Spatial Image Steganalysis using Transfer Learning Approach on Embedded JPEG Images

Nirav Jain[1], Yash Javeri[1],
Sudhir Bagul[1]

[1]Dwarkadas J. Sanghvi College of
Engineering, Mumbai, India.

niravjain1709@gmail.com,
yjaveri99@gmail.com,
sudhir.bagul@djsce.ac.in

Abstract. Steganalysis is the process of revealing the hidden messages embedded inside an image. For many years Ensemble Classifiers were used to identify the presence of hidden messages in images until recently, researchers started using deep learning, specifically Convolution Neural Networks, for this purpose. Deep learning is a method of training a model architecture over a large-scale, relevant and diverse dataset. In this paper, we have studied the effects of 3 transfer learning models in detecting the presence of hidden messages in images. MobileNet, EfficientNetB3, and ResNet50 models are used as the base-models pre-trained over the imagenet dataset. These models are then combined with our outer model and trained over our proposed dataset, hence accomplishing the aim of using transfer learning. Previous researches experimented with BOSSbase, WOW and/or S-UNIWARD images with different payload rates. In this paper, the dataset consists of the combination of images on which embedding algorithms, namely JMiPOD, J-UNIWARD and UERD are applied. We have also aimed to determine the possible reasons behind the results we observed in our research.

Keywords: computer vision, convolution neural networks, deep learning, image steganalysis, transfer learning

1 Introduction

Steganography can be seen as an advanced version of cryptography. In cryptography, the text to be sent is encoded by the sender and decoded by the receiver. If an attacker tries to steal the message midway, the message will be in the encoded form, but the attacker knows that it is the required message and hence he can try to decode it. On the other hand, steganography refers to hiding the message in the first place (the message can be initially encrypted before hiding to increase the security) in an image. So now, if an attacker tries to steal the message midway, it will be tough for him as he does not even know that the message exists.

With the increase in social media content like images, videos, etc. the importance of steganography has increased to a considerable extent. Not only is it used to share messages in a private and secure way it is also used by attackers to attack normal users who download images online, which might contain malicious scripts embedded in them. Different steganographic methods hide the data in different ways [1], which might range from changing the LSB's of the image to changing the DCT (Discrete Cosine Transform) coefficients of the image. Hence developing a generalized method

to find stego-images becomes important.

Steganalysis is defined as the set of methods used to detect hidden contents in an image. The research in steganalysis has increased a lot recently owing to the advancements in steganographic techniques. Using all the different statistical methods for steganalysis is not feasible and hence a generalized solution is desired. The use of Convolutional Neural Networks for image steganalysis has seen a rise because of the generalization and the efficient results that they provide [2]. CNN's have the ability to detect very high-level features from images and then use these features for further classification.

Over the past few years, a lot of robust and efficient CNN architectures have been developed. The Imagenet competition has invited many researchers to solve a multiclass classification problem, which has led to the development of models like the MobileNet [3], EfficientNet [4], ResNet [5], VGG [6], Xception [7], etc. All of these models have their own benefits and unique architectures to obtain the desired accuracy and each one of these models has been trained on the imagenet dataset for millions of iterations. Hence the trained models have very rich feature extractors that can be used for different classification models using a method called transfer learning. In transfer learning, we use the knowledge of a model in a particular domain to solve problems of a different but similar domain. Transfer learning has changed the way CNN's are trained. Instead of training everything from scratch every time a new model is created, which naturally uses up a lot of resources and time as well, we can simply leverage the power of a pre-trained model and train only the layers we require. Transfer learning has proved to be useful in solving different problems and hence using it for image steganalysis might prove beneficial.

The method that we propose uses transfer learning for image steganalysis. Transfer learning has been used before for image steganalysis like in [8] using ResNet but the possibility of using other pre-trained models for the same is yet to be explored. For this purpose, we propose the effects of using different pre-trained models as feature extractors for our classification problem. In this paper, we only discuss the effects of using different pre-trained models, viz ImageNetV2, EfficientNetB3 and ResNet50, as the base model and hence the outer model is defined only for classification purposes.

