Cover-source Mismatch in Deep Spatial Steganalysis

Xunpeng,Zhang[1][0000−0001−9415−6759], Xiangwei Kong[2][0000−0002−0851−6752], Pengda Wang[3][0000−0001−6779−574X], and Bo Wang[4][0000−0003−1057−7973]

[1] Dalian University of Technology, Dalian LN 116081, China
[2] Zhejiang University, Hangzhou ZJ 310058, China
[3] Dalian University of Technology
[4] Dalian University of Technology
zhangxunpeng@mail.dlut.edu.cn

Abstract. In conventional image steganalysis, cover-source mismatch is a serious problem restricting its utility. In our work, we validate that in deep steganalysis, cover-source mismatch still exists. But unlike in conventional scenarios, sharp accuracy reduction just exists in a part of cover-source mismatch scenarios in deep steganalysis. To explain this phenomenon, we use A-distance to measure the texture complexity between databases. Furthermore, to ease the accuracy reduction caused by the mismatch, we adapt JMMD into deep steganalysis and design a new network (J-Net). Extensive experiments prove A-distance and J-Net works well.

Keywords: Steganalysis · Deep learning · Cover-source mismatch.

1 Introduction

Steganography is the technique which embeds information into cover image imperceptibly to carry out secret and safe communication. Steganalysis is the technique to detect the existence of steganography, i.e., to judge whether the image is embedded with secret information. When taking steganalysis into real world, the images you want to detect and the images used to train the steganalysis model always come from different distributions. This phenomenon is called cover-source mismatch, which is a serious problem restricting the utility of steganalysis. In conventional steganalysis, cover-source mismatch always causes sharp accuracy reduction, and there are many researchers focusing on solving it [10, 13, 15]. In recent years, with the development of deep learning,researchers started to solving steganalysis problem using deep neural networks. But in steganalysis based on deep learning, cover-source mismatch problem has not attracted much attention [4], and there are rare works discussing that. Hence, in this paper we study the cover-source mismatch scenario in deep steganalysis.

At first, we do analysis on cover-source mismatch scenario in deep steganalysis and validate that cover-source mismatch still exists. But not like the scenario in conventional steganalysis, sharp accuracy reduction just exists in a part of

situations in deep steganalysis. According to that texture complexity is positively related to the steganalysis difficulty, we think it's caused by the discrepancy of texture complexity among databases. To explain this phenomenon, we adapt A-distance [1] to measure the texture complexity among the databases we use.

Now that there is still cover-source mismatch problem in deep steganalysis, we want to address it. But unlike in conventional steganalysis, deep steganalysis models unify preprocessing, feature extraction and classifying into one framework. So methods for conventional cover-source mismatch can't be used to solve the problem in deep steganalysis. Then inspired by researches in domain adaptation which has similar scenarios as cover-source mismatch, we adapt JMMD [12] into deep steganalysis and design a deep adaptive network (J-Net) to address the cover-source mismatch problem. To our best knowledge, there is no research working for solving this problem in deep steganalysis. And experiments prove that J-Net can relieve the accuracy reduction caused by cover-source mismatch.

The contributions of this paper are concluded as follows :

1.We validate that cover-source mismatch still exists in deep steganalysis, and use A-distance to explain it quantitatively, which is instructive for future works.

2.To ease the accuracy reduction caused by cover-source mismatch, We adapt JMMD into deep steganalysis and design a deep adaptive architecture (J-Net).

3.Extensive experiments prove that J-Net can ease the accuracy reduction caused by cover-source mismatch effectively.

2 Related Works

2.1 Deep steganalysis

With the impressive performance of deep learning in other fields, scholars started to utilize deep neural networks into steganalysis. In 2015, Qian et al. [18] first adapted CNN(convolutional neural network) to abstract the features used for steganalysis. Based on it, according to the speciality of steganalysis, Xu et al. [21] adjusted the details in CNN layers to promote its performance. Inspired by the idea in conventional steganalysis, Ye-Net [22] adapted selection channel knowledge into deep steganalysis and achieved much better performance than conventional methods. Utilizing the residuals, Fridrich et al. [3]designed a network which can be used both in spacial domain and JPEG domain. Adapting spatial pyramid pooling, Zhu-Net [25] could take images in random sizes as input.

