Intelligent Automation
& Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.029983
Article

A Deep Learning Driven Feature Based Steganalysis Approach

Yuchen Li[1], Baohong Ling[1,2,*], Donghui Hu[1], Shuli Zheng[1] and Guoan Zhang[3]

[1]College of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230009, China
[2]College of Information Engineering, Anhui Broadcasting Movie and Television College, Hefei, 230011, China
[3]Department of Informatics, Faculty of Natural & Mathematical Sciences, King's College London, London, WC2R2LS, UK
*Corresponding Author: Baohong Ling. Email: bhling2020@amtc.edu.cn

Received: 16 March 2022; Accepted: 25 May 2022; Published: 23 June 2023

Abstract: The goal of steganalysis is to detect whether the cover carries the secret information which is embedded by steganographic algorithms. The traditional steganalysis detector is trained on the stego images created by a certain type of steganographic algorithm, whose detection performance drops rapidly when it is applied to detect another type of steganographic algorithm. This phenomenon is called as steganographic algorithm mismatch in steganalysis. To resolve this problem, we propose a deep learning driven feature-based approach. An advanced steganalysis neural network is used to extract steganographic features, different pairs of training images embedded with steganographic algorithms can obtain diverse features of each algorithm. Then a multi-classifier implemented as lightgbm is used to predict the matching algorithm. Experimental results on four types of JPEG steganographic algorithms prove that the proposed method can improve the detection accuracy in the scenario of steganographic algorithm mismatch.

Keywords: Image steganalysis; algorithm mismatch; convolutional neural network; JPEG images

1 Introduction

Steganography is a technique to hide secret information in public carriers (text, image, audio, video etc) for specific purposes [1–6], and this secret information is difficult to be identified with human eyes and technical means. Digital images [7–9], especially JPEG (Joint Photographic Experts Group) images, are one of the most widely spread media on the internet. The corresponding JPEG steganographic algorithms are developing rapidly, from traditional embedding algorithms such as Outguess [10], MB [11] and nsF5 [12] to the adaptive algorithms based on syndrome-tellis codes (STC) [13] technique, such as J-UNIWARD [14], UED [15] and UERD [16], which make steganalysis more difficult.

As the opposite of steganography, the goal of steganalysis is to detect whether the cover carries secret information. With the rapid development of machine learning (including deep learning) [17–21], steganalysis models began to be constructed using these tools. Kodovský et al. proposed the ensemble classifier implemented as random forests to build a steganalyzer with improved detection accuracy [22], Zeng et al.

proposed a hybrid deep learning framework for large-scale JPEG steganalysis [23], and it was the first time that quantization and truncation were applied to deep learning based steganalysis. More recently, Boroumand et al. proposed SRNet [24], which can automatically compute noise residuals without using high-pass filters and obtain superior performance. However, in the real environment, we do not know which steganographic algorithm is used for embedding in advance. Without this important prior knowledge, we will face the problem of steganographic algorithm mismatch, and it is difficult to train and obtain the corresponding effective steganalysis model.

To solve or alleviate the algorithm mismatch problem, Pevný et al. summarized some approaches for the construction of universal steganalysis [25], the one-against-all classifier was trained on the image set which contains the stego images created by a variety of known (existing) steganographic algorithms, the one-class classifier used anomaly detection to identify stego images and the approach of multi-classifier was trained by a group of binary classifiers. For example, to construct a support vector machine (SVM) classifier for recognizing which steganographic algorithm was used [26–28], Pevný et al. used different handcrafted features such as calibrated DCT (Discrete Cosine Transform) features [29] and Markov features [30]. However, with the growth of feature dimensions, SVM is no longer the most appropriate choice. At the same time, deep neural network shows the excellent capabilities of feature extraction, and this end-to-end method does not need handcrafted features anymore.

Based on transfer learning, Kong et al. designed an iterative multi-order feature alignment (IMFA) algorithm to reduce the maximum mean discrepancy of feature distributions between training and testing sets [31]. Feng et al. presented a contribution-based feature transfer (CFT) algorithm to learn two transformations to transfer learning set features by evaluating both the sample feature and dimensional feature [32]. In this paper, we propose a different approach to relieve the impacts of steganographic algorithm mismatch. We first use the deep convolutional neural network as a feature extractor to obtain more representative steganographic features from different dimensions, then train a multi-classifier for different steganographic features, and finally use the steganalysis model based on the matched steganographic algorithm to detect the suspicious images. We conduct experiments on several classical JPEG steganographic algorithms which contain both traditional and adaptive algorithms. The experimental results show that the proposed method significantly improves the detection accuracy of steganalysis in the case of steganographic algorithm mismatch.

Section 2 describes the details of the proposed method. Section 3 presents the experimental configuration and results, and Section 4 concludes this paper.

2 Proposed Approach

In this section, we introduce our proposed approach in detail. First, we construct a steganographic library consisting of different types of stego and cover images, then we use convolutional neural networks trained by different pairs of training images to extract diverse features automatically. Based on the feature library, we train a multi-classifier for different steganographic algorithms. Finally, we estimate the most suspicious steganography method used in the testing images and detect the testing images by the matched steganalysis model. Fig. 1 shows the framework of the proposed blind steganalysis method.

