Ensemble Deep Learning Features for Real-World Image Steganalysis

Ziling Zhou[1], Shunquan Tan[1*], Jishen Zeng[2], Han Chen[2], Shaobin Hong[2]
[1]College of Computer Science and Software Engineering, Shenzhen University
[e-mail: tansq@szu.edu.cn]
[2] College of Information Engineering, Shenzhen University
*Corresponding author: Shunquan Tan

Received June 26, 2020; revised October 1, 2020; accepted November 1, 2020; published November 30, 2020

Abstract

The Alaska competition provides an opportunity to study the practical problems of real-world steganalysis. Participants are required to solve steganalysis involving various embedding schemes, inconsistency JPEG Quality Factor and various processing pipelines. In this paper, we propose a method to ensemble multiple deep learning steganalyzers. We select SRNet and RESDET as our base models. Then we design a three-layers model ensemble network to fuse these base models and output the final prediction. By separating the three colors channels for base model training and feature replacement strategy instead of simply merging features, the performance of the model ensemble is greatly improved. The proposed method won second place in the Alaska 1 competition in the end.

Keywords: Steganalysis, Deep learning, Color JPEG images, Feature fusion, Ensemble model

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (2019B010139003), NSFC (61772349,U19B2022, U1636202, 61872244), Guangdong Basic and Applied Basic Research Foundation (2019B151502001), and Shenzhen R&D Program (GJHZ20180928155814437, JCYJ20180305124325555).

http://doi.org/10.3837/tiis.2020.11.017 **ISSN : 1976-7277**

1. Introduction

In recent years, researchers have achieved brilliant results in image steganalysis using homologous datasets. With the development of deep learning technology, researchers have proposed many more steganalyzers by deep convolutional neural networks (CNNs) [1][2][3][4][5]. Their performance is even better than traditional rich model feature sets [6][7][8], but most of them are evaluated on single-source datasets. In practice, images usually come from a variety of sources and have different processing histories. Most of the existing steganalysis algorithms do not consider mixed datasets from different sources that are close to the scenario of the real world.

The Alaska competition[1] provides a good opportunity to bring steganalysis into the scenario of the real world. It provides a multi-sources RAW image dataset. In the competition, participants are required to solve various embedding schemes, inconsistency JPEG Quality Factor (QF) and various processing pipelines that appear in actual steganalysis.

We propose a method to ensemble various deep learning steganalysis. We select XuNet [3] and RESDET [4] as basic deep learning steganalysis models. We separate the three color channels (Y, Cb, and Cr) of color JPEG images to train the deep learning base models, respectively. Then the output features of all base models are merged to train a model ensemble network. The model ensemble network greatly improves the steganalysis performance. Meanwhile, we generate several fixed-QF compressed datasets with high-frequency QFs. By replacing the features of base models trained on the original mixed QF Alaska dataset by the ones of base models trained on the fixed-QF datasets, the final performance can be further improved. The final result of the proposed method won 2nd place in the Alaska 1 competition.

In this summary, we first introduce some related works and the proposed method. Then we describe experiments in Alaska 1 and Alaska 2 datasets with our proposed method, alone with the result and our analysis. The final part is the concluding remarks and conclusions.

2. Related Work

There are several deep convolutional neural networks such as XuNet, RESDET and SRNet that have a good performance on single-source JPEG datasets. It takes a lot of time to train a good SRNet, so we don't use SRNet in the Alaska 1 competition and only use SRNet in the Alaska 2 experiment. We choose XuNet and RESDET as base models in the Alaska 1 competition. In this section, the structure features of XuNet and RESDET are first explained. Since the feature fusion method is important in the model ensemble stage, this section also describes common feature fusion strategies.

2.1 The Structure Features of XuNet

XuNet is a 20-layers CNN. First of all, it uses an undecimated DCT of size 4x4 to project every single input to 16 different frequency bands. The DCT kernels are fixed in training. The output from the DCT layer is passed to 20 convolutional layers and a global average pooling layer. This part of the model learns an optimized function to transform each of the pre-processed inputs to a 384-D feature vector for classification. Every convolutional layer is followed by a Batch-Normalization (BN) to reduce the internal covariant shift [9]. The non-linear activation function is the most widely used Rectified Linear Unit (ReLU). The convolutional kernels have a unified size of 3x3. In CNN, pooling is achieved by a

[1] https://alaska.utt.fr/

convolutional layer with a stride of 2. After going through the pooling layer, the spatial sizes of data are cut by half and the number of channels doubles.

2.2 The Structure Features of RESDET

RESDET contains four parts: a DCT layer, three RBLOCK (resnet-like blocks), three DBLOCK (densenet-like blocks), and a global average pooling layer. The input of RESDET is filtered and truncated images with a shape of (511,511,16). Then we use 16 DCT basis patterns to convolve the decompressed image.

RBLOCK has two branches: a body branch and a shortcut branch. The body branch contains two convolutional layers, whose kernel size is 3x3. Batch-Normalization (BN) follows both convolutional layers with ReLU as the non-linear activation function. The shortcut path has a convolutional layer to resize the input x to a different dimension to match that of the body path. The stride of the convolutional layer is set to be 2. The operation of "add" is used to connect two branches. Through each RBLOCK, the number of the output feature maps increases by 12.

DBLOCK is composed of one convolutional layer, a BN layer and a ReLU layer. Each convolution kernel size is 3x3. The number of output feature maps is fixed to 12. The learned feature maps are concatenated with the input feature maps, like the dense connection in DenseNet [10] does. After the final DBLOCK, we use a global average-pooling to calculate the spatial average of each feature map.

2.3 Feature Fusion Strategies

After training the base models and using them to extract image features, we fuse features and train the ensemble network. Usually, there are two strategies of feature fusion based on two methods of feature combination [11]:

1. Serial feature fusion: The serial feature fusion is a process of feature extraction based on the serial feature combination method, and the resulting feature is called a serial fused feature.

