganographic Distortion Learning framework with GAN (ASDL-GAN for short) was introduced by Tang *et al.* in [43]. In this architecture, the generator is used to learn the probabilities for each pixel from the input cover image and the authors have proposed to use a novel activation function Ternary Embedding Simulator (TES) for generating the stego images from the generated probabilities. The discriminator helps in differentiating between the real and fake images. XuNet based architecture is used for discriminator D.

A sender-receiver type of architecture is another way for image steganography using GAN [33], [43] and [44]. SteganoGAN, a GAN model with encoding network for creating the stego images and decoding network to get the secret message from the stego image has been proposed by Zhang *et al.* in [45]. The secret message is embedded in the cover image using the encoder and the secret message is recovered back by the decoder and a critic is used to evaluate the quality of the images generated. Three variants of the encoder – basic, residual and dense are also discussed to deal with different payload capacity.

Three models – one basic model and two enhanced models are introduced by Chen *et al.* in [43]. All the three models are intended to perform steganography using the encoding network and the safe retrieval of the secret image using decoding network where the enhanced models are more secure and robust. The basic model uses the encoding network to hide grayscale secret image into the color channel B of the cover image to produce the stego image. In the same way, the decoding network reveals the secret image from the color channel B of the stego image. The reasons for using color B channel is that the impact of blue color on human eyes is less than red and green. A steganalysis network and attack network are added in the basic model to make it more secure and robust in enhanced models. XuNet with a Spatial Pyramid Pooling layer is used in the steganalysis network.

Generator model is used as sender and the discriminator is used as the receiver by Naito et al [44]. The generator in the sending end generates the stego images and the discriminator eliminates the images that are not realistic. At the receiving end, the discriminator classifies the sender of the stego image received to being true sender or third party to prevent the generator from wasting time on third party stego images. The generator in the receiving end decodes the stego image and uncovers the secret data. Both the sender and the receiver shares the same trained generator and discriminator for consistent results.

HIDDeN is a GAN based method with four main components – encoder, decoder, discriminator and a noise layer proposed by Zhu *et al.* [46]. The encoder takes the cover image and the secret message as input and creates an encoded image which is fed to the noise layer to produce the noised image. The decoder decodes the secret message from the noised image and the discriminator helps in giving the probability of the given image being encoded or not. Ke *et al.* [47] have proposed a slightly different architecture by using a Generative steganography with Kerckhoffs' principle (GSK). Instead of modifying the cover image and converting it into stego image, a new stego image with the secret message is generated. The extraction key and the generated stego image are given to the discriminator for decoding. Without extraction key the discriminator cannot decode the secret image. The generator is publicly available, the sender first gives the cover image to get the extraction key. Then the cover image and the extraction key are sent to the receiver. In the case where the cover image or extraction key are sent alone, only noise is given as output. The discriminator can output the secret message only when both the image and the extraction key are present.

CycleGAN [48] is well-suited for image steganography where the input image is given and the output similar to the given input image but with hidden information using the adversarial training is generated. The input image is first converted to a target domain image and then back to the source image eradicating the necessity for output image. The original cycleGAN method has been modified a little to fit the image steganography methodology perfectly for hiding secret message inside the cover image [49]–[51], and [52]. Firstly, the RGB cover image is converted to grayscale image and then the luma regions in the image are extracted [49]. The secret message is embedded in the LSB bit of the luma field. The generator then creates the new image with the secret message embedded. The generated image and the cover image are given to the discriminator to classify it either fake or real. A denoiser is introduced between the two generators of the cycle GAN to reduce and filter the low amplitude, high frequency messages and noise that the discriminator cannot see [52]. A cycle GAN is used for steganography and steganalysis in a covert communication to prevent the security breach and privacy preservation in IoT by Meng *et al.* in [52].

ACGAN [53] can generate realistic images for a given label and also recognize the label of the generated images. Three steps are followed to hide information which are generated by the model and extract the hidden information [54] and [55]. First, a word segmentation dictionary and image database is established. A generative hiding model named Stego-ACGAN is developed. Finally, a hiding and extracting algorithm for hiding and retrieving the information is designed. A database is set-up pairing the word segmentation as the label and the corresponding images. Like in any other ACGAN, stego-ACGAN also has three neural networks, namely. The generator, the discriminator and the auxiliary classifier. After training the stego-ACGAN, a secret channel is used to share the model parameters and the constructed image and word segment database to the other party. The secret information is divided into segments and a binary code is generated. Each code is then given a label and the mode generates a sequence of images for the segments. The image sequences and the input noise are given to the generative model to generate the stego images. At the receiving end, the noise is removed and the image sequences are extracted. Then the image sequence is given to the auxiliary classifier to get the secret information.

Similar to [54] and [55], the authors in [56] Duan *et al.* have proposed a coverless steganographic method using a generative model. Instead of transmitting the secret image as such, a new meaning-normal image totally not related to the cover image is generated using the generative model. At the receiving end, the transmitted meaning-normal image is fed to produce the secret image. A WGAN [57] is used as the generative model by Li *et al.* in [58]. A framework where a textural image is generated by a generative model and acts as a cover image is proposed. Then, this cover image and the secret image is given to the concealing network for hiding the secret image inside the cover image. So the final image is a texture based image concealed with secret information.

An adversarial learning based method with three components – Alice, Bob and Eve is proposed by Hayes *et al.* [40].

TABLE 3. Summary of the details on the GAN-based steganography methods.

