IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier*

# Provenance Inference for Instagram Photos through Device Fingerprinting

**YIJUN QUAN[1], (Student Member, IEEE), XUFENG LIN[2], AND CHANG-TSUN LI.[2], (Senior Member, IEEE)**
[1]Department of Computer Science, University of Warwick, Coventry, CV4 7AL United Kingdom (e-mail: Y.Quan@warwick.ac.uk)
[2]School of Information Technology, Deakin University, Geelong VIC 3216 Australia (e-mail: {xufeng.lin,changtsun.li}@deakin.edu.au)

Corresponding author: Yijun Quan (e-mail: Y.Quan@warwick.ac.uk).

This work is supported by the EU Horizon 2020 Marie Sklodowska-Curie Actions through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY)

## ABSTRACT

Sensor pattern noise (SPN) has been extensively studied in the scientific community and has found its applications in many practical scenarios in the law-enforcement sector. However, the emergence of photo-sharing social networking sites (SNSs) poses new challenges to SPN-based digital image provenance analysis. One particular issue is that the SNSs' built-in image editing tools tend to inflict distortion on SPNs. One well-known example of such tools is the image filters on Instagram. We observed that some Instagram image filters manipulate the high-frequency bands of the images and hence damage the SPNs, making source-oriented clustering (SOC) of the filtered images unsatisfactory. To address this issue, we propose to first separate the images processed by different filters beforehand into two groups, with Group Malignant (M) containing the filters that significantly distort SPNs and Group Benign (B) covering the other filters that have no significant impact on SPNs. We then cluster the images processed by Group B filters and calculate the centroid of each cluster, with one centroid representing the reference SPN of the corresponding camera. Finally, we use the centroid of each cluster to attract the images processed by the Group M filters in order to complete the SOC task. To identify the filter applied to each image so as to facilitate the clustering, a convolutional neural network based filter-oriented image classifier is proposed. Tested on 19,332 images processed by 18 different filters, the classifier delivers a very promising accuracy of 98.5%. Moreover, compared to the F1-measure of 47.74% by directly clustering on 1,800 filtered images, our proposed clustering framework achieves a much higher F1-measure of 90.33%.

**INDEX TERMS** sensor pattern noise, image clustering, social network, digital forensics, provenance analysis.

## I. INTRODUCTION

With the rapid development of mobile networks and the ever-increasing prevalence of smartphones, photo-sharing social networking sites (SNSs), such as Instagram, Facebook and Flickr, have become ubiquitous in our daily life. With millions of daily active users, these SNSs not only provide effective platforms for information sharing but also exert huge influence on commerce and politics. However, due to the convenient and broad reach of these platforms, they have been increasingly exploited for various malicious purposes, e.g. fraudulent advertisement, fictitious news or even terrorism. Meanwhile, the sheer volume of user-generated content on these platforms provides a rich source of evidence acquisition for forensic investigations. Thus, in recent years there have been growing interests in developing forensic tools and techniques to facilitate the investigations on the data collected from SNSs. One important related topic is the provenance analysis of images from SNSs. The provenance information of digital images is essential for forensic investigations. For example, when a forensic investigator is dealing with a set of images of unknown sources from multiple social network accounts, revealing the source devices of the images can help the investigator to focus on the images from the same source. In addition, linked and fake social network accounts can be discovered by finding images from the same source device across different accounts. This is because different accounts with photos taken with the camera are likely to be closely linked (e.g., between family members or friends) or fake accounts used in sybil attacks. With these telltale provenance information, more effective investigations can then be carried out. Though occasionally, one may use the metadata of an image to retrieve its provenance information, these information can still be questionable as the metadata could be edited easily. Moreover, many SNSs deliberately delete metadata from the images when they are uploaded. The unavailability of the provenance information may entail content-based analyses when rigorous forensic investigations are required. Many optical and camera artifacts left in images during in-camera processing or post-processing (e.g. lens aberration [1], [2], CFA interpolation [3], JPEG compression [4], [5], machine-extracted features [6], [7] and etc.), have been used to attribute the source devices.

Among various techniques used for analyzing images' provenance, the ones based on sensor pattern noise (SPN) [8] have drawn extensive attention from researchers. As its name indicates, SPN is a unique noise-like pattern introduced into the image content by an imaging device's sensor and can be used as the fingerprint of the device. SPN has been proved to be a powerful tool for provenance analysis, such as source camera identification (SCI) [9]–[13], source-oriented clustering (SOC) [14]–[22] or forgery detection [23], [24] . Almost all the above-mentioned methods were only evaluated on images straight from cameras without undergoing any post-processing except mild JPEG compression. However, there have been increasing doubts cast on the effectiveness of SPN-based methods on images from SNSs, where users can apply various built-in photo-editing tools, e.g. 'Filters' on Instagram and 'Effects' on Facebook, to the images before posting them online. SNSs users can easily manipulate the images by selecting their desired visual styles with just a few finger taps. Such manipulations may contaminate or attenuate the SPN embedded in the images, which may potentially compromise its effectiveness for forensic investigations. Thus, in this work, we aim to investigate how these built-in editing tools may affect the quality of SPN. We are particularly interested in the effects of image filters on Instagram as it is one of the most popular photo-sharing platforms. Our recent work in [25] has preliminarily shown that some Instagram image filters are particularly harmful to SPN-based clustering. In this work, we conduct further investigation on images from Instagram to gain a better understanding of how SPN-based clustering methods are affected by these filters. As we shall see later in this work, the investigation shows that though the devices' SPNs may survive the impact by the filters, the artifacts introduced by some filters can lead to significant performance deterioration in SOC. To address this problem, we propose a three-step clustering framework by segregating the images based on the filters applied to them. The proposed method can significantly improve the clustering performance on images filtered by different filters and provide us with a feasible solution to perform SOC on images from social network sites.

The rest of this work is organized as follows. An introduction to the background and related work is given in Section II. Section III shows the preliminary test of the existing SPN-based source camera identification and clustering methods on images from Instagram. The proposed three-step clustering method is shown in Section IV. Section V presents the experimental results while Section VI draws the conclusion.