2.Parallel feature fusion: The parallel feature fusion is a process of feature extraction based on the parallel feature combination method, and the resulting feature is called parallel fused feature.

3. Proposed Method

The proposed framework is illustrated in **Fig. 1**. The entire framework consists of two stages. The first stage is to train different base models. The second stage is to perform ensemble on these trained base models. In this part, by concatenating each base model feature, three full connection layers are applied for further training to obtain the final prediction.

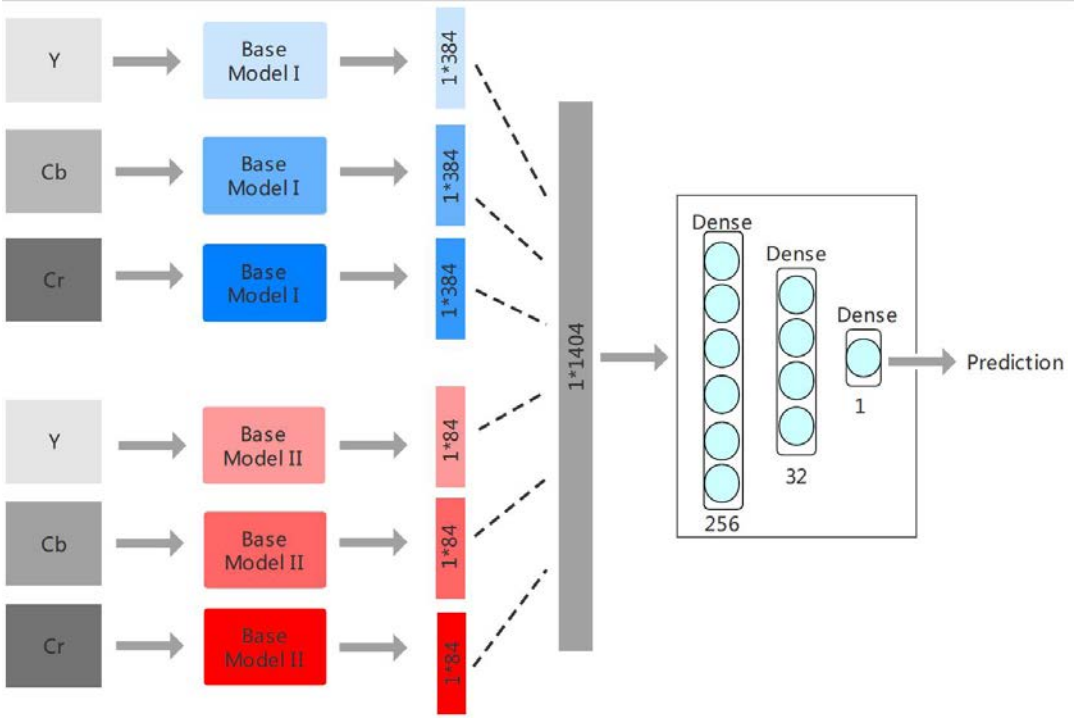

Fig. 1. The proposed framework architecture. The three channels (Y, Cb, and Cr) of a color JPEG image are separately trained using two base models. Here we only use two competitive base models, the Base Model I is XuNet [3] and the Base Model II is RESDET [4].

3.1 Base Model Training Stage

We select the XuNet and the RESDET as two base models. These two deep learning steganalysis models have competitive performance in JPEG domain steganalysis. Since the target object is color JPEG images, the three channels (Y, Cb, and Cr) are separately trained by each base model. Just like existing JPEG steganalysis methods, JPEG images are decompressed to the spatial domain before feeding into the network. For XuNet, we apply a 20-layers convolution with a bottleneck structure. For RESDET, we apply a 12-layers RESDET with a bottleneck structure and dense connection. Same as XuNet [3] and RESDET [4], images are also convolved by sixteen 4 × 4 DCT basis patterns of the first layer of the network, to help the CNN architecture focus on steganography artifacts rather than image contents for faster training convergence. The sixteen DCT basis patterns are defined as $B^{(k,l)} = \left(B_{mn}^{(k,l)}\right), 0 \le k, l \le 4, 0 \le m, n \le 4;$

$$B_{mn}^{(k,l)} = \frac{\omega_k \omega_l}{4} cos\frac{\pi k(2m+1)}{8} cos\frac{\pi l(2n+1)}{8}, \omega_0 = \frac{1}{2}, \omega_k = 1 \; for \; k > 0. \quad (1)$$

After the convolution of these sixteen 4×4 DCT basis patterns, we also apply truncation with threshold $T = 8$. Both networks receive 512 × 512 input. Considering that the Alaska dataset contains images of different sizes, which leads to inconsistencies in training and testing, we crop all the images to size of 512 × 512. As mentioned in Gibouloto et al. [12], the JPEG Quality Factor and Processing pipeline of the dataset have a major impact on the performance of JPEG steganalysis. The inconsistency of the JPEG Quality Factor between the training and inference phases results in a significant drop in detection performance. The Alaska dataset's

JPEG Quality Factors range from 60 to 100. **Fig. 2** shows the QF distribution of the Alaska dataset. From **Fig. 2** we find that the QFs in {75, 80, 85, 90, 95, 98, 100} account for 75.19% of the total number of images.