Method	Architecture	Dataset	Advantages	Disadvantages
[39]	Alice, Bob and Eve	BOSSbase and celebA	- Domain knowledge is not required for embedding	- Message is used rather than images - Grid search for selecting the embedding scheme is time consuming
[37]	Info-GAN	MNIST and celebA	- Additional security is provided by adding an extraction key along with the cover image for the generator	- Message is encrypted using extraction key generator. Both the extraction key and generator trained are publicly available which makes it prone to attacks - No privacy plans in place to avoid this attacks
[2]	DCGAN	CelebA and BOSSBase	Game-theoretic formulation is used	- Use of three components. Steganalyzer is used to provide the probability alone which leads to more computational overhead
[54]	WGAN	CelebA	- Image is secret message	- If the generative model is not working properly, the secret image will be lost
[53]	ACGAN	MNIST	- Highly Secure and robust with increased hiding capacity	- Complicated design for hiding the secret information. First images are generated as per the secret information and additionally noise to be added - A secure channel is required to pass the database of the word segment and the corresponding image
[47]	Cycle GAN	USC-SIPI	- Better performance and security than LSB methods	- Text information are hidden. The main question here is how to extract the hidden message back from the stego image. The proposed method is not complete - Convergence loss of GAN can cause mode collapse
[56]	DCGAN	DTD and COCO2017	- Secret image can be of any width and height	-Texture based images are only produced
[50]	Cycle GAN	ImageNet and Photo2Monet	- Securing of the medical data to enhance the privacy of the medical records	- Cycle GAN is used for masking the real data in IoT. The use of steganography in the place of encryption is not explained
[43]	GAN	CelebA	- Generation of unlimited number of cover images are possible	- Shared generator and discriminator. User has to choose the seed value instead of the cover image - Generated images are not natural
[38]	DCGAN	CelebA	- Generation of more realistic images - Highly secure	- The steganalyzer gives probability rather than the secret information - No explanation on how to uncover the secret message
[42]	ASDL GAN	BOSSBase	- New activation function to generate the stego images	- ASDL-GAN is still inferior to state-of-the-art hand-crafted steganographic algorithms

The general working principle is that Eve eavesdrops between Alice and Bob to check if there are any secret message embedded in the communication channel between them. The authors in [40] have used neural networks to train all the three components. For the steganographic scenario, Alice is trained to create the steganographic image while Bob recovers the secret message from the stego images. Eve helps Bob by giving the probability of the given image being a stego image. A model with four parts – Alice, Bob, Dev and Eve has been proposed by Wang *et al.* in [59]. Since the model is an unsupervised generative model, the authors have named the model Self-supervised Steganographic GAN (SSteGAN). Like in any other communication security paradigm, Alice and Bob try to communicate secretly while Eve eavesdrops on the communication channel. As such, Alice acts as the generator, Bob as the decoder, eve as the steganalyzer to classify if the given image is normal or stego image, and Dev here acts as the discriminator actively competing against Alice. Along with secret message, input noise is also given as input to Alice, to avoid generating identical images if the same secret message is given twice. This diverts any suspicion created and enhances the security of the model. Eve and Dev gets both the real image and the generated image. While Eve helps in distinguishing the stego and the cover image, Dev helps in classifying the image as real or fake. A detailed summary of the methods reviewed under GAN-based method topic is given in table 3.

III. DATASETS USED

There exist one dataset, BOSSBase, that was specifically created to deal with the problems of steganography. To further evaluate the performances of the algorithms some existing datasets, which are used for other purposes including object recognition and face recognition, are re-modeled to fit for the purpose of our experiments. A detailed explanation on the datasets are described in table 4.

A. BOSSBASE

Break Our Steganographic System (BOSS) [60] is the first scientific challenge conducted to take image steganography

TABLE 4. Information details of the dataset used in the literature of steganography.

Dataset	Number of samples	Images format	Image Size	Purpose
BOSSBase	9074 training and 1000 testing	tiff	512 × 512 × 1	Steganography and steganalysis
CelebA	More than 200K	jpg		Face attribute recognition, face detection, landmark (or facial part) localization, and face editing and synthesis
ImageNet	More than 14M	Arbitrary	Arbitrary	Computer Vision
MNIST Handwritten Digits	60,000 training and 10,000 testing	idx	28 × 28 × 1	Image processing and computer vision
COCO	330K	jpg	640×640×3	Object detection, segmentation and image captioning
Div2K	800 training, 100 validation and 100 testing	png	1020×678×3	Single Image Super-Resolution
SZUBase	40000	-	512×512×1	Steganography and steganalysis
USC-SIPI	170	tiff	256×256×1, 512×512×1, or 1024×1024×1	Image processing, image analysis, and machine vision
DTD	5640	jpg	300×300×3 and 640×640×3	Textural analysis
LFW	13233	jpg	150×150×3	Face verification
Pascal VOC	11530	jpg	500×300×3	Object detection and classification

from being a research topic to a practical application. The main aim of the competition was to develop a better steganalysis method that can break the steganographic images created by the HUGO (Highly Undetectable steGO) algorithm [61]. The dataset consists of a training set and testing set along with the HUGO algorithm that can be used to create the steganography images. The training dataset consists of 10,000 grayscale cover images with dimensions 512 × 512. The testing set consists of 1000 grayscale images with dimensions 512 × 512. There is an option to download the datasets with steganography images solely for the purpose of steganalysis. Firstly, the raw images are captured using 7 different cameras and they are converted to PGM images. The links to download the raw images, PGM images, the script used to convert the raw images into PGM, EXIF data of the raw images can be found in the official website.[1]

B. CELEBA

Large-scale CelebFaces Attributes dataset, also known as CelebA dataset [62], is a vast dataset with more than 200K images that can be used for face recognition, face detection, face localization and other face-related operations. The dataset consists of images from various sources, locations, background and poses and is best suitable for steganography also. The probability of using a photo/face image as the cover for hiding secret images is very high. Along with the images, there are 40 different annotations available like with/without glasses, emotions, hair styles, other accessories like hat.