## II. BACKGROUND AND RELATED WORKS

SPN is the fixed noise pattern in an image introduced by the imaging sensor. Despite that different sources may contribute to SPN, the most dominant component is the photo response non-uniformity (PRNU) noise, which arises due to the non-uniform pixel sensitivities to the incident photons. Such non-uniformity is inevitable due to the inhomogeneity of silicon wafers during sensor manufacturing process. Besides, such pixel-to-pixel discrepancy makes SPN unique to its sensor. Thus, it becomes a feasible choice for source camera fingerprinting. Typically, the SPN is approximated as the noise residual $\boldsymbol{n}$, which can be extracted by subtracting the original image $\boldsymbol{I}$ from its de-noised version $\hat{\boldsymbol{I}}$:

$$\boldsymbol{n} = \boldsymbol{I} - \hat{\boldsymbol{I}} \tag{1}$$

For SCI, its goal is to identify the source camera of the image in question among a number of candidate cameras. To serve this purpose, the normalized cross-correlation is usually used to measure the similarity between the noise residual $\boldsymbol{n}$ of the image and a set of reference SPNs $\{\boldsymbol{r}_i\}_{i=1}^{K}$ of $K$ candidate cameras:

$$\rho_i = \text{corr}(\boldsymbol{n}, \boldsymbol{r}_i) = \frac{(\boldsymbol{n} - \bar{\boldsymbol{n}}) \cdot (\boldsymbol{r}_i - \bar{\boldsymbol{r}_i})}{\|\boldsymbol{n} - \bar{\boldsymbol{n}}\| \|\boldsymbol{r}_i - \bar{\boldsymbol{r}_i}\|}, \tag{2}$$

where the reference SPN $\boldsymbol{r}_i$ is constructed by averaging the noise residuals extracted from another set of images (usually blue-sky or flat-field images) taken by camera $i$. The image under investigation is deemed to be from the camera corresponding to the highest similarity that exceeds a predefined threshold.

Generally speaking, SCI is a relatively easy task provided that the high-quality reference SPNs are available. In comparison, it is more challenging for SOC, where we aim to group a set of images of unknown sources into a number of clusters, such that the images in the same cluster are taken by the same camera. For this task, we often face the challenges of an unknown number of source devices and low-quality of the SPNs extracted from single images. Many techniques or combinations of them have been proposed for SPN-based SOC following the early work from [14], including the methods based on hierarchical clustering [15], [26], graph-based approaches [16]–[19], constraint optimization [20] and Markov random field [21]. However, due to the unavailability of the reference SPNs, these algorithms have to rely on the pairwise correlations between individual noise residuals, which are more susceptible to SPN-irrelevant interferences, especially for the images from SNSs that may have undergone a series of post-processing operations. This raises doubts about whether SPN-based provenance analysis

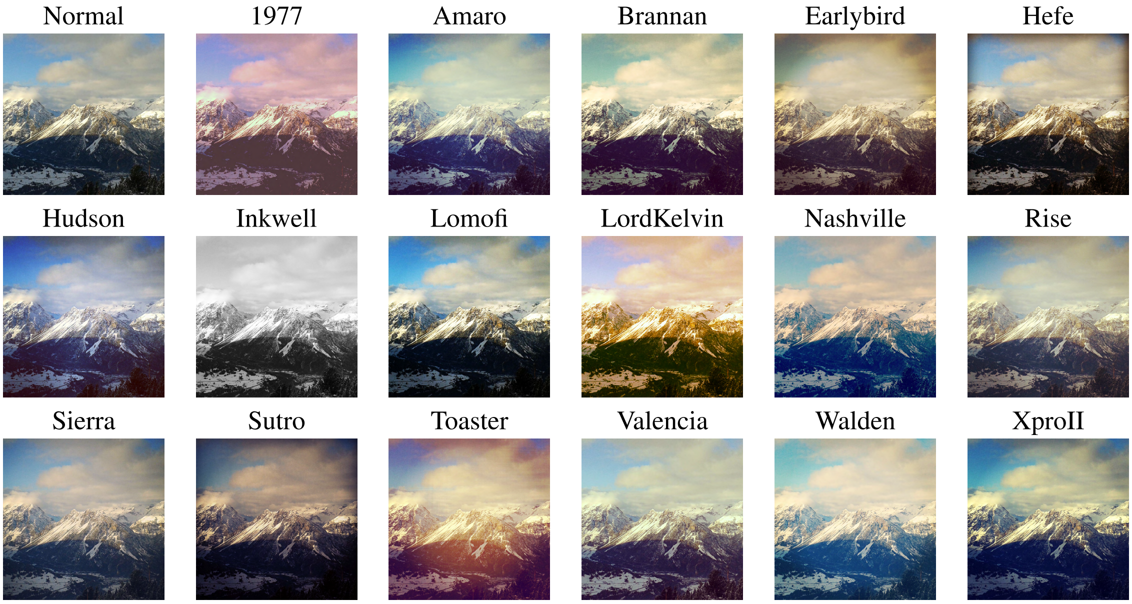


FIGURE 1: Example images of the 17 Instagram filters together with the original image (Normal) used in our experiment [25].

methods remain effective on images from social network sites.

Goljan *et al.* [27] perform a large-scale test of SPN-based camera identification on images downloaded from Flicker and show very promising results with a small false rejection rate $<0.0238$ at a false acceptance rate $<2.4\times10^{-5}$ for 6896 cameras with 150 different camera models. However, comparing to other social networking platforms, Flickr allows the uploaded images to be stored in their original resolution with no or very little compression, so it does not fully reflect the difficulty of the problem we usually face when performing image provenance analysis on other SNSs. Satta and Stirparo [28] use SPNs to build the link between a photo and the user accounts of the person that has shot the photo. A probe photo is considered to be from the account containing the image with the highest matching score to the probe photo. Their method achieves a recognition rate of $\sim 50\%$ by evaluating 2896 images from 30 different accounts across different SNSs, namely Flickr, Facebook, Google+ and personal blogs. The low recognition rate and the lack of in-depth investigation into the effect of image operations make it necessary to conduct further studies on the SPN-based provenance analysis of images from SNSs.