MobileNetV2 [9] was introduced in 2018 as an improvement to the original MobileNet. MobileNets were created with an aim to reduce the computational cost of computer vision models to a limit that could be easily handled by mobile phones while still maintaining good accuracy. EfficientNet was created to improve the model's accuracy on the imagenet dataset by tuning network parameters like depth, width, and image resolution. ResNet basically uses skip connections to avoid vanishing gradients and improve the model's accuracy.

We have used Alaska Image Steganalysis Dataset [10] from Kaggle for image steganalysis. It contains cover as well as images embedded using J-UNIWARD, JMiPOD and UERD algorithms. In this paper, we try to determine the effects that transfer learning has on the models trained on the described dataset by using the discussed base models. We then try to find the various reasons for the differences in results and identify the one that is best suited for image steganalysis.

2 Related Work

In [8], the authors have applied transfer learning in Image Steganalysis and compared it with a model without transfer learning. They have used the ResNet50 model by the Keras Framework (pre-trained over the 'imagenet' dataset) as their model architecture for the same. An output layer is added at the end of the model with a sigmoid activation function followed by the compilation of the model, which is performed with binary

cross-entropy as the loss function and AdamW as the model's optimizer. The paper proposes to reduce the learning rate to 10 -5 if validation loss does not decrease for 2 continuous epochs of training. They have used two separate datasets for training and testing purposes, BOSSbase v1.01 [11] and BOSSbase v0.92 for training, and BOWS2OrigEp3 for testing. After some preprocessing on the images, various steganographic algorithms are applied separately on the dataset namely WOW and HUGO algorithm. WOW and HUGO is used to create images with 10 different payload rates 0.1bpp, 0.2bpp, 0.3bpp, 0.4bpp, 0.5bpp, 0.6bpp, 0.7bpp, 0.8bpp, 0.9bpp and 1.0bpp. The model without transfer learning involves separately training their architecture on these 10 generated datasets, whereas the transfer learning model involves training their architecture, first over the images generated by applying the Least Significant Bit (LSB) algorithm, and then the 10 sets of images with different payload rates as discussed above, in a sequential manner. This way the transfer learning model learns to predict in the order of decreasing difficulty of the steganography images. The test results indicate that the predictions favored the model with transfer learning on the images in the latter range of payload rates, which is 0.5bpp to 1.0bpp, compared to the 0.1bpp-0.4bpp payload rates.

In [12], the paper's authors present another way to classify images as original or stego-images by proposing a CNN network called the Cover Image Suppression Network (CIS-Net). The main highlight of the paper includes the two new layers called the STL (Single-Value Truncation Layer) and the SPL (Sub-linear Pooling Layer). The STL is a part of the preprocessing block and it basically truncates the inputs to a predefined interval in order to filter out high amplitude, low-frequency cover image content, without affecting the embedded message greatly. The authors propose this layer in order to ease the process of classification. The sub-linear pooling layer is a part of the type-2 block of the network and is primarily used to suppress the cover image content while still preserving the embedded message. The bias for the network is initialized based on input cover-stego pairs. The network also uses different learning rates for different layers and is trained on the BOSSbase database using curriculum learning [13]. The model gives better results than previously built similar networks but it considers only the images having a payload in the range of 0.1bpp - 0.5bpp, and not above, as in the case of [8].