2.1 Feature Extraction

Here, the symbol N stands for the number of steganographic algorithms in this paper, S denotes the kind of steganography algorithms indexed by n $\in \{0, 1, \cdots, N\}$, and S_0 means the cover class. $\begin{pmatrix} N+1 \\ 2 \end{pmatrix}$ different feature extractors are trained by each pair of classes in S as Fig. 2 shown. For example, S_0 training set can be trained with $S_i, i \in \{1, 2, \cdots, N\}$. Different groups of feature extractors are used to extract the features from the training set and testing set. We use the feature extractors trained by the same class of steganographic algorithms. The features brought by these N feature extractors are used to represent a training image which means that we have N times data enhancement in terms of quantity.

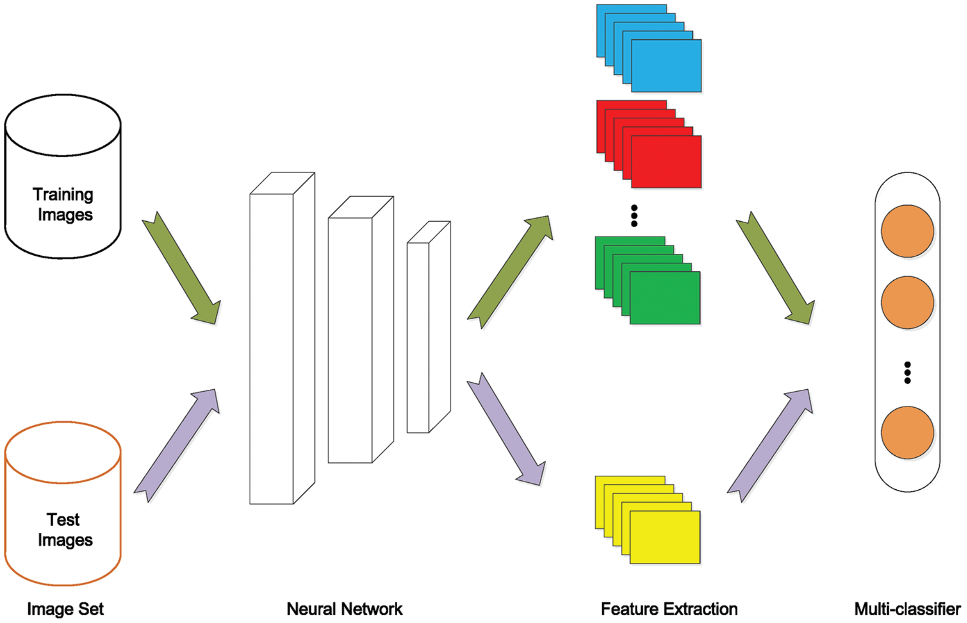

Figure 1: The framework of the proposed deep learning driven feature-based steganalysis method

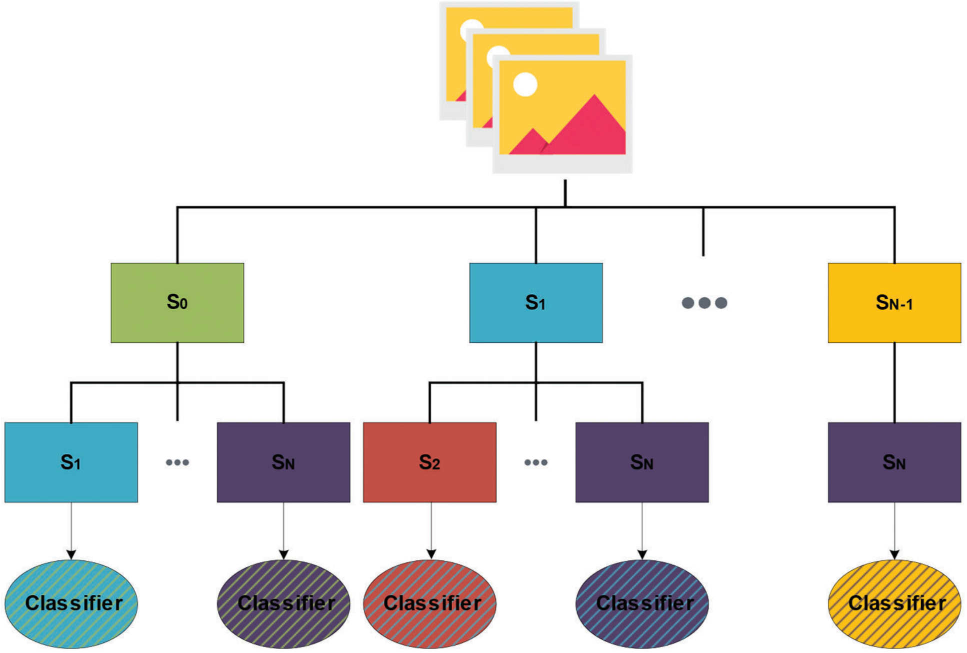

Figure 2: Different pairs of stego images to train classifiers

As for the testing set, all $\binom{N+1}{2}$ feature extractors are employed because we do not know which steganographic algorithm is used. For each image, we can obtain $\binom{N+1}{2}$ sets of different features. By using the multi-classifier, we can obtain prediction results with the same number of feature groups.

