Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 5041–5061
DOI: 10.3934/mbe.2019254
Received: 24 January 2019
Accepted: 09 May 2019
Published: 01 June 2019

Research article

Joint statistics matching for camera model identification of recompressed images

Bo Wang[1], Yabin Li[1], Xue Sui[2,*], Ming Li[1] and Yanqing Guo[1]

[1] School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China

[2] School of Psychology, Liaoning Normal University, Dalian, 116024, China

* **Correspondence:** Email:suixue@lnnu.edu.cn; Tel: +86-0411-8215-6430.

Abstract: Source camera identification has been well studied in laboratory environment where the training and test samples are all original images without recompression. However, image compression is quite common in the real world, when the training and test images are double JPEG compressed with different quantization tables, the identification accuracy of existing methods decreases dramatically. To address this challenge, we propose a novel iterative algorithm namely joint first and second order statistics matching (JSM) to learn a feature projection that projects the training and test features into a low dimensional subspace to reduce the shift caused by image recompression. Inspired by transfer learning, JSM aims to learn a new feature representation from original feature space by simultaneously matching the first and second order statistics between training and test features in a principled dimensionality reduction procedure. After the feature projection, the divergence between training and test features caused by recompression is reduced while the discriminative properties are preserved. Extensive experiments on public Dresden Image Database verify that JSM significantly outperforms several state-of-the-art methods on camera model identification of recompressed images.

Keywords: camera model identification; double JPEG compression; transfer learning; image forensics; recompressed images

1. Introduction

Due to the advance of digital technologies, the use of digital imaging devices for taking high quality pictures, such as digital cameras, camcorders and smart phones has incredibly become widespread in our daily life. At the same time, the digital images taken by these devices could be used as evidences in criminal investigation and many other fields, such as judicial system, news media and military. However, digital images can easily be tampered and forged with no visible traces, making it essential

to identify the authenticity and the source of digital images in forensic applications. So, the technology of blind source camera identification emerges under this circumstance.

As a challenging branch of multimedia forensics, source camera identification aims to determine and authenticate the original sources of digital images to support forensics, to be specific, its goal is to prove whether a given image was taken by a specific imaging device [1]. It has been well studied over the last decade, and a significant number of outstanding methods have been proposed. Taking a general survey on source camera identification, most of the existing works could be divided into two categories: camera individual identification and camera model identification, seeing [1, 2, 3] for a detailed review.

1.1. Camera individual identification

For camera individual identification, most researches use sensor pattern noise (SPN) to identify the individual camera of a query image. Researches assume that every imaging device inevitably has some internal defects on camera lens due to the shortage of raw material or the limitation of manufacturing techniques. The defects would leave indelible mark in images, which is unique of camera individual and could be used as the fingerprint to identify the camera source. Thus, SPN based methods are often used for camera individual identification.

Lukas *et al*. demonstrated that it is possible to identify camera individuals by SPN in [4]. They extracted noise residual from the images using a wavelet denoising filter to construct reference SPN as the unique fingerprint of camera individuals, and calculated the correlation between noise residual of the query image and reference SPN to determine the source of the test image. Obviously, for SPN based methods, more accurate estimation of SPN generally means higher identification accuracy. Many attempts [5, 6, 7, 8, 9] have been made to extract purer SPN by means of assigning weighting factors according to the magnitude of scene details or adopting adaptive filters. In addition, the high storage requirement and computation complexity of SPN matching is also an important issue for SPN based methods, and many efforts [10, 11, 12, 13] have been made to address this problem.

Indeed, image content has a significant influence of the purity of SPN. In order to increase the reliability of reference SPN, many researchers tend to choose images without much scene details, such as blue sky, to generate reference SPN. That is one of the limitations of SPN based methods. Besides, a few images taken by the tested camera individuals are required as prior information to obtain the reference SPN, which may not realistic in practical forensic scenario. That is the other limitation of SPN based methods [1]. Technically speaking, these methods are not rigorous blind image forensics.

1.2. Camera model identification

For camera model identification, most researches are based on well designed statistical features and machine learning algorithms. The methodologies mainly rely on the statistical traces left in images by the imaging pipeline and post-processing process inside imaging devices, such as white balancing, demosaicing algorithms and JPEG compression. Figure 1 illustrates the internal structure and imaging model of camera devices. These well designed features are fed into support vector machine (SVM) or ensemble classifiers to classify the image's source camera. The statistical features are usually model specific because cameras of the same model have the same image processing algorithms, so they are often used for camera model identification. Our work aims at camera model identification, and the

detailed discussion about statistical features based camera model identification methods is shown in Chapter 2.

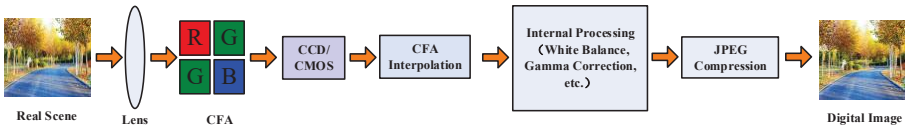


Figure 1. The internal structure and imaging model of camera devices.

Though statistical features based methods could achieve satisfactory identification performance (about 99%) in laboratory environment, there is still one challenging problem, the robustness against double JPEG compression. To prevent any potential confusion, we call the JPEG images directly exported from cameras the original images. The original images are all JPEG images and if the original images undergo another JPEG compression, we call them recompressed images. Existing methods all assume that the images from training and test set are all original images without recompression, and this implies that the features of training and test samples have to share the same statistical properties, or follow a common feature distribution. However, double JPEG compression is very common in practical forensic applications, and when the prerequisite of common feature distribution is not met, the identification accuracy of traditional methods will decrease dramatically [14]. For example, when the test samples are recompressed while the training samples are recompressed with another quantization table or even not recompressed, the features extracted from test set are very different from that of the training set because the statistical characteristics have been changed due to different manipulations, though small, quite influential. As a consequence, these statistical features based methods seem can not perform well on this condition, and it's still an open issue.

1.3. Goals and objectives

In this paper, we study the camera model identification in this more practical and challenging scenario, where the test samples are double JPEG compressed with different quality factors. To address this challenge, we propose a novel iterative algorithm namely joint first and second order statistics matching (JSM) to simultaneously match the first and second order statistics between training and test features in a principled dimensionality reduction procedure. Specifically, we propose to learn a feature projection that project the training and test features into a low dimensional subspace to align the characteristics of training and test features by means of matching both the first and second order statistics and this explicitly reduces the influence of double JPEG compression. After the projection, 1) the data properties are preserved by maximizing the variance of the union of training and test data, 2) both the marginal and conditional distribution (first order statistics) divergence are minimized to reduce the influence of recompression, and 3) the second order statistics divergence between training and test set is minimized to further reduce the divergence caused by double JPEG compression.

The rest of the paper is structured as follows. In Section 2, representative researches related to our work are discussed. The proposed JSM method and its optimization procedure are elaborated in Section 3. In Section 4, extensive experiments are carried out and comparisons of state-of-the-art are presented to show the superiority of proposed JSM method. And the conclusion is drawn in Section 5.

2. Related work

Since we focus on statistical features based camera model identification, we firstly elaborate representative researches in this scope. In order to illustrate the advantage of the proposed method, we also briefly introduce several transfer learning works related to our work in this section.

2.1. Statistical features based camera model identification

An earlier work is [15], Kharrazi *et al*. proposed 34-dimensional features consisting of image quality metrics (IQM), average pixel value, wavelet domain statistics, neighbor distribution center of mass, RGB pair energy ratio and pair correlation to identify the device origin of a given image. Experiments on five camera models showed an average identification accuracy of 88.02%. Taking the influence of JPEG compression into consideration, San Choi *et al*. [16] proposed to use the bit per pixel and the percentage of non-zero integers in each discrete cosine transform (DCT) coefficients to examine the JPEG compression statistics left in images, and these features are used as the trace to classify images origin. Wang *et al*. [17] engaged statistical characteristics of wavelet coefficients co-occurrence features and higher-order wavelet features, and applied Sequential Forward Feature Selection (SFFS) algorithm to reduce the redundancy and correlation of features. Their method improved the distinguishing ability for images taken by same brand but different models compared with [15], and relaxed the restrictions of image samples compared with [16], with a higher identification accuracy of 98% on 6 camera models .