More recent work [29], [30] discover that different SNSs may apply different image manipulations, which leave distinctive artifacts that can be used to trace the origin SNSs of the images. Moreover, they show how common it is for the SNSs to apply 'hidden' image manipulations, such as resizing and re-compression, to fulfill the system requirement, which may affect the SPN and pose challenges to SPN-based provenance analysis. Apart from the above-mentioned image manipulations, many SNSs also provide explicit image manipulation tools to allow the users to edit image effects according to their own preferences, with the 'Filters' from Instagram being the most famous example. While these tools enrich the user experience, they may also manipulate the images in a way that may make the SPN-based provenance analysis method ineffective. As a preliminary investigation in [25], we found some image filters of Instagram may cause significant performance drop of existing SPN-based SOC methods. In this work, we will further investigate the effects of Instagram filters and propose a new method to mitigate the impact of image filtering.

To carry out this work, we prepared a dataset $\mathscr{D}$ with a large number of images of known sources and applied different image filters to them. We selected $5,370$ images captured by $25$ cameras, with at least $137$ images from each camera, from the VISION image dataset [31]. The images are aligned to the same horizontal orientation according to their EXIF data and cropped to the size of $1080\times1080\times3$ pixels to match the default image size on Instagram. For each image, we applied $17$ different Instagram image filters by running the Instagram application on an iOS simulator. Thus, together with the original version, we generated $18$ different versions of each image and in total $96,660$ images for the use in our work. Fig. $1$ shows a sample image for each filter together with the original image (labelled as 'Normal' filter as it is termed on Instagram). In addition, we also processed images from Dresden Image Dataset [37] and form various subsets of $\mathscr{D}$ to carry out tests on different aspects of device fingerprint based provenance analysis and our proposed framework. An overview of these datasets are shown in Table 1.

## III. EXISTING SPN-BASED PROVENANCE ANALYSIS ON INSTAGRAM IMAGES

TABLE 1: An overview of different datasets used for different parts of the work with information including the source of the original images. $\mathscr{D}$, which is derived from VISION dataset, is the main dataset used in this work, including the training and testing of the proposed CNN-based filter classifier in Section V-A. $\mathscr{D}_{\text{SCI}}$ is a subset of $\mathscr{D}$, which is used to test device fingerprint based SCI in Section III-A. $\mathscr{D}_{\text{Dresden}}$ is derived from Dresden Image Database and used to show the proposed CNN-based filter classifier is not outfitted t the training cameras in Section V-A. $\mathscr{D}_2$, $\mathscr{D}_3$, $\mathscr{D}_4$, $\mathscr{D}_5$, $\mathscr{D}_6$ are subsets of $\mathscr{D}$ with different sizes, used to test proposed clustering framework in Section V-C

| Dataset | Source | No. of Devices | No. of Images | | |
|---|---|---|---|---|---|
| | | | total | per device | per filter |
| $\mathscr{D}$ | VISION [31] | 25 | 96,660 | $> 2466$ | 5370 |
| $\mathscr{D}_{\text{SCI}}$ | VISION [31] | 25 | 22,500 | 900 | 1250 |
| $\mathscr{D}_{\text{Dresden}}$ | Dresden [37] | 11 | 29,700 | 2700 | 1650 |
| $\mathscr{D}_2$ | VISION [31] | 25 | 900 | 36 | 50 |
| $\mathscr{D}_3$ | VISION [31] | 25 | 1,350 | 54 | 75 |
| $\mathscr{D}_4$ | VISION [31] | 25 | 1,800 | 72 | 100 |
| $\mathscr{D}_5$ | VISION [31] | 25 | 2,250 | 90 | 125 |
| $\mathscr{D}_6$ | VISION [31] | 25 | 2,700 | 108 | 150 |

### A. SPN-BASED SCI FOR INSTAGRAM IMAGES

In this section, we investigate the effect of different Instagram image filters on the task of SPN-based SCI. Specifically, we perform SCI by examining the correlations between the noise residuals extracted from the *filtered* images with the reference SPNs, each of which is estimated from 50 flat-field images taken by the same camera. Note that these flat-field images are original images to ensure the high quality of reference SPNs. Thus, the performance of the source camera identification task can serve as the baseline for the quality of the SPN embedded in the Instagram images. BM3D denoising algorithm [32] is used to extract the noise residual for each image. For the reference SPN of each camera, its correlations with $21,150$ inter-class *original* images (i.e. the ones from different source cameras) are computed to estimate the inter-class correlation distribution. Then we determine a decision threshold $\{\tau_i\}_{i=1}^{25}$ for each camera according to the corresponding inter-class correlation distribution based on the Neyman-Pearson criterion (by setting the false positive rate as $1{\times}10^{-3}$). We formed a testing dataset $\mathscr{D}_{\text{SCI}}$ with 50 test images $\{\boldsymbol{I}_l^{ij}\}_{l=1}^{50}$ for each camera $i$ processed by each filter $j$ randomly selected from $\mathscr{D}$. For each test image $\boldsymbol{I}_l^{ij}$, the largest correlation $\rho_{i^\star}$ among the correlations $\{\rho_i\}_{i=1}^{25}$ between its noise residual $\boldsymbol{n}_l^{ij}$ and the reference SPNs $\{\boldsymbol{r}_i\}_{i=1}^{25}$ of candidate cameras is compared with the pre-defined threshold $\tau_{i^\star}$ to examine whether the image is from the camera $i^*$ or from an unknown source. The accuracy of the SCI for each filter is shown in Table 2. In addition, to explicitly demonstrate the quality of SPN embedded in the filtered images, we select one camera (an iPhone4s) and plot the intra-class correlation distributions between the test images with the reference SPN for different filters in Fig. 2, where we use central points and error bars to represent the means and standard deviations of the distributions, respectively.

Table 2 shows that for each filter, the identification accuracy for the images processed by the same filter is comparable to that for the 'Normal' images. This is no surprise when we look at the correlation distribution plot in Fig. 2. As different devices show similar behaviour, we use an iPhone4s as an example. First, we notice that different Instagram image filters have almost no impact on the inter-class distribution. Secondly, when we compare the intra-correlation distributions from different image filters to the original images ('Normal'), we can only notice small reductions in intra-class correlation values and such reductions are insignificant compared to the difference between intra- and inter-class distributions. This explains why SCI remains accurate when image filters are applied to the images. Most importantly, these results imply that *the SPN is well preserved in the filtered images though it may be affected differently by filtering operations*. In other words, SPN is still useful for image provenance analysis even after the Instagram image filters have been applied.