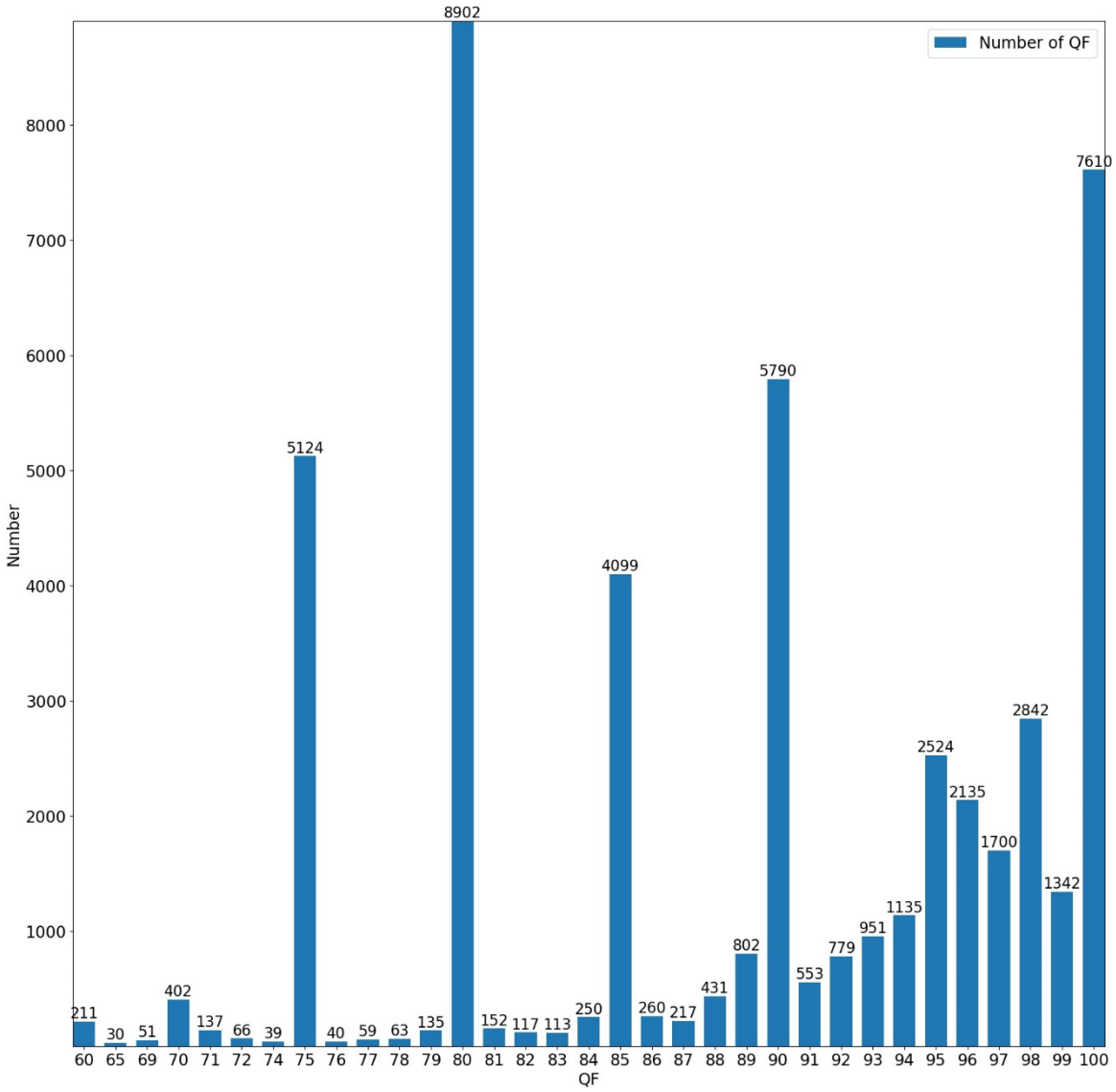

Fig. 2. The numbers of images compressed with corresponding QFs in the Alaska dataset.

Therefore, we generate two types of dataset to train the base models. The first dataset is the original Alaska dataset, which contains 49,061 cover stego pairs of mixed QF images. Here we randomly select 39,248 for training and the remaining 9813 as the validation set. The second type of datasets contains seven datasets where the images in the same dataset have the same QF. The seven QFs are {75, 80, 85, 90, 95, 98, 100}. In each dataset, all the images are obtained by re-developing the original raw with the scripts from the competition website2. In addition to the QF settings, other parameters such as random image size, preprocessing

[2] https://alaska.utt.fr/material

methods, and embedding methods remain the same. For each base model, there will be 3 CNNs trained on the original Alaska data set and 7 × 3 CNNs trained on the seven fixed-QF data sets, corresponding to three color channels.

3.2 Model Ensemble Stage

The above base models act as feature extractors in the Model Ensemble Stage. After training the above base models, we introduce 3 feature selection methods. **No replacement** means that all the features are obtained from base models trained on the original Alaska data set. **Replace** means that for images with QFs in 75, 80, 85, 90, 95, 98, and 100, the features obtained from the base models trained by the original Alaska data set are replaced with the features obtained from the base models trained on corresponding QF data sets. **Merged all feature** means that all the trained base model features are merged and fed into the model ensemble network.

3.3 Preprocessing and Parameters Setting

For each input image, a single XuNet outputs 384-dimension feature vector, and a single RESDET outputs 84-dimension feature vector. For each image with three color channels, the total feature dimension is 384 × 3 + 84 × 3 = 1404. During training, we additionally added a random rotation of 0, 90, 180, or 270 degrees and then apply a random upper or lower mirroring to the input image inside each training epoch. The detailed parameter settings of the model ensemble network and each base model are as follows:

- **XuNet and RESDET**: Batch size is set to 50 during training. The learning rate is initialized with 0.001 and the maximum number of epochs is set to 200.
- **Model ensemble network**: Batch size is set to 128 during training. The learning rate is initialized with 0.0001 and the maximum number of epochs is set to 250.

Both base models receive 512 × 512 input. We apply Sigmoid Cross Entropy Loss as the loss function in all networks. Batch normalization is also involved in each layer. During all training phase, we apply the stochastic gradient descent with warm restarts (SGDR) [13] to adjust the learning rate. SGDR can speed up convergence and jump out of the local minimum to some extent. The learning rate $alpha$ decays as follows:

$$\alpha = \alpha_{initial} * \left(\frac{1}{2}(1-\beta)\left(1+\cos\left(\frac{T_{current}\pi}{T_1}\right)\right)+\beta\right) \quad (2)$$

where $\alpha_{initial}$ is the initial learning rate. T_1 is the initial decay step. $T_{current}$ records the current decay process. We set $\beta = 0$, which is the minimum learning rate value as a fraction of the learning rate. In all experiments we set the first decay step as the ratio of the total number of training images to the batch size: $T_1 = \frac{\text{number of training image}}{\text{batch size}}$. $T_{current}$ is initialized to 0 and increased by one every time the variables have been updated during iteration.