Considering the color filter array (CFA) and demosaicing algorithms may be different of different camera manufacturers, and even of different camera models with same manufacturer, Bayram *et al*. [18] proposed to identify the source camera based on traces of the proprietary interpolation algorithms. After estimating the weighting (interpolation) coefficients which designate the amount of contribution from each pixel in the interpolation kernel by Expectation- Maximization (EM) algorithm, they combined the weighting coefficients, the peak location and magnitudes in frequency spectrum as distinguishing features. But there is a limitation in [18], when we do not know the CFA pattern of an image in advance, we must try various CFA models and interpolation algorithms for coefficients estimation. To address this problem, Ho *et al*.[19] proposed to use the variance of color difference planes to measure the inter-channel correlation for source camera identification and achieved a promising result. Based on [19], Hu *et al*. [20] developed an improved algorithm using inter-channel demosaicing traces for camera model identification. The shape and texture features, which are the combination of energy, contrast, correlation and entropy of Gray Level Co-occurrence Matrixes (GLCM) of the variance of color difference planes, were fed into Stumps AdaBoost classifier, and experimental results showed it's superiority.

Xu *et al*. [21] proposed to use uniform gray-scale invariant local binary patterns (LBP) from spatial domain of red and green color channels, their prediction-error 2D arrays and the 1st-level diagonal wavelet subband of each image as statistical features. LBP can effectively capture the characteristics or artifacts generated by image processing algorithms which are block-wise implemented inside cameras, such as demosaicing, filtering and JPEG compression. The average identification accuracy reported in their paper is 98.08% of 18 camera models. Different from [21], the recent work [22] also investigated the discriminative ability of local phase quantization (LPQ), a LBP like texture descriptor, to distinguish imaging devices. The combined texture features of LBP and LPQ, extracted from original images, residual noise images and contourlet transform coefficients of the residual noise images in

HSV color space, were fed into a multi-class SVM classifier, and reached better detection accuracy than [21].

In addition, several studies aimed at using high dimensional features (more than 1000 dimensions) to classify the image's camera model. Chen *et al.* [23] built a rich model of a camera's demosaicing algorithm to identify the model of an image's source camera. For each CFA pattern and interpolating algorithm, they utilized two co-occurrence matrixes to capture the reconstructed error between the original image and the reconstructed versions. The full feature dimension is 1372 and they fed them into a multi-class ensemble classifier. The average identification accuracy over 12 camera models is 99.2% reported in their paper. Tuama *et al.* [24] developed their method for camera model identification by extracting high order statistics features with 10932 dimensions consisted of co-occurrences matrix features, traces of color dependencies features related to CFA interpolation arrangement features, and conditional probability statistics features. A better identification result compared with the correlation based method is demonstrated in their paper. Recently, Roy *et al.* [25] proposed to use Discrete Cosine Transform Residual (DCTR) features combined with PCA and ensemble classifier for high accuracy camera source identification. They proved DCTR features can capture the compression artifacts imposed by the camera model dependent quantization tables, which could be used to identify the image source. The identification accuracy reported in their paper is 96.5%. Note that the camera models used in their experiments are all from different brands.

In summary, there are many researches studying camera model identification, but existing methods can't handle the problem of double JPEG compression. Actually, these statical features based methods suffer from double JPEG compression severely. Our work aims to solve this problem by learning a feature projection that projects the training and test features into a low dimensional subspace to suppress the influence of recompression.

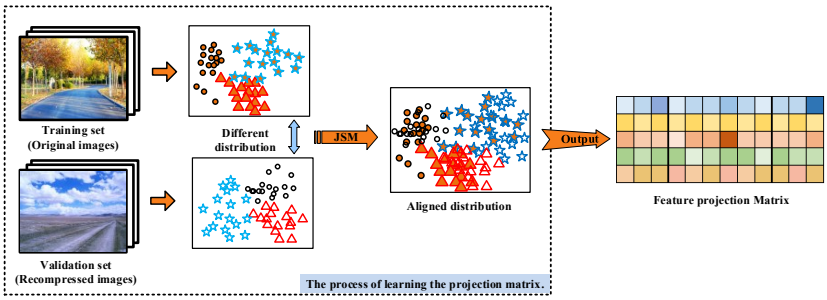
2.2. Transfer learning

The source camera model identification problem of compressed images is very similar to the mismatch of steganalysis and many other transfer learning based text classification tasks and computer vision tasks [26]. In these papers, the mismatched training and test sets are named as source and target domain respectively. These methods mainly focus on projecting the original features of two domains to another subspace with small domain shift by exploring correlations and other statistical information hidden in the samples between training and test set.

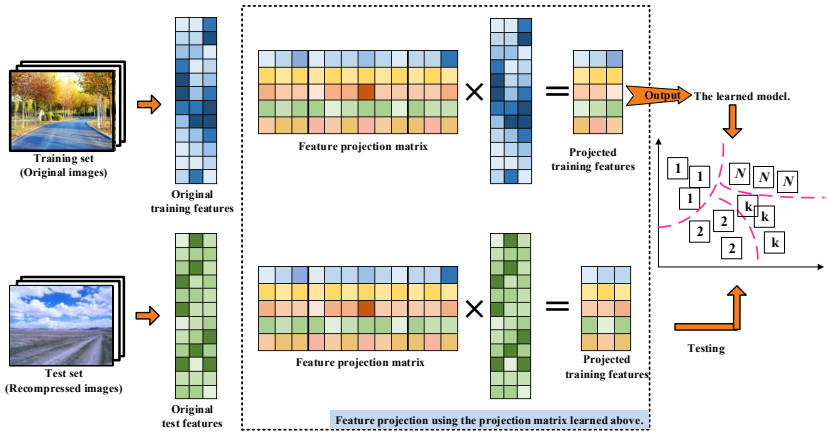
In [27], Pan *et al.* proposed a method named transfer component analysis (TCA) to learn a common feature transformation for source and target domain where the marginal distribution difference is minimized in projected feature space. Based on [27], Long *et al.* [28] took conditional distribution into consideration, and put forward joint distribution adaptation (JDA) to simultaneously reduce both the marginal and conditional distribution difference between domains. Inspired by these studies, Li *et al.* [29] proposed generalized transfer component analysis (GTCA) to solve the mismatched problem in steganalysis. Using an intermediate domain before TCA, GTCA can learn more discriminate representations for mismatched steganalysis. In [30], Luo *et al.* also introduced a repulsive force term to drag the sub-domains with different labels far away from each other to increase the discriminative power of the adapted domain. They named it as close yet discriminative domain adaptation (CDDA).

A recent transfer learning based research studying mismatched camera model identification is [31], Zhang *et al.* manually designed a feature transformation to minimize the distribution deviation between

training and test features. The experimental results reported in their paper are quite encouraging. Note that our proposed method is very different from [31], the transformation matrix in [31] is manually set based on test features and the transformation is only applied on training features while the feature projection matrix in JSM is automatically learned based on statistics matching and JSM projects both training and test features into a low dimensional subspace with small statistics divergence. Actually, the statistics divergence between training and test features caused by double JPEG compression in our task sometimes is quite influential, especially when the second compression quality factor is small. Directly applying the aforementioned methods to our problem is far from enough. More powerful methods need to be proposed to solve the camera model identification problem of recompressed images.



(a) Phase 1: The learning process of the projection matrix.



(b) Phase 2: The training and test step on projected features.

Figure 2. The framework of camera model identification through JSM.

3. Proposed joint statistics matching

The framework of camera model identification through JSM is shown in Figure 2. Firstly, we should learn the feature projection matrix through JSM on training and validation set as show in Figure 2a. Suppose there are a training set consisting of original images and a validation set consisting of recompressed images whose distribution is different from the former due to the recompression, our goal is to learn a feature projection through statistics matching to suppress the distribution divergence

caused by recompression. Once the feature projection matrix is available, we can project the original training features into the learned low dimensional subspace and train the identification model by SVM. Thus, given a query image, we could project the feature of the test image using the same feature projection matrix and then get the identification result using the model learned on the low dimensional subspace as show in Figure 2b.

