International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.13, No.1 (Apr-23)

http://dx.doi.org/10.12785/ijcds/130169

A Review on Deep Learning Solutions for Steganalysis

Ankita Gupta[1], Rita Chhikara[2] and Prabha Sharma[3]

[1]*Department of Computer Science, The NorthCap University, Gurugram, Haryana, India*
[2]*Department of Computer Science, The NorthCap University, Gurugram, Haryana, India*
[3]*Department of Computer Science, The NorthCap University, Gurugram, Haryana, India*

Received 29 Apr. 2022, Revised 21 Dec. 2022, Accepted 6 Feb. 2023, Published 16 Apr. 2023

Abstract: Steganalysis methods have developed to attack steganography, a technique used to hide secret information in a digital media. The traditional way of steganalysis is performed as feature extraction followed by classification. With the popularity of Deep Learning (DL) in the field of computer vision, researchers started applying deep learning for steganalysis problems also. Soon they found promising results with DL as it automates the feature extraction step and classification results can be used to better learn the features. Thus, the tedious task of manual extraction of features with a separate classification step is unified in deep learning giving optimistic results. This work provides a better insight into steganalysis evolution using deep learning and provides a broad review on how researchers have successfully applied Convolutional Neural Network (CNN) by using steganalysis specific activation functions, different convolutional layers and others. Researchers have compared their results with each other as well as state-of-the-art before deep learning (Rich Models + Ensemble Classifier). Initially, CNNs were created from scratch in the field of steganalysis but later researchers moved to highly efficient pretrained networks such as SRNet, ResNet and EfficientNet and found significant improvement in results on more challenging datasets such as ALASKA-I and ALASKA-II. The reason for such improvement is that pretrained networks are already trained on a very large dataset of images for some classification tasks and thus can be finetuned easily to other classification tasks with improved results.

Keywords: Steganography, Steganalysis, Convolution Neural Network, Deep Learning, Rich Models, Ensemble Classifier, Pretrained Networks, Spatial Domain, JPEG Domain.

1. INTRODUCTION

With the tremendous growth of communication over the internet through images, audio, video and other means, its misuse has been started in the form of hiding information in the above media. Steganography is the name given to such a technique. There are various such methods available publicly in which pixels of the images are changed such a way that distortion in the image is hard to detect [1][2][3]. Forensic experts need some technique to detect steganography as various criminal activities could take place by hiding important information in digital media. Steganalysis then comes into the picture to detect the changed image through steganography. Initially, Specific Steganalysis was targeted to detect known steganographies [4]. Soon Universal Steganalysis was developed to detect any kind of steganography with its dual strategy of feature extraction and classification [5][6]. Thus, with this approach information hidden through any steganography algorithm can be easily perceived. However, with the emergence of content adaptive steganographies in which information is concealed in such regions of images where the distortion between cover and stego image is minimal [7][8], higher-order statistical features need to be extracted. This gives rise to universal steganalysis extracting high dimensional features such as by Fridrich and Kodovsky [9], in Spatial Rich Model (SRM) features of the spatial domain and by Kodovsky and Fridrich [10] in J+SRM features of the transform domain. SRM features are extracted in the form of 106 submodels, each submodel with approximately 300 features, giving a total of 34671 features. The various variants of SRM such as Projection Spatial Rich Model (PSRM) by Holub and Fridrich [11], maxSRM by T. Denemark et al. [12] and others are also used for providing better results. The manual designing of such high-dimensional features followed by extraction is quite challenging. Moreover, it burdens the classification task in universal steganalysis. This redirected the researchers towards finding a way of automatically getting out of useful features from images through deep learning. Deep learning (DL) has already proven its potential in various other fields as computer vision, pattern recognition, and image classification. Researchers found the twofold benefit of using deep learning in the field

E-mail address: ankita17csd005@ncuindia.edu, ritachhikara@ncuindia.edu, prabhasharma@ncuindia.edu

of universal steganalysis. The two disconnected steps of feature extraction and classification get connected in deep learning as information from the classification step can be used to guide feature extraction. This guidance helped the researchers in attaining more useful features as compared to manual features. Tan and Li [13] made use of the Convolution Neural Network (CNN), one of the deep learning methods for classifying stego and cover images. After him, almost all researchers have made use of CNN in the field of steganalysis. Before the application of deep learning for steganalysis problems, Rich Model features such as SRM and its variants along with Ensemble Classifier (EC) [14] were giving the best results for various content adaptive steganographies. So, researchers using deep learning for steganalysis problems have compared their results with these state-of-the-art methods.

The main contribution of this review is to gain an insight on how deep learning has evolved over time for both spatial domain and transform domain steganalysis. Researchers have customized CNNs and put forward different activation functions such as Gaussian neuron (GN), Truncated linear Unit (TLU) and others for addressing the minute differences of cover and stego images in steganalysis. Similarly different preprocessing filters were used for gaining advantage in steganalysis domain. To improve the results, researchers first have created these customized CNNs but later moved to pretrained networks and found significant improvement in results. The pretrained networks are highly efficient as they are trained on millions of images for some classification problems. These pretrained networks are then finetuned for steganalysis.

The rest of the paper is organized into sections 2-10 as follows: Section 2 discusses why CNN is mainly used for steganalysis. Section 3 gives steganalysis in spatial domain. Section 4 gives steganalysis in transform domain. Details of deeper CNN networks for steganalysis is provided in section 5. Section 6 discusses an insight of how steganalysis moves from assumptions to reality. Section 7 explains benefits of using pretrained networks for steganalysis. Section 8 discusses how steganalysis can be used for images of arbitrary size. Section 9 presents open challenges and emerging directions in this area and Section 10 concludes the review.

2. CREATING CNN FROM SCRATCH FOR STEGANALYSIS PROBLEMS

The very first researchers applying deep learning for steganalysis problems [13] pointed out that CNN exhibited the same structure as that of the popular SRM framework in the same field. The various steps in the SRM framework are: A total of 39 filters are used for calculating the noise residuals from each image. The filters are manually designed in SRM by its researchers. Each filter act as a predictor of the central pixel from the neighbouring pixel. The residual is then obtained by subtracting the original value of the pixel from the predicted value. In this way residuals of all the pixels are calculated using 39 such filters, the purpose of which is to capture the dependencies among the neighbouring pixels in every possible way. Applying filters is equivalent to the convolution of filters with the image. This operation is very much similar to applying filters in the convolution layer of CNN.

After calculating the residuals of the images, quantization and truncation operations are performed to bring non-linearity into the SRM model. Finally, the feature submodels are constructed from the 4th order co-occurrence matrices calculated with the quantized and truncated residuals obtained. This step can be a possible analogue with the calculation of pointwise sigmoid nonlinearities in the convolution layer of CNN followed by the average/max pooling and subsampling layer.

The better performance of the SRM model as compared to other features like Subtractive Pixel Adjacency Matrix (SPAM) [6] is due to a diverse set of special structured filters. Since the SRM filters are handcrafted so a possible advantage of CNN can be that filters are learnable in CNN with the help of training data. Thus, if a large and diverse set of images are used to train CNN, the filters learned can better capture the nuances of cover and stego images as compared to the fixed filters.

3. STEGANALYSIS WITH CNN IN SPATIAL DOMAIN

The authors, Tan and Li [13] put forward a nine-layer 3-stage CNN network with 40 trainable filters to keep resemblance with SRM filters at the first stage of CNN. With the limitations of hardware they had, and extensive experimentation with their CNN, they found that using random initializations in CNN could make CNN stuck in local optimum and even result in CNN diverging to poor solutions. So, they multiplied their 40 kernels initialized with random values at the first stage with edge detector kernel K_5 used in SRM with the thought that it would help in better convergence of their network. This was thought of due to the success of the filters of SRM in Universal Steganalysis because of their well thought special structure provided by the SRM researchers. The second consideration was to incorporate Stacked Convolutional Auto Encoders (SCAE) for unsupervised pretraining of CNN. With both the above measures including in their CNN, they found improved results as compared to their basic CNN. Also, the performance surpassed SPAM performance but was still far behind performance achieved by the SRM model. The author attributed this to the lack of infrastructure and GPU facilities they had and believed once they would overcome this limitation, deep learning could provide better results than SRM which they initially believed theoretically. Soon after them, Qian et al. in [15] came up with a customized CNN model known as Gaussian Neuron CNN (GNCNN). They have made various changes to the basic CNN model which are essential for steganalysis problems. Firstly, they have added an image processing layer before the actual CNN network. They also believed that applying CNN with random initialization of filters would not serve any purpose for steganalysis problems as they were completely different from other computer science problems for which CNN proved their potential.

A. Adding Image Processing Layer before CNN for Steganalysis

The purpose of this layer is to initialize filters of the initial layer of CNN with values used in SRM filters instead of random values. Thus, in deep learning also, directly providing cover and stego images make CNN convergence difficult so noise residuals of both cover and stego images are given as input to the initial convolution layer and subsequent features are learnt on these noise residuals so that clear difference between cover and stego images can be found. Thus, each CNN is preceded by an image processing layer which convolves input images with different filters to calculate these residuals. Initial weights in these filters are the same as those used in SRM filters. Some researchers have kept the initial weight of these filters fixed in the image processing layer which are not changed during training of CNN. While many other researchers have used these weights initially, though instead of keeping them fixed, they are also tuned during CNN training. Fig. 1 below summarizes the difference between traditional learning and deep learning for steganalysis problems.

