Camera Model Identification Based on Local Co-Occurrence Features

Parv Parkhiya, Navyasri Reddy, Akshay Vyas, Vivek Jain, Diplav Srivastava, V. Lalitha
IIIT Hyderabad, India
Email: parvparkhiya@gmail.com, navyasrireddy1996@gmail.com,
manu4vyas@gmail.com, vivek13jain@gmail.com,
srivastavadiplav@outlook.com, lalitha.v@iiit.ac.in

***Abstract*—We apply the algorithm proposed by Chen etal to identify camera models. The algorithm assumes that CFA pattern used by the device is GBRG. Local co-occurrence features are computed using multiple interpolation algorithms (example nearest neighbour, bilinear). A multi-class linear SVM is trained with these features and employed to classify the given image to one of the camera classes. Some observations have been made with respect to validation accuracy of the model and the results obtained on Kaggle.**

I. Introduction

The 2018 SP Cup challenge poses the problem of forensic camera model identification. The problem lies at the intersection of machine learning, image processing and signal processing techniques. Information about which type of camera captured an image can be used to help determine or verify the origin of an image, and can form an important piece of evidence in forensics. Before describing the algorithm which we employ for camera model identification, we will quickly describe the image acquisition pipeline.

A. Image Acquisition Pipeline

The light from the scene is focused on the sensor array captured from a single lens or a combination of lenses. In a few cameras, the lens is followed by an interposed optical filter which reduce the unwanted light noise (for instance the infrared filter). The sensor is nowadays a CMOS sensor or a CCD sensor, which converts the optical light to electrical signal. There is a Color Filter Array (CFA) just before the sensor. The CFA allows only one color of light to pass through at each position before reaching the sensor. The most commonly used CFA patterns are the Bayer CFA patterns. The Bayer's CFA pattern has 4 patterns, each CFA is a matrix of 2×2 pixels and has two green filters in diagonal locations, one red filter and one blue filter in the off-diagonal locations (BGGR, RGBG, GRGB, or RGGB).

The image after CFA has only one color per pixel, but every pixel of image must have components of all the three colors. The reconstruction of complete image from the color filter array is called demosaicing. Generally CFA pattern is unique to the make of the device and it is not instance specific. Demosaicing algorithm is also one of the most important signatures of camera and is unique to manufacturer. However, it can be instant specific too. The features based on demosaicing are widely in literature to identify camera model [1], [2].

II. Local Co-occurrence Feature Extraction

We have used an algorithm proposed by Chen etal in [1], which classifies the image taken from a camera to model based classes. The algorithm assumes that the CFA pattern of the device is the most common Bayer pattern which is (GBRG). Consider an RGB image I of size $m \times n$. We will refer to the red color value at location (i, j) as $R(i, j)$. Given an RGB image, we resample the image I to result in an image I' as follows:

- At a given location (i, j) in the image, if green color filter is used, then the sample value $G(i, j)$ is the retained in the resampled image I', i.e., $G'(i, j) = G(i, j)$. The values of $B(i, j)$ and $R(i, j)$ are discarded and treated as missing values. Similar operation is repeated at all locations (i, j) of the image.
- After the re-sampling values, the missing values in the resultant image I' are interpolated from the neighbouring pixel values using one of the following interpolation techniques: nearest neighbour interpolation, bilinear interpolation.

The following difference images are computed between the orginial image I and the interpolated image I'.

$$\begin{aligned} E_R(i,j) &= R(i,j) - R'(i,j) \\ E_G(i,j) &= G(i,j) - G'(i,j) \\ E_B(i,j) &= B(i,j) - B'(i,j). \end{aligned}$$

In principle, the values of E_R, E_G and E_G can have high resolution and high dynamic range. The resolution is reduced by quantizing the values using the following expression:

$$E_{Rq}(i,j) = \text{round}\left(\frac{E_R(i,j)}{Q}\right).$$

The range of the values in the difference images can be confined to an interval $[-T, T]$ by using the following mapping:

$$E_{Rq}(i,j) = \begin{cases} E_{Rq}(i,j), & -T \leq E_{Rq}(i,j) \leq T \\ -T, & E_{Rq}(i,j) < -T \\ T, & E_{Rq}(i,j) > T. \end{cases}$$