As discussed above, the core content of the proposed method is the learning of feature projection matrix. So we explain detailedly how the feature projection matrix could be learned by statistics matching in this section. Overall, we propose to align the characteristics by matching both the first and second order statistics of training (original images) and validation (recompressed images) features while the important discriminative properties are preserved. To be exact, the distribution measurement in Section 3.2 could be simplified as the distance between the sample means of elements in original and recompressed images. So this item is named as the first order statistics matching. To further reduce the distribution divergence caused by recompression, we also investigate the high order statistical information hidden in the data. Specifically, we minimize the difference of variance between original and recompressed images to reduce the distribution divergence. And this item is named as second order statistics matching. The original intention of each item involved in the calculation and how our proposed method works for effective feature projection could be found in the following subsections.

3.1. Data properties preserving

Suppose we have a set of n_s labeled training samples (source domain, original images) $\mathbf{X}_s = \{\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, ..., \mathbf{x}_{s_{n_s}}\}$, $\mathbf{x}_{s_i} \in \mathcal{R}^d$, $i = 1, ..., n_s$, where d is the dimension of original features. $\mathbf{Y}_s = \{y_{s_1}, y_{s_2}, ..., y_{s_{n_s}}\}$ is the set of labels of $\mathbf{X}_s$ and $y_{s_i} \in \{1, ..., C\}$, where C is the number of classes in training set. Similarly, we have a set of n_t unlabeled test samples (target domain, recompressed images) $\mathbf{X}_t = \{\mathbf{x}_{t_1}, ..., \mathbf{x}_{t_{n_t}}\}$, $\mathbf{x}_{t_j} \in \mathcal{R}^d$, $j = 1, ..., n_t$. We assume the feature space and the label space of the two domains are the same: $\chi_s = \chi_t, \mathcal{Y}_s = \mathcal{Y}_t$. In our source camera identification problem, test samples (target domain) are recompressed while the training samples (source domain) are recompressed but with another quantization table or even not recompressed. Let $P_s(x)$ and $P_t(x)$ denotes the marginal distribution of $\mathbf{X}_s$ and $\mathbf{X}_t$ respectively. Due to the recompression of training and test images, the marginal distribution of source and target domain does not equal any more, that is $P_s(x) \neq P_t(x)$. So we aim to learn a unified projection $\varphi(\bullet)$ to match the distribution between domains in a latent feature space such that $P_s(\varphi(\mathbf{X}_s)) = P_t(\varphi(\mathbf{X}_t))$.

Although projected into a latent feature space, the data properties should be preserved and the reconstruction error should be minimized. It is a general choice to maximize the data variance as performed by principal component analysis (PCA). Denote $\mathbf{X} = [\mathbf{x}_{s_1}, ..., \mathbf{x}_{s_{n_s}}, \mathbf{x}_{t_1}, ..\mathbf{x}_{t_{n_t}}] \in \mathcal{R}^{d \times n}$ the input data matrix, where $n = n_s + n_t$, the covariance of $\mathbf{X}$ can be computed as $\mathbf{XHX}^{\mathrm{T}}$, where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$ is the centering matrix, $\mathbf{I} \in \mathcal{R}^{n \times n}$ is the identity matrix, and $\mathbf{1}$ is the $n \times n$ matrix of ones. Hence, when we project the original data into the latent feature space $\mathbf{A}^{\mathrm{T}}\mathbf{X}$, the variance maximization of the projected data could be achieved as follows

$$\max_{\mathbf{A}} \operatorname{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{XHX}^{\mathrm{T}}\mathbf{A}) \tag{3.1}$$

where $\operatorname{tr}(\bullet)$ denotes the trace of a matrix, $\mathbf{A} \in \mathcal{R}^{d \times k}$ is the projection matrix, and k is the dimension of the learned low dimensional subspace where $k < d$.

3.2. First order statistics matching

3.2.1. Marginal distribution matching

Only maximizing the data variance of projected data does no help in aligning the distribution between training and test set, the divergence between source and target domain is still pretty large. We need to reduce the domain shift in the latent feature space by explicitly minimizing the distribution difference through a proper distance measure. We adopt a non-parametric criteria Maximum Mean Discrepancy (MMD) as the distance measure to compare the distribution distance between domains. MMD is introduced by Gretton *et al.* to measure the distribution difference in [32] and will approach zero if and only if the two distribution are exactly the same [27]. Since we assume there is a unified transformation $\varphi(\bullet)$ for both source and target domain, the empirical estimate of MMD between projected source and target data could be computed as

$$\begin{aligned} &\mathrm{MMD}\left(\varphi(\mathbf{X}_s), \varphi(\mathbf{X}_t)\right) = \\ &\left\| \frac{1}{n_s} \sum_{\mathbf{x}_{s_i} \in \mathbf{X}_s} \varphi(\mathbf{x}_{s_i}) - \frac{1}{n_t} \sum_{\mathbf{x}_{t_j} \in \mathbf{X}_t} \varphi(\mathbf{x}_{t_j}) \right\|^2 \end{aligned} \tag{3.2}$$

Using the k-dimensional embeddings mentioned in (3.1), the MMD distance could be reformulated as

$$\begin{aligned} &\left\| \frac{1}{n_s} \sum_{\mathbf{x}_{s_i} \in \mathbf{X}_s} \mathbf{A}^{\mathrm{T}} \mathbf{x}_{s_i} - \frac{1}{n_t} \sum_{\mathbf{x}_{t_j} \in \mathbf{X}_t} \mathbf{A}^{\mathrm{T}} \mathbf{x}_{t_j} \right\|^2 \\ &= \mathrm{tr}(\mathbf{A}^{\mathrm{T}} \mathbf{X} \mathbf{M}_0 \mathbf{X}^{\mathrm{T}} \mathbf{A}) \end{aligned} \tag{3.3}$$

where $\mathbf{M}_0$ is the MMD matrix and is computed as follows

$$(M_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_s \\ \frac{1}{n_t n_t}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise} \end{cases} \tag{3.4}$$

Since we have found the proper distance measure MMD and (3.3) measures the distribution difference of the two domains, we then minimize (3.3) to achieve our goal of reducing the domain shift between training and test data. In another word, the domain shift will be reduced under the new feature representation $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$ when (3.3) is minimized.

3.2.2. Conditional distribution matching

After matching the marginal distribution, the domain shift has been reduced to a certain extent. But there is another thing causing domain shift between source and target domain, that is the difference of conditional distribution $Q_s(y_s|\mathbf{x}_s) \neq Q_t(y_t|\mathbf{x}_t)$. If the conditional distribution between domains are matched, the domain shift caused by double JPEG compression could be further reduced, which means better identification performance. Some methods have already been proposed to match the conditional distributions via kernel density eatimation [33] and sample selection in kernel mapped space [34]. But they all have a limitation of requiring some labeled samples in target domain, which is unrealistic in practical forensics scenario.