### B. SPN-BASED SOC FOR INSTAGRAM IMAGES

While the above SCI results show that the SPN is preserved in the filtered images, SOC relying on the pairwise similarities between the noise residuals of individual images can be more challenging. For SCI, as the reference SPN is immune from the filter-related artifacts, the inter-class correlation is unlikely to be altered. However, for SOC, the common artifacts introduced by the same filter may falsely increase the pairwise correlations between inter-class images, which might lead to *filter-oriented* rather than *source-oriented* clustering results. Thus, SOC is more vulnerable to these filter-related artifacts.

To investigate further, we test the images with the fast clustering (FC) method from [21], which has shown good precision and recall rates when applied on unedited original images from public image datasets. As a whole, we perform the SOC task on a test dataset, namely $\mathscr{D}_4$, which consists of 1800 images with 72 images from each of the 25 cameras in $\mathscr{D}$. The 72 images of each camera consist of 4 images randomly selected from those filtered by each of the 18 filters, which results in $4{\times}25{=}100$ filtered images for each filter. As shown in Table 4, the precision, recall and F1-measure are $61.11\%$, $39.17\%$ and $47.74\%$, respectively, which are much

TABLE 2: Source camera identification result for different Instagram image filters

| Filter | Normal | 1977 | Amaro | Brannan | Earlybird | Hefe | Hudson | Inkwell | Lomofi |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy (%) | 95.52 | 95.04 | 95.04 | 95.04 | 95.28 | 95.12 | 95.04 | 95.20 | 95.12 |
| Filter | LordKelvin | Nashville | Rise | Sierra | Sutro | Toaster | Valencia | Walden | XproII |
| Accuracy (%) | 95.20 | 95.04 | 95.04 | 95.20 | 95.44 | 95.36 | 95.52 | 95.20 | 95.28 |

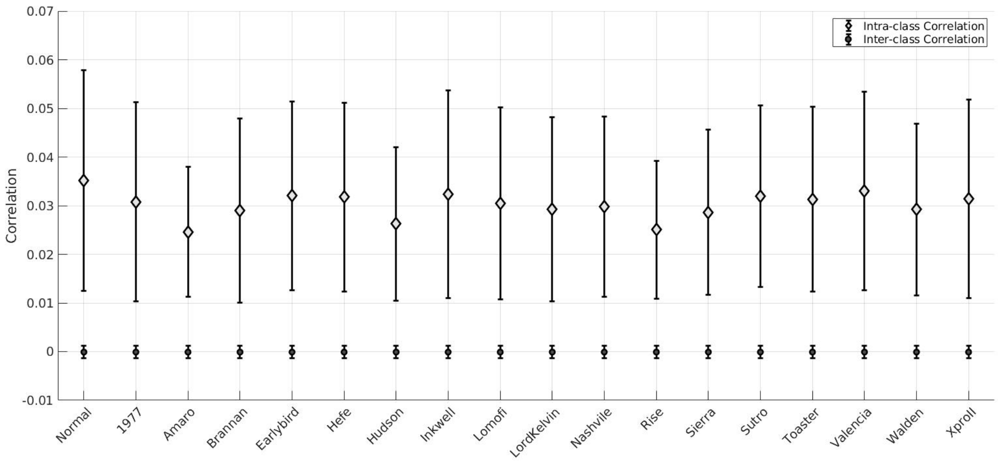


FIGURE 2: The correlation distributions for filtered images from an iPhone4s with its reference SPN with the central points representing the means and the error bars for the standard deviations. The distributions of the correlations between filtered inter-class images with the smartphone's original reference SPN are also shown in the figure.

lower even than the results (precision: $92.1\%$, recall: $81.2\%$, F1-measure: $86.3\%$) reported for the hard dataset $\mathcal{D}_4$ in [21]. To show that the performance is not biased to a specific algorithm, results are also shown in Table 3 for the hierarchical clustering (HC) method [15], the normalized cut-based clustering (NCUT) method [17] and consensus correlation clustering (CCC) method [18]. The low F1-measure rates for all the algorithms clearly show that it is a common challenge for existing SPN-based SOC algorithms to analyze Instagram images.

Additionally, to investigate how each filter affects the clustering results, we also perform separate clustering on the images filtered by the same filter. For each filter, we select 40 images from each of the 25 cameras in $\mathscr{D}$. Thus, for this experiment, the clustering for each filter is evaluated on 1000 images. The results are shown in Table 3. An interesting observation made from Table 3 is that, among the filters we have tested, some filters dramatically deteriorate the clustering performance while the others result in comparable clustering performance to that on *original* images. We, therefore, refer to the former set of filters as Group M because the filters are malignant for SPN-based SOC and the latter set of filters as Group B because the filters are 'benign'. When there is no ambiguity, we will also use Group M and Group B to refer to the images filtered by the former and the latter set of filters, respectively. We find that for Group M filters, the images are clustered into a single cluster, which is responsible for the low precision rate of $4.0\%$, i.e. each of the 25 camera accounts for 40 images in the resultant single cluster. This can be largely attributed to the common artifacts shared between the images filtered by the same filter. An example is shown for filter 'Hefe' in Fig. 3(a), where we plot the intra- and inter-class correlation distributions for original images (i.e. 'Normal') and the images processed by filter 'Hefe'. We also show the two corresponding grayscale plots of the $1000\times1000$ pairwise correlation matrices computed with 1000 'Normal' and 'Hefe' images, respectively, in Fig. 3(b). We can see that there is an apparent increase in mean and variance for both inter- and intra-class distributions for filter 'Hefe'. It is noteworthy that the increase of intra-class correlations is caused by the filter-related artifacts, thus it is not beneficial for *camera-oriented* clustering but rather gives rise to misleading *filter-oriented* results. Therefore, a clustering algorithm that can mitigate the effect of the artifacts introduced by the filters in Group M is needed for the effective provenance analysis of Instagram images.