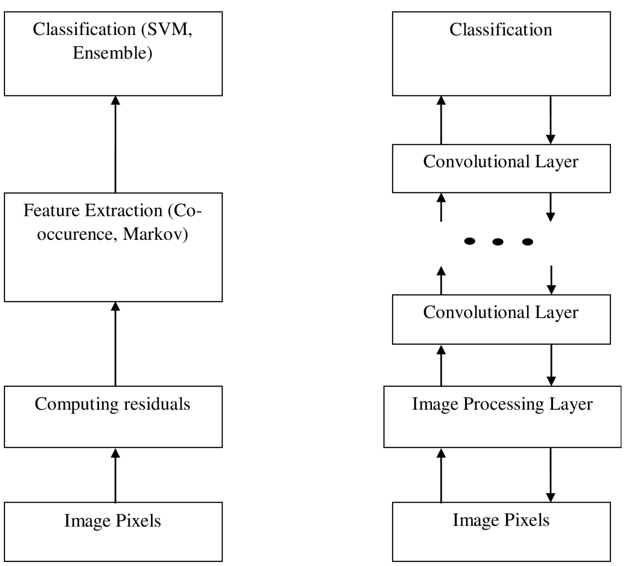

Figure 1. Difference between traditional and deep learning for steganalysis

In the image processing layer used by Qian et al. [15], the image is first convolved with a high pass filter which is similar to one of the filters used in SRM. This is required to strengthen the weak stego signals. Authors have kept this filter fixed during CNN training and then subsequent CNN layers help to aggregate stego signals from local to global. The KV kernel used as a filter by the authors in the image processing layer is provided by (1) as:

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

After the generation of an initial noise residual in the image processing layer, the various convolution operations in consecutive convolutional layers capture dependencies among a larger neighbourhood in a hierarchical fashion to make accurate predictions. Each convolution layer is generally represented by three kinds of operations: Convolution, non-linearity, and pooling in a sequential fashion which is described below in (2):

$$X_j^l = pool\left(f\left(\sum_i X_i^{l-1} * K_{ij}^l + b_j^l\right)\right) \tag{2}$$

where X_j^l is the j^{th} feature map in layer l, $pool()$ denotes pooling, $f()$ represents non-linearity, X_i^{l-1} is the i^{th} feature map in layer $l-1$, k_{ij}^l is trainable convolution filter or kernel between j^{th} output feature map and i^{th} input feature map and b_j^l is a bias parameter for the j^{th} output feature map.

The typical structure of CNN for steganalysis problems having an input image size of 256*256 provided in [15] is described in Fig. 2.

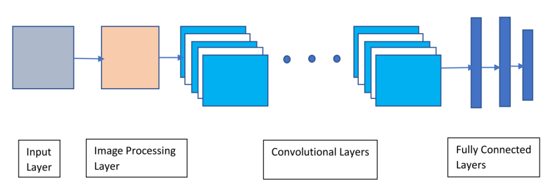

Figure 2. A typical architecture of CNN for steganalysis

Generally, the activation functions used in CNN are sigmoid and different forms of it, but authors in [15] have provided Gaussian Neuron (GN) activation function customized for steganalysis problems. GN function is defined with the help of (3):

$$f(x) = e^{\frac{-x^2}{\sigma^2}} \tag{3}$$

where σ defines the width of the Gaussian curve.

The final operation in the convolution layer is pooling which helps to move to a global area in subsequent convolution layers. For steganalysis problems, authors have found average pooling better than max pooling operation. Finally, the authors have used 2 fully connected layers followed by a softmax activation function for classification or classifying the initial input images into two classes as cover and stego. Authors have concluded that on the BOSSBase 1.01 dataset [16], the performance of their architecture is much better than the SPAM model but 2 -3 percent lower than the SRM model, however, for more diverse ImageNet database [17], results of the SRM model and GNCNN architecture are approximately same. They have demonstrated their results on images using wavelet obtained weights (WOW) [7], highly undetectable steganography (HUGO) [8], and universal wavelet relative distortion in spatial domain (S-UNIWARD) [18] stego algorithms. Thus, authors have concluded that deep learning architecture can handle large diverse datasets and since steganalysis is completely different from other AI tasks, deep networks can be better than handcrafted features if steganalysis properties are explored more to customize deep networks.

Further Qian et al. [19] applied deep transfer learning. First, they have learned the feature representation by training CNN on stego images with a higher payload. After this, the same CNN networks are used to refine the features for stego images with a lower payload. They have demonstrated that feature learning can be improved with pretrained CNN networks and proved their results on stego images converted by WOW and S-UNIWARD stego algorithms.

Xu et al. [20] also observed that the architecture of CNN used for various computer vision tasks might not be suitable for steganalysis work. Thus, they also customized the CNN architecture incorporating the knowledge of steganalysis. They have used the same high pass filter as used in [15] to compute the noise residuals of all images. Their architecture involves five groups of convolution layers. In the first group of convolution layers, various trainable filters are applied on the noise residual of the input image in the convolution layer to generate initial feature maps. After that, ABSolute (ABS) layer is applied on the feature maps generated in the previous step. The purpose of the ABS layer is to leverage the sign symmetry present in the noise residual of the input image. The output of the ABS layer is provided to the Batch Normalization (BN) Layer, the aim of which is to prevent CNN from falling into local minima by reducing the magnitude spread of the input data to limit their magnitude range to zero mean and unit variance. BN layer is followed by TanH activation layer and then average pooling layer. In the second group, the ABS layer is not present. In the next three groups, the Rectified Linear Unit (ReLU) activation layer is used in place of the TanH activation layer. In the last group, the global average pooling operation is performed to reduce the overall features to 128-D followed by a fully connected (FC) layer and then a softmax layer to classify the given input into two classes. On the BOSSBase 1.01 dataset, their CNN proved competitive with the SRM method against two steganographic algorithms HILL [21] and S-UNIWARD [18] . Their future work would be to improve their network by incorporating selection channel knowledge during embedding operations of various content-adaptive steganographies. Due to this selection channel knowledge models such as maxSRM and others have higher accuracy than the SRM model as well as various CNN discussed till this point.

Xu et al. further have used this CNN in their ensemble method [22]. Instead of using a single CNN they have trained many CNNs on different random subsamples of training images and then develop new features by combining some intermediate representations of features of these CNNs. They have found better results of the ensemble over using single CNN on S-UNIWARD stego algorithm at a payload of 0.4bpp using BOSSBase 1.01 dataset. Qian et al. in [23] further experimented with CNN to produce their results using five steganographic techniques HUGO, WOW, S-UNIWARD, MiPOD [24], and HILL-CMD [25]with random embedding keys on cover images. Their network architecture included an image processing layer, five convolution layers, and 3 fully connected layers. They tried different activation functions in convolution layers as ReLU, TanH, Gaussian, and 1-Gaussian, and found better results with 1-Gaussian. They used the dropout technique for the regularization of fully connected layers [26]. ReLU was used for the first two fully connected layers and softmax was used after the final layer. Their first contribution was to provide the feature visualization after each convolution layer to prove that the generated features effectively captured the stego noise after CNN training. Their next contribution was to use four different kinds of kernels picked from SRM work in the image processing layer to train many CNNs, one CNN with one kind of filter kernel. They then combined the output of these CNNs to give the final prediction. Thus, their combination strategy worked better due to different noise residuals captured in the image processing layer. Their final contribution was to handle images of arbitrary sizes by extracting fixed size patches from each image. They also augmented their dataset to effectively train CNN. They compared their results with SRM [9] + EC [14] and maxSRMd2 [12] + EC [14] and found that their results were close to the results of these methods but could not outperform them. This they attributed to the use of only four kinds of residual filters K 5*5, while about 30 basic linear filters were used by SRM and maxSRMd2 to calculate the noise residuals. Moreover, the incorporation of selection channel information as in maxSRMd2 could also boost the performance of their network. Authors, Ye et al. in [27]came up again with customized CNN. Their major addition was using the complete filter bank used in SRM in the initial convolution layer to compute all the noise residuals which greatly helped to suppress the image content and improved Signal to Noise Ratio (SNR) (stego signal to image content). Additionally, these initial filters were also trainable like other parameters of the network. Next, they incorporated Truncated Linear Unit (TLU) as an activation function in the first convolution layer which suited steganalysis problems. In all other layers, a ReLU activation unit was used. They also incorporated selection channel information in their network and all these additions helped the network to learn better, resulted in superior performance than SRM and maxSRMd2.

Yedroudj et al. in [28] also used all the basic 30 filters of SRM having size 5*5 in the initial convolution layer but weights in these filters were kept fixed and they were not tuned during the CNN training. The filters whose size were 3*3 in SRM were padded with zeros to make them 5*5. Like XuNet [20], they used 5 convolution layers, made use of the ABS layer and BN layer in each convolution layer. Like YeNet [27], they used the TLU activation function in the initial convolution layer. Unlike XuNet they used a scale layer accompanied by a BN layer and they found improved results with this inclusion. Also, to further increase the performance of the network BOSSBase1.01 dataset was augmented first with the BOWS2 dataset [29] and then both the datasets were further virtually augmented with rotations and flips. Against WOW and S-UNIWARD steganographies, their network achieved lower detection error as compared to SRM+EC, XuNet and YeNet. Further, Zhang et al. in [30] incorporated many changes to the CNN network to boost performance. Firstly, they used all the basic 30 filters of SRM in the initial image processing layer but instead of taking all filters of size 5*5 as in [28], only 5 such filters were used by them and all other 25 filters were of size 3*3. All these 30 filters were also tuned during network training. Thus, smaller size of 3*3 helped them to reduce the parameters of the network as well as to better capture information in a local neighbourhood. Secondly, they considered channel information also, as used in maxSRM by using two separable convolution blocks to capture spatial correlation as well as channel correlation. Their network could also handle arbitrary-sized images by performing Spatial Pyramid Pooling (SPP). The dataset used was BOSSBase 1.01 which was augmented with BOWS2 and further both were augmented with rotations and flips. Thus, with all the above additions to their network named as ZhuNet, it was able to outperform SRM, XuNet, YeNet as well as YedroudjNet using WOW and S-UNIWARD spatial steganographies on a reasonable variation of payloads. The structures of these famous networks are provided in Fig. 3, Fig. 4 , Fig. 5, Fig. 6, and can be compared directly. They are derived from their original architectures in a way so that they can be compared in terms of the structure of filters and outputs, activation functions, pooling operations and fully connected modules.