Since we have no labeled data in target domain, we can not directly estimate the conditional distribution of target domain $Q_t(y_t|\mathbf{x}_t)$ to match the. So we turn to utilize the pseudo labels of test samples y'_{t_j}, which can be obtained by applying a base classifier such as nearest neighbor classifier (NN) or support vector machine (SVM) trained on the source data to the unlabeled target data. With the pseudo labels in target domain and true labels in source domain, the class-conditional distribution shift in projected feature space is computed as

$$\begin{aligned} &\left\| \frac{1}{n_s^{(c)}} \sum_{\mathbf{x}_{s_i} \in \mathbf{X}_s^{(c)}} \mathbf{A}^{\mathrm{T}} \mathbf{x}_{s_i} - \frac{1}{n_t^{(c)}} \sum_{\mathbf{x}_{t_j} \in \mathbf{X}_t^{(c)}} \mathbf{A}^{\mathrm{T}} \mathbf{x}_{t_j} \right\|^2 \\ &= \mathrm{tr}(\mathbf{A}^{\mathrm{T}} \mathbf{X} \mathbf{M}_c \mathbf{X}^{\mathrm{T}} \mathbf{A}) \end{aligned} \tag{3.5}$$

where $\mathbf{X}_s^{(c)} = \{\mathbf{x}_{s_i} : \mathbf{x}_{s_i} \in \mathbf{X}_s \ \wedge \ y_{s_i} = c\}$ and $\mathbf{X}_t^{(c)} = \{\mathbf{x}_{t_j} : \mathbf{x}_{t_j} \in \mathbf{X}_t \ \wedge \ y'_{t_j} = c\}$ represent the subset of samples belonging to class c in the source and target domain respectively. y_{s_i} and y'_{t_j} is the true label of $\mathbf{x}_i$ and pseudo label of $\mathbf{x}_j$ correspondingly. $n_s^{(c)}$ and $n_t^{(c)}$ is the number of samples in $\mathbf{X}_s^{(c)}$ and $\mathbf{X}_t^{(c)}$. $\mathbf{M}_c$ is the conditional MMD matrix for class c which is computed as

$$(M_c)_{i,j} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} \mathbf{x}_i \in \mathbf{X}_s^{(c)}, \mathbf{x}_j \in \mathbf{X}_t^{(c)} \\ \mathbf{x}_i \in \mathbf{X}_t^{(c)}, \mathbf{x}_j \in \mathbf{X}_s^{(c)} \end{cases} \\ 0 \quad , & \text{otherwise} \end{cases} . \tag{3.6}$$

Similar to (3.3), we utilize (3.5) to measure the conditional distribution difference of the two domains, and the domain shift is reduced under the new feature representation $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$ when (3.5) is minimized. Based on the above analysis, we find that the MMD measurement in (3.3) and (3.5) between two domains could be simplified as the distance between the sample means of elements in source and target domain. So we call marginal distribution matching and conditional distribution matching the first order statistics matching.

3.3. Second order statistics matching

Now we have just aligned the marginal and conditional distribution between domains. Through the matching of first order statistics, the distribution divergence should be explicitly reduced. But it is still not good enough for our mismatched source camera identification problem, and there are plenty of statistical information hidden in the data of source and target domain we could use to further reduce the distribution divergence.

Here, we aim to match the second order statistics of projected data to reduce the domain shift. To reach this goal, we minimize the difference of variance between source and target domains as follows

$$\min_{\mathbf{A}} \mathrm{tr}\left(\mathbf{A}^{\mathrm{T}} \mathbf{X}_s \mathbf{H}_s \mathbf{X}_s^{\mathrm{T}} \mathbf{A} - \mathbf{A}^{\mathrm{T}} \mathbf{X}_t \mathbf{H}_t \mathbf{X}_t^{\mathrm{T}} \mathbf{A}\right) \tag{3.7}$$

where $\mathbf{A}$ is the transformation matrix and $\mathbf{H}_s = \mathbf{I}_s - \frac{1}{n_s}\mathbf{1}$, $\mathbf{H}_t = \mathbf{I}_t - \frac{1}{n_t}\mathbf{1}$ is the centering matrix of source and target domain respectively. Rewrite $[\mathbf{X}_s, \mathbf{X}_t]$ as $\mathbf{X}$, (3.7) can be rewritten as

$$\min_{\mathbf{A}} \mathrm{tr}(\mathbf{A}^{\mathrm{T}} \mathbf{X} \mathbf{L} \mathbf{X}^{\mathrm{T}} \mathbf{A}) \tag{3.8}$$

where $\mathbf{L}$ is the modified centering matrix, and is rewritten as

$$\mathbf{L} = \begin{bmatrix} \mathbf{H}_s & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_t \end{bmatrix} \tag{3.9}$$

When (3.8) is minimized, the second order statistics is matched between domains in projected feature space. With this improvement, JSM matches both the first and second order statistics between domains and could be quite effective for our mismatched source camera identification problem even when the distribution divergence caused by recompression is pretty large. Note that the goal of first and second order statistics matching could be achieved simultaneously.

3.4. Overall objective function

For JSM, we aim to reduce the domain shift between training and test set by simultaneously matching the first and second order statistics. According to the above analysis, we learn the feature projection by maxmizing Eq. (3.1) for data properties preserving (DPP) while minimizing Eq. (3.3) for marginal distribution matching (MDM), Eq. (3.5) for conditional distribution matching (CDM) and Eq. (3.8) for second order statistics matching (SOSM). So we incorporate the above four equations to formulate our JSM method as follows

$$\min \frac{\{\text{MDM}\} + \{\text{CDM}\} + \{\text{SOSM}\}}{\{\text{DDP}\}} \tag{3.10}$$

We further impose the constrain that $\text{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{A})$ is small to control the scale of $\mathbf{A}$. So we find the projection $\mathbf{A}$ by solving the following optimization function

$$\min_{\mathbf{A}} \frac{\text{tr}\left(\mathbf{A}^{\mathrm{T}}\left(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \alpha\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\right)\mathbf{A}\right) + \beta\text{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{A})}{\text{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A}\right)} \tag{3.11}$$

where α is the trade-off parameter to balance the importance of first and second order statistics matching, β is the parameter of regularization on $\mathbf{A}$ to guarantee the optimization problem to be well defined.

According to the generalized Rayleigh quotient, minimizing the numerator of (3.11) such that the denominator of (3.11) is maximized is equivalent to minimizing the numerator of (3.11) such that the denominator of (3.11) is fixed. So the final constrained objective function is as follows

$$\begin{aligned} &\min_{\mathbf{A}} \text{tr}\left(\mathbf{A}^{\mathrm{T}}\left(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \alpha\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I}\right)\mathbf{A}\right) \\ &s.t.\ \ \mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A} = \mathbf{I} \end{aligned} . \tag{3.12}$$

By minimizing (3.12), the distribution divergence between source and target domain is reduced through first and second order statistics matching simultaneously. Through feature projection, the learned feature representation is insensitive to recompression and that is why JSM is effectiveness in identifying the source cameras of recompressed images. It is worth noting that JSM seems similar to JDA, but they are essentially different, because we also take the second order statistics matching between domains into consideration, which can further reduce the domain shift caused by double JPEG compression. JDA is only a special case of JSM with $\alpha = 0$.

3.5. Optimization

To optimize (3.12), we solve the optimization problem according to the constrained optimization theory. The Lagrange function of (3.12) is

$$\begin{aligned} G = &\operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\left(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \alpha\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I}\right)\mathbf{A}\right) \\ &+ \operatorname{tr}\left(\left(\mathbf{I} - \mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A}\right)\mathbf{\Phi}\right) \end{aligned} \tag{3.13}$$

where $\mathbf{\Phi}$= diag $(\phi_1, ..., \phi_k) \in \mathcal{R}^{k\times k}$ are the Lagrange multipliers. By setting $\frac{\partial G}{\partial \mathbf{A}} = \mathbf{0}$, we can get a generalized eigendecomposition problem

$$\left(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \alpha\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I}\right)\mathbf{A} = \mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A}\mathbf{\Phi}. \tag{3.14}$$

Thus, the optimal projection matrix $\mathbf{A}$ can be found by solving (3.14) for the k smallest eigenvectors of $(\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}})^{-1}(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \alpha\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I})$. The pseudo code is summarized in Algorithm 1. Once the projection matrix $\mathbf{A}$ is available, we can project source and target data into the latent feature space $\mathbf{A}^{\mathrm{T}}\mathbf{X}_s$ and $\mathbf{A}^{\mathrm{T}}\mathbf{X}_t$ with domain shift reduced. And traditional machine learning methods such as SVM can be applied on the projected data to identify the source of double JPEG compressed images. It is worth noticing that the pseudo labels samples in target domain are iteratively refined throughout the JSM method in order to learn a better feature projection.