In "Yedroudj-Net: An Efficient CNN for Spatial Steganalysis" [14], the researchers have proposed their own CNN architecture for Spatial Steganalysis. The model consists of three major parts. The first part is the pre-processing layer consisting of 30 basic high-pass filters (to increase the rate of convergence). This filter is added to reduce the noise with respect to the signal, suppress the image's content, and reduce the dynamic range. The second part is the major chunk of the architecture consisting of 5 blocks each having a convolution layer followed by Batch Normalization, Scaling, and ReLU activation function, and Average Pooling (except for the last block). They have also added a Truncation activation function along with the scaling layer for the second block. The last part consists of 3 fully connected layers followed by softmax activation. The pre-processing, Truncation activation, Batch Normalization with a scaling layer are the elements of the architecture that have played a crucial role in the steganalysis performance. They have reported the error probability for their own model Yedroudj-Net along with Ye-Net [15], Xu-Net [16], SRM+EC [17, 18] for the datasets WOW and S-UNIWARD and payload rates of 0.2bpp and 0.4bpp each, and the observations indicate that Yedroudj-Net has the least error probability of 27.8%, 14.1%, 22.8% amongst all 4 models, for images with 0.2bpp(WOW), 0.4bpp(WOW), 0.4bpp(S-UNIWARD) payload rates respectively.

In the paper "Structural Design of Convolutional Neural Networks for Steganalysis" [19], the authors propose a way to design Convolutional Neural Networks for Image

Steganalysis. The proposed CNN architecture in the paper consists of 5 groups of layers where each group has been selected for better performance. Group 1 and group 2 consist of non-linear activation functions like tanh for better statistical modeling whereas groups 3 to 5 consist of the ReLU activation function. The abs layer in each group takes into account the symmetry in noise residuals for better statistical modeling. Also, a batch normalization layer is applied before each nonlinear activation to avoid the problems of local minima. The researchers have trained the model on the Boss base dataset with embedding rates of 0.1 and 0.4bpp. The model achieves an accuracy of 57.33% on 0.1bpp images and an accuracy of 80.24% on 0.4bpp images. The proposed model still requires modifications as models with more complex steganalysis methods have proved more accurate.

3 Proposed Methodology

The proposed methodology involves studying the effects of various transfer learning models in Image Steganalysis. We propose to make use of the ResNet50, MobileNet, and the EfficientNetB3 CNN models as our base models. For this experiment, we chose the functional API of the deep learning framework Keras written in Python. For each model, the methodology is divided into 3 parts. The first step is to load the weights of the respective model, pre-trained over the 'imagenet' dataset. The second step is to combine the base model with an outer model consisting of DenseLayers. The third step involves compiling the model followed by training the model over the proposed dataset. Then all 3 models are evaluated individually over 3 types of stego-images (J-UNIWARD, JMiPOD, UERD).

3.1 Dataset

The dataset used here is the ALASKA2 Image Steganalysis dataset from Kaggle by the Troyes University of Technology. The dataset includes a great number of pure images (cover images) along with stego-images, where the information is hidden within the images using various steganography algorithms. In this case, the 3 algorithms used for hiding the secret messages are JMiPOD, J-UNIWARD, UERD with the same probability. Figure 1 represents the pixel differences between the cover image and the respective stego-images. Message length, on an average, is 0.4 bit per non-zero AC DCT coefficient. The payload rates are adjusted in such a way that the difficulties of the images (i.e. measure of how different is the stego-image from the initial pure image) are nearly the same.

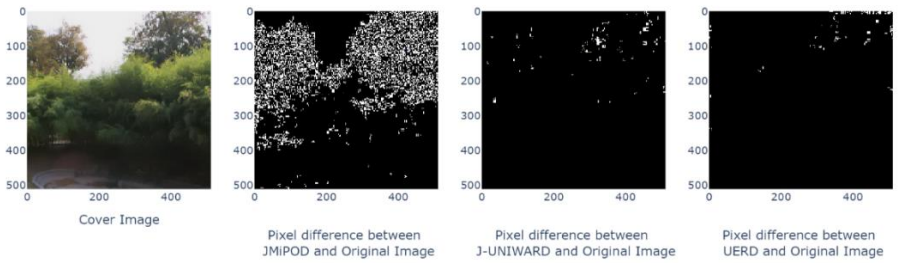

Fig. 1. Pixel difference between different stego-images and the corresponding cover image

3.2 Base Model

The following is the detailed information about the 3 models that are selected for this experiment as the base models. These models are also optimized to fit with our input shape 512 x 512.