## IV. PROPOSED METHOD

In the previous section, we have demonstrated the difficulty in SPN-based SOC on Instagram images, which arises mainly because of the artifacts introduced by the filters in Group M. Inspired by the success of the SPN-based SCI, for which the reference SPNs are available, we develop a framework that first performs clustering on the images in Group B and use the resultant clusters to process the images in Group M. We, therefore, propose a three-step strategy for the SOC on Instagram images. In the first step, a classifier is

TABLE 3: SOC results for different Instagram Image filters. The filters in Group M are highlighted with gray background.

| % | Precision | Recall | F1-measure |
|---|---|---|---|
| Normal | 94.70 | 78.92 | 86.09 |
| 1977 | 93.90 | 78.25 | 85.36 |
| Amaro | 4.00 | 100 | 7.69 |
| Brannan | 95.30 | 79.42 | 86.64 |
| Earlybird | 93.60 | 80.69 | 86.67 |
| Hefe | 4.00 | 100 | 7.69 |
| Hudson | 4.00 | 100 | 7.69 |
| Inkwell | 95.60 | 74.69 | 83.86 |
| Lomofi | 93.40 | 75.32 | 83.39 |
| LordKelvin | 91.30 | 81.52 | 86.13 |
| Nashvile | 95.10 | 78.92 | 86.45 |
| Rise | 4.00 | 100 | 7.69 |
| Sierra | 4.00 | 100 | 7.69 |
| Sutro | 4.00 | 100 | 7.69 |
| Toaster | 4.00 | 100 | 7.69 |
| Valencia | 93.30 | 75.24 | 83.30 |
| Walden | 93.50 | 75.40 | 83.48 |
| XproII | 90.30 | 83.61 | 86.83 |

TABLE 4: Clustering result on 1800 images with mixed filters and native images using the fast clustering (FC) method [21], the hierarchical clustering (HC) based method [15], the normalized cut-based clustering (NCUT) based method [17] and the consensused correlation clustering (CCC) based method [18].

| % | Precision | Recall | F1-measure |
|---|---|---|---|
| FC | 61.11 | 39.17 | 47.74 |
| HC | 56.06 | 37.88 | 45.21 |
| NCUT | 5.61 | 46.76 | 10.02 |
| CCC | 98.95 | 2.74 | 5.33 |

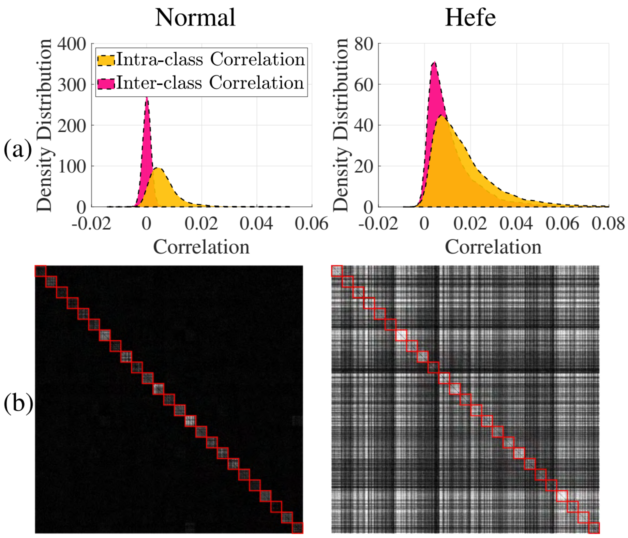


FIGURE 3: Comparison of the pairwise correlations between images with no filters applied ('Normal') and between images filtered by 'Hefe' filter. (a) Distributions plot for the pairwise intra- (yellow) and inter-class (red) correlations from 25 different cameras. (b) Visualization of the pairwise correlations for images from 25 different cameras. The intra-class correlations are surrounded by red squares. The brighter color indicate larger correlation values.

constructed for *filter-oriented* image classification to separate the images processed by Group B filters from the rest. In the second step, SOC is performed only on the images classified as processed by the filters in Group B. In the final step, we use the centroids of the clusters discovered in the second step as the reference SPNs to identify the source cameras for the remaining images, similarly to the task of SCI as described in Section III-A.

The three steps of our proposed framework are illustrated in Fig. 4. Specifically, we first pass the images to a convolutional neural network (CNN) based classifier to identify the image filter that has been applied to each image. Based on the classification result, we can separate the images into two sets, $S_B^\dagger$ and $S_M^\dagger$ for the images filtered by a filter from Group B and M, respectively. Due to classification errors, there might be images filtered by a Group M filter left in $S_B^\dagger$. To further purify $S_B^\dagger$, we refine the images $S_B^\dagger$ by comparing the pairwise correlations and the number shared nearest neighbours (SNN) [33] for images in $S_B^\dagger$. If we found that some images in $S_B^\dagger$ are more likely to be from $S_M^\dagger$, we will remove them (i.e. $S_M^{\dagger\dagger}$) from $S_B^\dagger$ to form a purified $S_B$. Then we apply the clustering algorithm to the images in $S_B$ to find the set of clusters $C$. Using the centroids of the clusters in $C$ as the reference SPNs $\{c_i\}$, we can approach the clustering as a SCI problem by attracting the images remained in $S_M^\dagger$ and $S_M^{\dagger\dagger}$ with $\{c_i\}$ to form the final clustering result. We will present the details about the CNN-based classifier and the classification refinement step in the following parts of this section.

### A. CNN-BASED INSTAGRAM FILTER CLASSIFIER

The proposed method mainly mitigates the negative impact of the filters in Group M by segregating the images according to the filter classification result. Thus, the performance of the classifier is key to the proposed framework and the classifier needs to be designed carefully. As the Instagram filters may differ from each other greatly, manual feature engineering requires a great amount of study for each filter and the fixed definition of image features might not be helpful when we need to deal with forthcoming filters that are not covered by this study. Moreover, the artifacts introduced by the filters can be content dependent, which may result in very different artifacts for the same filter. Therefore, we use a Convolutional Neural Network (CNN) based classifier to automatically extract features for the filter-oriented image classification task. The CNN architecture used in this work takes inspiration from the well-known Very Deep Neural Networks (VGG-net) [34], which has shown great performance on different image classification tasks. Particularly, Gatys *et al.* [35] manage to use VGG-net to extract and transfer the artistic styles of an image from one artwork to another, which is similar to adding