Algorithm 1 : Joint First and Second Statistics Matching

Input: Source/target data: $\mathbf{X}_s, \mathbf{X}_t$, source label: $\mathbf{Y}_s$, regularization parameter: α, β, subspace base: k.
Output: Transformation matrix: $\mathbf{A}$, embedding: $\mathbf{Z}$, adaptive classifier: f.

1: Construct MMD matrix $\mathbf{M}_0$ by (3.4), and set $\{\mathbf{M}_c = 0\}_{c=1}^{C}$.
2: Construct centering matrix $\mathbf{L}$ by (3.9).
3: **repeat**
4: Solve the generalized eigendecomposition problem in (3.14) and select the k corresponding eigenvectors of the k smallest eigenvalues to construct the transformation matrix $\mathbf{A}$, and set $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$.
5: Train a base classifier f on $\{\mathbf{A}^{\mathbf{T}}\mathbf{X}_s, \mathbf{Y}_s\}$ to update the pseudo labels in target domain $\{y'_j = f(\mathbf{A}^{\mathrm{T}}\mathbf{x}_{t_j})\}$.
6: Update MMD matrix $\{\mathbf{M}_c\}_{c=1}^{C}$ by (3.6).
7: **until** 10 iterations;
8: Obtain the final adaptive classifier f trained on $\{\mathbf{A}^{\mathrm{T}}\mathbf{X}_s, \mathbf{Y}_s\}$.

3.6. Theoretical performance analysis

Here we analyze the theoretical performance of Algorithm 1 by the computational and space complexity using the big O notation. We denote T the number of iterations. k is the dimension of subspace and C is the number of classes. Typically, $T \ll \min(m, n), k \ll \min(m, n), C \ll \min(m, n)$. The computational cost are detailed as follows: $O(TCn^2)$ for constructing the MMD and centering matrices in

Line 1, 2 and 6; $O(Tkm^2)$ for solving the generalized eigendecomposition problem in Line 4; $O(Tmn)$ for updating the pseudo labels in Line 5; The cost for training the final adaptive classifier in Line 8 can vary from the training algorithms which is denoted as $O(tra)$. So the overall computational complexity is $O(TCn^2 + Tkm^2 + Tmn + tra)$. As for the space cost: $O(n^2)$ for the MMD and centering matrices constructing; $O(n^2)$ for solving the generalized eigendecomposition problem; $O(mn)$ for updating the pseudo labels and $O(tra)$ for the final adaptive classifier. In total, the overall space complexity is $O(2n^2 + mn + tra)$.

4. Experiments

In this part, we conduct extensive experiments to evaluate the JSM method on source camera model identification problem of recompressed images. We compare our JSM method with several representative forensic methods and related transfer learning methods. The representative forensic methods are: three SVM based methods with color filter array features (CFA) [18], local binary pattern features (LBP) [21], and texture features (TF) [22], two ensemble classifiers based methods with ensemble of demosaicing features (EDF)[23] and discrete cosine transform residue features (DCTR) [25]. The compared transfer learning based methods are: transfer component analysis (TCA)[27], joint distribution matching (JDA)[28], generalized transfer component analysis (GTCA)[29], close yet discriminative domain adaptation (CDDA) [30] and cross-class alignment (CCA)[31]. We choose nearest neighbor classifier (NN) as the base classifier to predict pseudo labels in our experiment since it does not require tuning cross-validation parameter.

4.1. Experimental setup

4.1.1. Database

We evaluate our proposed method on a public JPEG image database, 'Dresden Image Database'[35]. The image database is specifically built for the purpose of development and benchmarking of camera-based digital forensic techniques [35], whose images are captured by different cameras indoor/outdoor with various camera settings. Some examples used in the experiments are shown in Figure 3.

Figure 3. Sample images used in the experiments.

We choose 16 camera models from the Dresden Image Database to form the dataset used in the experiments, the details can be found in Table 1. In order to verify the method will not affected by different individuals of the same model, we select one camera individual of each model to form the training set, and use the images taken by another individual of the same model as the test set. This test setting is more practical in that we aim to detect the source model of the given images.

Table 1. The camera models used in the experiments.

Camera Model	Abbr.	Format	#train	#test
Canon_Ixus70	C1	JPEG	180	180
Casio_EX-Z150	C2	JPEG	180	180
FujiFilm_FinePixJ50	F1	JPEG	180	180
Kodak_M1063	K1	JPEG	180	180
Nikon_CoolPixS710	N1	JPEG	180	180
Nikon_D70	N2	JPEG	180	180
Nikon_D200	N3	JPEG	180	180
Olympus_mju_1050SW	O1	JPEG	180	180
Panasonic_DMC-FZ50	P1	JPEG	180	180
Praktica_DCZ5.9	P2	JPEG	180	180
Rollei_RCP-7325XS	R1	JPEG	180	180
Samsung_L74wide	S1	JPEG	180	180
Samsung_NV15	S2	JPEG	180	180
Sony_DSC-H50	SD1	JPEG	180	180
Sony_DSC-T77	SD2	JPEG	180	180
Sony_DSC-W170	SD3	JPEG	180	180

4.1.2. Implementation details

In our experiments, we use the LBP features proposed in [21] as the original features for transfer learning based methods to identify the source camera model. The reason why we choose LBP features is that the basic identification accuracies in our experiment are highest compared with other traditional forensic methods, and the identification accuracies can be found in Table 2. Since the image size in dataset is different, we extract the LBP features in the 1024×1024 subimage cropped from the upper left corner. To have a fair comparison, other compared methods have the same setting of the proposed method.

To evaluate the performance of JSM in mismatched JPEG compression condition, we conduct an experiment with training set consisting of original images and test set consisting of recompressed images. We named it Experiments 1, and you can find the detailed description and experimental results in the next subsection. We also conduct an experiment on synthetic data to visually show how JSM works. What's more, we then conduct Experiment 2 with mixed training set consisting of images recompressed with various quality factors to see whether our proposed method is suitable for more complicated forensic scenario,

Note that there are 3 parameters involved in the objective function (3.12), we tune them through cross validation by empirically searching the parameter space $\alpha, \beta \in \{0.001, 0.01, 0.1, 1\}$ for the optimal

parameter setting, and the best results are reported in the experimental results. For TCA, GTCA, JDA, and CDDA, we also tune their parameters using the same way with JSM. As for subspace dimension k, we empirically fix $k = 50$ for all the experiments because it have little influence of the experimental results.

4.2. Results and discussion

To evaluate the effectiveness of our proposed method, we conduct extensive experiments under different situations as mentioned above. The classification accuracies of JSM and other baseline methods with different training samples are shown in Table 2-5. The highest accuracy for each mismatched task is highlighted in bold.

4.2.1. Experiment 1

In this experiment, we firstly compress the test images listed in Table 1 using stand quantization tables with quality factors from 70 to 100 with a internal of 5. Thus, we get several training sets named T70, T75, T80, T85, T90, T95, and T100. The original training and test sets are named as SOri and TOri. We train the identification model using SVM on the original training set SOri and apply it to the recompressed test sets T70, T75, T80, T85, T90, T95 and T100 to evaluate the performance of the proposed method when training and test set are mismatched.

Note that for transfer learning based methods, we should firstly learn the feature projection matrix $\mathbf{A}$. So we randomly choose 20% samples of each class in SOri and then compress them using the same quantization table of the test set to form the corresponding validation set. The regularization parameters are tuned by cross validation as mentioned in subsection implementation details, and we get $\alpha = 0.01, \lambda = 0.01$ in this experiment. With feature projection matrix, the training and testing procedure is conducted by SVM under the projected feature representation for transfer learning based methods. And the experimental results are listed in Table 2.