The above procedure is repeated for the green and blue channels as well. The local co-occurrences of the residual images have been shown to be important features in identifying camera class in [1]. Let N_t denote the number of 2×2 blocks in the given image. The local-occurrences can be obtained by considering intra-channel and inter-channel correlations. Intra-channel co-occurrences are described below:

- **Red Channel Co-occurrence:** The residual image $E_{Rq}(i,j)$ is considered for calculating the red channel co-occurrence. The complete residual image $E_{Rq}(i,j)$ is divided into 2×2 blocks. The co-occurences are calculated by counting the number of occurrences of the 3-tuple (d_1, d_2, d_3) in $E_{Rq}(i,j)$ $(-T \leq d_i \leq T)$ in the configuration shown below:

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline d_1 & d_2 \\ \hline R & d_3 \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline \ & B \\ \hline \ & \ \\ \hline\end{array}$$

  The number of occurrences of the 3-tuple (d_1, d_2, d_3) in $E_{Rq}(i,j)$ is denoted by $N_R(d_1, d_2, d_3)$. The red channel co-occurrence is given by the following expression:

$$C_R(d_1, d_2, d_3) = \frac{N_R(d_1, d_2, d_3)}{N_t}, \quad -T \leq d_k \leq T.$$

- **Blue Channel Co-occurrence:** The blue-channel co-occurrences are calculated by counting the number of occurrences of the 3-tuple (d_1, d_2, d_3) in $E_{Bq}(i,j)$ $(-T \leq d_i \leq T)$ in the configuration shown below:

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline \ & \ \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline d_1 & B \\ \hline d_2 & d_3 \\ \hline\end{array}$$

  The number of occurrences of the 3-tuple (d_1, d_2, d_3) in $E_{Rq}(i,j)$ is denoted by $N_R(d_1, d_2, d_3)$. The blue channel co-occurrence is given by the following expression:

$$C_B(d_1, d_2, d_3) = \frac{N_B(d_1, d_2, d_3)}{N_t}, \quad -T \leq d_k \leq T.$$

- **Green Channel Co-occurrence:**

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline \ & \ \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & d_1 \\ \hline d_2 & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline \ & B \\ \hline \ & \ \\ \hline\end{array}$$

  The number of occurrences of the 2-tuple (d_1, d_2) in $E_{Gq}(i,j)$ is denoted by $N_G(d_1, d_2)$. The green channel co-occurrence is given by the following expression:

$$C_G(d_1, d_2) = \frac{N_G(d_1, d_2)}{N_t}, \quad -T \leq d_k \leq T.$$

Note that due to the structure of the Bayer CFA pattern, blue and red channel co-occurrences are three-dimensional whereas green channel co-occurrence is two-dimensional.

Now, we compute inter-channel co-occurrences. We consider at most three-dimensional inter-channel co-occurrences for the sake of keeping the feature space low. Also, only those co-occurences are considered which do not subsume the intra-channel co-occurrences described above. Inter-channel co-occurrences considering two channels at a time are described below:

- **Red-Green Channel Co-occurrence:** The co-occurrences are calculated by counting the number of occurrences of the 3-tuple (d_1, d_2, d_3) jointly in $E_{Rq}(i,j)$ and $E_{Gq}(i,j)$ $(-T \leq d_k \leq T)$ in the configurations shown below:

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline d_1 & d_2 \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & d_3 \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline \ & B \\ \hline \ & \ \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline \ & d_2 \\ \hline R & d_1 \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & d_3 \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline \ & B \\ \hline \ & \ \\ \hline\end{array}$$