B. An Approach for Steganalysis of Arbitrary Size Images

Training of deep networks for steganalysis problems requires a substantial amount of GPUs' memory and thus it is difficult to train the networks on large sized images. Hence larger images are generally resized to smaller tiles of size 256*256 or 512*512 which sometimes leads to the loss of stego information. Tsang and Fridrich in [31] resolved the problem of arbitrary image sizes in a unique way other than the common way of extracting different tiles of smaller size from large sized images and then results can be fused or pooled with the help of pooled steganalysis as done in [32][33]. The network architecture is made scalable with respect to the input image sizes. YeNet is modified to extract the features from the last convolution layer and before the fully connected Inner Product (IP) layer. The 9th convolution layer of the modified YeNet produces 16 feature maps of size 7*7 instead of 3*3. Instead of directly feeding these feature maps to the IP layer, four different statistical moments are computed from these feature maps and are considered as the output of the last convolution layer. These statistical moments like sample variance, sample average, minimum and maximum contain most of the information about images like their resolution, size etc. and thus these are fed to the fully connected IP layer. It is believed and further proved by the authors that the convolution part needs not be retrained for bigger size images and only the IP part of the network needs to be retrained for large size images.

Thus, the first part of the network named as Universal feature extractor is trained along with IP layer on smaller sized images obtained by cropping BOSSBase 1.01 images of size 1024*1024. Then for larger image sizes only IP layer is retrained. The authors have named this network as TRIP (fixed Tile detector and Retrained IP layer). According to square root law [34], payload/change rate is adjusted with respect to image sizes for constant statistical detectability. Thus, authors have created another architecture named as RTRIP (Retrained Tile detector and Retrained IP layer) in which first part of the network is also retrained for smaller payloads embedded in large sized images along with retraining of the IP layer on moments extracted from large sized images. Results are computed on WOW and non-adaptive LSBM [2] stego techniques across a wide variety of payloads and change rates. Results are almost comparable for both TRIP and RTRIP detectors and authors have finally provided a size independent detector (SID) for steganalysis of arbitrary size images. Thus, first part of the network can be trained on smaller images without need of retraining on larger size images whereas only IP layers need to be retrained.

4. STEGANALYSIS IN TRANSFORM DOMAIN

Most of the researches discussed so far involved steganalysis in the spatial domain, but simultaneously researchers focused on applying deep learning in the transform or Joint Photographic Expert Group (JPEG) domain. The first representative work in this domain without JPEG-phase awareness was put forward by Zeng et al. [35] and with JPEG-phase awareness by Chen et al. [36]. Inspired by XuNet, Zeng et al. in [35] incorporated deep learning framework for steganalysis in frequency domain. With extensive experimentation on ImageNet database, they proved superior performance of their architecture than XuNet but comparison was not direct as both were using different domains. Their architecture consists of two stages. First

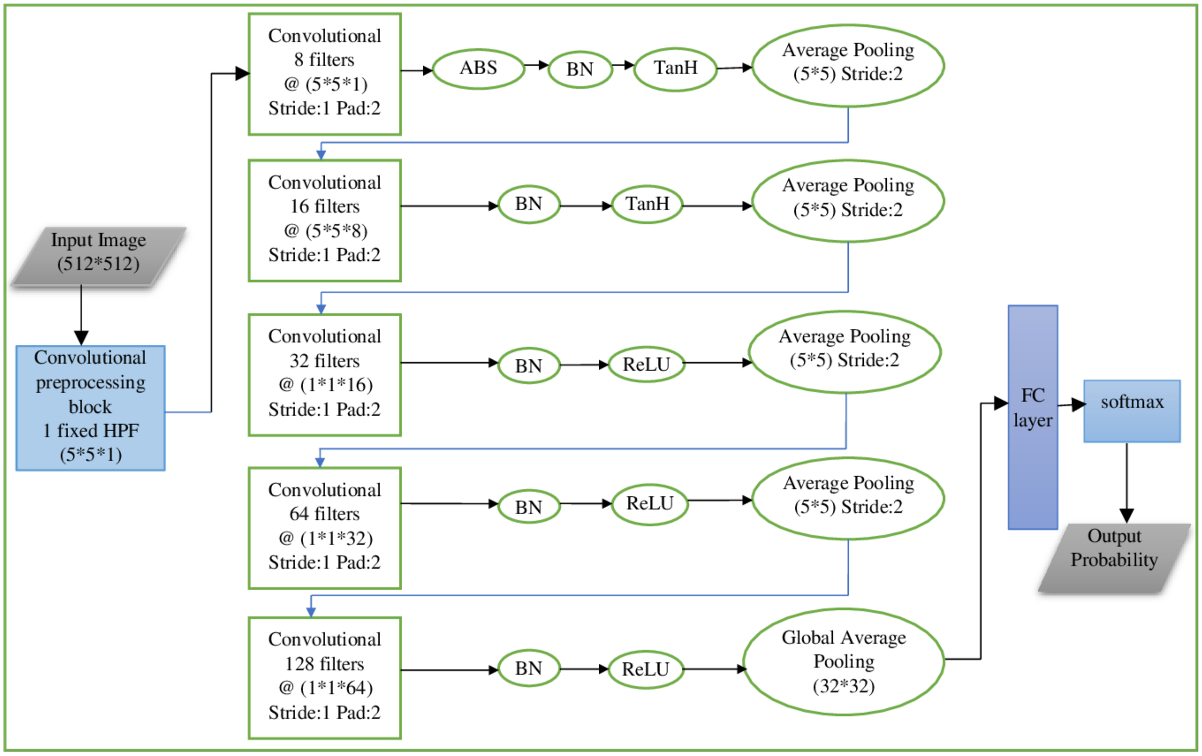

Figure 3. XuNet architecture [20]

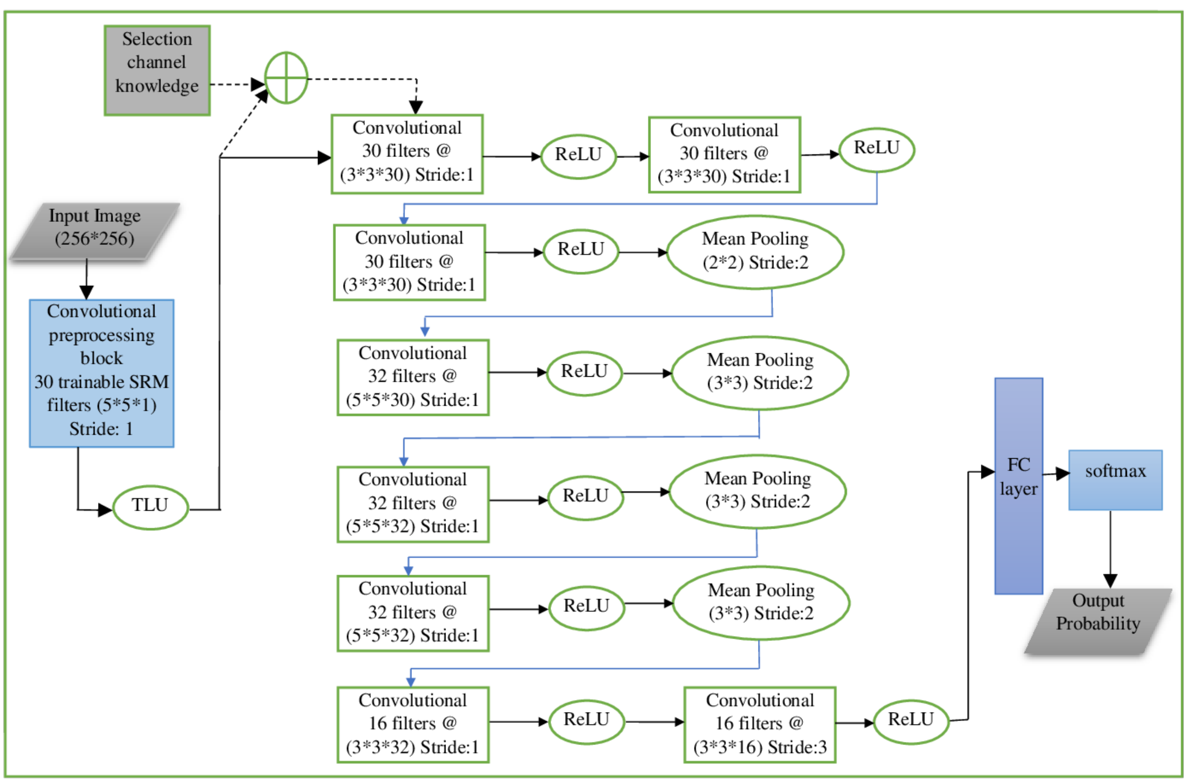

Figure 4. YeNet architecture [27]