Table 2. Identification accuracy (%) when training samples are not re-compressed.

	traditional forensic methods					transfer learning based methods					
Train→Test	LBP	LPQ	CFA	DCTR	EDF	CCA	TCA	GTCA	JDA	CDDA	JSM
SOri→TOri	**96.25**	93.26	92.53	92.85	94.97	90.97	92.05	89.24	94.44	93.99	95.31
SOri→T100	85.42	34.86	81.70	92.88	73.85	89.55	89.79	87.33	92.50	92.33	**94.44**
SOri→T95	62.95	36.35	61.32	63.78	38.96	66.25	68.58	67.47	85.94	83.33	**87.33**
SOri→T90	37.78	22.57	35.14	28.37	20.21	36.04	48.26	44.72	65.66	65.10	**68.26**
SOri→T85	23.23	18.78	14.86	13.78	13.61	29.03	32.81	41.11	52.57	51.63	**59.20**
SOri→T80	18.96	13.30	11.70	11.15	11.22	22.29	25.73	31.35	44.62	44.31	**52.40**
SOri→T75	11.98	13.13	10.52	9.38	11.11	16.74	21.25	25.10	42.36	39.62	**46.91**
SOri→T70	10.49	10.49	8.44	8.02	10.87	15.17	18.54	20.10	32.26	32.88	**39.44**
Ave.	43.38	30.34	39.53	40.03	34.35	45.76	49.63	50.80	63.79	62.90	**67.91**

Firstly, we notice that traditional forensic methods perform very well when training and test set are matched, but poorly when mismatched especially when the domain shift between training and test set is substantially large. As we all know, traditional forensic methods are mostly based on SVM

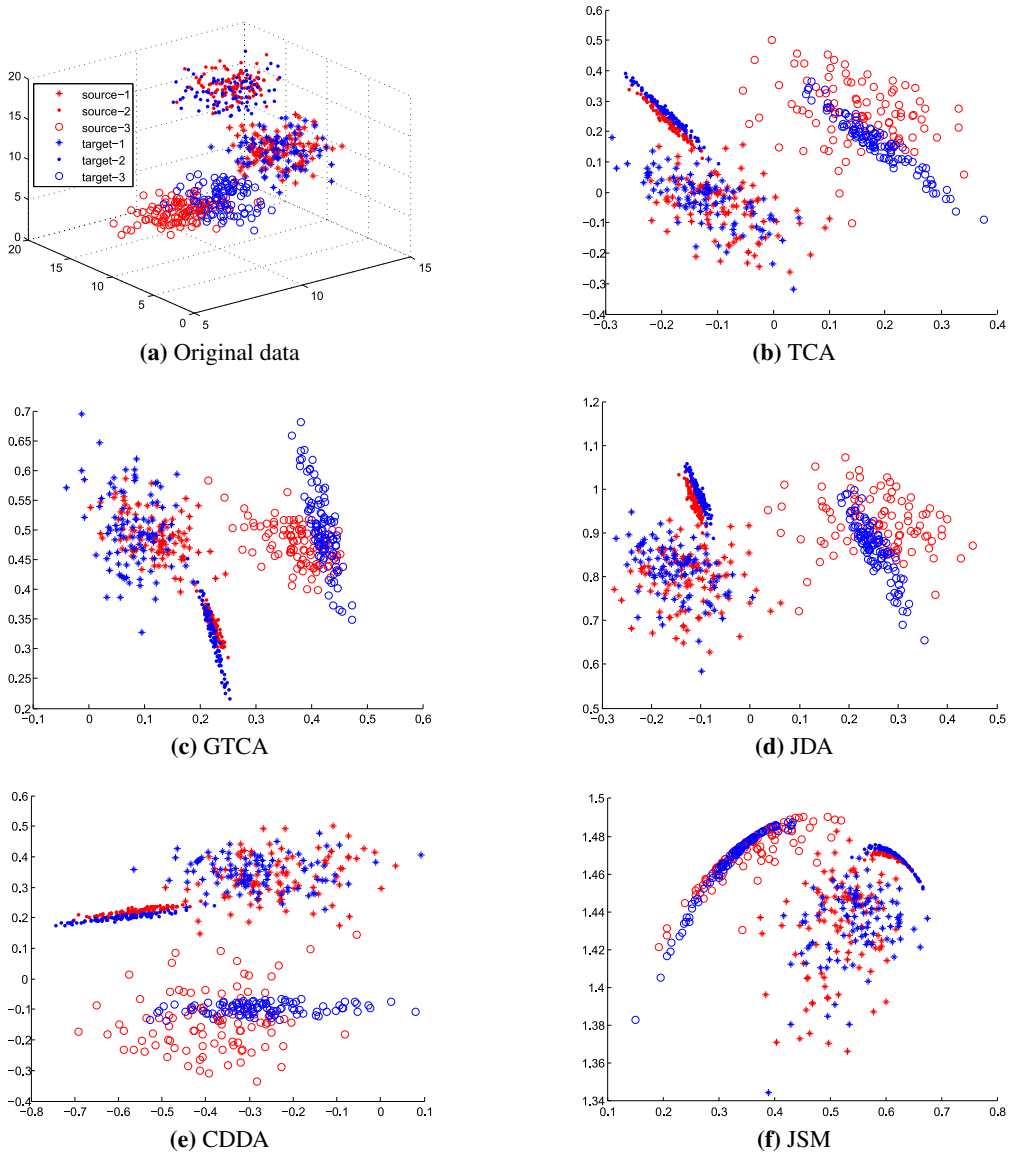
or ensemble classifiers, and they are powerful and reliable classification algorithms which can fit the training data well. When the test samples are drawn from the same distribution of training samples like case SOri→TOri, a satisfactory classification result can be achieved. However, when the test images are recompressed, the distributions of test samples are different from that of the training samples, the classification accuracy will decrease dramatically. The larger the domain shift, the worse the classification performance, as illustrated in Table 2. Though performing worse when test samples are recompressed, LBP [21] is still the best traditional forensic method in term of average identification accuracy, and that's why we choose LBP features as the original features for our proposed method. For a fair comparison, all the compared transfer learning based methods have the same experimental setup.

It can be observed that JSM achieve much better performance than the compared methods (both traditional forensic methods and transfer learning based methods) in each mismatched cases when test samples are recompressed. For SOri→T95, the proposed method shows an identification accuracy of 94.44% , gaining a significant improvement compared with 85.42% of LBP, 92.88% of DCTR, 92.50% of JDA and 92.33% of CDDA. When training set and test set are matched (case SOri→TOri), the result of JSM is slightly lower than that of LBP, and that is reasonable because there is no domain shift between training and test set. For other transfer learning based methods, there is a obvious gap from the basic LBP method for case SOri→TOri. When the quality factor of test images decreases, the identification accuracy of all the methods has declined due to the growing domain divergence, but JSM is still the best among the methods. JSM depicts an overall average accuracy of 67.91%, with an improvement of 4.12% compared with the result of best baseline method JDA. So JSM is more powerful in identifying recompressed image's source camera model.

4.2.2. Synthetic data

To visually demonstrate why JSM performs better than other transfer learning based methods, we also conduct an experiment on synthetic data. The source and target data are both draw from a mixture of several Gaussian distributions but with different means or variances. For class one, the variance of source and target data is different but with the same mean. It is exactly the opposite for the second class, the means are shifted between domains but with the same variance. As for the data of the third class, the domain shift is pretty large because both the mean and variance of source and target data are different. For the sake of simplicity, the original data are 3-dimensional and the subspace dimension k is set to 2.

Figure 4 illustrates the original synthetic data and the results of transfer learning methods applied on the synthetic data. It is obvious that after TCA the data of first class lies quite closer, but the domain divergence is still apparent for the third class due to the difference in variance. The method JDA seems only have little and not obvious improvement over TCA after aligning the conditional distribution. Because the marginal and conditional distribution matching are not enough to reduce the domain divergence caused by variance shift. Similarly, the repulsive force term of CDDA didn't help much in reducing the domain shift of the third class if the divergence caused by variance shift is not considered. Similar results can be observed in GTCA. But the result of JSM is quite encouraging compared with the methods mentioned above. It can keep the samples of different classes separated as well as align the distribution of the two domains effectively even when the domain shift is pretty large. From the above discussion, we can clearly see why JSM performs better than other transfer learning based methods from Figure 4.



(a) Original data **(b)** TCA

(c) GTCA **(d)** JDA

(e) CDDA **(f)** JSM

Figure 4. Comparisons of baseline methods and the proposed JSM method on synthetic data.