  The number of occurrences of Type-1 (first configuration) is denoted by $N_{RG}^{(1)}(d_1, d_2, d_3)$ and that of Type-2 (second configuration) is denoted by $N_{RG}^{(2)}(d_1, d_2, d_3)$. The red-green channel co-occurrence is given by the following expression:

$$C_{RG}(d_1, d_2, d_3) = \frac{N_{RG}^{(1)}(d_1, d_2, d_3) + N_{RG}^{(2)}(d_1, d_2, d_3)}{N_t}.$$

- **Blue-Green Channel Co-occurrence:**

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline \ & \ \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline d_3 & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline \ & B \\ \hline d_1 & d_2 \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline \ & \ \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline d_3 & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline d_2 & B \\ \hline d_1 & \ \\ \hline\end{array}$$

  The number of occurrences of Type-1 (first configuration) is denoted by $N_{BG}^{(1)}(d_1, d_2, d_3)$ and that of Type-2 (second configuration) is denoted by $N_{BG}^{(2)}(d_1, d_2, d_3)$. The blue-green channel co-occurrence is given by the following expression:

$$C_{BG}(d_1, d_2, d_3) = \frac{N_{BG}^{(1)}(d_1, d_2, d_3) + N_{BG}^{(2)}(d_1, d_2, d_3)}{N_t}.$$

- **Red-Blue Channel Co-occurrence:**

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline d_1 & d_3 \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline d_2 & B \\ \hline \ & \ \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline G & B \\ \hline R & G \\ \hline\end{array} = \begin{array}{|c|c|}\hline d_1 & \ \\ \hline R & \ \\ \hline\end{array} + \begin{array}{|c|c|}\hline G & \ \\ \hline \ & G \\ \hline\end{array} + \begin{array}{|c|c|}\hline d_2 & B \\ \hline d_3 & \ \\ \hline\end{array}$$

  The red-blue channel co-occurrence is given by the following expression:

$$C_{RB}(d_1, d_2, d_3) = \frac{N_{RB}^{(1)}(d_1, d_2, d_3) + N_{RB}^{(2)}(d_1, d_2, d_3)}{N_t}.$$

Note that the values of intra-channel co-occurrences are at most 1 and those of inter-channel co-occurrences are at most 2. Out of all the intra and inter-channel co-occurrences described above, only some of them are used as features for camera model identification.

Feature Set For Camera Model Identification: The following co-occurrences [1] are computed as features:

- Red Channel Co-occurrence
- Red-Green Channel Co-occurrence

The number of features is given by $2(2T+1)^3$.

III. Camera Model Identification Based on Extracted Features

Database: The database for training the classifier was posted by organisers of SPCup Challenge. The database was constructed using 10 camera models are given in Table I.

The database consists of 275 images of each camera class. We crop each image into multiple sub-images of size 512×512. For each of these 512×512 sub-images, we

Number	Class
1	HTC-1-M7
2	iPhone-4s
3	iPhone-6
4	LG-Nexus-5x
5	Motorola-Droid-Maxx
6	Motorola-Nexus-6
7	Motorola-X
8	Samsung-Galaxy-Note3
9	Samsung-Galaxy-S4
10	Sony-NEX-7

TABLE I
CAMERA MODELS USED IN THE CHALLENGE

calculate the $2(2T + 1)^3$ features using each of nearest neighbour interpolation and bilinear interpolation. We would like to note that since we are taking ratios for calculating co-occurrences, we can potentially extract the features from varying sub-image sizes. However, we used 512×512 size for obtaining sub-images and calculating features. We trained a multi-class one-vs-one linear SVM classifier with 75% sub-images from the database and validate on the other 25% sub-images.