In conventional steganalysis, cover-source mismatch attracted much attention [10, 13, 15]. But in deep steganalysis, there are rare works discussing it. [4] said mismatch in deep steganalysis is not yet really well treated and understood. And [16] proposed that there is no mismatch phenomenon when using CNNs in steganalysis, but we have found sufficient experimental evidence to prove their conclusion might be ill-considered to some degree.

2.2 Deep domain adaptation

Domain adaptation focuses on the problem that how to transfer the model trained on labeled source database to the unlabeled target database without sharp accuracy reduction, where the source and target database have different distributions, which is very similar to the cover-source mismatch scenario in steganalysis. To measure the distance between the source database and the target database, Ben-David et al. [1] proposed A-distance to measure the domain discrepancy. With the development of deep learning, researchers did lots of works about the generality of deep neural networks [2,5,7,24]. Glorot et al. [7] proposed that while deep neural networks are more general than conventional networks, they still can't remove the discrepancy across domain. Furthermore, [24] proposed that the features in deep CNNs will transform from general to specific along the network, i.e, the deeper are the layers, the less transferable are the features.

Based on these theories, scholars further studied how to improve the domain adaptation performance of deep neural networks. VRNN [17] learned temporal dependencies to create domain invariant representations; Madasu et al. [14] designed GCN to filter out domain dependant knowledge; Deng et al. [6] proposed an active transfer learning network to get competitive performance using minimally labeled training data.

Among deep domain adaptation researches, MMD(maximum mean discrepancy) is a very popular tool to restrict the discrepancy between source and target domain. MMD is proposed by [19] to measure the distance between two statistic distributions. In 2012, Chen et al. [20] started to adapt MMD into deep domain adaptation. And Long et al. [11] proposed a variant of MMD called multi-kernel maximum mean discrepancy(M-MMD). Then based on M-MMD, Long et al. [12] proposed JMMD which takes the joint distribution of the input images and the predicted labels into account. Inspired by researches above, we adapt JMMD into deep steganalysis to address cover-source mismatch problem.

3 Methodology

3.1 Analysis of cover-source mismatch in deep steganalysis

Steganographies with high concealment always embed information into the high frequency part of the image, which has less probability of being detected by steganalysis [9]. It means that, the texture complexity of database is positively correlated to the steganalysis difficulty [16]. Based on it, we believe that there will be cover-source mismatch when the training set and testing set come from different database with different texture complexity. In experiment part, we prove that in spatial domain, there is sharp accuracy reduction when the steganalysis model is trained on less textured set and tested on more textured set.

To measure the texture complexity among databases, we adapt A-distance [1] to be the measurement tool. Next we give a simple introduction of A-distance.

A-distance is proposed to measure the discrepancy between two databases, which is calculated using the following formula:

$$\hat{d}_A = 2(1 - 2 \times error) \tag{1}$$

where *error* stands for the generalization error of a binary classifier (fully connected layers with 2 outputs here) trained on the binary problem to distinguish input samples between the training and testing database. More implementation details will be given in the experiments part.

Although A-distance is just a linear form of binary classifier, it can be used to measure the discrepancy between 2 databases in the latent space depending on the features used for the classifier. When the features for the classifier in A-distance are features used for steganalysis, the latent space where the discrepancy is measured in A-distance is steganalysis-relevant. Therefore, we think that A-distance can measure the attributes which is relevant to steganalysis between 2 databases, including texture complexity. And the experimental results prove its effectiveness.

3.2 J-Net for cover-source mismatch in deep steganalysis

Since there is cover-source mismatch problem in deep steganalysis, we try to ease the accuracy reduction caused by cover-source mismatch. Note that cover-source mismatch in steganalysis is really similar to the scenario in domain adaptation: the model is trained on the labeled source database and tested on the unlabeled target database which is in different distribution with the source. Hence, inspired by the impress performance of JMMD(joint maximum mean discrepancy). [12] in domain adaptation, we adapt it into deep steganalysis and design J-Net(Fig.1).The structure of J-Net can be divided into four part: preprocessing, feature extraction, classifier and JMMD.