4. Large-scale Experiments on Alaska v1

4.1 Experiment Result and Analysis

This part of the experiments is implemented based on Tensorflow (https://github.com/tensorflow). The GPU used in all experiments is NVIDIA Tesla P100. The training set of the processed Alaska dataset contains a total of 49,061 JPEG images. We randomly select 39,248 images as the experimental training set, and the remaining 9813 as the validation set, which gives a ratio of 4:1. The target steganography algorithm includes J-UNIWARD [14], UED [15], EBS [16] and ns-F5[17]. The left-top 512 × 512 regions of all

the images are cropped to fit the input size of the network.

Different feature selection method has a different impact on the final result. **Table 1** shows the comparison of different ways to select features for the model ensemble network.

Table 1. Comparison of feature selection.

Method	No replacement	Replace	Merged all feature
Feature Dimensions (XuNet and RESDET)	(384+84)x3=1404	(384+84)x3=1404	1404x8=11232
False alarm	18.50%	16.73%	22.38%
Miss Detection	27.49%	23.07%	35.11%
Accuracy	77.00%	80.10%	71.26%

The results show that the Replace strategy achieves a larger improvement from 77% validation accuracy to 80.1%. Simply merging features results in performance degradation. One of the possible reasons is that the base models trained with the fixed-QF dataset have improved the performance on corresponding QF images but have a greater performance degradation on other QFs. Therefore, we apply Replace to select the features entering the model ensemble network.

Since the magnitudes of these features obtained by different classifiers vary, we adopt linear normalization which maps feature values to [0,1]. Then all the features are concatenated into one feature vector as the input of the model ensemble network. As shown in **Fig. 1**, the model ensemble network consists of three fully connected layers. The successive layers contain 256, 32, and 1 neuron, respectively. The predicted value of the target image is output by a Sigmoid function which labels 1 for stego images and 0 for cover images. The loss function is Sigmoid Cross Entropy Loss. The final prediction for each image is obtained by the model ensemble network. Each image only gets one final prediction value. The final ranking is obtained by sorting these predictions in descending order. The more likely an image is stego, the higher it ranks.

Table 2. Performance comparison of models with color channels and model ensemble.

Color channel	Y channel		Cb channel		Cr channel		
Model	XuNet	RESDET	XuNet	RESDET	XuNet	RESDET	Model ensemble
False Alarm	28.51%	27.19%	20.31%	28.52%	20.44%	30.34%	16.72%
Miss Detection	42.53%	41.44%	36.30%	28.55%	38.27%	30.34%	23.07%
Accuracy	64.47%	65.67%	71.69%	71.46%	70.64%	70.84%	80.01%
Local MD005	66.5%	69.6%	56.9%	58.2%	58.2%	56.4%	37.4%

Table 3. Comparison of image cropping to 512 × 512 in different ways.

Method	Left-top crop	Central crop	Random crop	L1 crop
Validation accuracy	71.69%	71.2%	69.46%	70.40%
Local MD005	56.71%	57.10%	59.01%	57.60%

In **Table 2**, we compare the validation performance of each color channel on both base models. XuNet and RESDET have similar performance on each color channel. Among the three color channels, the Cb channel performs the best. The performance of the Cb channel is close to the Cr channel, since their intrinsic properties are similar. In all models, the miss detection rate is higher than the false alarm rate, which indicates that these models are more error-prone to the stego image. After the model ensemble, the False Alarm Rate reaches a minimum of 16.72%, while the Miss Detection Rate reaches a minimum of 23.07%. This shows that the model ensemble can effectively reduce the validation False Alarm Rate and Miss Detection Rate.

Table 4. Comparison between early fusion and separating the color band on XuNet.

Color channel	Y channel	Cb channel	Cr channel	Early fusion
False alarm	28.51%	20.31%	20.44%	24.17%
Miss detection	42.53%	36.3%	38.27%	40.75%
Accuracy	64.47%	71.69%	70.64%	67.54%
Local MD005	66.5%	56.9%	58.2%	67.30%

Table 5. Final submission result

Submit ID	MD005(Ranking)	minPE	FP50
Yyousfi1	24.37(1st)	14.26	0.62
2016130231	50(2nd)	25.03	5.18

Table 6. Training epoch numbers when getting the best score

Training epochs when getting the best score			
Model	XuNet	RESDET	Model ensemble
Epoch Number	62	62	206

Since the Alaska dataset contains images of different sizes, we crop all the images in the same size to avoid size mismatch. Table 3 shows the comparison of different ways to crop images to 512 × 512. In Table 3, Left-top/Central crop indicates that all images have been left-top/central cropped with size 512 × 512. Random crop indicates that the image is randomly cropped with size 512 × 512 under the block structure satisfying 8 × 8. The L1 crop from Tsang and Fridrich [18] is achieved by cropping the image with size 512x512 with the closest local variance histogram to the whole image local variance histogram in the L1 norm sense. The results show that various cropping methods achieve similar validation accuracy. The Left-top crop achieves the highest validation accuracy, so we finally decide to apply it to all images.

One of the main reasons that we train CNN separately with each color band is that there are no channel-related embedding schemes in the Alaska competition. All the embedding algorithms independently embed the information in each color channel. Each of these channels is treated as a separate image for embedding. Table 4 show the comparison between early fusion and separating the color band. The results show that the performance of early fusion (which inputs 3 color channels to the same network) has declined a lot. The validation accuracy of early fusion drops by nearly 3% compared to the Cb and Cr channels. For the efficiency reason, we choose to separate the color channels and use one CNN per channel.