C. IMAGENet

ImageNet [63] is also a very large dataset containing images from the WordNet hierarchy with each node containing more than 500 to 1000 images. ImageNet does not have any copyrights to the image and contains only the links or thumbnails to the original image. The dataset consists of images of varying size. Based on the requirement, the number of images, classes they belong to, background and the image size can be selected from the wide range available.

D. MNIST HANDWRITTEN DIGITS

Modified National Institute of Standards and Technology database (MNIST) [64] is another dataset that can be used for various computer vision and image processing applications. MNIST handwritten dataset consists of a training and testing set with images of handwritten digits 0 to 9. Images in this dataset are normalized, black and white with dimensions 28 × 28 pixels. The training set consists of 60,000 images and testing set consists of 10,000 images.

E. COCO

Common Objects in Context (COCO) dataset [65] was mainly developed for object detection, segmentation and image captioning purposes. This again is a huge dataset with images from 80 object categories. Each class contains at least 5 images. This dataset comes along with the class annotation and the segmentation annotation. There is no predefined training and testing split. The dataset split can be carried out based on the research topic and the user convenience.

F. OTHERS

Other datasets that can be used for image steganography and steganalysis are Div2K, SZUBase, USC-SIPI, DTD, LFW, and Pascal VOC. Div2K [66] is a commonly used dataset for Single Image Super-Resolution introduced in the NTIRE 2017 Challenge on Single Image Super-Resolution. It has a total of 1000 images split into 800 for training, 100 for

[1] http://agents.fel.cvut.cz/boss/index.php?mode=VIEW&tmpl=materials

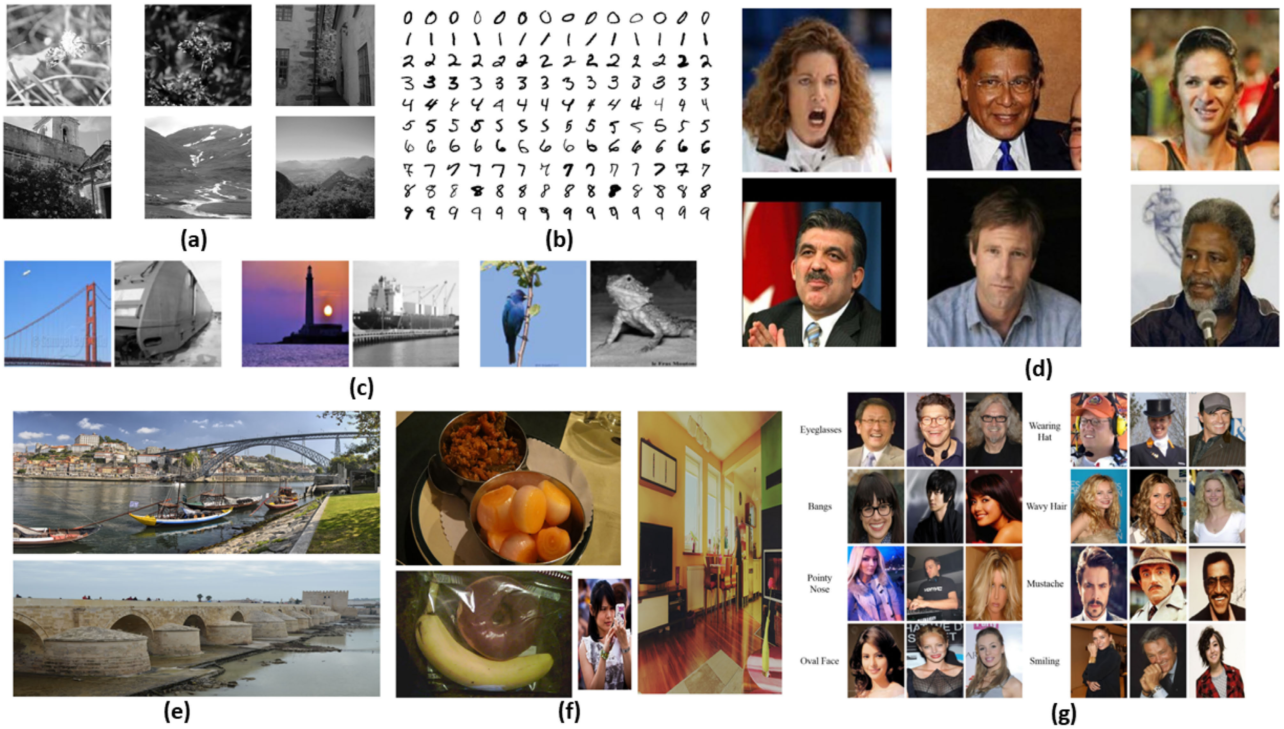


FIGURE 4. Sample images from datasets (a) BOSSBase (b) MNIST (c) ImageNet (d) LFW (e) Div2K (f) COCO and (g) CelebA.

validation and 100 for testing. Images with high resolution (1024 × 678) and three grades of low resolution can be found in this dataset. SZUBase is a dataset collected in [67] with 40000 grayscale images of size 512 × 512. USC-SIPI [68] has a variety of different resolution images and sizes for Image processing, image analysis, and machine vision purposes. Describable Textures Dataset (DTD) [69] is used to analysis the textural components containing 5640 jpg images of two sizes - 300 × 300 and 640 × 640. LFW dataset is used for face recognition and verification tasks [70]. PASCAL Visual Object Classes (VOC) is a challenge conducted for object detection and classification [71]. Figure 4 below represents some of the images collated from the dataset described.