2.2 Feature Combination

Due to its excellent detection ability, we use SRNet [24] as our feature extractor and the structure of feature extractor is shown in Fig. 3. We intercept the feature map of the twelfth layer of SRNet as steganography features. To obtain more diverse image feature representations, we use different reduction strategies, including *max*, *min* and *mean* as shown in Fig. 4. Combining these sub-features can produce new features such as *max_mean*, *min_mean*, *max_min* and *max_min_mean*. The combination is done by directly concatenating different types of features. We conduct experiments presented in Section 3 to choose the optimal sub-feature combination.

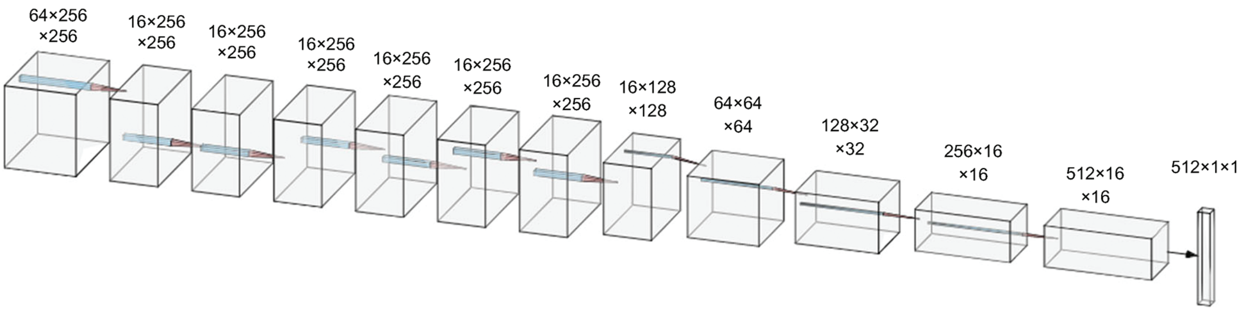

Figure 3: The structure of feature extractor

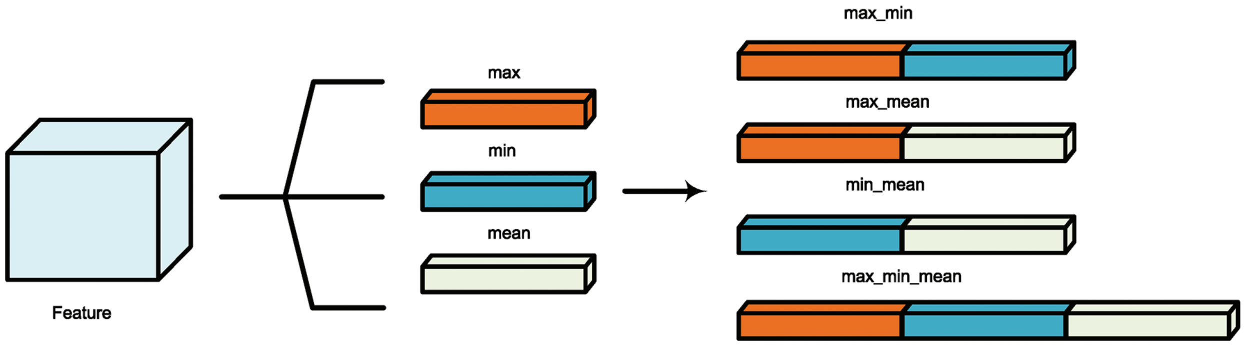

Figure 4: Different reduction strategies of feature map

2.3 Feature Matching

We train a multi-classifier fed by N stego image sets and one cover image set. In this paper, we choose lightgbm [33] as our multi-classifier because of its excellent learning and generalization capability. For each testing image, we have $\binom{N+1}{2}$ different feature extractors to obtain 512-dimensional feature respectively. Based on these features, we obtain $\binom{N+1}{2}$ prediction results by the multi-classifier. Then, we adopt the strategy of max-wins which means the max number of votes is the final prediction result. It is worth noting that if we obtain the same number of the max votes, we make a random choice. The whole matching process is described in Fig. 5, where the table T records the number of votes for each steganographic algorithm, and *Algo* records the best matched steganographic algorithm used in those test images. The ratio of matching can be described as follows:

$$Ratio_i = \frac{T_i}{\sum_{i=1}^{N} T_i} \tag{1}$$

where T_i represents the number of votes for the steganographic algorithm, N represents the total number of steganographic algorithms. The corresponding algorithm with the highest *Ratio* score is the best matched steganographic algorithm.