4.2.3. Experiment 2

In practical forensic scenario, we always know nothing about the compression history of the query image in prior. The model trained on SOri is not suitable for complicated forensic demand, especially when query images are recompressed with various quality factors. A intuitive solution is to retrain a classifier using images compressed with different quality factors. So we also compress the training images listed in Table 1 using standard quantization tables with quality factors from 70 to 100 with a interval of 5. Thus, we get several compressed training sets named S70, S75, S80, S85, S90, S95 and S100. And we mix up these compressed training sets with SOri and get a new mixed training set named SMixed. Then, we rebuild the identification model using SMixed as training set and report the experimental results in Table 3-5.

Similar to experiment 1, we randomly choose 20% training samples in SMixed as validation set to learn the feature projection matrix **A**. This validation construction strategy no longer relies on the prior information of the test samples and is more practical in real forensic scenario. And the regularization parameters $\alpha = 0.01, \lambda = 0.01$ are tuned by cross validation with the validation set.

The identification accuracies of each method when test samples are recompressed with aforementioned quality factors are shown in Table 3. The performance of traditional forensic methods performs barely satisfactory, LBP achieves an average accuracy of 89.06% and DCTR achieve a better result of 89.25% while LPQ, CFA and EDF performs worse. For transfer learning based methods, the average identification accuracies of CCA, GTCA and CDDA are lower than LBP, that's because the quality factors in Table 3 are the same with the training set, which means there is no clear shift between training and test set. The results show that CCA, GTCA and CDDA can't not handle the situation when there is no clear domain shift between training and test set.

Table 3. Identification accuracy (%) when training samples are mixed (SMixed).

	traditional forensic methods					transfer learning based methods					
Train→Test	LBP	LPQ	CFA	DCTR	EDF	CCA	TCA	GTCA	JDA	CDDA	JSM
SMixed→TOri	94.62	91.39	90.03	92.01	92.78	94.38	94.24	91.22	94.10	91.74	**94.93**
SMixed→T100	**94.58**	90.63	88.51	92.33	92.64	94.41	93.75	90.87	93.89	91.01	**94.58**
SMixed→T95	92.92	89.90	87.01	91.32	91.56	92.85	93.23	89.83	93.40	87.99	**94.31**
SMixed→T90	91.08	88.44	80.35	90.45	89.51	90.42	90.83	84.31	90.42	85.87	**91.91**
SMixed→T85	87.95	86.63	72.43	89.06	86.18	88.37	88.58	79.93	89.13	84.34	**90.10**
SMixed→T80	85.83	85.56	64.65	**88.02**	82.40	86.35	87.36	77.08	87.33	84.13	87.60
SMixed→T75	84.27	83.58	58.85	**86.42**	78.19	84.03	85.14	72.92	85.35	83.33	85.73
SMixed→T70	81.25	81.53	51.42	**84.38**	71.08	81.22	83.26	69.41	82.81	81.22	83.82
Ave.	89.06	87.20	74.16	89.25	85.54	89.00	89.55	81.94	89.55	86.20	**90.37**

Different from the three aforementioned transfer learning based methods, TCA, JDA and JSM all performs better than LBP, especially our proposed method JSM. JSM depicts an average identification accuracy of 90.37%, 1.31% higher than the basic LBP method. The promotion illustrates that the matching of the first and second statistics helps a lot in reducing the domain shift between training and test set and that's a strong demonstration of the effectiveness of the proposed JSM method. What's more, JSM also outperforms all other transfer learning based methods and that's another proof that demonstrates the effectiveness of our proposed method JSM to identify the source camera of recompressed images.

The average confusion matrix for the result of JSM in Table 3 is shown in Table 4. The average identification accuracy is 90.37% for 16 camera models. Note that the training and test samples are mixed of images recompressed with different quality factors, the identification accuracy of this experiment is lower than the result of SOri→TOri in Table 2. From the confusion matrix, we can see that the highest identification accuracy is 96.46% recorded by FujiFilm FinePixJ50 while the least identification accuracy is 66.67% recorded by Sony_DSC-W170. As we can see form Table 4, there is a clear gap between the accuracy of two Sony camera models (Sony_DSC-H50 and Sony_DSC-W170) and the average accuracy. Also these two models are always misclassified into the other class. Similarly, we also find the same phenomena in the identification result of all other compared methods, which means it is not a specific problem happened in JSM. The reason of this interesting phenomena is probably that

it is difficult to distinguish the images from these two models of the same brand especially when these models share strong similarity of features.

Table 4. Average confusion matrix for experiment 2. '-' means less than 0.5%.

Ave. 90.37%		Predicted															
		C1	C2	F1	K1	N1	N2	N3	O1	P1	P2	R1	S1	S2	SD1	SD2	SD3
Actual	C1	**94.86**	1.88	-	-	1.11	-	-	-	0.90	0.56	-	-	-	-	-	-
	C2	-	**91.39**	-	0.56	2.92	-	0.76	0.69	-	0.69	-	-	0.90	-	-	-
	F1	-	-	**96.46**	-	-	-	1.53	-	-	-	-	-	-	-	-	0.76
	K1	-	-	-	**95.63**	-	1.32	1.25	-	-	-	0.97	-	-	-	-	-
	N1	1.32	-	-	-	**92.43**	-	2.43	-	-	-	0.69	1.32	0.63	-	-	-
	N2	1.04	-	-	-	2.01	**89.86**	5.21	0.56	-	-	-	0.56	-	-	-	-
	N3	-	-	-	-	0.56	1.67	**94.79**	1.60	-	-	-	0.83	-	-	-	-
	O1	-	-	-	-	0.69	-	-	**96.04**	-	-	0.56	0.90	-	-	-	-
	P1	-	-	-	-	-	0.56	1.60	-	**92.36**	-	0.83	-	0.56	-	0.63	1.46
	P2	-	-	-	-	0.76	-	0.90	-	0.56	**96.25**	-	0.63	-	-	-	-
	R1	-	-	-	-	-	-	0.69	-	-	0.63	**94.86**	1.39	-	-	-	-
	S1	-	-	-	-	0.83	-	1.39	1.67	-	0.83	1.74	**92.01**	0.83	-	-	-
	S2	-	2.64	-	-	1.88	-	-	2.29	-	0.56	2.01	2.22	**85.49**	0.90	-	0.56
	SD1	0.69	0.76	-	-	-	-	0.97	-	-	-	-	-	-	**79.72**	5.14	11.46
	SD2	-	0.97	-	-	-	-	-	-	-	-	-	-	-	6.11	**87.15**	4.44
	SD3	0.76	0.83	0.83	-	-	-	0.56	1.46	1.81	-	-	-	0.56	18.54	6.81	**66.67**

As mentioned before, the practical forensic demand is quite complicated. A situation that needs to be taken into consideration is that the quality factors of the query image are not included in the training set. To evaluate the performance of JSM under this situation, we also recompress the test images with quality factor 72, 77, 82, 87, 92 and 97, and report the identification accuracy in Table 5 using the same model learned above. As one might expect, the identification accuracies of all the 11 methods have decreased compared with the result in Table 3, but the average identification of JSM is still the highest compared with other methods. The average identification of JSM have decreased to 87.34% from 90.37% while JDA have decreased to 86.52% from 89.55% and LBP have decreased to 85.83% from 89.06%. Under this situation, the promotion compared with the basic LBP method is 1.51% while the promotion compared with the basic LPQ method is 1.22%. Similar with the results in Table 3, both TCA, JDA and JSM performs better than LBP, and JSM outperforms TCA and JDA, which mean JSM is still the best compared with the baseline methods.