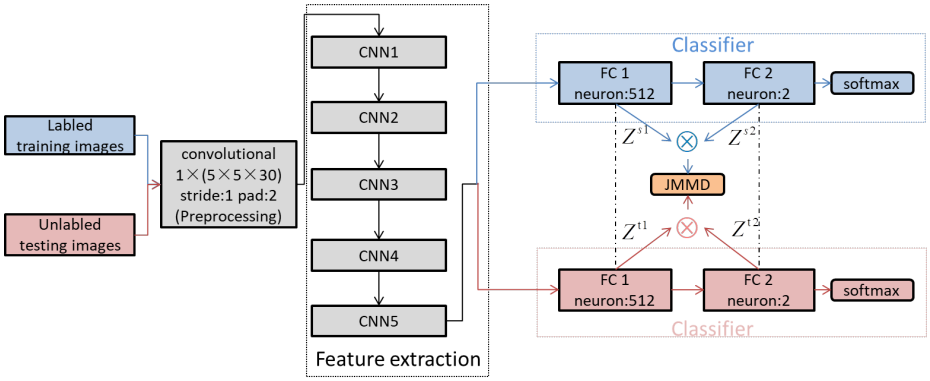

Fig. 1. The structrue of J-Net. The dotted line stands for sharing parameters.

As is known to us, the stego signal is always embedded in the high frequency part of image. Hence, to improve the signal to noise ratio(the ratio of stego signal to image signal), we use 30 5×5 high pass filters to preprocess the input images just as Yedroudj-Net [23].

Then the feature extraction part consists of 5 CNN (convolutional neural network) layers. The implementation details of the CNN layers are shown in Table 1. Note that, because the average pooling operation acts as a low pass filter [3] while the stego signal acts as high frequency noise, we get rid of the pooling operation in CNN1. In addition, since in the bottom CNN layers, relatively to image signal, stego signal is very small, ReLU(Rectified Linear Unit) is not suitable for the weak stego signal. Hence, we use TLU(truncate linear unit) [22] as the activation in CNN1 and CNN2. TLU function is defined as:

$$f(x) = \begin{cases} -T, x < -T \\ x, -T \leq x \leq T \\ T, x > T \end{cases} \tag{2}$$

The classifier in J-Net concludes 2 fully connected layers followed with a softmax function which is always used for classifier:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}} \tag{3}$$

Note that, the FC layers here have no difference from the fully connected layers in other model. What is special is that,after pre-trained procedure, the FC layers in J-Net will be fine-tuned under the constraint of JMMD.

At last, we use JMMD(joint maximum mean discrepency) [12] to measure and restrict the discrepancy of features in 2 fully connected layers, separately extracted from the training and testing database which come from different distribution. JMMD measures the distance of P and Q in reproducing kernel Hilbert space (RKHS), which is defined as:

$$D_L(P,Q) \triangleq ||L_{Z^{s,1:|L|}}(P) - L_{Z^{t,1:|L|}}(Q)||^2_{\otimes_{l=1}^{|L|} H^l} \tag{4}$$

where P and Q stand for the distribution of the training and testing database respectively, which are called source and target domain respectively here. $L_{Z^{s,1:|L|}}(P)$ are the features in layer L extracted from P, which are in distribution P, and H represent the reproducing kernel Hilbert space. Assuming that the source domain $\mathrm{D_s}$ has labeled $\mathrm{n_s}$ points drawn i.i.d from P, while the target domain $\mathrm{D_t}$ has $\mathrm{n_t}$ unlabeled points drawn i.i.d from Q. The CNN will get features in layer 1 to L as $\{(\mathrm{z_i^{s1}},...,\mathrm{z_i^{sL}})\}_{\mathrm{i=1}}^{\mathrm{n_s}}$ and $\{(\mathrm{z_i^{t1}},...,\mathrm{z_i^{tL}})\}_{\mathrm{i=1}}^{\mathrm{n_t}}$. In empirical calculation, we use the estimate of $\mathrm{D}_L(P,Q)$, which is defined as :

$$\begin{aligned} \hat{D}_L(P,Q) =& \frac{2}{n}\sum_{i=1}^{n/2}(\prod_{l\in L} k^l(z_{2i-1}^{sl}, z_{2i-1}^{sl}) + \prod_{l\in L} k^l(z_{2i-1}^{tl}, z_{2i-1}^{tl})) \\ & -\frac{2}{n}\sum_{i=n/2}^{n}(\prod_{l\in L} k^l(z_{2i-1}^{sl}, z_{2i-1}^{sl}) + \prod_{l\in L} k^l(z_{2i-1}^{tl}, z_{2i-1}^{tl})) \end{aligned} \tag{5}$$

Table 1. Implementation details of CNNs in J-Net.