Algorithm 1 The process of obtaining the best matched steganographic algorithm

Input: The test images $Test$; the multi-classfier C; the total number of steganographic algorithms N.
Output: Prediction table T; the best matched steganographic algorithms $Algo$.

```
$T[1 \cdots N] = 0$, $K = \binom{N+1}{2}$;
for $i = 0$; $i < length(Test)$; $i++$ do
    $Vote[1 \cdots N+1] = 0$;
    Extracts $K$ different steganographic features of $Test_i$, which are denoted as $F_{ik}$, $k = 1, 2, \cdots, K$;
    for $j = 0$; $j < K$; $j++$ do
        Uses $C$ to predict the class of the test image features $F_{ij}$, and the prediction result is denoted as $n$;
        $Vote[n] += 1$;
    end for
    Obtains the index $n$ of highest voting score;
    $T[n] += 1$;
end for
Obtains the $Algo$ by maximum value of the $T$.
```

Figure 5: Algorithm to obtain the best matched steganographic algorithm

Finally, we use the model of the best matched steganographic algorithm to detect test images. We can achieve a relatively high accuracy as without algorithm mismatching if the matching result is correct.

3 Experiment

3.1 Experimental Setup

In the experiments, the Graphics card we use is RTX2080 TI with 11 G memory. The raw dataset is BOSSbase 1.01 [34], which has a total of 10000 grayscale images. These images are divided into three parts, 60% as a training set and the rest are equally used as a validation set and a testing set, respectively.

The images are first resized from their original size 512 × 512 to 256 × 256, then are compressed with JPEG quality factors (QF) 75 and 95. The embedding rates are set from 0.1 to 0.4 bits per non-zero AC DCT coefficient (bpnzac). For each combination of embedding rate and quality factor, we use four types of steganographic algorithms in the frequency domain, J-UNIWARD [14], UED [15], UERD [16] and nsF5 [12], and the steganographic algorithm feature library is composed of five types of steganographic algorithm features (including cover). Totally, there are 6000 × 5 × 4 × 2 = 240000 groups of stego features.

All feature extractors are built via curriculum training [35]. We first train the network for 0.4 bpnzac as it is the easiest task for extractors, and these parameters are seeded for the network for 0.3 bpnzac, and the rest can be inferred in the same manner. The feature extractors are trained for 200 k iterations with an initial learning rate of 0.001 and another 50 k iterations with the learning rate are cut to one-tenth of the original.

There are some special features in the training, the network for J-UNIWARD with JPEG quality factor 95 at 0.4 bpnzac is seeded by the network trained for J-UNIWARD with JPEG quality factor 75 at 0.4 bpnzac. When we train the network of two types of stego images, we experience convergence problems, so we initialize the network with the parameters trained by stego and cover images as usual.

3.2 Optimal Classifier

In the experiments, we choose some traditional classifiers. We take the testing set with QF 95 and embedding rate 0.1 bpnzac as an example to compare their effects. The matching ratios of experiments are shown in Table 1. The higher the value, the stronger the confidence of being successfully matched. Red numbers indicate that the corresponding algorithm predictions are not accurate. Support vector machine, lightgbm and neural network are denoted as SVM, LGB and NN, respectively. We use the default parameters of SVM and LGB in this experiment. As for NN, we design a three-layer fully connected network with activation function of ReLU [36], a loss function of cross entropy and an optimizer of RMSprop [37]. Among these classifiers, lightgbm can match the steganography algorithm most correctly and achieve a highest confidence level, so we use lightgbm as our multi-classifier.

Table 1: Matching ratios of different classifiers for QF 95 and bpnzac 0.1

Test algorithm	Classifier		
	SVM	LGB	NN
UED	0.2883	0.3343	0.3244
JUNI	0.5175	0.3480	0.3938
UERD	0.2032	0.3010	0.2931
nsF5	0.3444	0.4478	0.3964

3.3 Optimal Sub-Feature Combination

To obtain more diverse representations of image features, we conduct experiments to compare the matching performance of steganographic algorithm by using different combinations of sub-features. We also take the testing set with QF 95 and embedding rate 0.1 bpnzac as an example. Figs. 6–9 show the matching results of different steganographic algorithms. The results show that all the combinations of sub-features achieve excellent performance except the combination of *min_max*. Considering the time consumption, we prefer to choose the single sub-feature. Among the three single sub-features, we use sub-feature *mean* in the following experiments because it has the most powerful capability of matching steganographic algorithm.

3.4 Steganography Matching Results

In this section, we conduct the experiments to verify the effectiveness of the steganographic algorithm matching method. In detail, for each of the four steganographic algorithms, J-UNIWARD, UED, UERD and nsF5, we use all 2000 pairs of images with cover images coming from the aforementioned testing set. The experiments of different QFs and embedding rates are conducted separately. Algorithm J-UNIWARD is denoted as JUNI. Tables 2 and 3 show the matching results of the four JPEG steganographic algorithms when QF = 75 and QF = 95, respectively.