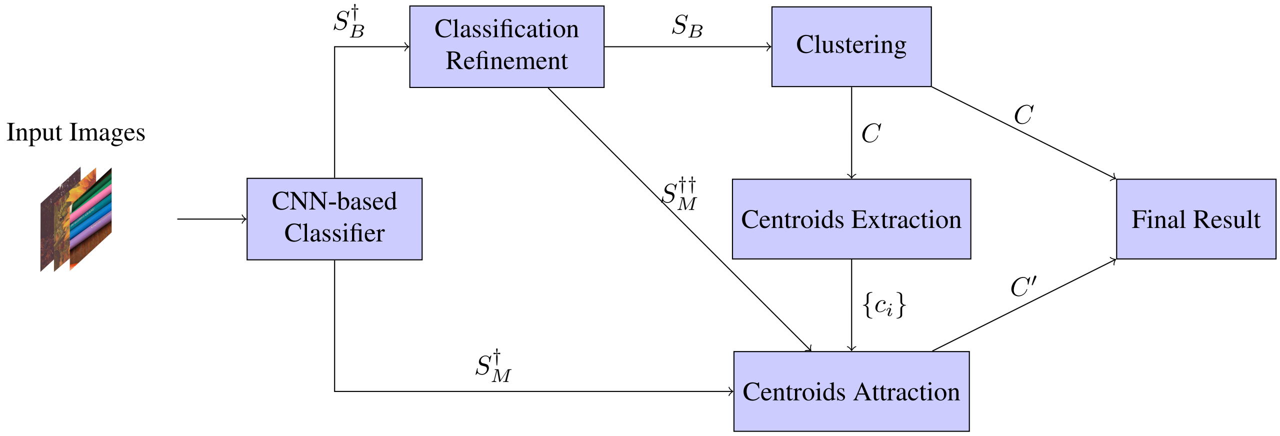


FIGURE 4: Flowchart of the proposed method for SPN-based source oriented clustering on Instagram images

visual effects to an image by applying Instagram filters. This shows that the network architecture is capable of extracting the features from dissimilar styles and inspires us to adopt a similar network architecture for classifying Instagram filters.

The network architecture used in this work is shown in Fig. 5. It consists of 7 convolutional layers for feature extraction and 3 fully connected layers for classification. Batch normalization [36] is applied to all the hidden layers. The input size of the network is set to $1080 \times 1080 \times 3$, which is the default image size of Instagram. As we aim to classify the images into 18 different classes, the network produces a vector of 18 elements. Softmax function is applied to the vector such that each element in the vector represents the probability of the corresponding image filter being applied to the input image. The network design shares a few similar characteristics with the VGG-net. The VGG-net features small kernel size for the convolutional layers (e.g., $3 \times 3$ pixels). This enables the convolutional layers to focus on microscopic features such as texture. Combined with a large number of layers resulting in a large receptive field, the network can extract the macroscopic feature such as color tone at the same time. This makes VGG-net an ideal choice to distinguish the filters. However, the requirement of large input size makes directly adopting the ordinary VGG-net very memory-consuming. Hence, our proposed network has two major differences compared to the ordinary VGG-net. The first difference is that the number of channels for each layer in the proposed network is much smaller than that used in VGG-net. Secondly, in our proposed network, each convolutional layer is followed by a max-pooling layer with a stride of 2. The max-pooling layers help the network extract features more efficiently and the input size of each layer is reduced significantly as the network gets deeper. With these two modifications in place, the memory consumption and the computational cost are significantly reduced, making the network more practicable.

### B. IMAGE FILTER CLASSIFICATION REFINEMENT BASED ON SNN-CORRELATION DIFFERENCE

Though a CNN-based classifier is proposed to distinguish Group M and B image filters and its effectiveness can be seen in the following section, its imperfect accuracy does not guarantee a complete separation between images with filtered by Group M and B filters. Thus, some images with Group M filters applied could remain in $S_2^\dagger$, which can affect the performance of the proposed clustering method. For example, a cluster contains a significant proportion of images processed by a Group M filter, the centroid the cluster is more likely to mistakenly attract images processed by the same filter later. Thus, to alleviate this problem, a classification refinement step is proposed below.

The main challenge we face by the inclusion of Group M filter applied images in $S_B^\dagger$ is that they may falsely increase the correlations between inter-class images and ultimately bring the risk of grouping the inter-class images into the same cluster. However, for these falsely increased inter-class correlations, the image pairs corresponding to them may share very different neighbours with each other. Figure 6 shows three clusters of images from three different cameras (represented by three orange circles) and four images (node $a$, $b$, $c$ and $d$) filtered by the same Group M filter. The dashed lines between $a$, $b$, $c$ and $d$ indicate the correlations between them might be falsely increased due to the same applied filter. Statistically, the intra-class correlations should be higher than the inter-class correlations which makes the intra-class image pairs to be closer neighbours to each other. As a result, even though $a$ and $b$ may have a large correlation between them, these two images share very few close neighbours with only $c$ and $d$ as the shared neighbours. It gives us a clue that the disagreement between the pairwise correlations and SNN [33] can be used to discover the images with Group M filters applied left in $S_B^\dagger$. Thus, we aim to find the image pairs with large correlation between their noise residuals but sharing few neighbours by the measure of correlation distances. More specifically, we remove the

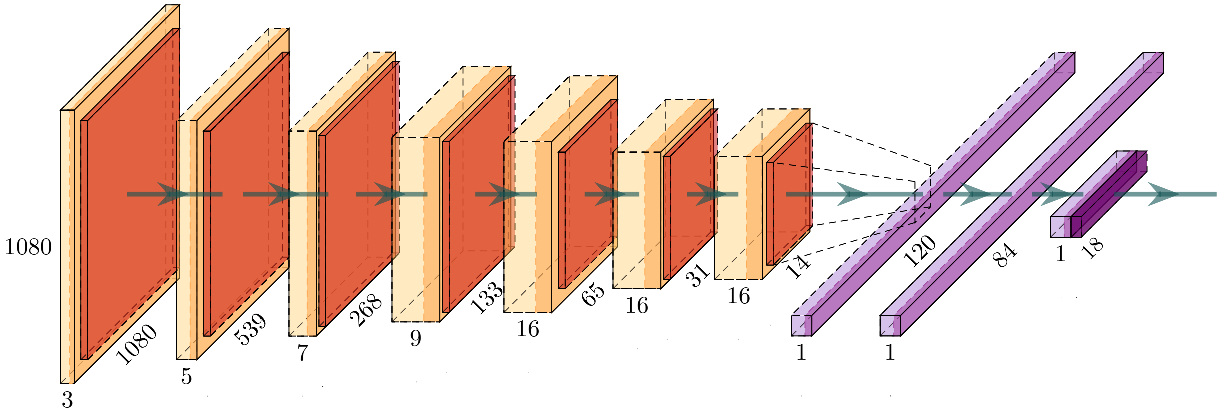


FIGURE 5: The network architecture of the proposed filter-oriented image classifier. The network takes 1080 × 1080 × 3 images as input and outputs a vector of length 18 for the classification. The network consists of 7 convolutional layers (shown in yellow) and 3 fully connected layers (shown in purple). In addition, every convolutional layer is followed by a max-pooling layer. The kernel size for the convolutional layers is 3 × 3 pixels throughout the network. The number at the bottom is the number of channels for the layer while the number at the sides are the dimension of the layer.