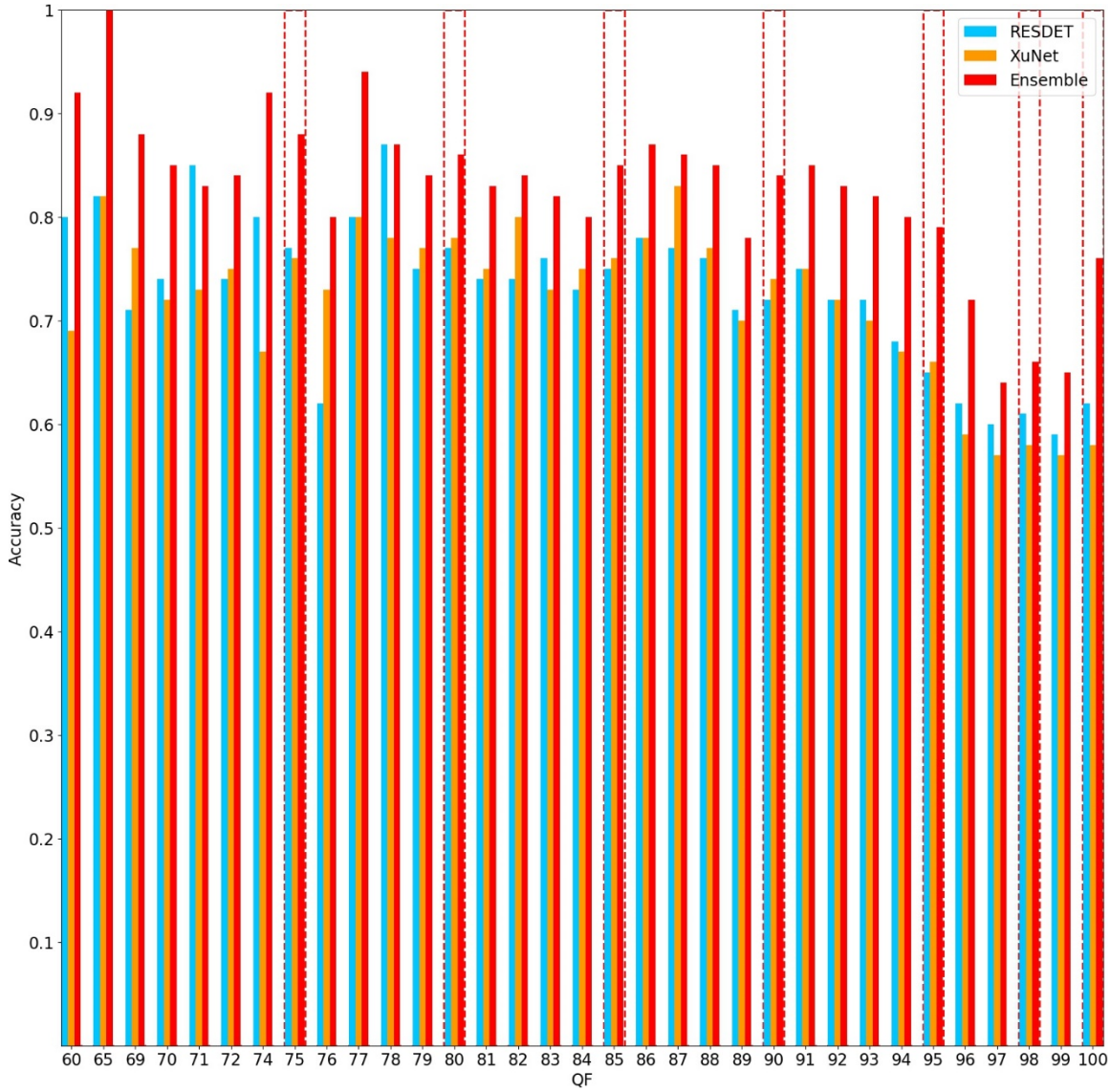

Fig. 3. Comparison of validation accuracy of the two base models and model ensemble on each JPEG Quality Factor (QF). The red bin is the performance of the model ensemble, the blue bin is the performance of XuNet, and the orange bin is the performance of RESDET. The red dashed box indicates that the features are obtained from base models trained on fixed-QF datasets.