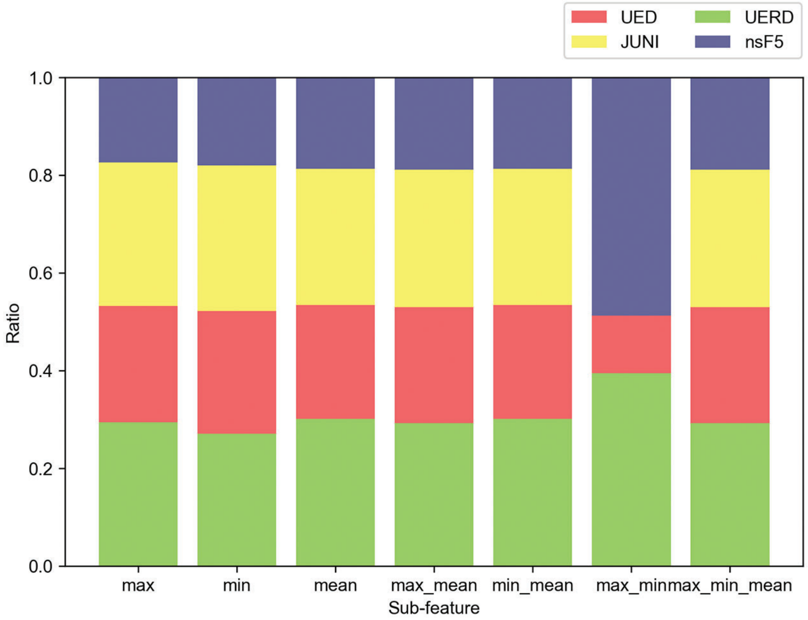

Figure 6: Matching ratio of each sub-feature combination for UERD

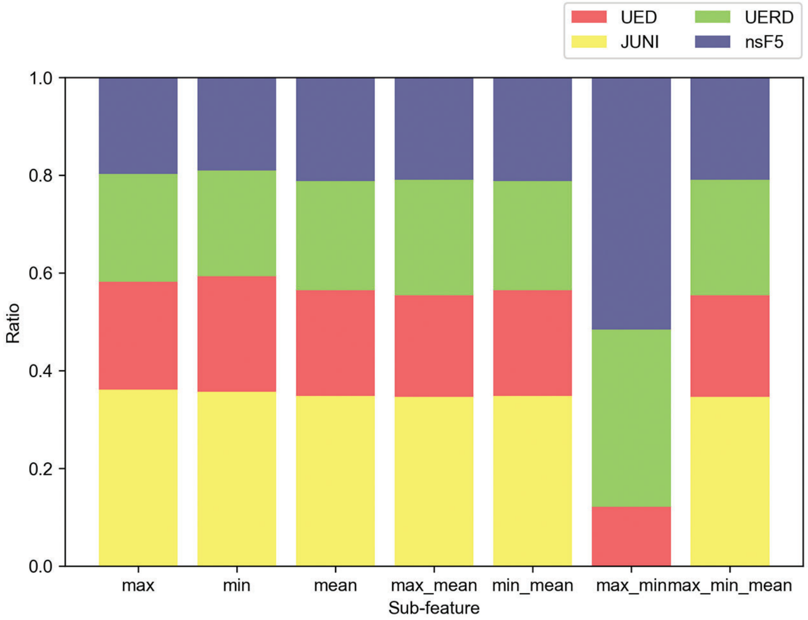

Figure 7: Matching ratio of each sub-feature combination for JUNI

For both Tables of 2 and 3, the first column *ER* represents the embedding rate of images, the second column *TA* represents the steganographic algorithm used on test images, the third to sixth columns represent the matching ratios of four categories (UED, JUNI, UERD and nsF5 respectively), and the last column *Result* is the final prediction result. We take the first row as an example, for the steganographic algorithm of UED, our approach predicts it as UED, JUNI, UERD and nsF5 with probabilities of 0.5074, 0.2083, 0.1228 and 0.1615, respectively. The bold number is the highest probability in each row and we choose the corresponding steganographic algorithm as the matched algorithm. As we can see, four test algorithms are all matched successfully, which means we can handle the algorithm mismatch problem effectively by using the network trained by the matching steganographic algorithm.