IV. EVALUATION

Evaluation metrics are used to measure the invisibility, security, robustness and capacity of the proposed methods. The most commonly used metrics and the functionality they measure are described below.

A. EXPERIMENTAL SETUP

Image steganography models use two inputs and two output images mostly. In addition, the datasets used are usually very large. A GPU-based computer with a powerful graphic card is required to perform the training and testing. The trained model can then be deployed in a CPU or hand held standalone computers for deployment. All the experiments are conducted using either python 3.5 and above versions with pytorch [27], [32], [56] and [28] or MATLAB [16], [17], [49], and [10]. Another popular library used is the tensorflow [40], [52], [55] and [39]. Chainer library from python is also used [44].

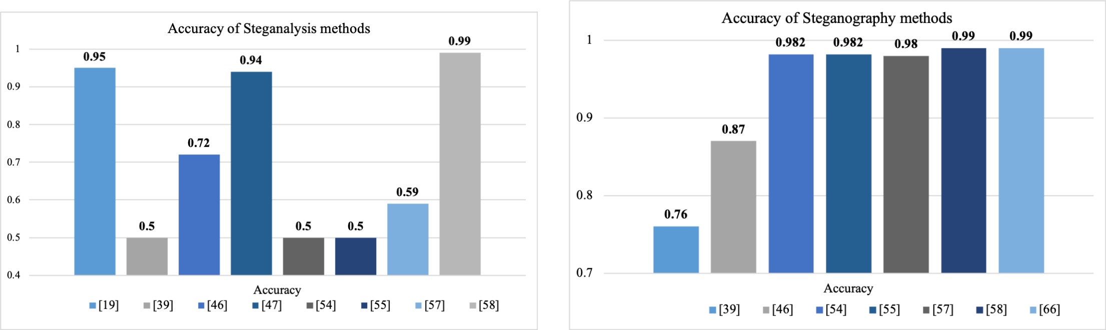
B. METRICS USED

1) PSNR

Peak Signal to Noise Ratio (PSNR) is used to determine the quality, robustness and invisibility of the proposed steganography method. PSNR is the ratio between the maximum quality representation of the cover image and the stego image. In the case of steganalysis, it is the ratio of the maximum quality measurement of the original secret image and the extracted secret image. PSNR is used to measure the peak error of the proposed method. The value of PSNR has to be high which implies that the quality of the reconstructed stego image is good. Mean Squared Error (MSE) is another metric to measure the quality of the stego image reconstructed. MSE is the cumulative squared error between the stego image and the original cover image. For better quality images, PSNR has to be high whereas the value of MSE has to be low indicating that the error is low. The formulas for calculating MSE and PSNR are given below.

$$MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N} \tag{1}$$

where, M and N is the number of rows and columns in the input image respectively.



(a) (b)

FIGURE 5. Steganalyser accuracy comparison between different methods.

After calculating MSE, its value is used in the calculation of PSNR.

$$PSNR = 10 * log_{10}\frac{(R^2)}{MSE} \tag{2}$$

where R is the fluctuation in the input image. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255. Table 5 summarizes the PSNR, MSE and SSIM values obtained by various methods in the study.

TABLE 5. PSNR and MSE values of the methods. The values indicate stego images and * for cover and ** for secret images.

Method	PSNR	MSE	SSIM
[16]	62.53	-	-
[17]	56.95	0.13	-
[31]	-	-	6.4* and 3.6**
[27]	40.47* and 40.66**	-	0.9794* and 0.9842**
[47]	64.7	-	-
[56]	-	3.54 and 11.46**	-
[10]	32.09	-	-

2) ACCURACY

Accuracy/bit accuracy is the commonly used metric to measure the security and robustness of the methods. Steganographic accuracy is defined as the accuracy of the proposed method correctly identifying the image as steganography image or not. Accuracy is calculated using four terms: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). True positive and true negative are the correct predictions made by the model while False positive and false negative are the errors made by the model. The table 6 is used to calculate the measures from the confusion matrix.

The accuracy is calculated as the total number of correct predictions made by the model against all classifications. Though accuracy is a simple and commonly used, it is not a good measure when the data is imbalanced. That is the cost for both false positives and false negatives are same. In addition to accuracy, other evaluation metrics are used to measure the performance of the model truly.

TABLE 6. Confusion Matrix for calculating accuracy.

Actual	Predictions	
	True	False
True	True Positive	False Negative
False	False Positive	True Negative

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{3}$$

For a stronger steganography method, a lower value of the steganalysis accuracy indicates higher security. The detection rate is another term used to calculate the testing accuracy of the steganalysis model used. It is used to determine the secrecy of the proposed method. Figure 5 shows the accuracy of the steganography and steganalysis methods used in the study.

A comparison on the steganalysis accuracy methods for different methods for stego images produced by popular methods like WOW [72], HUGO [61], S-UNIWARD are given in the figure 6.