MobileNetV2: The first model that we use is MobileNetV2. The main highlight of the MobileNet architecture is the presence of depth-wise separable convolutions. MobilenetV2 extends these benefits by including a shortcut or residual connection, or more specifically an inverted residual block, along with linear bottlenecks which not only improve the training accuracy but also help in dealing with the vanishing gradient problem. The presence of depth-wise separable convolution helps maintain the information across all the channels while the pointwise convolution helps combine the outputs of the depth-wise separable convolution with minimum information loss. We use MobileNetV2 for our image steganalysis problem because of its rich feature extractor and the presence of depth-wise separable convolutions and average pooling layer,s which may help preserve the noise required for classification than eliminating it.

EfficientNetB3: The second model that we have used is EfficientNetB3. The main idea behind all the EfficientNets is the compound scaling of network parameters like depth, width, and resolution, on a base architecture, in an optimum way, to increase the network's accuracy and efficiency. The network consists of MBConv blocks which are nothing but the inverted residual blocks as used in MobileNet. The optimally tuned parameters for increasing a network's efficiency is what we use this network for. The model that we use is EfficientNetB3 as it has fewer number of parameters, hence computationally efficient for the number of resources available for the research, compared to models like EfficientNetB4 to EfficientNetB7 which have a huge number of parameters. On the other hand, EfficientNetB3 performs better in terms of accuracy on the imagenet dataset as compared to models from EfficientNetB0 to EfficientNetB2.

ResNet50: ResNet stands for Residual Networks. ResNet was the winner of the 2015 Imagenet challenge. Many previous models like AlexNet (winner of 2012) or VGG (winner of 2014) have inferior performance when compared to ResNet in some cases. When stacking up layers over each other, the gradient value starts diminishing and becomes significantly small. This problem is known as the vanishing gradient problem. ResNet solved this by introducing the identity blocks, where the model is divided into multiple convolution blocks and apart from stacking the convolution layers it also adds the output of each block with the original input for that respective block. This ensures that there is no significant vanish in the gradient values and the upper layers perform as good as the lower layers. This is also known as 'skip connection'. These modifications to the ResNet model as compared to the previous models make it a strong pillar in computer vision applications.

3.3 Outer Model

The outer model as shown in Figure 2 consists of a fully connected layer with 3 blocks, each having 1 Dense layer followed by ReLU activation followed by a dropout as seen in Figure 2. The output of the Dense layers in the first 2 blocks have1024 units each and the third Dense layer has 512 output units. These 3 blocks are followed by a final Dense layer followed by a sigmoid activation function which is defined in (1).

$$Sigmoid = \frac{1}{1+e^{-z}} \quad (1)$$

Sigmoid, a nonlinear function, is widely used in applications involving supervised learning. The function returns values between 0 and 1, and since in this experiment,

probabilities are expected at the output, sigmoid can be a good fit. We use this outer model followed by a final sigmoid layer for all the three base models because we aim to find the effects of using transfer learning on image steganalysis with the discussed base models.