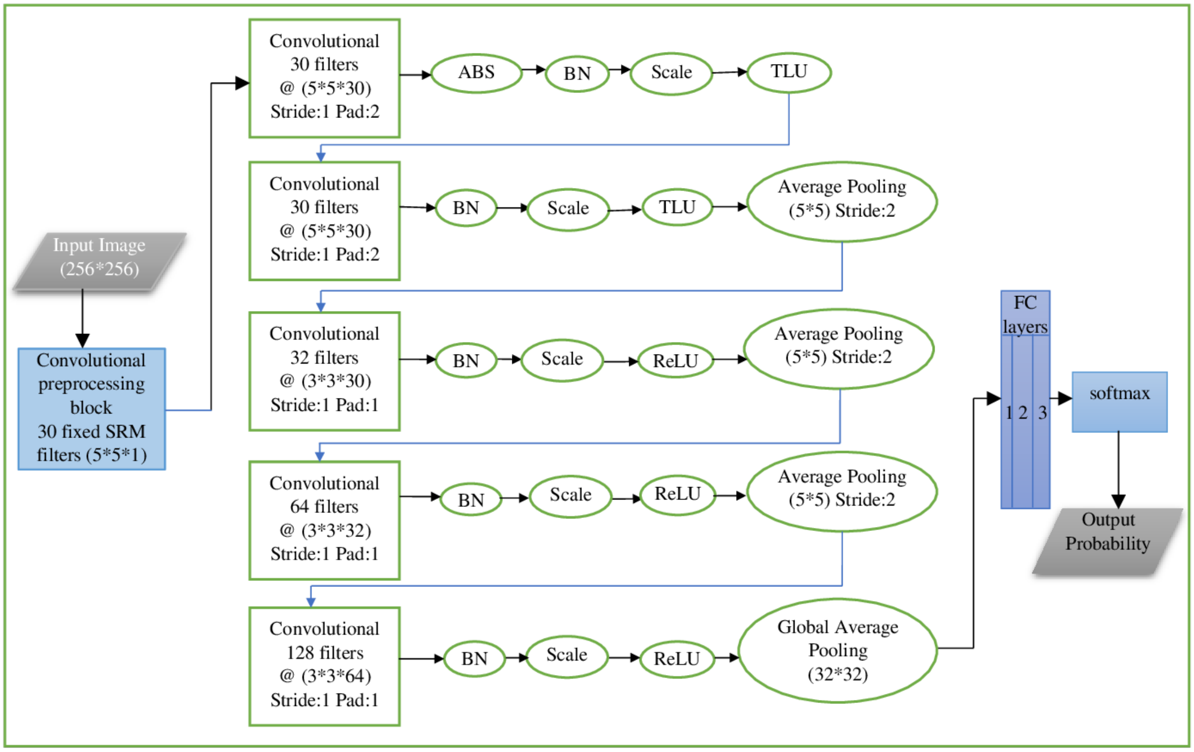

Figure 5. YedroudjNet architecture [28]

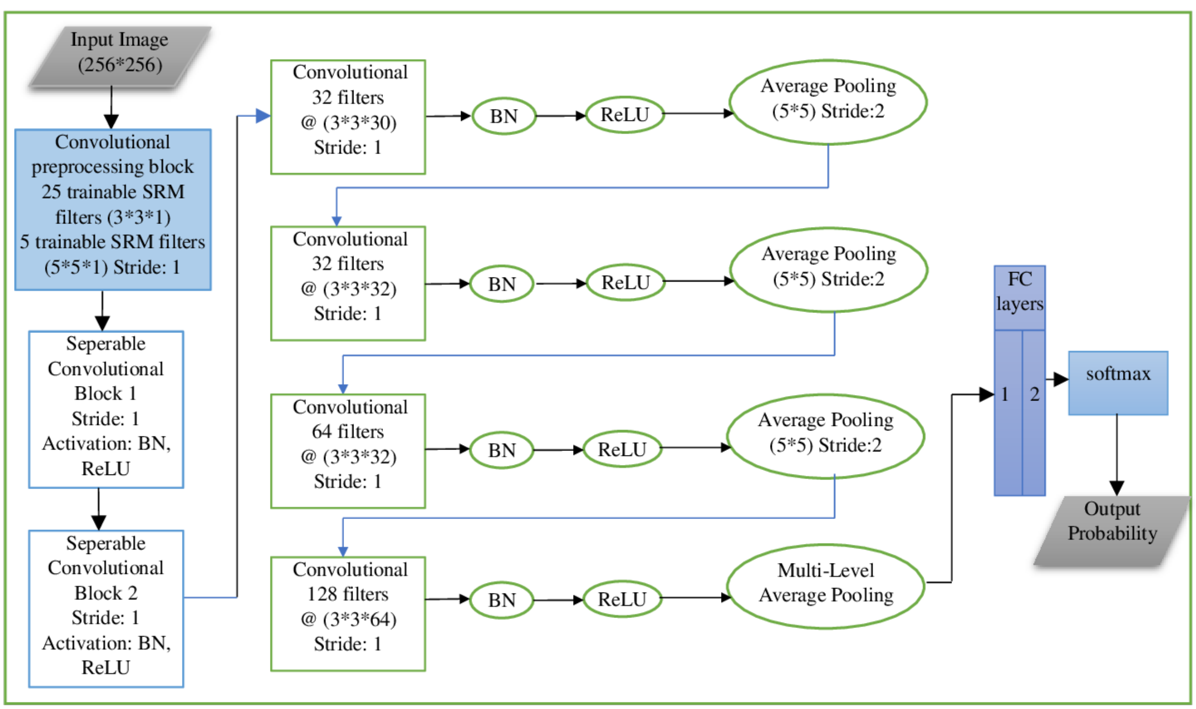

Figure 6. ZhuNet architecture [30]

stage is similar to DCTR [37] but it contains fixed 25, 5*5 DCT (Discrete Cosine Transform) kernels followed by 3 different quantization and truncation whereas 64 DCT kernels are used in DCTR. This stage is fixed as the parameters are not learnable. The output of this stage is passed to 3 identical learnable convolution networks which are finally merged with 4 fully connected layers. The inputs to the first stage are decompressed JPEG images of size 256*256. They found that inclusion of quantization and truncation step enhanced the performance of the architecture. The researchers performed modifications in XuNet [20] to make it successful for JPEG steganalysis. On the other hand, Chen et al. in [36] split the feature maps into 64 parallel channels to immerse knowledge of the jpeg phase into the architecture. Their network design considered JPEG noise residuals initially provided by Holub and Fridrich in [37][38] and later refined by using Gabor filters [39][40] and selection channel knowledge [41]. Xu in [42] used a deep residual network as explained in [43][44] for steganalysis of JPEG images distorted with the J-UNIWARD [18] steganographic technique. However, the authors also used fixed DCT kernels in the preprocessing step of the initial convolution layer.

Thus, it can be easily observed from the above survey that to make steganalysis successful with deep learning techniques, a lot of heuristics were used which were taken from SRM or J+SRM construction such as the number of filters and their weights in the image processing layer, mimicking the quantization and truncation step of SRM by incorporating activation functions such as TLU, etc. and including JPEG phase awareness in the existing architecture.

5. MAKING CNN DEEPER

Bouromand et al. [45] first came up with a deep residual architecture to overcome these limitations. Instead of using handcrafted filters, their network architecture included a broad front part to learn noise residuals automatically by training the network. Pooling was disabled in this front part so that stego signals would not get suppressed. With extensive experimentation and bringing various changes to their network, authors proved the superior performance of their work in both spatial and JPEG domain with detection accuracy much improved in JPEG steganalysis.

They used images of size 256 * 256 from BOSSBase 1.01 and BOWS2 datasets for performing their experiments and comparing their results. With a wide variety of content-adaptive steganographies in spatial as well as JPEG domain on diverse payloads, their architecture performed better than maxSRMd2, YeNet in the spatial domain, and various state of the arts in JPEG domain [36][41][42]. However, they have found that without selection channel knowledge the detection error of their network is slightly higher on WOW steganography as compared to YeNet which makes use of selection channel knowledge. Thus, their network could perform better after using selection channel information but since they did not want to use the heuristics, their future work would be to bring required modifications to the network. Thus, they have proved that with a minimum of heuristics they have used in their network for steganalysis problems, their network generalization ability is much better than other networks so far developed in this field and thus can be used to solve diverse problems instead of only steganalysis.

Li et al. [46] named their network as ReST-Net, have proposed that increasing the width of CNN as done in the inception module by Szegedy et al. [47] boosts the network performance. Authors in [47] did it by using different kernel sizes but in ReST-Net, authors have incorporated Diverse Activation Modules (DAM) in which different activation functions such as ReLU, Sigmoid, and TanH are used in different convolution groups based on which they have named their network as ReST-Net. Moreover, they have used parallel subnets, the input of which are residual maps generated from different classes of filters like Linear SRM filters, Non-linear SRM filters, and Gabor filters. Each subnet is individually trained with a classification module to optimize the parameters of the individual subnets on the BOSSBase 1.01 dataset of cover and stego images. Once the parameters are optimized, their individual classification modules are discarded and the final features of these pretrained subnets are concatenated and passed to a new classification module which will be trained for these features. In this way, authors have tried to imbibe diversity both concerning filters as well as activation functions and have proved that performance is drastically improved over XuNet2 [42] and TLUCNN version of YeNet [27] with three steganographic methods as S-UNIWARD, HILL, and HILL-CMD over a variety of payloads. They have further concluded that their network performance could be enhanced by considering the structure of ResNet [43] and Densely Connected CNNs [48].

Pathak et al. [49] have used a pretrained CNN AlexNet [50] to extract features and then used Levi Flight Based Grey Wolf Optimization (LFBGWO) to reduce the features. They have compared their results with other metaheuristic techniques [51][52][53] and opened the way for using CNN for feature extraction from images which can then be further reduced and classified with a variety of methods.

Gowda and Yuan [54] have also used the same LFGWO for feature selection along with the colorspace approach to determine if an image is hiding information or not. By using ColorNet [55], each colorspace generates its feature maps, and then weighted averaging is applied to achieve a final feature map. In ColorNet authors have found that images in different color spaces exhibit some features which are unique to that colorspace. Thus, if an image is changed then it can be detected easily by combining features from different colorspaces. After final features from different color spaces, LFGWO (meta-heuristic approach) is applied for reducing the features. The reduced features are then used to classify an image into cover or stego. The authors have named their network as StegColNet and have proved the better performance of their network than popular color steganalysis methods as WISERNet [56], DeepCNN or XuNet2 [42], DHR [27].