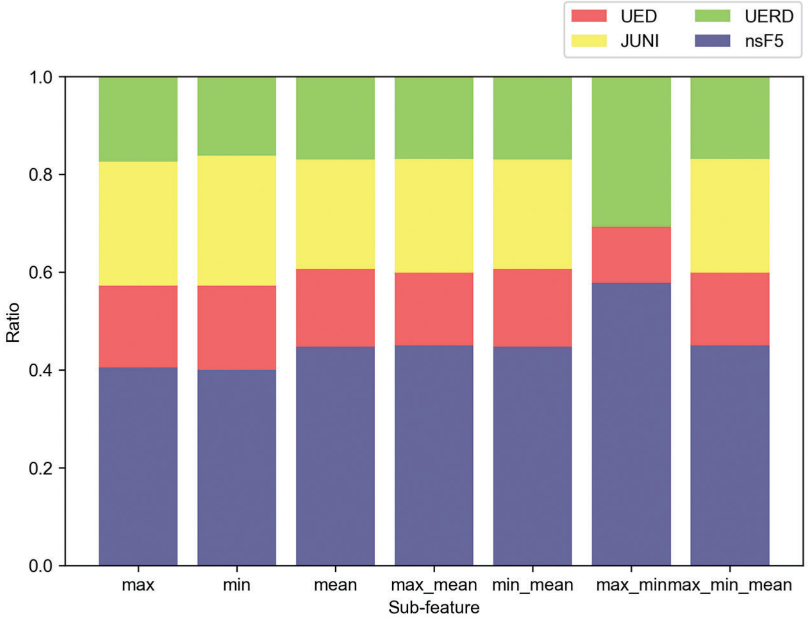

Figure 8: Matching ratio of each sub-feature combination for nsF5

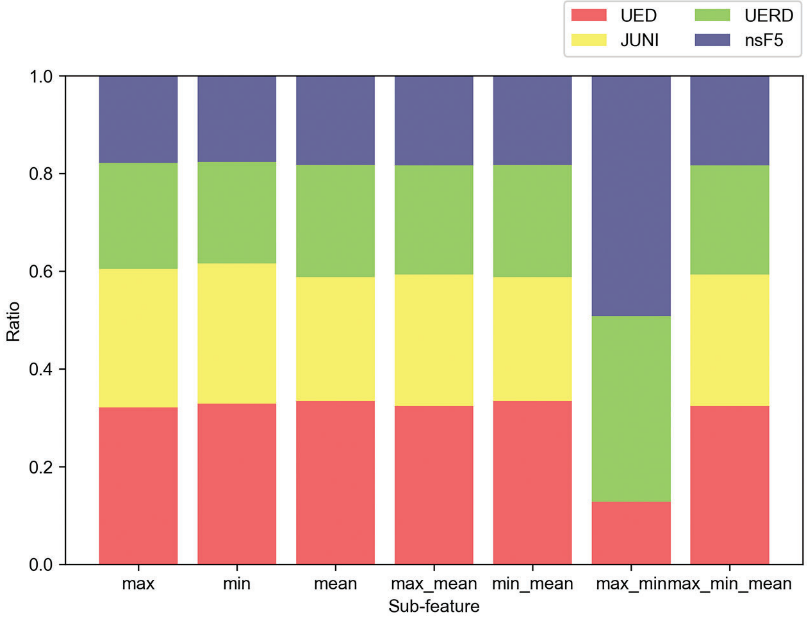

Figure 9: Matching ratio of each sub-feature combination for UED

Table 2: Matching result for QF 75

ER	TA	UED	JUNI	UERD	nsF5	Result
0.1	UED	**0.5074**	0.2083	0.1228	0.1615	UED
	JUNI	0.1115	**0.4718**	0.1943	0.2223	JUNI
	UERD	0.1336	0.2971	**0.3594**	0.2099	UERD
	nsF5	0.0930	0.2839	0.1434	**0.4797**	nsF5
0.2	UED	**0.6120**	0.1961	0.0925	0.0994	UED
	JUNI	0.0635	**0.6486**	0.1633	0.1245	JUNI
	UERD	0.0869	0.2702	**0.5181**	0.1248	UERD
	nsF5	0.0510	0.2395	0.0827	**0.6267**	nsF5
0.3	UED	**0.7245**	0.1629	0.0635	0.0491	UED
	JUNI	0.0370	**0.8018**	0.1192	0.0421	JUNI
	UERD	0.0693	0.2253	**0.6546**	0.0509	UERD
	nsF5	0.0403	0.1668	0.0520	**0.7409**	nsF5
0.4	UED	**0.8127**	0.1039	0.0526	0.0308	UED
	JUNI	0.0296	**0.8517**	0.1032	0.0155	JUNI
	UERD	0.0443	0.1630	**0.7731**	0.0196	UERD
	nsF5	0.0315	0.1000	0.0283	**0.8403**	nsF5

Table 3: Matching result for QF 95

ER	TA	UED	JUNI	UERD	nsF5	Result
0.1	UED	**0.3343**	0.2538	0.2295	0.1824	UED
	JUNI	0.2168	**0.3480**	0.2229	0.2123	JUNI
	UERD	0.2333	0.2790	**0.3010**	0.1866	UERD
	nsF5	0.1588	0.2236	0.1698	**0.4478**	nsF5
0.2	UED	**0.4392**	0.2787	0.2098	0.0723	UED
	JUNI	0.2180	**0.4758**	0.2254	0.0808	JUNI
	UERD	0.2338	0.2969	**0.3961**	0.0732	UERD
	nsF5	0.1186	0.2152	0.1207	**0.5455**	nsF5
0.3	UED	**0.5098**	0.2858	0.1812	0.0232	UED
	JUNI	0.1447	**0.6268**	0.2074	0.0210	JUNI
	UERD	0.1787	0.3012	**0.4996**	0.0204	UERD
	nsF5	0.0598	0.2104	0.0826	**0.6472**	nsF5
0.4	UED	**0.5901**	0.2664	0.1274	0.0161	UED
	JUNI	0.1151	**0.7233**	0.1507	0.0109	JUNI
	UERD	0.1482	0.2749	**0.5641**	0.0128	UERD
	nsF5	0.0456	0.1990	0.0544	**0.7010**	nsF5

3.5 Blind Steganalysis Results

In this section, we use our proposed method to evaluate the detection accuracy of steganalysis in the case of algorithm mismatch. We compare our work with state-of-the-art works, including the subspace learning-based method [38] and SRNet with algorithm mismatch. The detection performance is measured as follows:

$$P_E = \min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD}) \tag{2}$$

Tables 4 and 5 present the detection results with QFs 75 and 95 of 0.1 bpnzac, respectively. As we can see in both tables, our method has a significant improvement over the detection of SRNet with algorithm mismatch, the performances of different steganographic algorithms are improved by more than 15% with QF 75 and approximately 10% with QF 95 averagely. Compared with the method proposed by [38], our method also achieves better performance in the majority of cases. We also conduct experiments on other conditions. Table 6 presents the detection results with embedding rates ranging from 0.2 to 0.4, the first column *ER* represents the embedding rate of images, the third column *TA* represents the steganographic algorithm used on test images, the fourth column shows the detections results of SRNet and the fifth columns shows the results of our proposed method. Our method also has better performance than SRNet under different embedding rates and quality factors.