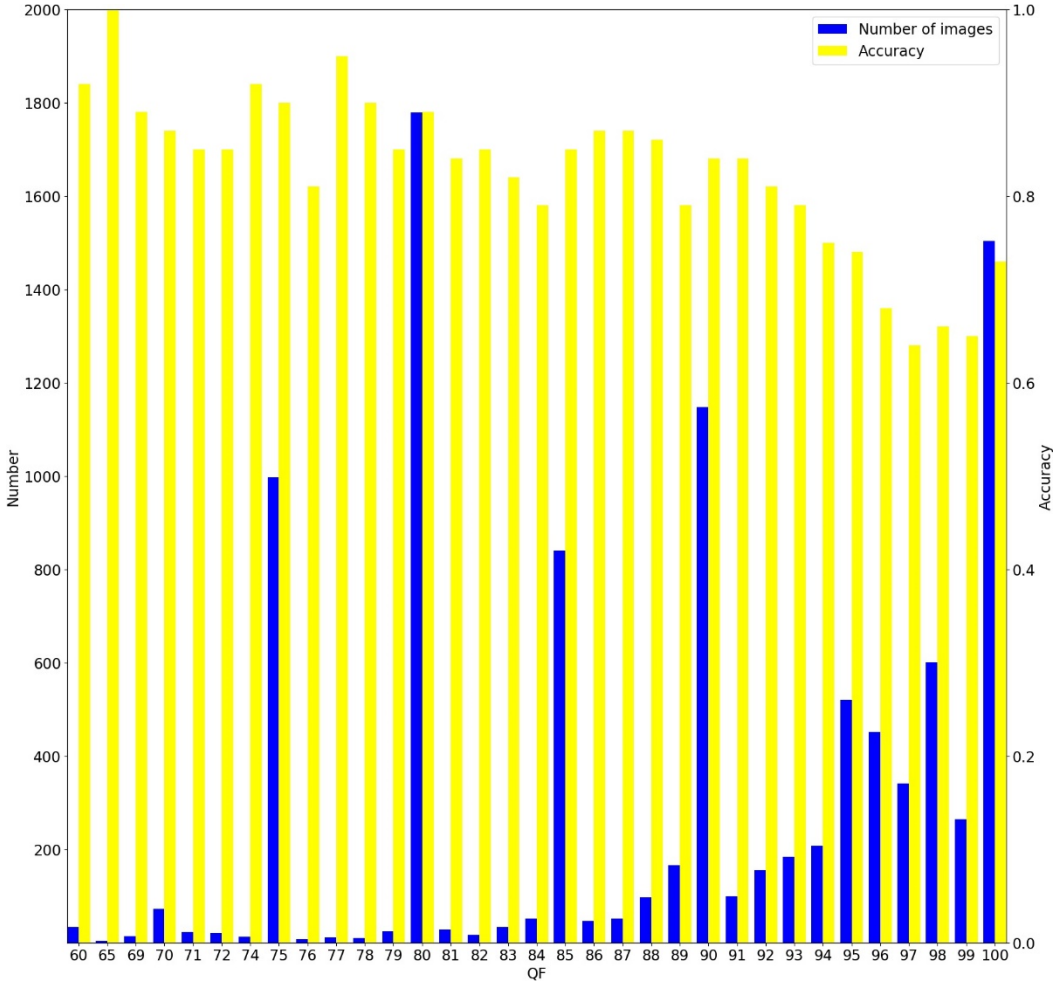

Fig. 4. The validation accuracy of the model ensemble of each QF. The yellow bin is the validation accuracy, and the blue bin is the number of validation images corresponding to each QF.

Fig. 3 shows the validation accuracy comparison between base models and model ensemble on different QFs. The performance of the two base models is relatively close. On most QF values, the accuracy of RESDET is slightly higher than that of XuNet. After adopting the model ensemble, the performance of each QF has been greatly improved. It shows that the model ensemble can effectively improve the overall classification ability. In the case of training on the fixed-QF dataset, the model ensemble has also achieved a relatively high improvement.

Fig. 4 shows the validation result of the model ensemble on each JPEG Quality Factor. As the JPEG Quality Factor declines, the accuracy of the validation set increases. Note that the accuracy of model ensemble on images with QFs of {75, 80, 85, 90, 95, 98, 100} is {90%, 89%, 85%, 84%, 74%, 66%, 73%}, respectively.

Table 5 is the final submission result of the model ensemble in the Alaska 1 competition and **Table 6** shows the training epoch numbers when getting the best score. The testing dataset of the Alaska competition consists of 5000 JPEG images. After the XuNet iterates for 62 epochs, RESDET iterates for 62 epochs, and the model ensemble network iterates for 206 epochs, the model ensemble network achieves our best submission result. The final model

ensemble result reaches an MD005(Missed detection for False alarm of 0.05) of 50% and a minPE of 14.26%. The final submission of our proposed method achieves 2nd place in the Alaska competition. The first place in the competition finds a better solution which reaches an MD005 of 24.37%.

4.2 Further Discussions

Comparing to the 1st place in the competition, we still have a lot of room for improvement. We also experiment on SRNet with the Cb channel. The input size is set to 512 × 512. According to the GPU memory limit, we set the batch size to 10 in training. We set the maximum iterations to 500,000. SRNet takes much more time to converge than XuNet. The final accuracy of the local validation set is 69.73%. Compared to XuNet's 71.69% validation accuracy, the accuracy of SRNet is slightly lower. The possible reason for this result is that we have reduced the batch size so that the SRNet does not fully converge within 500,000 iterations. It takes us one week to reach the maximum iterations of SRNet. Therefore, we did not apply SRNet as our base model in the competition considering the huge time consumption. If we do have had more time and more computational power, SRNet seems to be a better choice as a base model.

Other parameters including image size and embedding schemes also play a key role in steganalysis. As we all know that steganalyzers usually require a fixed input image size. Initially we tried to use Global Average Pooling at the fully connected layer in XuNet, which forces the final feature maps to size 1 × 1, but the validation accuracy drops to 50%. Another way to solve this problem is to train deep learning models for each size, but it is time consuming. As for embedding schemes, the Alaska dataset randomly selects one of the four embedding schemes to embed the image with different payloads. Therefore, it is difficult for us to know which embedding schemes and payloads are used for the testing samples. We conduct a simple mismatch experiment where we train the XuNet on Cb channels on mixed embedding schemes dataset, while test on only JUNIWARD but with randomly payloads embedded dataset. The validation accuracy drops to 65.83%, lower than the case without mismatching. Considering the mismatch problems, if we do have more time, we would provide more matching dataset and train the base models in the matching situation, then ensemble those base models in the same way. The proposed model ensemble method can also satisfy with such a scenario.

5. Large-scale Experiments on Alaska v2

5.1 Parameters Setting and Data Preprocessing

We use the Alaska 2 competition datasets[3] to validate our ensemble multiple deep learnig steganalyzers. We choose 30000 JPEG images of Alaska 2 datasets as cover with QFs of 75, 90, and 95 (10000 each). The stego is generated through information embedding by J-UNIWARD with embedding rates of 0.4 bpnzAC. The embed payload is 0.4 bits per non-zero AC DCT coefficient. The image size is 512x512. Thus, we get 60000 JPEG images where cover and stego both have 30000. The datasets are randomly split into training (36000), validation (18000), and test sets (6000).

The base model also uses XuNet and RESDET as above and adds SRNet to improve the ensemble performance. When we use different channel data to train base models, we find that

[3] https://www.kaggle.com/c/alaska2-image-steganalysis/

the DCT layer and the difference in JPEG read format have a great impact on the final training result. To test the ensemble performance with more channel data, we add CrCb, and YCrCb features to train the ensemble model.