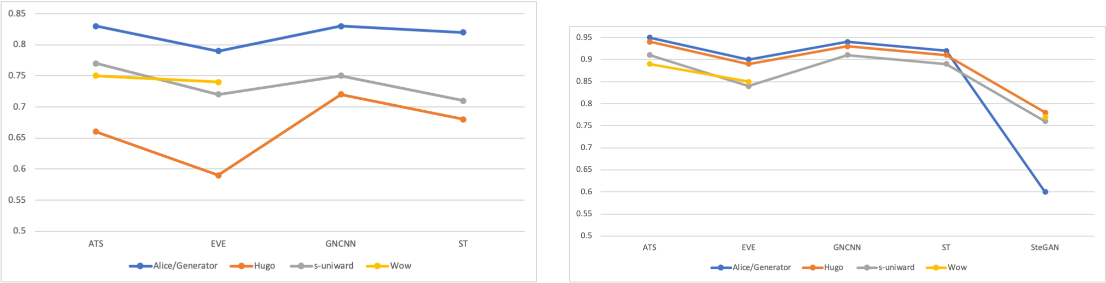
3) BPP

Hiding Capacity of the steganography algorithm is calculated using the Bits Per Pixel (BPP). BPP represents the number of bits that are hidden in every pixel of the cover image to produce the stego image. For a higher hiding capacity, the value of BPP has to be high. The below equation is used to calculate the value of BPP.

$$BPP = \frac{L}{HWC} \tag{4}$$

where L is the length of the hidden message, H is the height, W is the width and C is the number of channels of the cover image.

Many of these metrics compete against each other. Models which have higher capacity typically sacrifice secrecy, since hiding more information in images naturally leads to larger image distortions; models that are very robust to noise typically sacrifice capacity or secrecy, since the message must be



(a)

(b)

FIGURE 6. Accuracy comparison between different methods (a) Steganalysis and (b) Steganography.

encoded in the image redundantly. In some sense, steganography and watermarking are at two ends of a spectrum of problems prioritizing these different axes; steganography stresses secrecy while watermarking emphasizes robustness. In cases of hiding text inside an image, BPP used are either **0.1** or **0.4**.

C. OBSERVATIONS

The observation made from the results reported by the methods are delineated here. Basically, the hiding capacity, security and robustness factors are taken into account while discussing the observations.

Hiding Capacity: It is generally noted that the hiding capacity of the methods are in the following order. The methods with least hiding capacity are the traditional methods where text is the primary form of secret communication. Following that is the GAN-based methods, where only text message is used as secret. Unlike the traditional and GAN-based methods, the hiding capacity of the CNN-based methods are far better and is almost 1 [27]. The size of the secret image is same as the cover image. Even when a grayscale image is used for hiding [33], the size of the secret and the cover images are equal. In terms of the hiding capacity, CNN-based methods clearly outperform other methods.

Security and Robustness: Security is associated with embedding and robustness is associated with the extraction of the secret image. From our observations, CNN-based methods and GAN-based methods yield higher security. It is worth noting that the extraction of the secret images is prone to loss in information in deep learning methods. However, in traditional methods, the security is less but the robustness is high. PSNR measure is used to correlate the security and the robustness. From 5 and it can be noted that [49] has the highest PSNR value explaining the higher security of the GAN based method. The highest value of PSNR being 64.7, given by the cycle GAN based image steganography method.

From the observations made, GAN-based deep learning methods have the best performance in terms of the hiding capacity (1), PSNR (64.7) [49]. The discriminators are trained in a way to overcome any steganalysis attack and has better anti-detection property compared to traditional and CNN-based methods.

V. CHALLENGES

The following are some challenges for consideration in image steganography problems.

Data Availability - Though image steganography is an unsupervised learning and the main goal is image reconstruction, there is no proper benchmark dataset available except BOSSBase [60]. The number of images may be large in the BOSSBase but the images are of grayscale stored available in tiff format. Most of the methods deal with hiding RGB images inside RGB cover images. Finding a suitable dataset can be challenging. ImageNet is the most commonly used dataset with a major drawback being the image size. The images are very small in size 64 × 64.

Convergence of GAN Convergence is a major drawback for GAN where the model does not converge irrespective of the parameters chosen. Mode collapse also happen often as the generator and discriminator are inter-dependent.

Comparison with other methods Evaluation metrics used by different methods are different and hence comparing the proposed method with the state-of-the-art methods are not easy.

Real-time steganography Steganography models are trained on a huge amounts of datasets, like in, [27], 45000 training images are used. However, when it comes to the real-time steganography, it gets difficult. The implementation of the trained model for performing the steganography and steganalysis requires transferring the stego image through an untrusted channel to the receiving end. The capability of the trained model in dealing with real time live images which may contain noises, skewing, blurring is not proved. The implementation of the model for real-time steganography is still questionable.

VI. DISCUSSION AND FUTURE WORKS

GAN is the most widely used architecture and specifically cycleGAN when compared to the CNN based methods. The most important factor to consider is that the GAN is a two part model where one model is used at the sender end for embedding the secret information while the second model is used at the receiving end to extract the secret information. Two models that are trained end-to-end under same training circumstances are required for the whole process of image steganography to work perfectly. Without or the loss of one model may affect the embedding/ extraction as they are interconnected. CNN-based methods use U-Net/Xu-Net autoencoder-decoder architecture for embedding and extracting. Some methods use the encoder for embedding and decoder for extracting, whereas, some use one autoencoder-decoder for embedding and another for extracting. Though there is some inter dependency, it is not totally linked like GAN methods.

Unbalance in the learning of generator-discriminator can happen where the generator is performing efficiently but the discriminator is struggling. Though the overall efficiency will not be affected, either sender or receiver will be prone to faults. This can be avoided by choosing the parameters carefully and avoiding over fitting during training.

Security capabilities of the GAN methods are higher when compared with CNN architectures and traditional LSB methods. GANs are basically used in the image reconstruction field which makes it conveniently suitable for image steganography compared to convNets.