Recently authors have also started making use of Dense CNN in the field of steganalysis. Yang et al. [57] have used densely connected CNN inspired by DenseNet [48]. Also, they have used the selection channel information in training the network as used in SCA-GFR approach [41]. The authors have named their network as CNN-SCA-GFR and have proposed an ensemble approach making use of dense connections combining with the SCA-GFR approach. An ensemble is created by extracting features from nine CNNs and SCA-GFR. An ensemble classifier is then applied on these different feature sets and output probabilities are then averaged to differentiate between cover and stego. Authors are able to outperform SCA-GFR approach and XuNet2 methods of JPEG steganalysis for UERD [58] and J-UNIWARD steganographies at different payloads. Thus, Residual and Dense CNNs are two promising architectures that are currently being explored by researchers in the field of both spatial and transform domain steganalysis. In Residual CNN, concept of residual network is used to solve the vanishing/exploding gradient problem. A sort of skip connections is used in this network. Dense CNN encourages the use of feature reuse by connecting each layer to all the subsequent layers in a feed-forward manner. The benefit is manifold as it ameliorates the vanishing gradient problem along with substantial reduction of network parameters by reusing of features.

6. FROM ASSUMPTIONS TO REALITY

In the above forementioned review authors have incorporated different changes in CNN networks to improve the results but Cogranne et al. [59] in their ALASKA challenge have explained that before ALASKA challenge all the steganalysis work was not performed using realistic assumptions faced by forensic experts in their day-to-day situations. Authors have observed that most of the works published in journals like IH-MMSec, IEEE-TIFS and IEEE-SPL from 2016 to 2019, stego contents are not detected in environments which are nearer to forensic context. Authors have found many assumptions in a major amount of published work which are not true for images to be processed by forensic analyst. Such assumptions are more focus on spatial domain steganalysis than JPEG steganalysis, applying stego algorithms with fixed embedding rate, average detection error P_E is only used as performance metric under the assumption of same number of cover and stego images, BOSSBase 1.01 dataset with default settings is mainly used for carrying out experiments, use of grayscale images instead of colour images and several others.

Thus, ALASKA challenge was created to make steganalysis environment closer to reality. This work renames ALASKA as ALASKA-I in the rest of the paper. The competition webpage of ALASKA-I provided a training set of 50000 RAW format images captured by 21 different cameras. To make this scenario realistic, all possible sensor sizes were used with these cameras. Pictures from very low ISO to very high ISO were present in this dataset. The processing pipeline used for producing images was embellished with diversity incorporated at each step of the pipeline a) Demosaicking b) Resizing c) Smart Crop d) Denoising e) JPEG compression, etc. Diversity was also incorporated in using stego schemes, payloads, etc. The performance metric used was MD_5 (Missed Detection rate for a 5% false positive rate) to make this competition more realistic. The authors have presented the results of top 5 winners and also displayed the results with respect to different demosaicking algorithms used, different resizing applied to images, different jpeg quality factors used etc. Apart from the metric MD_5, the author also reported the winners' other performance metrics as FP_{50} (False Positive rate for 50% missed detections) and P_E. The results from the competitors were taken on a testing set in the form of most likely stego image to the least likely stego image so that authors or creators of this challenge can compute all the performance metrics of the competitors.

To break ALASKA-I, Yousfi et al. [60] have made use of SRNet deep learning framework created for the field of steganalysis. However instead of single SRNet multiple SRNets are created. The statistical moments are produced as final features by SRNets which are fed to the multi-layered multiclass perceptrons. The approach is same as used by Tsang and Fridrich [31] for accepting images of arbitrary size.

The purpose of explaining ALASKA-I challenge in this review is to brief about how the winners of this challenge have modified deep learning architecture for this realistic dataset. Thus, for any arbitrary sized input image the final features are 4 statistical moments of 512 feature maps which are then fed for classification. 5 SRNets are created for different combinations of Y, Cr and Cb channels of the coloured JPEG images. Different such detectors are created for JPEG images with different Quality Factors (QFs) having range from 60-98. For QFs 99 and 100, different approach named as reverse JPEG compatibility attack is used. Thus, to handle a diverse testing set of 5000 images, authors have created diverse architectures. Four stego algorithms J-UNIWARD, UED-JC [61], EBS [62] and nsF5 [63] are used to produce stego images. Authors have produced most of the results graphically so this work has also derived the final results from the graphs on two test sets taken by authors with respect to 3 metrics P_E, MD_5 and FP_{50}.

Being diverse in nature researchers after ALASKA-I challenge started including the ALASKA-I dataset to publish the performance of their networks for steganalysis problems. Authors, Tan et al. [64] have adopted channel-pruning scheme to prove that the performance of different existing networks like SRNet and XuNet2 can be kept comparable even after pruning the network and reducing a lot of learnable parameters. Thus, inspired by different pruning schemes already introduced in deep learning area such as ThiNet [65], L1-Norm [66], and others, authors have created hybrid pruning based on 2 pruning schemes. With a tremendous reduction of computational cost as well as model size, authors are able to achieve comparable performance on a wide variety of datasets used in the field of the steganalysis till this research such as BOSSBase 1.01, BOWS2, ImageNet and ALASKA-I. The reduced architectures are named as CALPA-SRNet after pruning SR-Net architecture and CALPA-XuNet2 after pruning XuNet2 architecture.

7. USE OF PRETRAINED NETWORKS FOR STEGANALYSIS PROBLEMS

Soon after ALASKA-I challenge, Cogranne et al. [67] published ALASKA-II challenge and its ALASKA-II dataset available at https://alaska.utt.fr. The dataset is comprised of 3*25000 cover images which are compressed with 75, 90 and 95 quality factors, and the same number of stego images produced through J-UNIWARD [18], J-MiPOD [68] and UERD [58] stego methods respectively. Most of the top performers of this challenge have adopted a completely different strategy rather than training the CNN from scratch as done in SRNet. Yousfi et al. [69] have used already trained deep networks on ImageNet database which is very popular for computer vision tasks and also has gained popularity for steganalysis in recent years. Authors have proved that these pretrained networks on millions of images can be easily adapted to JPEG steganalysis in place of complete training of well performing CNNs such as XuNet2, SRNet etc. It helps to reduce the computational complexity as well as filters of the pretrained networks such as ResNet [43], TRes-Net [70], SK-ResNeXt [71], DenseNet [48], EfficientNet [72] and MixNet [73] are highly efficient in recognising noise patterns, textures, diversity of shapes and various other attributes in which stego signals are modulated. Thus, authors have given a new direction to the task of steganalysis. Instead of training the CNNs from scratch, they took advantage of pretrained networks which were trained on millions of images for other classification tasks.

Butora et al. [74] have further taken this concept of pretraining used for steganalysis in a systematic way. Various researchers have observed that random initialization of network weights for steganalysis tasks make the CNN networks difficult to converge. Butora et al. have visualized pretraining as a different way to initialize the weights of these networks. Pretraining helps in the extraction of certain features from the natural images, e.g., edges, periodic patterns, textures etc. This is very helpful for steganalysis problems as stego signal is actually a noise moderated by the content of the image. Also, without pretraining, networks for steganalysis converges only when same cover and stego images are put in the same minibatch, which is known as Pair Constraint (PC). But this PC hinders in the network generalization ability and impairs the network performance in the last. Boroumand et al. [45] have also proved that SRNet without PC constraint helps boosting the performance of the network. In [74], authors have used SRNets which are pretrained on 3 different tasks of classification named as IN, JIN and QIN. ImageNet dataset is used for pretraining whereas pretrained networks are verified on ALASKA-II dataset and combined BOSSBase 1.01 and BOWS2 dataset. This can be also seen as transfer learning where the network is trained on one task while it is refined for the target task. Transfer learning is already proved successful in many areas as reviewed by Zhuang et al. [75]. Authors have also transferred the learning from colour images to grayscale images. In one of the results provided by the authors, results are superior for pretraining on JIN classification task as compared to IN and QIN and without pretraining at all. Yousfi et al. [76] have also used the concept of transfer learning by improving EfficientNet pretrained on ImageNet classification task for JPEG steganalysis. They have proved that certain modifications in EfficientNet architecture significantly improves steganalysis in JPEG domain. They have limited themselves to EfficientNet family because of its use by top competitors of the ALASKA-II challenge. Also, they compared their results with the winner of ALASKA-II, Xu [77], who has used the SE-ResNet architecture provided by Hu et al. [78]. The various performance metrics used for comparison are wAUC (weighted area under ROC curve), FLOPs (number of floating-point operations), number of parameters, and memory consumption.

8. A SIAMESE SOLUTION TO STEGANALYZE IMAGES OF ARBITRARY SIZES

Few researchers have focused on steganalyzing images of different sizes till now [31], [60]. You et al. [79] provided an end-to-end solution for handling arbitrary size images without the need of parameters retraining. A Siamese CNN architecture is used for providing a robust solution. Siamese networks are actually used to find similarities between two inputs. In steganalysis, this helps in finding the relationship between different parts of an image which in turn can be used to differentiate cover images from stego images. A Siamese network consists of two parallel subnets with shared weights and parameters. Authors have named their network as SiaStegNet for steganalyzing images of arbitrary sizes. Each subnet consists of preprocessing and feature extraction phase. The output of each subnet is a 128-dimensional vector results from a global pooling operation.

Thus, size of output is independent of input image size. Moreover, inspired by SID [31], the outputs of both subnets are fused by calculating elementwise four statistical moments, max, min, variance and mean. Two supervisory signals are used to update the parameters of the network, one is L_{SML} to find the similarities between different parts of an image, other is L_{CLS}, which works on statistical moments to classify the image as stego or cover. Contrastive loss is used in L_{SML} whereas cross-entropy loss is used in

L_{CLS}, and to update the network both are combined using a hyperparameter $\lambda(L_{CLS} + \lambda L_{SML})$. Authors have proved that unlike SID, their networks need not be retrained in the final layers for arbitrary size images. When comparing with SRNet on fixed size images, their network requires few parameters to be learned as compared to SRNet, while accuracy is comparable with SRNet. For arbitrary size images, their network outperforms SID for a wide variety of images. Similarly, they have proved the superiority of their network on variations of SiaStegNet architectures. They have used different sizes of images from BOSSBase 1.01 and ALASKA-II dataset and a variety of steganography algorithms for proving the robustness of their network.