Table 5. Identification accuracy (%) for other recompression cases when training samples are mixed (SMixed).

	traditional forensic methods					transfer learning based methods					
Train→Test	LBP	LPQ	CFA	DCTR	EDF	CCA	TCA	GTCA	JDA	CDDA	JSM
SMixed→T97	91.42	89.83	84.17	82.43	91.81	91.81	91.74	85.87	92.12	87.60	**93.33**
SMixed→T92	87.36	88.13	80.45	75.97	**90.97**	87.01	90.03	83.44	89.76	84.55	90.80
SMixed→T87	87.95	87.40	73.96	68.37	87.47	88.09	88.68	80.03	88.30	83.51	**89.58**
SMixed→T82	85.17	85.28	64.31	75.59	84.24	85.17	85.03	77.15	86.08	81.84	**86.60**
SMixed→T77	82.36	**83.51**	59.90	74.58	79.58	82.92	82.99	72.74	81.46	78.65	81.63
SMixed→T72	80.69	**82.57**	54.24	75.59	73.37	80.59	82.29	68.30	81.39	79.58	82.12
Ave.	85.83	86.12	69.50	75.42	84.57	85.93	86.79	77.92	86.52	82.62	**87.34**

5. Conclusion

In this paper, a novel iterative algorithm JSM has been introduced to identify the source camera model of recompressed images. JSM aims to learn a new feature representation from original features by simultaneously matching the first and second order statistics between training and test features. Comprehensive experiments on synthetic data and Dresden Image Database demonstrate the effectiveness of JSM and show that JSM significantly outperforms several state-of-the-art methods when identifying the recompressed images. Our future work includes further improving identification performance of recompressed images by analyzing the intrinsic characteristics and difference among these recompressed images.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 61502076, No. 61772111).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. M. C. Stamm, M. Wu and K. J. R. Liu, Information forensics: An overview of the first decade, *IEEE Access*, **1** (2013), 167–200.
2. A. Piva, An overview on image forensics, *Isrn Signal Process.*, **2013**.
3. M. Kirchner and T. Gloe, Forensic camera model identification, in *Handbook of Digital Forensics of Multimedia Data and Devices*, Chichester, U.K.: Wiley, 2015, 329–374.
4. J. Lukáš, J. Fridrich and M. Goljan, Determining digital image origin using sensor imperfections, in *Proc. SPIE*, San Jose, CA, USA, 2005, 249–260.
5. M. Chen, J. J. Fridrich and M. Goljan, Digital imaging sensor identification (further study)., in *Proc. SPIE*, San Jose, CA, USA, 2007, 65050P–65050P13.
6. C. T. Li, Source camera identification using enhanced sensor pattern noise, *IEEE Trans. Inf. Foren. Secur.*, **5** (2010), 280–287.
7. X. Kang, Y. Li, Z. Qu, et al., Enhanced source camera identification performance with a camera reference phase sensor, *IEEE Trans. Inf. Foren. Secur.*, **7** (2012), 393–402.
8. A. Lawgaly, F. Khelifi and A. Bouridane, Weighted averaging-based sensor pattern noise estimation for source camera identification, in *Proc. IEEE Int. Conf. Image Process.*, Paris, France, 2014, 5357–5361.
9. A. Lawgaly and F. Khelifi, Sensor pattern noise estimation based on improved locally adaptive dct filtering and weighted averaging for source camera identification and verification, *IEEE Trans. Inf. Foren. Secur.*, **12** (2017), 392–404.

10. Y. Hu, C. T. Li and Z. Lai, Fast source camera identification using matching signs between query and reference fingerprints, *Mult. Tool. Appl.*, **74** (2015), 7405–7428.
11. S. Bayram, H. T. Sencar and N. Memon, Sensor fingerprint identification through composite fingerprints and group testing, *IEEE Trans. Inf. Foren. Secur.*, **10** (2015), 597–612.
12. D. Valsesia, G. Coluccia, T. Bianchi, et al., Compressed fingerprint matching and camera identification via random projections, *IEEE Trans. Inf. Foren. Secur.*, **10** (2015), 1472–1485.
13. R. Li, C. T. Li and Y. Guan, Inference of a compact representation of sensor fingerprint for source camera identification, *Patt. Recogn.*, **74** (2018), 556 – 567.
14. Y. Hu, C. T. Li, C. Zhou, et al., Source camera identification issues: forensic features selection and robustness, *Int. J. Digit. Crime Foren.*, **3** (2011), 1–15.
15. M. Kharrazi, H. T. Sencar and N. Memon, Blind source camera identification, in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Singapore, 2004, 709–712.
16. K. San Choi, E. Y. Lam and K. K. Wong, Source camera identification by jpeg compression statistics for image forensics, in *Proc. IEEE Region 10 Conf.*, Hong Kong, China, 2006, 1–4.
17. B. Wang, Y. Guo, X. Kong, et al., Source camera identification forensics based on wavelet features, in *Proc. IEEE Fifth Int. Conf. Intelligent Inf. Hiding and Multimedia Signal Process.*, Kyoto, Japan, 2009, 702–705.
18. S. Bayram, H. Sencar, N. Memon, et al., Source camera identification based on cfa interpolation, in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Genova, Italy, 2005, 69–72.
19. J. S. Ho, O. C. Au, J. Zhou, et al., Inter-channel demosaicking traces for digital image forensics, in *Proc. IEEE Int. Conf. Multimedia Expo*, Suntec, Singapore, 2010, 1475–1480.
20. Y. Hu, C. T. Li, X. Lin, et al., An improved algorithm for camera model identification using inter-channel demosaicking traces, in *8th Int. Conf. Intelligent Inform. Hiding and Multimedia Signal Process.*, Piraeus, Greece, 2012, 325–330.
21. G. Xu and Y. Q. Shi, Camera model identification using local binary patterns, in *Proc. IEEE Int. Conf. Multimedia Expo*, Melbourne, VIC, Australia, 2012, 392–397.
22. B. Xu, X. Wang, X. Zhou, et al., Source camera identification from image texture features, *Neurocomputing*, **207** (2016), 131–140.
23. C. Chen and M. C. Stamm, Camera model identification framework using an ensemble of demosaicing features, in *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, 1–6.
24. A. Tuama, F. Comby and M. Chaumont, Camera model identification based machine learning approach with high order statistics features, in *Proc. 24th European Signal Processing Conference (EUSIPCO)*, 2016, 1183–1187.
25. A. Roy, R. S. Chakraborty, U. Sameer, et al., Camera source identification using discrete cosine transform residue features and ensemble classifier, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.
26. L. Zeng, X. Kong, M. Li, et al., Jpeg quantization table mismatched steganalysis via robust discriminative feature transformation., in *Proc. SPIE Media Watermarking, Security, and Forensics*, San Francisco, CA, USA, 2015, 94090U–94090U9.

27. S. J. Pan, I. W. Tsang, J. T. Kwok, et al., Domain adaptation via transfer component analysis, *IEEE Trans. Neural Netw.*, **22** (2011), 199–210.

28. M. Long, J. Wang, G. Ding, et al., Transfer feature learning with joint distribution adaptation, in *Proc. IEEE Int. Conf. Comput. Vision*, San Francisco, CA, USA, 2013, 2200–2207.

29. X. Li, X. Kong, B. Wang, et al., Generalized transfer component analysis for mismatched jpeg steganalysis, in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, 2013, 4432–4436.

30. L. Luo, X. Wang, S. Hu, et al., Close yet distinctive domain adaptation, *arXiv:1704.04235*.

31. G. Zhang, B. Wang and Y. Li, Cross-class and inter-class alignment based camera source identification for re-compression images, in *Proc. IEEE Int. Conf. Image Graphics*, Shanghai, China, 2017.

32. A. Gretton, K. M. Borgwardt, M. Rasch, et al., A kernel method for the two-sample-problem, in *Proc. Advances Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2007, 513–520.

33. B. Quanz, J. Huan and M. Mishra, Knowledge transfer with low-quality data: A feature extraction issue, *IEEE Trans. Knowl. Data Eng.*, **24** (2012), 1789–1802.

34. E. Zhong, W. Fan, J. Peng, et al., Cross domain distribution adaptation via kernel mapping, in *Proc. 15th ACM SIGKDD Int. Conf. on Knowl. Discovery Data Mining*, Paris, France, 2009, 1027–1036.

35. T. Gloe and R. Böhme, The dresden image database for benchmarking digital image forensics, *J. Digit. Forensic Pract.*, **3** (2010), 150–159.

AIMS Press