In deep learning methods, the working principle of the image steganography is to extract features from the cover and secret image and concatenate them to produce an end result closer to the cover image. However, where and how the secret image pixels are embedded cannot be understood clearly. Without the counterpart extracting model trained, it may be difficult to crack the steganography image. This increases the security but becomes difficult when the extraction model is not working or crashed.

Some of the major disadvantages are the time taken for training, the computational time during testing and the storage capabilities. The models take two images or one image and a text message converted into bits as input. The features are extracted from both the inputs which increases the computational time in both embedding and extraction. The number of parameters also increases by double at least when compared to a normal architecture which in turn increase the storage space required by the model.

RGB secret images are used by a handful methods when others used gray scale images. When converting the gray scale image to RGB image for better understanding, there can be loss of information. Image enhancement techniques are required in addition to understand the secret information properly.

Some of the aspects that can be considered for future works are enumerated below,

- Usage of popular networks U-Net, cycleGAN and using DCT and DWT have been considered and more exploration on the other customized architectures can be attempted. For example, Recurrent Neural Networks (RNNs) instead of CNNs can be further explored. A customized WGAN can be replaced with other variants of GAN.
- The majority of image steganography methods use either text or gray scale image as the secret information and there is a need for more research in hiding image in image and image in video.
- Experiments related to optimizing the parameters and decreasing the storage capacities can be further conducted using various datasets.
- The era of quantum computing is not far away, more efforts on developing designs on quantum images can be explored.
- To benefit from a combination of methods, an ensemble of traditional and deep learning methods can be further studied.
- Efforts can be directed to form a benchmark dataset containing images from various source cameras, image formats. A compilation of all possible algorithms can also be done to create the steganography images.
- Many methods have considered the hiding capacity, security and robustness as the performance measure. However, there are possibilities for man-in-the-middle attacks when the transfer happens through untrusted channels. Tampering of the stego image can also happen during the transfer. These attacks and the performance of the designed algorithm against these attacks can be considered for evaluation along with other metrics.

VII. CONCLUSION

Image steganography is the method used in transmitting secret information by hiding it in plain sight inside a cover image. Deep learning methods are widely used in every field and has been used in the research of steganography. Review of all the related works led to categorizing them into three groups vastly. Most of the traditional based steganography methods use the LSB substitution and some of its variants. Other than LSB, PVD, DCT and EMD are commonly used. The hiding capacity of the traditional methods are limited as over burdening the cover image by exploiting more pixels for hiding the secret message may led to distortions. Also, the autoencoder-decoder structure with VGG as base, U-Net and Xu-Net are the most prevailing architectures used for CNN-based image steganography methods. More recently, GAN architecture has gained significant attention for their ability to deal with image reconstruction tasks. Image steganography can be considered one such image reconstruction task where the cover image and the secret information is taken as input to reconstruct a steganographic image which is close to the cover image in resemblance.

There is no benchmark image datasets to perform the image steganography while most of them use the ImageNet, CelebA or BOSSBase. Each of the methods have their own evaluation methods and metrics and hence there is no common platform for comparisons. Peak Signal-to-Noise Ratio (PSNR) value comparison shown in table 5 gives an idea on the security performance of the different methods. By far, the best PSNR value of 64.7 is obtained by Kuppusamy *et al.* [49] using cycle GAN. GAN based methods have proved to have better security performance and hiding capacity. GAN is the extensively used and most preferred architecture over the autoencoder and statistical methods. Traditional methods are less secure as it is only a matter of detection of presence of the secret message. The secret message can be easily extracted as the embedding used a statistical method.

In summary, this paper has elaborated on the techniques used in the recent times for image steganography, the current trends. Along with it, details on the datasets and evaluation metrics are detailed. Challenges faced, some discussions on the gaps and the scopes for future direction is also evaluated in this paper. It can be concluded that deep learning has tremendous potential in the image steganography field taking into consideration that all the challenges and gaps are filled.

ACKNOWLEDGMENT

The findings achieved herein are solely the responsibility of the author. Open Access funding was provided by the Qatar National Library.

REFERENCES

[1] Wikipedia. (2020). *Steganography*. [Online]. Available: https://en.wikipedia.org/wiki/Steganography

[2] H. Shi, X.-Y. Zhang, S. Wang, G. Fu, and J. Tang, "Synchronized detection and recovery of steganographic messages with adversarial learning," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2019, pp. 31–43.

[3] N. F. Hordri, S. S. Yuhaniz, and S. M. Shamsuddin, "Deep learning and its applications: A review," in *Proc. Conf. Postgraduate Annu. Res. Informat. Seminar*, 2016, pp. 1–6.

[4] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, Feb. 1998.

[5] S. Gupta, G. Gujral, and N. Aggarwal, "Enhanced least significant bit algorithm for image steganography," *Int. J. Comput. Eng. Manage.*, vol. 15, no. 4, pp. 40–42, 2012.

[6] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman encoding," in *Proc. 3rd Nat. Conf. Emerg. Trends Appl. Comput. Sci.*, Mar. 2012, pp. 14–18.

[7] A. Singh and H. Singh, "An improved LSB based image steganography technique for RGB images," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–4.

[8] Z. Qu, Z. Cheng, W. Liu, and X. Wang, "A novel quantum image steganography algorithm based on exploiting modification direction," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 7981–8001, Apr. 2019.

[9] S. Wang, J. Sang, X. Song, and X. Niu, "Least significant qubit (LSQb) information hiding algorithm for quantum image," *Measurement*, vol. 73, pp. 352–359, Sep. 2015.