The RESDET Y and YCrCb channel models need DCT layers and the JPEG toolbox to decompress images. The RESDET Cr, Cb, and CrCb channel models need DCT layers, but the input is directly read by OpenCV and round. Otherwise, the training process is hard to converge.

The XuNet Y and YCrCb channel models need DCT layers and the JPEG toolbox to decompress images. The RESDET Cr and CrCb channel models do not need the DCT layer, but the input is directly read by OpenCV and round. The RESDET Cb channel model does not need the DCT layer, but the input is directly read by OpenCV and not round.

The SRNet Y channel model needs the DCT layer and the JPEG toolbox to decompress images. The SRNet Cr channel model also needs the DCT layer and the input is directly read by OpenCV and round. The SRNet Cb, CrCb, and YCrCb channel models need DCT layers, but the input is directly read by OpenCV. **Table 6** shows the detail of the training preprocesses of different channel models.

Table 6. The training preprocesses of different channel models

Channel	Y	Cr	Cb	CrCb	YCrCb
RESDET	DCT, JPEG decompress	DCT, OpenCV read and round	DCT, OpenCV read and round	DCT, OpenCV read and round	DCT, JPEG decompress
XuNet	DCT, JPEG decompress	No DCT, OpenCV read and round	No DCT, OpenCV read	No DCT, OpenCV read and round	DCT, JPEG decompress
SRNet	DCT, JPEG decompress	DCT, OpenCV read and round	DCT, OpenCV read	DCT, OpenCV read	DCT, OpenCV read

5.2 Experiment Result and Analysis

We get the performance comparison of each model with each color channel and present the validation and test accuracy in the **Table 7** and **Table 8** respectively. From the two tables, we can see RESDET has the best performance in Y, Cr, and Cb channels. XuNet performs the best in the CrCb channel and SRNet performs the best in the YCrCb channel. Among the three color channels, the Y channels have the best validation accuracy and test accuracy. The Cb channel performance seems worse. The Y channel RESDET model has the highest validation accuracy 70.58% and test accuracy 70.4%.

Table 7. Validation accuracy of base models and color channels

Channel	Y	Cr	Cb	CrCb	YCrCb
RESDET	**70.58%**	63.32%	57.52%	56.72%	58.83%
XuNet	68.2%	63.2%	56.37%	67.49%	56.03%
SRNet	69.56%	59.31%	55.06%	59.5%	62.13%

Table 8. Test accuracy of base models and color channels

Channel	Y	Cr	Cb	CrCb	YCrCb
RESDET	**70.4%**	63.5%	57.58%	56.5%	58.9%
XuNet	68.3%	63.1%	56.7%	67%	56%
SRNet	69.3%	58.3%	54%	59.37%	60.73%

In the model ensemble stage, we find that using the model antepenultimate output tensor's mean, variance, maximum value, and minimum value across high and width axes as features is better than only using the model global average pooling layer output tensor. The forward method which has four groups can get feature dimension 84*8 (RESDET), 384*8 (XuNet) and 512*8 (SRNet). While the latter method which only has one group can get feature dimension 84*1 (RESDET), 384*1 (XuNet) and 512*1 (SRNet). **Table 9** shows that the four-group-feature model ensemble has a good performance than the one-group-feature model. Here, we use the serial without weights feature fusion method to combine the features.

Table 9. Different feature ensemble models' validation and test accuracy

Features	Validation accuracy	Test accuracy
One group	76.54%	76.83%
Four group	77.37%	78.17%

Then, we fuse the three-channel features to train the ensemble model. Because Alaska 2 competition only provides JPEG images, we can't produce identical QF datasets to use the replace feature selection method.

In this part, we use the no replacement feature selection method. We test serial feature fusion and parallel feature fusion with and without weights. Specifically, we experiment with six feature fusion methods as shown in Table 10, along with their validation and test accuracy. The weights for RESDET, XuNet, and SRNet are 84/980, 384/980, and 512/980 respectively.

Table 10. Different feature fusion methods ensemble model validation and test accuracy

Feature fusion	Dimensions	Validation accuracy	Test accuracy
Serial w/o weights	(84+384+512)x24	77.37%	**78.17%**
Serial w/ weights	(84+384+512)x24	**77.51%**	77.9%
Parallel w/o weights	(84+384+512)x8	77.46%	77.77%
Parallel w/ weights	(84+384+512)x8	77.47%	77.88%
0 padding w/ weights	512x8	77.46%	77.97%
0 padding w/o weights	512x8	77.48%	78.02%

From **Table 10**, we can see that all feature fusion methods have close validation accuracy. The serial fusion with weights gets the best validation accuracy of 77.51%. The serial fusion without weights has the best test accuracy of 78.17%. Comparing with the best channel test accuracy in **Table 8**---the Y channel's 70.4%--- the model ensemble improves the test accuracy for about 8%.

In the above experiments, we only fuse channels Y, Cr, and Cb. In this part, we try to fuse Y, Cr, Cb, CrCb, and YcrCb with serial fusion without weights. There are three feature selection plans: Y, Cr, Cb, and CrCb; Y, Cr, Cb, and YCrCb; Y, Cr, Cb, CrCb, and YCrCb. The experiment result is shown in **Table 11**.

Table 11. Different feature selection methods' ensemble model validation and test accuracy

Features selection	Validation accuracy	Test accuracy
Y,Cr, and Cb	77.37%	78.17%
Y,Cr,Cb, and CrCb	**78.16%**	**78.45%**
Y,Cr,Cb, andYCrCb	77.48%	78.05%
Y,Cr,Cb,CrCb, and YCrCb	78.07%	**78.45%**

Comparing the result of **Table 10** with **Table 11**, we can see that adding CrCb channel data to fusion slightly improves the validation and test accuracy. The best validation accuracy is 78.16% and the best test accuracy is 78.45%. However, adding YCrCb channel data does not have a positive effect on the final test result. The best feature selection method is using Y, Cr, Cb, and CrCb channel data.