Table I provides evolution of deep learning (DL) steganalysis in Spatial domain whereas Table II provides the results obtained as P_E (minimal testing error under equal priors) by applying various deep learning methods as well as other state-of-the-art methods in spatial domain. Similarly, Table III provides evolution of DL steganalysis in JPEG domain whereas Table IV provides the results obtained as P_E (minimal testing error under equal priors) by applying various deep learning methods as well as other state-of-the-art methods in JPEG domain.

TABLE I. Evolution of DL Steganalysis Methods in Spatial Domain

Authors	**Methodology**	**Dataset**	**Domain/Steganographic-algorithm**
1. Tan and Li [13]	CNN-9 layers divide into 3 stages, SCAE for unsupervised pre-training	BOSSBase 1.01	Spatial/HUGO
2. Qian et al. [15]	Gaussian Neuron CNN + Image preprocessing layer	BOSSBase 1.01 and ImageNet	Spatial/HUGO, WOW and S-UNIWARD
3. Qian et al. [19]	Deep transfer GNCNN from higher payload to low payload	BOSSBase 1.01	Spatial/WOW and S-UNIWARD
4. Xu et al. [20]	ABS+TanH+BN in convolution layer	BOSSBase1.01	Spatial/S-UNIWARD and HILL
5. Xu et al. [22]	Ensemble of CNNs	BOSSBase 1.01	Spatial/S-UNIWARD
6. Qian et al. [23]	5 filters in image processing layer and then use model combination to boost performance. Also use cropping strategy to handle arbitrary size images	BOSSBase 1.01	Spatial/ HUGO, WOW, S-UNIWARD, MiPOD and HILL-CMD
7. Ye et al [27]	High pass filters with hybrid activation function consist of Truncated Linear Unit + ReLU	BOSSBase 1.01, BOWS2 with Augmentation (AUG)	Spatial/ WOW, S-UNIWARD and HILL
8. Yedroudj et al. [28]	CNN with preprocessing filter bank, Augmented database to improve training of CNN	BOSSBase 1.01, BOWS2 and Virtual Augmented (VA)	Spatial/WOW and S-UNIWARD
9. Zhang et al. [30]	CNN with Spatial Pyramid Pooling	BOSSBase1.01, BOWS2 and AUG	Spatial/WOW and S-UNIWARD
10. Boroumand et al. [45]	Residual CNN named as SRNet	BOSSBase1.01, BOWS2 and AUG	Spatial/WOW, HILL and S-UNIWARD, JPEG/J-UNIWARD, UED-JC
11. Pathak et al. [49]	AlexNet with Levy Flight based Grey Wolf Optimization	BOSSBase1.01	Spatial/Not-mentioned
12. Gowda and Yuan [54]	Image Steganalysis with colorspace ensemble approach and Levy Flight based Grey Wolf optimization	BOSSBase 1.01 and BOWS2	Spatial/HILL and S-UNIWARD
13. You et al. [79]	Siamese architecture for steganalysis of arbitrary size images	BOSSBase 1.01 and ALASKA-II dataset	Spatial/WOW, S-UNIWARD and HILL

TABLE II. Results of DL Steganalysis Methods in Spatial Domain

Authors	**Results(% of error P_E)**
1. Tan and Li [13]	Payload-0.4 bpp SRM: -14, SPAM: - 42, CNN: - 31
2. Qian et al. [15]	Payload-0.4 bpp Dataset-BOSSBase 1.01 SRM: - HUGO (25.2), WOW (25.7), S-UNIWARD (26.3) SPAM: - HUGO (39.1), WOW (38.2), S-UNIWARD (35.1) CNN: -HUGO (28.29), WOW (29.3), S-UNIWARD (30.29) Dataset-ImageNet SRM: - HUGO (32.4), WOW (34.7), S-UNIWARD (34.4)) CNN: - HUGO (33.6), WOW (34.1), S-UNIWARD (34.7)
3. Qian et al. [19]	Pretrain-0.4bpp For WOW SRM: - 24.92(0.3 bpp), 31.75 (0.2bpp) and 39.77 (0.1bpp) CNN: - 24.87(0.3 bpp), 30.78 (0.2bpp) and 38.43 (0.1bpp) For S-UNIWARD SRM: - 24.95(0.3 bpp), 32.10 (0.2bpp) and 40.25 (0.1bpp) CNN: - 28.42 (0.3 bpp), 34.38 (0.2bpp) and 42.93 (0.1bpp)
4. Xu et al. [20]	Payload-0.1 bpp Dataset-BOSSBase 1.01 SRM: - HILL (43.56), S-UNIWARD (40.75) CNN: - HILL (41.56), S-UNIWARD (42.67) Payload-0.4 bpp Dataset-BOSSBase 1.01 SRM: - HILL (24.53), S-UNIWARD (20.47) CNN: - HILL (20.76), S-UNIWARD (19.76)
5. Xu et al. [22]	Perform better than single CNN and SRM at payload-0.4 bpp. CNNs with network size-256 AVE (Model averaging): - 18.97 CNNs with network size-128 AVE (Model averaging): - 20.39 SRM: - 20.47
6. Qian et al. [23]	Still could not compete some advanced steganalysis methods like XuNet but provide feature visualization which has not been provided in steganalysis previously. Some of the results reported are: At 0.4bpp CNN model Combination: - WOW (20.05), S-UNIWARD (21.72), MiPOD (26.07), HILL-CMD (31.7) SRM: - WOW (20.90), S-UNIWARD (20.92), MiPOD (24.32), HILL-CMD (29.83) K5*5-SRM: - WOW (26.72), S-UNIWARD (26.25), MiPOD (28.53), HILL-CMD (35.13) XuNet: - WOW (22.08), S-UNIWARD (20.70), MiPOD (23.72), HILL-CMD (30.42)
7. Ye et al. [27]	Results are better than SRM+EC with wide variety of stego algorithms and payloads. Results here are presented only for 0.1bpp. BOSSBase 1.01 and BOWS2(Train and Test) SRM: - WOW (31.63), S-UNIWARD (38.06), HILL (38.94) Proposed CNN: - WOW (24.42), S-UNIWARD (32.20), HILL (33.80)
8. Yedroudj et al. [28]	Significantly better than XuNet, YeNet and SRM+EC. For BOSSBase image size 256*256 Payload-0.2bpp Proposed YedroudjNet: - WOW (27.8), S-UNIWARD (36.7) SRM: - WOW (36.5), S-UNIWARD (36.6) XuNet: - WOW (32.4), S-UNIWARD (39.1) YeNet: - WOW (33.1), S-UNIWARD (40.0) For augmented dataset (BOSSBase 1.01, BOWS2 and VA) YedroudjNet: 20.8 YeNet: 22.2 XuNet: - 30.5

9. Zhang et al. [30]	Better than XuNet, YeNet, YedroudjNet and SRM+EC. For BOSSBase 1.01 dataset at payload 0.2bpp Proposed ZhuNet: - WOW (23.3), S-UNIWARD (28.3) SRM: - WOW (36.5), S-UNIWARD (36.6) XuNet: - WOW (32.4), S-UNIWARD (39.1) YeNet: - WOW (33.1), S-UNIWARD (40.0) YedroudjNet: - WOW (27.8), S-UNIWARD (36.7) For Augmented dataset ZhuNet: - WOW (13.1), S-UNIWARD (17.1) YeNet: - WOW (22.2), S-UNIWARD (33.5) YedroudjNet: - WOW (20.8), S-UNIWARD (31.1)
10. Boroumand et al. [45]	Results are superior both for spatial and JPEG domain and less need for steganalysis heuristics. Spatial Domain, BOSSBase 1.01 + BOWS2 dataset payload 0.1bpp Proposed SRNet: - WOW (.2877), HILL (.3134), S-UNIWARD (.3104) SCA-YeNet: - WOW (.2442), HILL (.3380), S-UNIWARD (.3220) maxSRM with Random Conditioning: - WOW (.2998), HILL (.3768), S-UNIWARD (.3817) payload 0.2bpp Proposed SRNet: - WOW (.1676), HILL (.2353), S-UNIWARD (.2090) SCA-YeNet: - WOW (.1691), HILL (.2538), S-UNIWARD (.2224) maxSRM with Random Conditioning: - WOW (.2144), HILL (.3168), S-UNIWARD (.2904) payload 0.4bpp Proposed SRNet: - WOW (.0893), HILL (.1414), S-UNIWARD (.1023) SCA-YeNet: - WOW (.0959), HILL (.1708), S-UNIWARD (.1281) maxSRM with Random Conditioning: - WOW (.1350), HILL (.2338), S-UNIWARD (.1783) JPEG Domain, BOSSBase + BOWS2 dataset, QF-75 Payload 0.1bpp Proposed SRNet: - J-UNIWARD (.3201), UED-JC (.1311) XuNet2: - J-UNIWARD (.4310), UED-JC (.2144) SCA-GFR: - J-UNIWARD (.4197), UED-JC (.3176) Payload 0.2bpp Proposed SRNet: - J-UNIWARD (.1889), UED-JC (.0568) XuNet2: - J-UNIWARD (.2849), UED-JC (.0972) SCA-GFR: - J-UNIWARD (.3257), UED-JC (.2154) Payload 0.4bpp Proposed SRNet: - J-UNIWARD (.0670), UED-JC (.0188) XuNet2: - J-UNIWARD (.1207), UED-JC (.0287) SCA-GFR: - J-UNIWARD (.1728), UED-JC (.0871)
11. Pathak et al. [49]	Results are better than other metaheuristic techniques
12. Gowda and Yuan [54]	Results are better than other steganalysis approaches used for colored images

13. You et al. [79]	SiaStegNet on image size 256*256 at 0.4bpp Learnable parameters: - 0.7Million Errors on WOW: - 7.81, S-UNIWARD: - 8.11 and HILL: - 14.03 SRNet on image size 256*256 at 0.4bpp Learnable parameters: - 4.7Million Errors on WOW: - 7.49, S-UNIWARD: - 7.78 and HILL: - 14.43 SiaStegNet for arbitrary sized images with different stego algorithms Train Size – 512 * 512 Test Sizes and Error: 512 * 512: - 24.64 512 * 640: - 24.77 640 * 512: - 25.23 640 * 640: - 24.66 SID for arbitrary sized images with different stego algorithms Train Size – 512 * 512 Test Sizes and Error: 512 * 512: - 35.14 512 * 640: - 36.99 640 * 512: - 36.74 640 * 640: - 35.97