A. Observations

- The value of C required for SVM training for good accuracy turned out to be very high (See Fig. 2). C is a regularisation parameter of SVM which controls the tolerance to misclassification in the training data.
- For lower values of $C \leq 20$, the accuracy of class 1 is much lower than those of the other classes (See Fig. 1). One possible reason could be that the camera class 1 may not be using CFA pattern GBRG. This is however a guess as the ground truth of the CFA patterns corresponding to the cameras is not available. The sensitivity of the algorithm to the assumption that the CFA pattern is GBRG has not been studied and it might be an interesting problem to study.
- We have used the model obtained using $C = 25000$ in Kaggle and we obtained a weighted score of 60.1% on public leaderboard. This indicates the model is overfitting because if that were not the case then the accuracy should have been close to 70% or higher. This is because the weighted accuracy computed is $0.7\times$accuracy of unmanipulated images$+0.3\times$ accuracy of manipulated images. However, our accuracy of 60.1% reflects the fact that the model is overfitting and hence not performing as good on the test data. One of the reasons again for these misclassifications could be the assumption of CFA pattern.
- It has been pointed out in [2] that the models obtained based on co-occurrence features perform poorly on manipulated images, where the manipulation is either JPEG compression or image resizing. We have done a similar experiment for the case of gamma correction and the results are summarised in Fig. 3. It can be seen that the accuracy of the model when gamma correction is applied is around 90%. We can infer that the models trained from unmanipulated images are more robust to gamma correction than to JPEG compression and image resizing.

REFERENCES

[1] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, 2015, pp. 1-6.
[2] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "A study of co-occurrence based local features for camera model identification." Multimedia Tools and Applications (2016): 1- 17.

Class	1	2	3	4	5	6	7	8	9	10
	76	0	1	1	2	0	1	0	0	0
	0	**99**	0	0	0	0	0	0	0	0
	1	0	**97**	0	0	0	0	0	0	0
	3	0	0	**97**	2	0	3	0	0	0
	4	0	0	0	**90**	0	1	1	2	0
	0	0	0	0	0	**97**	1	0	0	0
	11	0	1	1	3	1	**92**	2	1	0
	3	0	0	0	1	0	1	**96**	0	0
	2	0	0	1	1	1	1	1	**97**	1
	0	0	0	0	0	0	0	0	0	**99**

Overall Accuracy: 95.38%

Fig. 1. Confusion matrix for validation accuracy of Linear SVM with $C = 20$.

Class	1	2	3	4	5	6	7	8	9	10
	97.0	0.0	0.0	0.1	0.4	0.0	0.2	0.0	0.1	0.0
	0.0	**99.6**	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.1	0.1	**99.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.4	0.0	0.0	**99.1**	0.2	0.0	0.2	0.0	0.0	0.1
	1.6	0.0	0.0	0.1	**98.7**	0.0	0.2	0.0	0.0	0.0
	0.2	0.2	0.0	0.1	0.2	**99.8**	0.4	0.5	0.5	0.4
	0.8	0.0	0.0	0.5	0.5	0.0	**98.8**	0.1	0.2	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.1	**99.2**	0.0	0.0
	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.1	**99.2**	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**99.5**

Overall Accuracy: 99.19198176

Fig. 2. Confusion Matrix for validation accuracy of Linear SVM with $C = 25000$.

Class	1	2	3	4	5	6	7	8	9	10
	83.6	0.0	0.6	0.4	0.6	0.2	1.0	0.0	0.2	0.0
	0.0	**99.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.2	0.4	**98.4**	1.4	0.0	0.4	0.0	0.0	0.0	0.0
	1.2	0.0	0.0	**62.6**	0.6	0.6	2.2	1.4	0.2	0.0
	5.8	0.0	0.0	4.2	**91.0**	0.2	6.4	0.2	1.4	0.0
	0.0	0.0	0.0	0.0	0.6	**95.2**	0.4	1.0	0.4	4.6
	9.2	0.0	1.0	30.0	6.6	0.6	**90.0**	0.6	0.8	0.2
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**96.4**	0.0	0.0
	0.0	0.0	0.0	0.6	0.6	2.8	0.0	0.4	**97.0**	0.2
	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	**95.0**

Overall Accuracy: 90.88

Fig. 3. Confusion Matrix for Linear SVM trained on unmanipulated images and tested on gamma corrected images.