[10] N. Patel and S. Meena, "LSB based image steganography using dynamic key cryptography," in *Proc. Int. Conf. Emerg. Trends Commun. Technol. (ETCT)*, Nov. 2016, pp. 1–5.

[11] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 131–135.

[12] M. V. S. Tarun, K. V. Rao, M. N. Mahesh, N. Srikanth, and M. Reddy, "Digital video steganography using LSB technique," *Red*, vol. 100111, Apr. 2020, Art. no. 11001001.

[13] S. S. M. Than, "Secure data transmission in video format based on LSB and Huffman coding," *Int. J. Image, Graph. Signal Process.*, vol. 12, no. 1, p. 10, 2020.

[14] M. B. Tuieb, M. Z. Abdullah, and N. S. Abdul-Razaq, "An efficiency, secured and reversible video steganography approach based on lest significant," *J. Cellular Automata*, vol. 16, no. 17, Apr. 2020.

[15] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A robust and secure video steganography method in DWT-DCT domains based on multiple object tracking and ECC," *IEEE Access*, vol. 5, pp. 5354–5365, 2017.

[16] K. A. Al-Afandy, O. S. Faragallah, A. Elmhalawy, E.-S.-M. El-Rabaie, and G. M. El-Banby, "High security data hiding using image cropping and LSB least significant bit steganography," in *Proc. 4th IEEE Int. Colloq. Inf. Sci. Technol. (CiSt)*, Oct. 2016, pp. 400–404.

[17] A. Arya and S. Soni, "Performance evaluation of secrete image steganography techniques using least significant bit (LSB) method," *Int. J. Comput. Sci. Trends Technol.*, vol. 6, no. 2, pp. 160–165, 2018.

[18] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2995–3004, Apr. 2019.

[19] A. Qiu, X. Chen, X. Sun, S. Wang, and W. Guo, "Coverless image steganography method based on feature selection," *J. Inf. Hiding Privacy Protection*, vol. 1, no. 2, p. 49, 2019.

[20] R. D. Rashid and T. F. Majeed, "Edge based image steganography: Problems and solution," in *Proc. Int. Conf. Commun., Signal Process., Appl. (ICCSPA)*, Mar. 2019, pp. 1–5.

[21] X. Liao, J. Yin, S. Guo, X. Li, and A. K. Sangaiah, "Medical JPEG image steganography based on preserving inter-block dependencies," *Comput. Electr. Eng.*, vol. 67, pp. 320–329, Apr. 2018.

[22] W. Lu, Y. Xue, Y. Yeung, H. Liu, J. Huang, and Y. Shi, "Secure halftone image steganography based on pixel density transition," *IEEE Trans. Dependable Secure Comput.*, early access, Aug. 6, 2019, doi: 10.1109/TDSC.2019.2933621.

[23] Y. Zhang, C. Qin, W. Zhang, F. Liu, and X. Luo, "On the fault-tolerant performance for a class of robust image steganography," *Signal Process.*, vol. 146, pp. 99–111, May 2018.

[24] H. M. Sidqi and M. S. Al-Ani, "Image steganography: Review study," in *Proc. Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*, 2019, pp. 134–140.

[25] P. Wu, Y. Yang, and X. Li, "Image-into-image steganography using deep convolutional network," in *Proc. Pacific Rim Conf. Multimedia*. Cham, Switzerland: Springer, 2018, pp. 792–802.

[26] P. Wu, Y. Yang, and X. Li, "StegNet: Mega image steganography capacity with deep convolutional network," *Future Internet*, vol. 10, no. 6, p. 54, Jun. 2018.

[27] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a U-Net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.

[28] T. P. Van, T. H. Dinh, and T. M. Thanh, "Simultaneous convolutional neural network for highly efficient image steganography," in *Proc. 19th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Sep. 2019, pp. 410–415.

[29] R. Rahim and S. Nadeem, "End-to-end trained CNN encoder-decoder networks for image steganography," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–6.

[30] Z. Wang, N. Gao, X. Wang, J. Xiang, and G. Liu, "STNet: A style transformation network for deep image steganography," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2019, pp. 3–14.

[31] K. Yang, K. Chen, W. Zhang, and N. Yu, "Provably secure generative steganography based on autoregressive model," in *Proc. Int. Workshop Digit. Watermarking*. Cham, Switzerland: Springer, 2018, pp. 55–68.

[32] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2069–2079.

[33] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8559–8575, Apr. 2019.

[34] S. Islam, A. Nigam, A. Mishra, and S. Kumar, "VStegNET: Video steganography network using spatio-temporal features and micro-bottleneck," in *Proc. BMVC*, Sep. 2019, p. 274.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[36] D. Volkhonskiy, B. Borisenko, and E. Burnaev, "Generative adversarial networks for image steganography," in *Proc. ICRL Conf.*, 2016.

[37] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, vol. 11433, 2020, Art. no. 114333M.

[38] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," 2016, *arXiv:1602.05110*. [Online]. Available: http://arxiv.org/abs/1602.05110

[39] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "SSGAN: Secure steganography based on generative adversarial networks," in *Proc. Pacific Rim Conf. Multimedia*. Cham, Switzerland: Springer, 2017, pp. 534–544.

[40] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1954–1963.

[41] J. Yang, K. Liu, X. Kang, E. K. Wong, and Y.-Q. Shi, "Spatial image steganography based on generative adversarial network," 2018, *arXiv:1804.07939*. [Online]. Available: http://arxiv.org/abs/1804.07939

[42] J. Yang, D. Ruan, J. Huang, X. Kang, and Y.-Q. Shi, "An embedding cost learning framework using GAN," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 839–851, 2020.