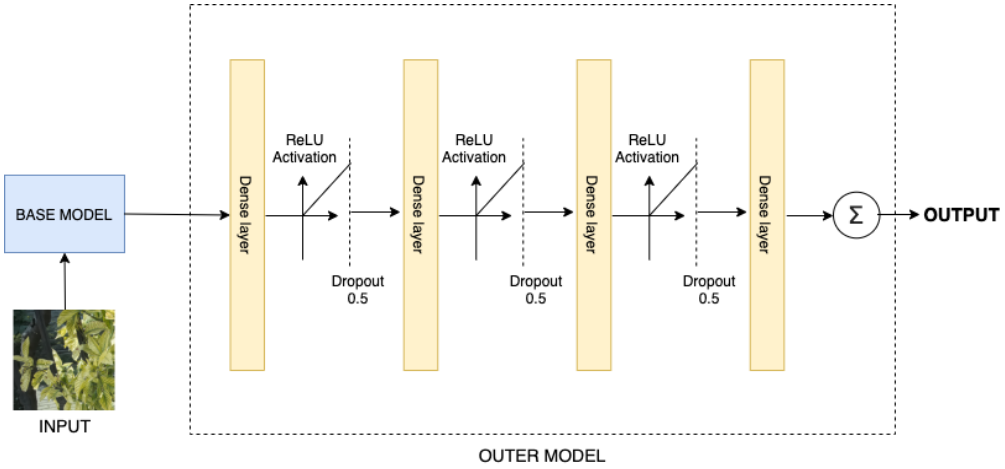

Fig. 2. Proposed Model Architecture

3.4 Training

Adam optimizer is used as the optimizer for the training of all 3 final models with $\beta 1$ = 0.9 and $\beta 2$ = 0.999. The reason behind having Adam as the optimizer is that it inherits from RMSprop and AdaGrad, which are not much sensitive to hyperparameters, and it also has adaptive learning rates. Moreover, it stores exponentially decaying past squared-gradients' average as well as the average of past gradients (similar to momentum).
The model is trained over the ALASKA2 dataset (consisting of steganographic images + cover images) for 50 epochs with a learning rate of 10-3. While training the entire model over the dataset, it is ensured that the base model is not being trained during this process, and it is just being used to predict the intermediate outputs(features). We were able to achieve this by setting the trainable parameter of the base models to 'False', as provided by the Keras framework.

3.5 Training and Testing Environment

The entire training and evaluation were performed in Kaggle's TPU environment. The TPU consists of 4 dual-core TPU (i.e. 8 cores) with a high-speed memory of size 128GB. Google Cloud Storage was used for efficient data feeding to the TPU so that the TPU never starves.

4 Results and discussion

As mentioned above, the 3 models are trained over the proposed dataset and the graph of accuracy vs epochs for all 3 models and their corresponding confusion matrices are shown in Figures 3-8. Figure 3 is the observation for the MobileNet model. The training accuracy starts increasing rapidly as compared to the validation accuracy, indicating that the model has started overfitting the training dataset after the 10th epoch. Similarly, in the case of EfficientNetB3 (Figure 5), the model starts overfitting the dataset after the 7th epoch. Although, the ResNet50 model shows hardly any change in the accuracy, giving the impression that the model is not getting trained at all. But the confusion matrix and the classification report for the individual image-types indicate that it has been trained but with inferior performance. As transfer learning is used, all the models are prone to overfitting if we unfreeze the layers of the base model and train.

The confusion matrices for the three models clearly indicate that the model having MobileNetV2 as its base model is pretty good at classifying the steganographic and cover images. However, as seen in Figure 4 it is prone to a high number of false negatives i.e. it is prone to classifying a large number of stego-images as cover images. EfficientNetB3 however, has a higher number of false negatives as seen in Figure 6, but is still pretty decent at image steganalysis. ResNet50 on the other hand is not able to classify the images properly; it has a very high number of false positives indicating that the model is pretty biased towards classifying images as steganographic.