CNN1	converlution 30×(5×5×30) stride:1 pad:2
	ABS
	BN
	TLU
CNN2	converlution 30×(5×5×30) stride:1 pad:2
	BN
	TLU
	average_pooling(5×5) stride:2
CNN3	converlution 30×(3×3×32) stride:1 pad:2
	BN
	ReLU
	average_pooling(5×5) stride:2
CNN4	converlution 32×(3×3×64) stride:2 pad:2
	BN
	ReLU
	average_pooling(5×5) stride:2
CNN5	converlution 64×(3×3×128) stride:1 pad:2
	BN
	ReLU
	global_average_pooling(32×32) stride:2

The entire loss function of J-Net is composed of two parts:

$$\min_{f} \frac{2}{n_s} \sum_{i=1}^{n_s} (J(f(x_i^s), y_i^s)) + \lambda \hat{D}_L(P, Q) \tag{6}$$

where $J(\bullet)$ is classifier loss, $f(x_i)$ represents the predicted output of the input image, $\hat{D}_L(P, Q)$ stands for the JMMD distance between P and Q, in layer L. Note that, in J-Net the features in fc layers are special and less transferable, so L=FC1,FC2 . By minimizing $\hat{D}_L(P, Q)$, the features in FC1 and FC2 can be as similar as possible in the reproducing kernel Hilbert space.

4 Experiment

4.1 Experimental settings

Database According to the analysis in Subsection3.1, we need textured and less textured database together to conduct experiments. We choose BOSSBase to be the textured database, which contains 10000 512×512 images in pgm format. As mentioned above, cover-source mismatch in steganalysis is a serious problem in real world, while ImageNet and MIRFlickr are good samples of the real world. Hence, to match the image format and amount of BOSSBase, we randomly select 10000 images from ImageNet and 10000 from MIRFlickr respectively and transform them into pgm format. Then, to maintain the consistency of image

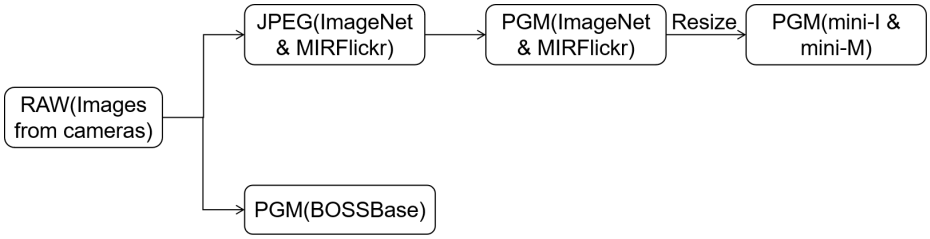

Fig. 2. Database processing programme.

size, we resize the images to 512×512. The database composed of 10000 512×512 images in pgm format from ImageNet (MIRFlickr) is called mini-I (mini-M).

Next, based on Fig.2, we'll utilize a simple inference to illustrate that images in mini-I and mini-M are less textured than images in BOSSBase. As we all known that, the compression from raw format to pgm format is less lossy than the compression from raw format to jpeg format; and the format conversion from jpeg to pgm will bring additional information loss; then the resizing process will further hurt the texture of image. Hence, mini-I and mini-M are less textured than BOSSBase. Note that it's not a serious inference because the images in BOSSBase, ImageNet and MIRFlickr are from different cameras, but we can still think that most images in mini-I and mini-M are less textured than images in BOSSBase(Fig.3).

Note that, most images in ImageNet and MIRFlickr are smaller than 512×512, so due to the texture hurting, the mini-I and mini-M will be detected with high accuracy. This scenario can not be used in normal steganalysis research, but in our experiments, no matter how we process it, what we need is less textured database. In real world scenario, there will be also many images which are processed in unknown way.

Implementation details All the experiments are implemented on pytorch with NVIDIA 1080Ti. And we adopt stochastic gradient descent (SGD) algorithm to update the parameters of J-Net, the learning rate is initialized as 0.001, and multiply 0.9 every 90 epochs. In addition, in all experiments shown in this paper, we use S-UNIWARD [9] and WOW [8] at 0.4 bpp(bits per pixel) as the stegaography methods.

4.2 Validation of cover-source mismatch

To validate the cover-source mismatch in deep steganalysis, we train and test J-Net without the JMMD module on BOSSBase, mini-I and mini-M respectively, which make up 9 scenarios totally(Table 2). Note that, without the JMMD module, the loss function of J-Net can be rewritten as:

$$\min_{f} \frac{2}{n_s} \sum_{i=1}^{n_s} (J(f(x_i), y_i)) \tag{7}$$

Fig. 3. Image samples.