[43] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Process. Lett.*, vol. 24, no. 10, pp. 1547–1551, Oct. 2017.

[44] H. Naito and Q. Zhao, "A new steganography method based on generative adversarial networks," in *Proc. IEEE 10th Int. Conf. Awareness Sci. Technol. (iCAST)*, Oct. 2019, pp. 1–6.

[45] K. Zhang, A. Cuesta-Infante, and K. Veeramachaneni, "SteganoGAN: Pushing the limits of image steganography," Jan. 2019, *arXiv:1901.03892*. [Online]. Available: https://arxiv.org/abs/1901.03892

[46] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–672.

[47] Y. Ke, M. Zhang, J. Liu, T. Su, and X. Yang, "Generative steganography with Kerckhoffs' principle based on generative adversarial networks," 2017, *arXiv:1711.04916*. [Online]. Available: http://arxiv.org/abs/1711.04916

[48] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[49] P. G. Kuppusamy, K. C. Ramya, S. Sheebha Rani, M. Sivaram, and V. Dhasarathan, "A novel approach based on modified cycle generative adversarial networks for image steganography," *Scalable Comput., Pract. Exper.*, vol. 21, no. 1, pp. 63–72, Mar. 2020.

[50] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a master of steganography," 2017, *arXiv:1712.02950*. [Online]. Available: http://arxiv.org/abs/1712.02950

[51] H. Porav, V. Musat, and P. Newman, "Reducing steganography in cycle-consistency GANs," in *Proc. CVPR Workshops*, 2019, pp. 78–82.

[52] R. Meng, Q. Cui, Z. Zhou, Z. Fu, and X. Sun, "A steganography algorithm based on CycleGAN for covert communication in the Internet of Things," *IEEE Access*, vol. 7, pp. 90574–90584, 2019.

[53] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[54] Z. Zhang, G. Fu, J. Liu, and W. Fu, "Generative information hiding method based on adversarial networks," in *Proc. Int. Conf. Comput. Eng. Netw.* Cham, Switzerland: Springer, 2018, pp. 261–270.

[55] M.-M. Liu, M.-Q. Zhang, J. Liu, Y.-N. Zhang, and Y. Ke, "Coverless information hiding based on generative adversarial networks," 2017, *arXiv:1712.06951*. [Online]. Available: http://arxiv.org/abs/1712.06951

[56] X. Duan, H. Song, C. Qin, and M. K. Khan, "Coverless steganography for digital images based on a generative model," *Comput., Mater. Continua*, vol. 55, no. 3, pp. 483–493, Jul. 2018.

[57] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[58] C. Li, Y. Jiang, and M. Cheslyar, "Embedding image through generated intermediate medium using deep convolutional generative adversarial network," *Comput., Mater. Continua*, vol. 56, no. 2, pp. 313–324, 2018.

[59] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "SSteGAN: Self-learning steganography based on generative adversarial networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 253–264.

[60] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Information Hiding*, T. Filler, T. Pevný, S. Craver, and A. Ker, Eds. Berlin, Germany: Springer, 2011, pp. 59–70.

[61] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding*, R. Böhme, P. W. L. Fong, and R. Safavi-Naini, Eds. Berlin, Germany: Springer, 2010, pp. 161–177.

[62] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[64] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[65] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[66] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.

[67] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Process. Lett.*, vol. 24, no. 10, pp. 1547–1551, Oct. 2017.

[68] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. ACM Int. Conf. Ubiquitous Comput. (Ubicomp), Workshop Situation, Activity Goal Awareness (SAGAware)*, Pittsburgh, PA, USA, Sep. 2012, pp. 1036–1043.

[69] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3606–3613.

[70] B. G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[71] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[72] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 234–239.

NANDHINI SUBRAMANIAN (Member, IEEE) received the bachelor's degree in electrical and electronics engineering from the PSG College of Technology, India, and the master's degree in computing from Qatar University, Doha. She is currently working as a Research Assistant with Dr. S. Al-Maadeed with Qatar University. Her interests include computer vision, artificial intelligence, machine learning, and cloud computing.

OMAR ELHARROUSS received the master's degree from the Faculty of Sciences, Dhar El Mehraz, Fez, Morocco, in 2013, and the Ph.D. degree from the LIIAN Laboratory, USMBA-Fez University, in 2017. His research interests include pattern recognition, image processing, and computer vision.

SOMAYA AL-MAADEED (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Nottingham, U.K., in 2004. She is currently the Head of the Computer Science Department, Qatar University, where she is also a Coordinator of the Computer Vision and AI Research Group. She enjoys excellent collaboration with national and international institutions and industry. She is a Principal Investigator of several funded research projects generating approximately five million. She has published extensively pattern recognition and delivered workshops on teaching programming for undergraduate students. She attended workshops related to higher education strategy, assessment methods, and interactive teaching. In 2015, she was elected as the IEEE Chair for the Qatar Section.

AHMED BOURIDANE (Senior Member, IEEE) received the Ingénieur d'Etat degree in electronics from the Ecole Nationale Polytechnique d'Alger, Algeria, in 1982, the M.Phil. degree in VLSI for signal processing from Newcastle University, U.K., in 1988, and the Ph.D. degree in computer vision from the University of Nottingham, U.K., in 1992. He joined Queen's University Belfast, in 1994, initially as a Lecturer and then as a Reader in Computer Science. He is currently a Professor of Computer Science and leads the Intelligent Systems and Security Group, Northumbria University, Newcastle upon Tyne, U.K. He has authored or coauthored over 350 publications. His research interests include imaging for security and medical engineering.

• • •