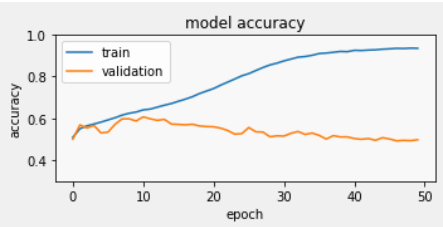

Fig. 3. Change in accuracy for MobileNetV2

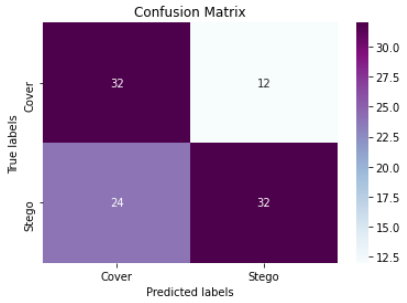

Fig. 4. Confusion matrix for MobileNetV2

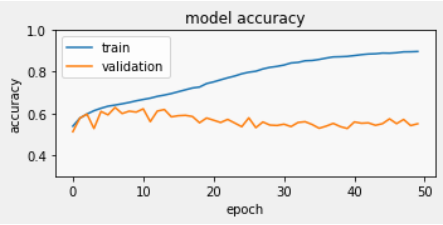

Fig. 5. Change in accuracy for EfficientNetB3

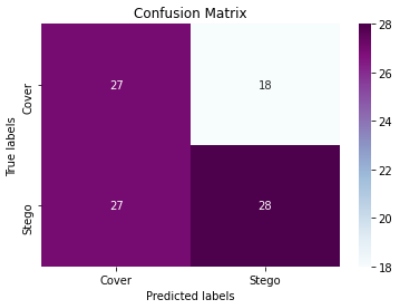

Fig. 6. Confusion matrix for EfficientNetB3

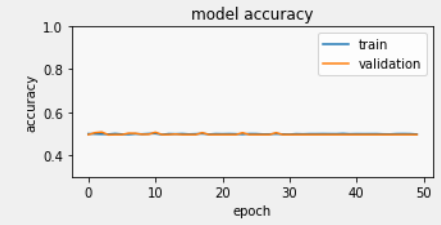

Fig. 7. Change in accuracy for ResNet50

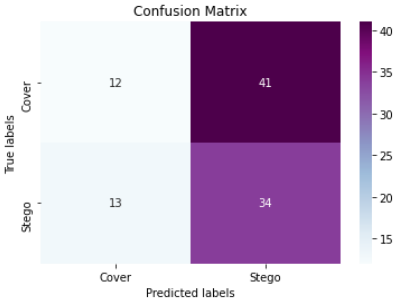

Fig. 8. Confusion matrix for ResNet50

After looking at the overall accuracy graphs and confusion matrices we evaluated the models individually to test how they perform in classifying the images as steganographic for each of the 3 steganographic algorithms that the dataset consists of. As can be seen from the following three tables, viz. Table-I, Table-II & Table-III, we can say that MobileNetV2 is the best at classifying images embedded using JMiPOD and UERD in terms of accuracy. EfficientNetB3 also shows decent results in terms of accuracy and is better than ResNet50. ResNet50 on the other hand shows high recall in all the three cases, and high F1-score in JMiPOD and J-UNIWARD but has very low precision values. Depending on the type of metric being considered we can either select EfficientNetB3 or ResNet50 for classifying J-UNIWARD images. For example, if not missing out on detecting the stego-images is our priority, ResNet50 can be a better option since it has a high recall. However, if we consider the overall performance of the three networks, we can say that MobileNetV2 might be the best for image steganalysis, followed by EfficientNetB3, which may be followed by ResNet50.

Table 1. Model performance for JMIPOD images.

	Accuracy	**Recall**	**Precision**	**F1-Score**
MobileNet	0.68	0.52	0.68	0.59
EfficientNetB3	0.62	0.68	0.56	0.61
ResNet50	0.51	0.86	0.47	0.61

Table 2. Model performance for J-UNIWARD images.

	Accuracy	Recall	Precision	F1-Score
MobileNet	0.59	0.23	0.48	0.31
EfficientNetB3	0.61	0.48	0.47	0.47
ResNet50	0.61	0.84	0.46	0.60

Table 3. Model performance for UERD images.

	Accuracy	Recall	Precision	F1-Score
MobileNet	0.74	0.66	0.72	0.69
EfficientNetB3	0.68	0.82	0.60	0.69
ResNet50	0.45	0.73	0.43	0.54

5 Conclusion

In this paper, we show the effects of MobileNetV2, EfficientNetB3 and ResNet50 on image steganalysis. Based on the results, we find out that MobileNetV2 is best suited for this purpose compared to the other two base models. We also try to find out the important reasons that might support the experimental results thereby bolstering the conclusion. It is interesting to see how a simple technique like transfer learning can be used to solve a complex task like image steganalysis. For future works, we can try to compare even more base models to leverage the benefits of transfer learning and thereby create a standard base model that can be further improved for better steganalysis.