Table 4: The detection errors of different steganalysis system with the embedding rate 0.1 bpnzac and QF 75

	JUNI	nsF5	UED	UERD
SRNet [24]	43.91%	44.65%	29.74%	37.37%
Method proposed in [38]	43.80%	42.00%	42.25%	NaN
Our method	**33.60%**	**31.28%**	**14.47%**	**19.73%**

Table 5: The detection errors of different steganalysis system with the embedding rate 0.1 bpnzac and QF 95

	JUNI	nsF5	UED	UERD
SRNet [24]	48.07%	47.41%	43.13%	42.24%
Method proposed in [38]	**40.40%**	42.85%	40.50%	NaN
Our method	43.53%	**31.05%**	**34.97%**	**32.45%**

The core of our method is to extract multi-dimensional steganographic features so that the multi-classifier can match the steganographic algorithm of the testing set. The reason why matching can be successful is that for each steganographic algorithm, we have the features extracted by the feature extractor obtained through training by pairs, which can help multi-classifier to distinguish different steganographic algorithms. When the feature extractor does not extract any useful features, the multi-classifier can only produce random prediction, and the probability of each steganography algorithm is $\frac{1}{N}$. When the feature extractor itself has a certain discrimination ability, it can extract useful steganographic features, the multi-classifier will also increase the selection probability of the matching algorithm in the N groups of features participating in the training, and using a voting ensemble can make it easier to obtain the correct results. When the steganography algorithm is relatively difficult to detect, the confidence of matching will also decline. For example, the matching value of quality factor 75 is always greater than quality factor 95.

Table 6: The detection errors of different steganalysis system

ER	QF	TA	SRNet [24]	Our method
0.2	75	UED	16.12%	**8.12%**
		JUNI	34.55%	**23.40%**
		UERD	24.72%	**11.59%**
		nsF5	32.32%	**17.02%**
	95	UED	31.81%	**24.18%**
		JUNI	43.15%	**37.73%**
		UERD	33.29%	**24.18%**
		nsF5	36.04%	**16.60%**
0.3	75	UED	10.76%	**4.50%**
		JUNI	31.91%	**14.80%**
		UERD	20.81%	**6.52%**
		nsF5	23.75%	**7.37%**
	95	UED	26.59%	**16.35%**
		JUNI	40.25%	**30.47%**
		UERD	28.06%	**16.45%**
		nsF5	27.50%	**6.45%**
0.4	75	UED	8.70%	**2.28%**
		JUNI	28.53%	**9.25%**
		UERD	19.13%	**3.68%**
		nsF5	14.38%	**2.88**%
	95	UED	23.34%	**9.35**%
		JUNI	36.14%	**20.95**%
		UERD	24.85%	**11.32**%
		nsF5	17.21%	**1.63**%

4 Conclusions and Future Work

In this paper, we propose a deep learning driven feature-based multi-classifier model to solve the steganalysis problem in the case of algorithm mismatch. Representative steganographic features extracted by neural networks are designed to train a multi-classifier for finding the most matched steganographic algorithm, then the model trained by this steganographic algorithm is used to detect the test images. Experimental results show that the proposed method outperforms state-of-the-art works significantly.

However, this method still has some problems, for example, the process of feature extracting takes a long time because deep neural networks need to be trained, and it can only handle steganographic algorithms that are already known. Therefore, designing a lightweight network to extract the features without the loss of accuracy is one of our future works. In addition, implementing incremental learning to deal with new steganographic algorithms is also under consideration.

Funding Statement: This work was supported by the National Natural Science Foundation of China (NSFC) under grant No. U1836102, Anhui Science and Technology Key Special Program under the grant No.201903a05020016, and 2020 Domestic Visiting and Training Program for Outstanding Young Backbone Talents in Colleges and Universities under the grant No. gxgnfx2020132.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Dhawan and R. Gupta, "Analysis of various data security techniques of steganography: A survey," *Information Security Journal: A Global Perspective*, vol. 30, no. 2, pp. 63–87, 2021.

[2] L. Shi, Z. Wang, Z. Qian, N. Huang, P. Puteaux *et al.,* "Distortion function for emoji image steganography," *Computers, Materials & Continua*, vol. 59, no. 3, pp. 943–953, 2019.

[3] Y. Tong, Y. Liu, J. Wang and G. Xin, "Text steganography on RNN-generated lyrics," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5451–5463, 2019.