$i$th and $j$th image from $S_B^\dagger$ if $\rho_{ij} > \tau_1$ and $s_{ij} < \tau_2$, where $\rho$ and $s$ are the pairwise correlation matrix and the SNN matrix, respectively. $\tau_1$ and $\tau_2$ are the two threshold determined from the estimated intra-class correlations and intra-class SNN for each image. To estimate the intra-class correlation and SNN values, we follow the method from [21] using k-means clustering with $k$ set to 2 to differentiate the pairwise correlations and SNNs. Empirically, we set $\tau_1$ to the top $5\%$ of the intra-class correlations and set $\tau_2$ to the smallest value of the intra-class SNNs.

In the demonstration, we assume that the Group M filter applied images left in $S_B^\dagger$ are from multiple cameras and there are only a few of them in $S_B^\dagger$. Apparently, it is not always the case as described by these two assumptions and they may not hold. However, when these two assumptions do not hold, though the proposed method may become less effective, its mechanism of finding the obvious disagreement between the pairwise correlations and SNNs prevents it from repeatedly removing Group B filter applied images and deteriorate the performance of the clustering step. Thus, it is beneficial to apply the proposed refinement to $S_B^\dagger$ to purify $S_B^\dagger$ after the classification step.

## V. EXPERIMENT

### A. CNN-BASED INSTAGRAM FILTER-ORIENTED IMAGE CLASSIFIER

We first perform a comprehensive evaluation for the proposed CNN-based Instagram filter-oriented image classifier before using it in our proposed three-step SOC framework. As mentioned in Section II, we generate a dataset by filtering $5,370$ images of 25 different source devices from the VISION image dataset using 18 different Instagram filters, which results in a dataset $\mathscr{D}$ consisting of $96,660$ images.

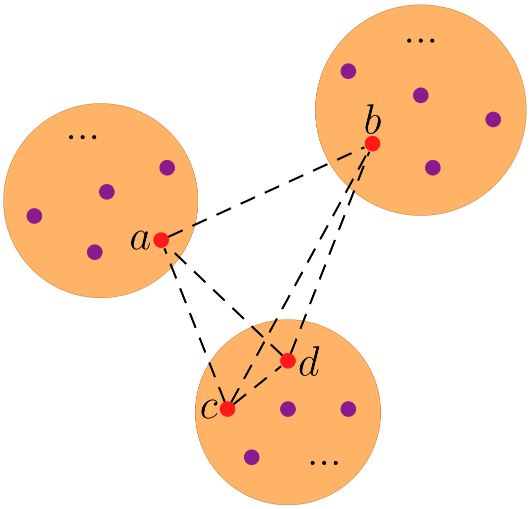


FIGURE 6: A demonstration of how the proposed filter classification refinement method may discover the images filtered by Group M filters remained in $S_B^\dagger$. Each node in the figure represents a candidate image to be clustered and the three circles represents the three ground truth clusters these images belonged to. Nodes $a$, $b$, $c$, $d$ are four images with the same Group M filter applied. Dashed lines are used to indicate the correlations between them may be falsely increased due to the filter.

These images are divided into training, validation and test sets with a ratio of $60\%$:$20\%$:$20\%$ by randomly selecting an equal number of images filtered by each filter. The proposed network is trained on a desktop with an Intel Core i7-9700K CPU and a Nvidia Geforce RTX 2080 Ti GPU. The special design of the network significantly reduces the consumption

TABLE 5: The precision ($\mathcal{P}$) and recall ($\mathcal{R}$) rates for different filters from the proposed CNN-based filter-oriented image classifier trained with different inputs ($\boldsymbol{I}$-net, $\hat{\boldsymbol{I}}$-net and $\boldsymbol{n}$-net). The best precision and recall rates for each filter are marked by gray background.

| Filters | $\mathcal{P}(\%)$ | | | $\mathcal{R}(\%)$ | | |
|---|---|---|---|---|---|---|
| | $\boldsymbol{I}$-net | $\hat{\boldsymbol{I}}$-net | $\boldsymbol{n}$-net | $\boldsymbol{I}$-net | $\hat{\boldsymbol{I}}$-net | $\boldsymbol{n}$-net |
| Normal | 52.89 | 76.70 | 96.61 | 65.55 | 58.85 | 98.04 |
| 1977 | 86.47 | 92.03 | 91.34 | 79.14 | 93.48 | 95.25 |
| Amaro | 83.74 | 85.54 | 87.49 | 69.55 | 79.33 | 87.90 |
| Brannan | 72.90 | 80.81 | 93.44 | 94.41 | 88.64 | 88.92 |
| Earlybird | 84.90 | 88.48 | 93.10 | 85.85 | 95.16 | 94.23 |
| Hefe | 83.00 | 81.96 | 92.10 | 85.94 | 91.81 | 94.41 |
| Hudson | 87.58 | 95.97 | 98.60 | 91.90 | 93.11 | 98.14 |
| Inkwell | 94.10 | 93.22 | 99.53 | 56.42 | 92.18 | 97.77 |
| Lomofi | 58.88 | 71.12 | 81.96 | 69.46 | 72.91 | 89.66 |
| LordKelvin | 90.19 | 95.41 | 98.22 | 94.97 | 94.88 | 97.86 |
| Nashville | 81.76 | 92.58 | 94.67 | 85.57 | 95.25 | 92.55 |
| Rise | 80.32 | 81.71 | 87.77 | 64.62 | 79.05 | 85.57 |
| Sierra | 72.44 | 79.93 | 97.84 | 77.09 | 86.41 | 96.83 |
| Sutro | 87.21 | 91.25 | 97.91 | 88.27 | 96.18 | 91.62 |
| Toaster | 98.21 | 96.93 | 97.13 | 92.27 | 96.93 | 97.77 |
| Valencia | 66.11 | 82.74 | 89.92 | 69.55 | 69.65 | 89.66 |
| Walden | 91.47 | 95.08 | 96.46 | 88.83 | 95.44 | 96.28 |
| XproII | 87.78 | 88.27 | 90.77 | 78.96 | 91.81 | 90.69 |

of GPU memory, which allows us to train the neural network with a batch size of up to 64. For the rest of this work, we will report the results generated with the classifiers trained with a batch size of 64. We train the classifier for 50 epochs using cross-entropy loss and a learning rate of $2\times10^{-3}$.