From Table 2, we can see that there is sharp accuracy reduction, when the model is trained on the less textured database(mini-I or mini-M) and tested on the more textured database BOSSBase; while there is little or even no accuracy reduction when using the model trained on BOSSBase to detect mini-I or mini-M. In addition, when using the model trained on mini-I (or mini-M) to test images from mini-M (or mini-I), although there is a little accuracy reduction, the accuracy on test database still reaches upper than 90%.

Unlike the scenario in conventional steganalysis, sharp accuracy reduction just exists in a part of situations in deep steganalysis. This phenomenon may be caused by the strong learning ability of deep neural networks: Since the texture complexity is positively related to the steganlysis difficulty, when trained on textured database, the deep model can learn more intrinsical features for classifier, which can also perform well on less textured database. But how to judge a database is textured or not? In the next part, We use A-distance [1] to measure the texture complexity between databases.

4.3 Texture complexity measurement by A-distance

Fig.4 shows that A-distance matches the texture complexity analysis in Subsection4.1 and the experimental results above very well, which will be described in detail next:

A-distance between 2 of the 3 databases we use is shown in Fig.4. Note that in Fig.4, negative B→I stands for that mini-I is less textured than BOSSBase,

Table 2. Validation of cover-source mismatch(%)

train:BOSSBase			
	BOSSBase	mini-M	mini-I
suni-0.4	81.3	79.9	87.25
wow-0.4	83.323	78.15	85.475
train:mini-M			
	BOSSBase	mini-M	mini-I
suni-0.4	54.425	97.975	94.2
wow-0.4	53.825	97.875	95.675
train:mini-I			
	BOSSBase	mini-M	mini-I
suni-0.4	61.85	93.275	97.325
wow-0.4	63.175	92.45	96.475

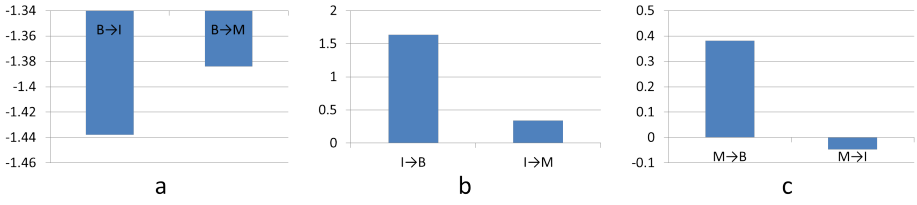

Fig. 4. A-distance among BOSSBase (B), mini-I (I) and mini-M (M).

and the larger is the absolute value of B→I, the less is the texture similarity of BOSSBase and mini-I. And in the process to get B→I, features used for the classifier are extracted from BOSSBase and mini-I by CNN trained on BOSSBase for steganalysis.

A-distance in Fig.4 shows that after the processing programme, mini-I and mini-M are less textured than BOSSBase, which demonstrate the inference in Subsection4.1. It matches the experimental results in Subsection4.2 well: there is sharp accuracy reduction when steganalysis model is trained on less textured mini-I(mini-M) and tested on BOSSBase, while there is just little accuracy reduction between mini-I and mini-M which have similar texture complexity. Note that to our best knowledge, there is no research measuring the texture complexity of databases numerically before, we believe that it's instructive for future works.

From experimental results above, we can concluded that not relying on the image content, the cover-source mismatch problem in deep steganalysis is mainly related to the texture complexity of the database, which is very different from mismatch problem in other computer vision fields.

4.4 J-Net for cover-source mismatch in deep steganalysis

Table 3. The accuracy promotion of J-Net(%).

train:mini-I test:BOSSBase			
	pre-train	J-Net	promotion
suni-0.4	61.85	68.95	7.1
wow-0.4	63.175	71.2	8.025
train:mini-I test:BOSSBase			
	pre-train	J-Net	promotion
suni-0.4	54.425	63.875	9.45
wow-0.4	53.825	63.725	9.9

At first, it's necessary to introduce the training process of J-Net:

1. Pre-train J-Net on training database without JMMD module, which has been done in Subsection4.2;

2. Except full connected layers, fix parameters of J-Net;

3. Fine-tune J-Net with the labeled training images and unlabeled testing images.

Table 3 gives the J-Net experimental results in cover-source scenarios where sharp accuracy reduction happens. From Table3 we can see that, in these cover-source mismatch scenarios, J-Net can promote the accuracy by 7%-10%, which demonstrate the effectiveness of J-Net. Note that, up to now, this is the first attempt to solve the cover-source mismatch problem in deep steganalysis.