6. Conclusions and Future Work

In this paper, we present an ensemble of deep learning steganalyzers. We apply XuNet and RESDET as our base models and design a model ensemble network to train the base models output features. We apply the replacing strategy rather than simply merging all the features. The experimental results show that the model ensemble can effectively improve the accuracy of the base steganalyzers. The main contribution of our proposed method is listed as follows:

- The model ensemble achieves better performance than a single base model and provides a unique ranking of the test images from multiple classifiers.
- The replacing strategy can better avoid QF mismatch than simply merging all classifiers features.
- We find that separating the color band is better than early fusion in detecting no channel-related embedding schemes.

In future work, we will continue to focus on a more effective model ensemble method and find a better solution for real-world steganalysis. For example, we will try to use Siamese CNN [19] as a base model and combine different models and feature fusion strategies to improve our ensemble model. Besides, using an adversarial network [20] to enhance the security of the model is a good research direction.

References

[1] J.S. Zeng, S.Q. Tan, B. Li and J.W. Huang, "Largescale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1200-1214, May 2018. Article (CrossRef Link)

[2] M. Chen, V. Sedighi, M. Boroumand and J. Fridrich, "JPEG-phase-aware convolutional neural network for steganalysis of JPEG images," in *Proc. of the 5th ACM Workshop on Information Hiding and Multimedia Security, Philadelphia, Pennsylvania, USA*, pp.75-84, June 2017. Article (CrossRef Link)

[3] G.S. Xu, "Deep convolutional neural network to detect J-UNIWARD," in *Proc. of the 5th ACM Workshop on Information Hiding and Multimedia Security, New York, United States*, pp.67-73, June 2017. Article (CrossRef Link)
[4] X.S. Huang, S.L Wang, T.F. Sun, G.S. Liu and X. Lin, "Steganalysis of Adaptive JPEG Steganography Based on ResDet," in *Proc. of the APSIPA Annual Summit and Conference, Honolulu, HI, USA*, Nov. 12–15, 2018. Article (CrossRef Link)
[5] M. Boroumand, M. Chen and J. Fridrich," Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181-1193, Sept. 2018. Article (CrossRef Link)
[6] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol.10, no. 2, pp. 219–228, Feb. 2015. Article (CrossRef Link)
[7] V. Holub and J. Fridrich, "Phase-aware projection model for steganalysis of JPEG images," in *Proc. of the Media Watermarking, Security, and Forensics, San Francisco, California, United States*, Mar 8-12, 2015. Article (CrossRef Link)
[8] X. Song, F. Liu, C. Yang, X.Y. Luo and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," in *Proc. of the 3rd ACM workshop on information hiding and multimedia security, Portland, Oregon, USA*, pp.15-23, June 2015. Article (CrossRef Link)
[9] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. Article (CrossRef Link)
[10] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. of the CVPR, Honolulu, HI, USA*, pp.2261-2269, July 2017. Article (CrossRef Link)
[11] J. Yang, J.Y. Yang, D. Zhang and J.F. Lu, "Feature fusion: parallel strategy vs. serial strategy," *Pattern recognition*, Vol. 36, no. 6, pp. 1369-1381, June, 2003. Article (CrossRef Link)
[12] Q. Gibouloto, R. Cogranneo, and P. Bas, "Steganalysis into the wild: How to define a source?," *Electronic Imaging*, no. 7, pp. 318-1-318-12, Jan. 2018. Article (CrossRef Link)
[13] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016. Article (Web Link)
[14] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, Jan. 2014. Article (CrossRef Link)
[15] L. Guo, J.Q. Ni, W.K. Su, C.P. Tang, and Y.Q. Shi, "Using Statistical Image Model for JPEG Steganography: Uniform Embedding Revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, Aug. 2015. Article (CrossRef Link)
[16] L. Guo, J.Q. Ni, and Y.Q. Shi, "Uniform Embedding for Efficient JPEG Steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, Mar. 2014. Article (CrossRef Link)
[17] F. Jessica, T. Pevný, and J. Kodovský, "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities," in *Proc. of the 9th MM&Sec'07, New York, United States*, pp.3-14, Sept. 2007. Article (CrossRef Link)
[18] C.F. Tsang and J. Fridrich, "Steganalyzing images of arbitrary size with CNNs," *Electronic Imaging*, no. 7, pp.121-1-121-8, Jan. 2018. Article (CrossRef Link)
[19] W.K.You, H. Zhang, and X.F. Zhao, "A Siamese CNN for Image Steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 291-306, July, 2020. Article (CrossRef Link)
[20] B.C. Chen, J.X. Wang, Y.Y. Chen, Z.L. Jin, H.J. Shim and Y.Q. Shi, "High-Capacity Robust Image Steganography via Adversarial Network," *KSII Transactions on Internet & Information Systems*, vol. 14, no. 1, pp. 366-381, Jan. 2020. Article (CrossRef Link)

ZiLing Zhou is currently a postgraduate student in the College of Computer Science and Software Engineering at Shenzhen University. His research interest involves in machine learning, deep learning and Steganalysis.

Shunquan Tan (M'10–SM'17) received the B.S. degree in computational mathematics and applied software and the Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2002 and 2007, respectively. He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2005 to 2006. He is currently an Associate Professor with College of Computer Science and Software Engineering, Shenzhen University, China, which he joined in 2007. His current research interests include multimedia security, multimedia forensics, and machine learning.

Jishen Zeng (S'16) received the B.S degree of electronic information science and technology from Sun Yat-sen University, Guangzhou, China in 2015. He received the Ph.D. degree in electronic Information engineering from Shenzhen University, China in 2020. His current research interests include steganography, steganalysis, multimedia forensics, and deep learning.

Han Chen received the B.S. degree in electronic Information engineering from Shenzhen University. He is currently a master student at the Shenzhen University majoring in information and communication engineering. His current research mostly focus on multimedia forensics and deep learning.

Shaobin Hong received the B.S. degree in electronic Information engineering from Shenzhen University. His current research mostly focus on multimedia forensics, machine learning and deep learning.