[4] S. Rahman, F. Masood, W. U. Khan, N. Ullah and F. Qudus, "A novel approach of image steganography for secure communication based on lsb substitution technique," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 31–61, 2020.

[5] A. M. Alhomoud, "Image steganography in spatial domain: current status, techniques, and trends," *Intelligent Automation & Soft Computing*, vol. 27, no. 1, pp. 69–88, 2021.

[6] D. Datta, L. Garg, K. Srinivasan, A. Inoue, G. T. Reddy *et al.,* "An efficient sound and data steganography based secure authentication system," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 723–751, 2021.

[7] J. Wang, M. Cheng, P. Wu and B. Chen, "A survey on digital image steganography," *Journal of Information Hiding and Privacy Protection*, vol. 1, no. 2, pp. 87–93, 2019.

[8] A. Baumy, A. D. Algarni, M. Abdalla, W. El-Shafai, F. E. Abd EI-Samie *et al.,* "Efficient forgery detection approaches for digital color images," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3257–3276, 2022.

[9] Z. Yan, P. Yang, R. Ni, Y. Zhao and H. Qi, "CNN-based forensic method on contrast enhancement with JPEG post-processing," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3205–3216, 2021.

[10] N. Provos, "Defending against statistical steganalysis," in *Proc. Usenix Security Symp.*, Washington DC, vol. 10, pp. 323–336, 2001.

[11] P. Sallee, "Model-based steganography," in *Proc. Int. Workshop on Digital Watermarking*, Berlin, Heidelberg, Springer, pp. 154–167, 2003.

[12] J. Fridrich, T. Pevný and J. Kodovský, "Statistically undetectable jpeg steganography: Dead ends challenges, and opportunities," in *Proc. of the 9th Workshop on Multimedia & Security*, New York, ACM, pp. 3–14, 2007.

[13] T. Filler, J. Judas and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[14] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proc. of the First ACM Workshop on Information Hiding and Multimedia Security*, Montpellier, France, pp. 59–68, 2013.

[15] L. Guo, J. 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, 2014.

[16] 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, no. 12, pp. 2669–2680, 2015.

[17] Q. Zhang, L. T. Yang, Z. Chen and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, no. 9, pp. 146–157, 2018.

[18] Y. Zhang and Z. Wang, "Hybrid malware detection approach with feedback-directed machine learning," *Information Sciences*, vol. 63, no. 139103, pp. 1–139103, 2020.

[19] Z. Li, J. Zhang, K. Zhang and Z. Li, "Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4478–4489, 2018.

[20] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.

[21] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.

[22] J. Kodovský, J. Fridrich and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2011.

[23] 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, no. 5, pp. 1200–1214, 2017.

[24] 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, 2018.

[25] T. Pevný and J. Fridrich, "Novelty detection in blind steganalysis," in *Proc. of the 10th ACM Workshop on Multimedia and Security*, Oxford, UK, pp. 167–176, 2008.

[26] T. Pevný and J. Fridrich, "Towards multi-class blind steganalyzer for JPEG images," in *Proc. Int. Workshop on Digital Watermarking*, Berlin, Heidelberg, Springer, pp. 39–53, 2005.

[27] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proc. Security, Steganaography, and Watermarking of Multimedia Contents IX*, vol. 6505, no. 2, pp. 650503, 2007.

[28] T. Pevný and J. Fridrich, "Multi-class blind steganalysis for JPEG images," *Proc. Security, Steganaography, and Watermarking of Multimedia Contents VIII*, vol. 6072, no. 2, pp. 60720O, 2006.

[29] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Proc. Int. Workshop on Information Hiding*, Berlin, Heidelberg, Springer, pp. 67–81, 2004.

[30] Y. Q. Shi, C. Chen and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Proc. Int. Workshop on Information Hiding*, Berlin, Heidelberg, Springer, pp. 249–264, 2006.

[31] X. Kong, C. Feng, M. Li and Y. Guo, "Iterative multi-order feature alignment for JPEG mismatched steganalysis," *Neurocomputing*, vol. 214, no. 1, pp. 458–470, 2016.

[32] C. Feng, X. Kong, M. Li, Y. Yang and Y. Guo, "Contribution-based feature transfer for JPEG mismatched steganalysis," in *Proc. 2017 IEEE Int. Conf. on Image Processing (ICIP)*, Beijing, China, pp. 500–504, 2017.

[33] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen *et al.,* "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, USA, pp. 3146–3154, 2017.

[34] P. Bas, T. Filler and T. Pevný, "break our steganographic system: The ins and outs of organizing boss," in *Proc. Int. Workshop on Information Hiding*, Berlin, Heidelberg, Springer, pp. 59–70, 2011.

[35] Y. Bengio, J. Louradour, R. Collobert and J. Weston, "Curriculum learning," in *Proc. of the 26th Annual Int. Conf. on Machine Learning*, New York, NY, USA, pp. 41–48, 2009.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc ICML*, Haifa, Isreal, pp. 807–814, 2010.

[37] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[38] Y. Xue, L. Yang, J. Wen, S. Niu and P. Zhong, "A subspace learning-based method for JPEG mismatched steganalysis," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8151–8166, 2019.