References

1. Sumathi, C.P. & Santanam, T. & Umamaheswari, G.A Study of Various Steganographic Techniques Used for Information Hiding. *International Journal of Computer Science & Engineering Survey*.2014.https://doi.org/10.5121/ijcses.2013.4602.
2. Y. Qian, J. Dong, W. Wang, and T. Tan, Deep learning for steganalysis via convolutional neural networks.*SPIE Media Watermarking, Security, and Forensics*.2015; 9409.
3. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig.MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.*ArXiv*.2017; 1704.0486.
4. Tan, Mingxing & Le, Quoc.EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.*ArXiv*. 2019; 1905.11946v5.
5. K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition.*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas,*

NV.2016.https://doi.org/10.1109/CVPR.2016.90.
6. Simonyan, Karen & Zisserman, Andrew.Very Deep Convolutional Networks for Large-Scale Image Recognition.*arXiv.2014*; 1409.1556.
7. F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions.*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*.2017.https://doi.org/10.1109/CVPR.2017.195.
8. S. Ozcan and A. F. Mustacoglu.Transfer Learning Effects on Image Steganalysis with Pre-Trained Deep Residual Neural Network Model.*2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA.2018.https://doi.org/10.1109/BigData.2018.8622437.
9. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen.MobileNetV2: Inverted Residuals and Linear Bottlenecks.*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT*.2018.https://doi.org/10.1109/CVPR.2018.00474.
10. Rémi Cogranne, Quentin Giboulot, and Patrick Bas.*The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec'19). Association for Computing Machinery. New York, NY, USA*.2019.https://doi.org/10.1145/3335203.3335726
11. P. Bas, T. Filler, and T. Pevny.*'Break Our Steganographic System': The Ins and Outs of Organizing BOSS," in Proceedings of the 13th International Conference on Information Hiding, IH'2011, Prague, Czech Republic.*2011; 6958, 59–70.
12. Songtao, Wu & Liu, Yan & Liu, Mengyuan.*CIS-Net: A Novel CNN Model for Spatial Image Steganalysis via Cover Image Suppression*.2019.
13. Y. Bengio, J. Louradour, R. Collobert, and J. Weston.*Curriculum learning.International Conference on Machine Learning*.2009.
14. 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), Calgary, AB*.2018.https://doi.org/10.1109/ICASSP.2018.8461438.
15. Jian Ye, Jiangqun Ni, and Y. Yi.Deep learning hierarchical representations for image steganalysis.*IEEE Transactions on Information Forensics and Security*.2017; 12, 2545–2557.
16. G. Xu, H. Z. Wu, and Y. Q. Shi.Structural Design of Convolutional Neural Networks for Steganalysis.*IEEE Signal Processing Letters.*2016; 23708–712.
17. J. Kodovsky, J. Fridrich, and V. Holu.Ensemble Classifiers for Steganalysis of Digital Media.*IEEE Transactions on Information Forensics and Security*.2012; 7, 432–444.
18. J. Fridrich and J. Kodovsk.Rich Models for Steganalysis of Digital of Images.*IEEE Transactions on Information Forensics and Security.2012;* 7, 868–882.
19. G. Xu, H. Wu and Y. Sh.Structural Design of Convolutional Neural Networks for Steganalysis.*IEEE Signal Processing Letters*.2016.https://doi.org/10.1109/LSP.2016.2548421.
20. R. Zhang, F. Zhu, J. Liu and G. Liu.Depth-Wise Separable Convolutions and Multi-Level Pooling for an Efficient Spatial CNN-Based Steganalysis.*IEEE Transactions on Information Forensics and Security*.2020.https://doi.org/10.1109/TIFS.2019.2936913.