TABLE III. Evolution of DL Steganalysis Methods in Transform Domain

Authors	**Methodology**	**Dataset**	**Domain/Steganographic-algorithm**
1. Zeng et al. [35]	1st fixed stage consisted of DCTR kernels and truncation and quantization. 2nd learnable stage consists of CNNs.	ImageNet	JPEG/ UED, UERD, J-UNIWARD
2. Chen et al. [36]	Two networks PNet and VNet are proposed.	BOSSBase 1.01 and BOWS2	JPEG/J-UNIWARD, UED-JC
3. Xu [42]	Deep residual networks and fixed DCT kernels as pre-processing step.	BOSSBase 1.01, ImageNet	JPEG/J-UNIWARD
4. Yang et al. [57]	32 layers CNN based on DenseNet.	BOSSBase 1.01, BOWS2 and ImageNet	JPEG/J-UNIWARD, UERD
5. Yousfi et al. [60]	Multiple SRNets are created with multi-layered multiclass perceptrons.	ALASKA-I dataset	JPEG/J-UNIWARD, UED-JC, EBS and nsF5
6. Tan et al. [64]	Channel Pruning of SRNet and XuNet2.	ALASKA-I dataset	Spatial- SUNIWARD, HILL JPEG- JUNIWARD, UERD
7. Yousfi et al. [69]	Use of pretrained networks which are finetuned for JPEG steganalysis.	ALASKA-II dataset	JPEG/ J-UNIWARD, J-MiPOD, UERD
8. Butora et al. [74]	Three pretrained networks IN, QIN and JIN for steganalysis.	ALASKA-II, combination of BOSSBase 1.01 and BOWS2	Spatial/ MiPOD, HILL JPEG/ J-UNIWARD
9. Yousfi et al. [76]	Modified EfficientNet architecture by disabling pooling function in initial layers.	ALASKA-II dataset	JPEG/ J-UNIWARD, J-MiPOD, UERD

TABLE IV. Results of DL Steganalysis Methods in Transform Domain

Authors	**Results(% of error P_E)**
Zeng et al. [35]	Payload-0.4 bpnzAC 3 different truncation and quantization values are: (4,1), (4,2) and (4,4) Testing error for 25 DCT kernels architecture: - 26.5 For 256 GFR kernels: - 26.4 Comparisons with state of the arts GFR, SCA-GFR and XuNet2 have plotted graphically in the paper.

Chen et al. [36]	Payload-0.1 bpnzAC BOSSBase 1.01 (Train, Test) BOWS2 (Test) CNN-PNet: - UED-JC (17.7), J-UNIWARD (35.75) CNN-VNet: - UED-JC (18.97), J-UNIWARD (36.15) SCA-GFR: - UED-JC (22.54), J-UNIWARD (35.54)	Payload-0.3 bpnzAC BOSSBase1.01 (Train, Test) BOWS2 (Test) CNN-PNet: - UED-JC (3.90), J-UNIWARD (12.28) CNN-VNet: - UED-JC (4.07), J-UNIWARD (13.32) SCA-GFR: - UED-JC (6.35), J-UNIWARD (13.44)
Xu [42]	For BOSSBase 1.01 Payload-0.1 bpnzAC QF-75 SCA-GFR: -35.98 CNN: -32.83 Payload-0.2 bpnzAC QF-75 SCA-GFR: -23.16 CNN: -19.47	Payload-0.3 bpnzAC QF-75 SCA-GFR: -14.09 CNN: -11.24 Payload-0.4 bpnzAC QF-75 SCA-GFR: -8.07 CNN: -6.41 For ImageNet Payload-0.4 bpnzAC QF-75 CNN: -16.8
Yang et al. [57]	Results are shown here on only BOSSBase 1.01 dataset on some of the payloads only. Stego method: - J-UNIWARD Payload: - 0.1 bpnzAC QF-75 Proposed CNN: - 0.3730 Proposed CNN-SCA-GFR: - 0.3596 SCA-GFR: - 0.4385 XuNet2: - 0.4163 Payload: - 0.4bpnzAC QF-75 Proposed CNN: - 0.1084 Proposed CNN-SCA-GFR: - 0.0975 SCA-GFR: - 0.1781 XuNet2: - 0.1416	Stego method-UERD Payload: - 0.1 bpnzAC QF-75 Proposed CNN: - 0.2659 Proposed CNN-SCA-GFR: - 0.2572 SCA-GFR: - 0.3600 XuNet2: - 0.3166 Payload: - 0.4bpnzAC QF-75 Proposed CNN: - 0.0573 Proposed CNN-SCA-GFR: - 0.0503 SCA-GFR: - 0.1242 XuNet2: - 0.0750
Yousfi et al. [60]	On test set mixTST MD_5: - 18.55 P_E : - 11.50 FP_{50}: - 0.09	On test set ALASKArank MD_5: - 25.2 P_E: - 14.63 FP_{50}: - 0.77
Tan et al. [64]	Stego method – JUNIWARD Payload-0.4 bpnzAC QF-75 Original SRNet Parameters: - 477.06 * 104 FLOPs: - 5.95 * 109 Error: - 7.02	CALPA-SRNet with tolerable accuracy loss 2% Parameters: - 8.43 * 104(1.77% of original parameters) FLOPs: - 2.29 * 109 (38.5% of original FLOPs) Error: - 6.88
Yousfi et al. [69]	Results are quite elaborative due to several pretrained networks used.	
Butora et al. [74]	Stego method – JUNIWARD Payload-0.2 bpnzAC, QF-75 SRNet without pretraining-0.2076 Pretrained SRNet (IN): - 0.2059 Pretrained SRNet (QIN): - 0.2260 Pretrained SRNet (JIN): - 0.1754	Stego method – JUNIWARD Payload-0.2 bpnzAC, QF-95 SRNet without pretraining: - 0.3433 Pretrained SRNet (IN): - 0.3679 Pretrained SRNet (QIN): - 0.3701 Pretrained SRNet (JIN): - 0.3294
Yousfi et al. [76]	Results are reported in terms of different performance metrics as wAUC, FLOPs etc.	So direct comparison in terms of error is not provided.

9. OPEN RESEARCH CHALLENGES AND EMERGING RESEARCH DIRECTION IN THE FIELD OF STEGANALYSIS

The open research challenges in the field of steganalysis are many: Steganalysts have to shift from various assumptions to reality while doing steganalysis. The realistic scenario faced by forensic experts for steganalysis is much different than the scenario used by different researchers. Different researchers have worked under various assumptions for steganalysis. Thus, the results they have provided are based on various assumptions such as fixed payload while converting cover to stego, more use of spatial domain steganalysis than transform domain steganalysis, preservation of cover-stego pairs and many others. With ALASKA challenge, researchers have started using steganalysis in a more realistic way. However, they have found that rather than creating CNN from scratch, pretrained networks work as a boon for steganalysis. Thus, research has shifted towards using various pretrained networks such as ResNet, EfficientNet and others. Researchers have also used pretrained networks such as ALEXNET for feature extraction in steganalysis as in [49][80]. Along with pretrained networks, recently researchers are also working on very deep networks which uses fractal net [81][82] and multi scale residual networks for steganalysis [83].

Also, performance metric such as P_E which is an average detection error is alone not sufficient to report the results in the field of steganalysis. More realistic performance metrics have come into action such as MD_5, FP_{50}, wAUC, FLOPS and others. Thus, in the field of steganalysis research has shifted from traditional approach to a more realistic approach with improved performance metrics.

10. CONCLUSION

As applying filters for feature extraction and classification in steganalysis is adaptable to CNN structure so researchers are moving towards exploring residual connections in CNN as well as densely connected CNNs. Different meta-heuristic techniques are also tried by researchers after extracting the features from CNN to optimize these features. In a nutshell, researchers have also come up with an ensemble of CNNs, and an ensemble of CNN with some meta-heuristic techniques for both spatial and transform domain steganalysis.

Very recently from ALASKA challenge, researchers came up with the idea of using highly efficient pretrained networks for steganalysis tasks. Researchers have started believing that instead of training deep networks from scratch on steganalysis tasks, networks which are already trained on some other tasks can be finetuned on steganalysis purpose. With this perspective, researchers have found excellent results and the top competitors of the ALASKA II challenge have used same strategy of fine tuning pretrained networks. From Table III and Table IV, it can be seen that pretrained networks have proved highly efficient for steganalysis. Table IV provides the results of pretrained SRNets [74] and demonstrates quite good results even at a very low payload rate of 0.2 bpp. Similarly, pretrained networks with improved performance metrics for steganalysis are also reviewed in this work as in [76], but results in numeric terms have not provided directly. Table IV provides the collective view of using pretrained networks both in spatial as well as JPEG domain. Therefore, results of pretrained networks have not included in Table II. Thus, a new angle of using pretrained networks is formed in the research of steganalysis and is the current trend in the area of applying deep learning for steganalysis purpose.

References

[1] C. K. Chan and L. M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[2] J. Mielikainen, "Lsb matching revisited," *IEEE signal processing letters*, vol. 13, no. 5, pp. 285–287, 2006.

[3] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE security & privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[4] J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.

[5] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE transactions on Image Processing*, vol. 12, pp. 221–229, 2003.

[6] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on information Forensics and Security*, vol. 5, pp. 215–224, 2010.

[7] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," *2012 IEEE International workshop on information forensics and security (WIFS)*, pp. 234–239, 2012.

[8] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," *International workshop on information hiding*, pp. 161–177, 2010.

[9] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on information Forensics and Security*, vol. 7, pp. 868–882, 2012.