Note that, domain adaptation strategy can only be used in the scenario where the images tested are a set of images. How to address the cover-source mismatch problem when the image tested is a single image will be the future work.

4.5 Parameter analysis

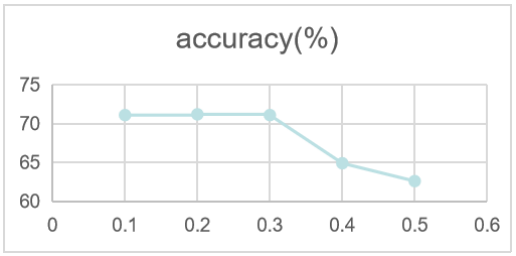

Fig. 5. Analysis of λ in the loss function of J-Net.

In this section, we do analysis on the tradeoff parameter λ in the loss function of J-Net(Figure 5). The experiment is implemented in the scenario where training and testing set is mini-I and BOSSBase respectively, and the steganography is wow(0.4bpp).

Fig.5 shows that, along the increasing of λ, the performance of J-Net rise first and then fall, which demonstrates that the adaptation of JMMD makes sense. According to Figure 5, we fixed all the λ in experiments as 0.2.

Table 4. Experiments in steganography mismatch scenario(%).

train:suni-0.4			
	BOSSBase	mini-I	mini-M
suni-0.4	81.875	97.325	97.975
wow-0.4	78.575	96.275	97.05
train:wow-0.4			
	BOSSBase	mini-I	mini-M
suni-0.4	77.433	97.775	95.375
wow-0.4	83.025	97.925	96.525

4.6 Bonus experiments

In addition to cover-source mismatch scenario, we do bonus experiments on steganography mismatch scenario(Table 4). Table 4 shows that there is just

little accuracy reduction when the training and testing database are embedded with different steganographies. It means that whether the steganography is WOW or S-UNIWARD, J-Net(without JMMD) can learn similar features, which demonstrates the strong learning ability and generality of deep neural network in steganalysis.

5 Conclusion

In this paper, we do analysis on cover-source mismatch scenarios in deep steganalysis, and find that unlike in conventional steganalysis, sharp accuracy reduction just exists in a part of situations in deep steganalysis. To explain this phenomenon, we utilize A-distance to measure the texture complexity between databases. To address the sharp accuracy reduction caused by cover-source mismatch, we adapt JMMD into deep steganalysis and design J-Net. Experimental results prove the effectiveness of J-Net.

References

1. Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.
2. Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
3. Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.
4. Marc Chaumont. Deep learning in steganography and steganalysis from 2015 to 2018. *arXiv preprint arXiv:1904.01444*, 2019.
5. Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.
6. Cheng Deng, Yumeng Xue, Xianglong Liu, Chao Li, and Dacheng Tao. Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1741–1754, 2018.
7. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.
8. Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, pages 234–239. IEEE, 2012.
9. Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1, 2014.
10. Jan Kodovskỳ, Vahid Sedighi, and Jessica Fridrich. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, page 90280J. International Society for Optics and Photonics, 2014.

11. Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.
12. Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.
13. Ivans Lubenko and Andrew D Ker. Steganalysis with mismatched covers: Do simple classifiers help? In *Proceedings of the on Multimedia and security*, pages 11–18. ACM, 2012.
14. Avinash Madasu and Vijjini Anvesh Rao. Gated convolutional neural networks for domain adaptation. In *International Conference on Applications of Natural Language to Information Systems*, pages 118–130. Springer, 2019.
15. Jérôme Pasquet, Sandra Bringay, and Marc Chaumont. Steganalysis with cover-source mismatch and a small learning database. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 2425–2429. IEEE, 2014.
16. Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. *Electronic Imaging*, 2016(8):1–11, 2016.
17. Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. 2016.
18. Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.
19. Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, Kenji Fukumizu, et al. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.
20. Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
21. Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016.
22. Jian Ye, Jiangqun Ni, and Yang Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.
23. Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont. Yedroudj-net: an efficient cnn for spatial steganalysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2092–2096. IEEE, 2018.
24. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
25. Ru Zhang, Feng Zhu, Jianyi Liu, and Gongshen Liu. Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis. *IEEE Transactions on Information Forensics and Security*, 2019.