[10] J. Kodovský and J. Fridrich, "Steganalysis of jpeg images using rich models," *Media Watermarking, Security, and Forensics 2012*, vol. 8303, pp. 81–93, 2012.

[11] V. Holub and J. Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Transactions on information forensics and security*, vol. 8, pp. 1996–2006, 2013.

[12] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 48–53, 2014.

[13] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," *Signal and information processing association annual summit and conference (APSIPA), 2014 Asia-Pacific*, pp. 1–4, 2014.

[14] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 432–444, 2011.

[15] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Media Watermarking, Security, and Forensics 2015*, vol. 9409, pp. 171–180, 2015.

[16] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: the ins and outs of organizing boss," *International workshop on information hiding*, pp. 59–70, 2011.

[17] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[18] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, pp. 1–13, 2014.

[19] Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," *2016 IEEE international conference on image processing (ICIP)*, pp. 2752–2756, 2016.

[20] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, pp. 708–712, 2016.

[21] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, 2014.

[22] G. Xu, H.-Z. Wu, and Y. Q. Shi, "Ensemble of cnns for steganalysis: An empirical study," *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 103–107, 2016.

[23] Y. Qian, J. Dong, W. Wang, and T. Tan, "Feature learning for steganalysis using convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, pp. 19 633–19 657, 2018.

[24] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 221–234, 2015.

[25] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 1905–1917, 2015.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[27] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 2545–2557, 2017.

[28] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient cnn for spatial steganalysis," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2092–2096, 2018.

[29] P. Bas and T. Furon, "Bows-2 contest (break our watermarking system). organized between the 17th of july 2007 and the 17th of april 2008," 2008. [Online]. Available: http://bows2.ec-lille.fr/

[30] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Efficient feature learning and multi-size image steganalysis based on cnn," *arXiv preprint arXiv:1807.11428*, 2018.

[31] C. F. Tsang and J. Fridrich, "Steganalyzing images of arbitrary size with cnns," *Electronic Imaging*, vol. 2018, p. 121, 2018.

[32] A. D. Ker, "Batch steganography and pooled steganalysis," *International Workshop on Information Hiding*, pp. 265–281, 2006.

[33] T. Pevný and I. Nikolaev, "Optimizing pooling function for pooled steganalysis," *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2015.

[34] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich, "The square root law of steganographic capacity," *Proceedings of the 10th ACM workshop on Multimedia and security*, pp. 107–116, 2008.

[35] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale jpeg image steganalysis using hybrid deep-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 1200–1214, 2017.

[36] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "Jpeg-phase-aware convolutional neural network for steganalysis of jpeg images," *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, pp. 75–84, 2017.

[37] V. Holub and J. Fridrich, "Low-complexity features for jpeg steganalysis using undecimated dct," *IEEE Transactions on Information forensics and security*, vol. 10, pp. 219–228, 2014.

[38] V. Holub and J. Fridrich, "Phase-aware projection model for steganalysis of jpeg images," *Media Watermarking, Security, and Forensics 2015*, vol. 9409, pp. 259–269, 2015.

[39] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive jpeg steganography using 2d gabor filters," *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*, pp. 15–23, 2015.

[40] C. Xia, Q. Guan, X. Zhao, Z. Xu, and Y. Ma, "Improving gfr steganalysis features by using gabor symmetry and weighted histograms," *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pp. 55–66, 2017.

[41] T. D. Denemark, M. Boroumand, and J. Fridrich, "Steganalysis features for content-adaptive jpeg steganography," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 1736–1746, 2016.

[42] G. Xu, "Deep convolutional neural network to detect j-uniward," *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, pp. 67–73, 2017.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *European conference on computer vision*, pp. 630–645, 2016.

[45] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network

for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 1181–1193, 2018.

[46] B. Li, W. Wei, A. Ferreira, and S. Tan, "Rest-net: Diverse activation modules and parallel subnets-based cnn for spatial image steganalysis," *IEEE Signal Processing Letters*, vol. 25, pp. 650–654, 2018.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[48] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[49] Y. Pathak, K. V. Arya, and S. Tiwari, "Feature selection for image steganalysis using levy flight-based grey wolf optimization," *Multimedia Tools and Applications*, vol. 78, pp. 1473–1494, 2019.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[51] R. R. Chhikara, P. Sharma, and L. Singh, "A hybrid feature selection approach based on improved pso and filter approaches for image steganalysis," *International Journal of Machine Learning and Cybernetics*, vol. 7, pp. 1195–1206, 2016.

[52] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[53] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information sciences*, vol. 179, pp. 2232–2248, 2009.

[54] S. N. Gowda and C. Yuan, "Stegcolnet: Steganalysis based on an ensemble colorspace approach," *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 313–323, 2021.

[55] S. N. Gowda and C. Yuan, "Colornet: Investigating the importance of color spaces for image classification," *Asian Conference on Computer Vision*, pp. 581–596, 2018.

[56] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "Wisernet: Wider separate-then-reunion network for steganalysis of color images," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 2735–2748, 2019.

[57] J. Yang, X. Kang, E. K. Wong, and Y. Q. Shi, "Jpeg steganalysis with combined dense connected cnns and sca-gfr," *Multimedia Tools and Applications*, vol. 78, pp. 8481–8495, 2019.

[58] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient jpeg steganography," *IEEE transactions on Information Forensics and Security*, vol. 9, pp. 814–825, 2014.

[59] R. Cogranne, Q. Giboulot, and P. Bas, "The alaska steganalysis challenge: A first step towards steganalysis," *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pp. 125–137, 2019.

[60] Y. Yousfi, J. Butora, J. Fridrich, and Q. Giboulot, "Breaking alaska: Color separation for steganalysis in jpeg domain," *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pp. 138–149, 2019.

[61] L. Guo, J. Ni, W. Su, C. Tang, and Y. Q. Shi, "Using statistical image model for jpeg steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 2669–2680, 2015.

[62] C. Wang and J. Ni, "An efficient jpeg steganographic scheme based on the block entropy of dct coefficients," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1785–1788, 2012.

[63] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities," *Proceedings of the 9th workshop on Multimedia & security*, pp. 3–14, 2007.

[64] S. Tan, W. Wu, Z. Shao, Q. Li, B. Li, and J. Huang, "Calpa-net: channel-pruning-assisted deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 131–146, 2020.

[65] J. H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

[66] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[67] R. Cogranne, Q. Giboulot, and P. Bas, "Alaska# 2: Challenging academic research on steganalysis with realistic images," *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–5, 2020.

[68] R. Cogranne, Q. Giboulot, and P. Bas, "Steganography by minimizing statistical detectability: The cases of jpeg and color images," *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pp. 161–167, 2020.

[69] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, "Imagenet pre-trained cnns for jpeg steganalysis," *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2020.

[70] T. Ridnik, H. Lawen, A. Noy, E. B. Baruch, G. Sharir, and I. Friedman, "Tresnet: High performance gpu-dedicated architecture," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1400–1409, 2021.

[71] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 510–519, 2019.

[72] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *International conference on machine learning*, pp. 6105–6114, 2019.

[73] M. Tan and Q. V. Le, "Mixconv: Mixed depthwise convolutional kernels," *arXiv preprint arXiv:1907.09595*, 2019.

[74] J. Butora, Y. Yousfi, and J. Fridrich, "How to pretrain for steganalysis," *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, pp. 143–148, 2021.

[75] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and

Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, pp. 43–76, 2020.

[76] Y. Yousfi, J. Butora, J. Fridrich, and C. F. Tsang, "Improving efficientnet for jpeg steganalysis," *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, pp. 149–157, 2021.

[77] G. Xu. (2020) 1st place solution. [Online]. Available: https://www.kaggle.com/c/alaska2-image-steganalysis/discussion/168548

[78] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[79] W. You, H. Zhang, and X. Zhao, "A siamese cnn for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 291–306, 2020.

[80] I. T. Ahmed, B. T. Hammad, and N. Jamil, "Image steganalysis based on pretrained convolutional neural networks," *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA)*, pp. 283–286, 2022.

[81] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[82] B. Singh, A. Sur, and P. Mitra, "Steganalysis of digital images using deep fractal network," *IEEE Transactions on Computational Social Systems*, vol. 8, pp. 599–606, 2021.

[83] H. Chen, Q. Han, Q. Li, and X. Tong, "Image steganalysis with multi-scale residual network," *Multimedia Tools and Applications*, pp. 1–23, 2022.

Ankita Gupta is working as a Research Scholar in the Department of Computer Science and Engineering, The NorthCap University, Gurugram, Haryana, India. She did her M.Tech. in CSE from The NorthCap University in 2017. She has cleared UGC-NET and JRF exam and is now working as SRF under UGC scheme. Her research areas include Machine Learning, Data Science and Steganalysis.

Rita Chhikara is working as a professor in the department of CSE at The NorthCap University, Gurugram. She has an excellent teaching experience of more than 20 years in various esteemed institutions. She has completed her Ph.D. from The NorthCap University (NCU) in Data Mining. She has completed two projects titled 'Neural Network based Steganalysis' funded by the Department of Science and Technology and 'Detecting Dementia using Deep Neural Network' funded by DST, CSRI. Her current areas of research include Data Mining, Pattern Recognition, Machine Learning, Image Processing and Deep Learning. She has published around 50 papers in peer reviewed international journals with good indexing and reputed national/international conference proceedings. She is a member of ISTE, ACM, IEEE and IET.

Prabha Sharma is currently an emeritus faculty at The NorthCap University, Gurugram, Haryana. She did her Ph.D. from Northwestern University, USA in 1970 and was post-doctoral fellow at University of Florida, Gainesville, USA from 1970 to 1971. She taught at University of Florida, Gainesville, USA and at IIT Kanpur till 2011. Her areas of research include Design and Analysis of Algorithms for Combinatorial Optimisation problems, Machine Learning, Steganalysis and, Steganography. She has published in national and international journals and guided, nine Ph.D. students.