Instead of altering the semantic content of an image, most Instagram filters change the image's visual style and introduce different levels of textures, which mainly affects the high-frequency components of the image, where SPNs reside. This motivates us to investigate the contributions of the image content itself and the high-frequency components (noise residual) to the classification result. Thus, we pre-process the $96,660$ images and generate two more versions of input to the network, namely the denoised image and the noise residual of the image, i.e. $\hat{\boldsymbol{I}}$ and $\boldsymbol{n}$ in Equation 1. Again, we use BM3D denoising algorithm [32] to generate the denoised version of the images and extract the noise residuals from three color channels of each image. In such a way, $\boldsymbol{n}$ will have the same dimension as $\boldsymbol{I}$ and $\hat{\boldsymbol{I}}$, which allows them to be fed to the network without changing the network structure. Finally, we train three networks with these three different inputs, namely $\boldsymbol{I}$-net for the original images, $\hat{\boldsymbol{I}}$-net for the denoised images and $\boldsymbol{n}$-net for the noise residuals.

The precision $\mathcal{P}$ and recall $\mathcal{R}$ rates for 18 filters are reported in Table 5. Interestingly, we notice that $\boldsymbol{n}$-net, which takes the noise residuals as the input, outperforms the other two networks for almost all image filters. Though for some filters, $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net have higher precision or recall rates than $\boldsymbol{n}$-net, the performance gap is very small (within about $1\%$ for $\mathcal{P}$ and $1\% \sim 5\%$ for $\mathcal{R}$). Furthermore, both $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net have problems identifying 'Normal' images, which are the unedited original images. In comparison, the $\boldsymbol{n}$-net has a precision rate of $96.61\%$ and $98.04\%$ for the 'Normal' class.

TABLE 6: Confusion matrix for the classification of Group M and B applied images produced by the proposed CNN-based filter-oriented image classifiers.

| Real/Predict | $\boldsymbol{I}$-net | | $\hat{\boldsymbol{I}}$-net | | $\boldsymbol{n}$-net | |
|---|---|---|---|---|---|---|
| | M | B | M | B | M | B |
| M | 0.889 | 0.111 | 0.925 | 0.075 | 0.985 | 0.015 |
| B | 0.047 | 0.953 | 0.048 | 0.952 | 0.010 | 0.990 |

The high performance of $\boldsymbol{n}$-net can help the forensic investigators to better identify unedited images. Overall, the $\boldsymbol{n}$-net achieves a precision of $93.52\%$ for all filters while $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net reach $79.92\%$ and $87.29\%$, respectively. The high precision of $\boldsymbol{n}$-net shows the effectiveness of the proposed CNN-based classifier. We also show the confusion matrix for the classification of Group M and Group B filters as a whole in Table 6. Again, $\boldsymbol{n}$-net shows superior performance with only $1.5\%$ of the images in Group M misidentified. Due to the better performance of $\boldsymbol{n}$-net, we will use it as the filter-oriented image classifier for the following experiments of this work.

Despite the proposed network's high accuracy on filter classification, we have concerns about the generalization of the network to new *cameras* and *filters*. First, in many realistic forensic scenarios, the training and test images are quite unlikely to be from the same cameras. If a trained network is overfitted to the cameras in the training set, it will not perform well on the images from another set of cameras. To show that our trained network is not overfitted to the cameras in the training set, we test the trained $\boldsymbol{n}$-net on images captured by 11 different cameras of the Dresden Image Database [37]. We form a testing dataset $\mathscr{D}_{\text{Dresden}}$ with 18 different versions for each image from the cameras by applying the 18 different filters, resulting in a total of $29,700$ images. The classification results on $\mathscr{D}_{\text{Dresden}}$ are shown in Table 7, where $\boldsymbol{n}$-net shows similar performance as on the images from the VISION dataset, confirming that the trained model is not overfitted to cameras in the training set.

Secondly, new filtering features of Instagram are being developed continually. Thus, despite the 18 filters could be representative for studying the impact of filters on provenance analysis, we would like the classifier to be adaptive and robust to the filters that are not included in the training set. Thus in this experiment, we aim to show that the proposed network trained on a certain number of filters can be easily adapted for other filters by applying transfer learning. We test the $\boldsymbol{n}$-net by training it with images processed by 10 filters first and then apply transfer learning to the trained network to make it available for images processed by other filters as well. To facilitate transfer learning, we change the length of the last layer of the network to match the number of filters the network needs to predict for and keep the rest of the structure unchanged. The weights for the first five convolutional layers are fixed for the transfer learning process and each network is trained for another 10 epochs only. The performance of the network is shown in Table 8. It shows that despite the

TABLE 7: Filter classification result on images from Dresden Image Database [37] predicted by $n$-net trained with images from VISION dataset [31].

| Filters | $\mathcal{P}$ (%) | $\mathcal{R}$ (%) | F1-measure (%) |
|---|---|---|---|
| Normal | 97.28 | 99.76 | 98.50 |
| 1977 | 87.34 | 87.02 | 87.18 |
| Amaro | 90.63 | 89.75 | 90.19 |
| Brannan | 92.34 | 83.32 | 87.60 |
| Earlybird | 93.22 | 92.54 | 92.88 |
| Hefe | 95.65 | 84.05 | 89.48 |
| Hudson | 99.08 | 98.18 | 98.63 |
| Inkwell | 99.70 | 99.82 | 99.76 |
| Lomofi | 66.11 | 91.42 | 76.73 |
| LordKelvin | 92.98 | 88.36 | 90.61 |
| Nashvile | 87.70 | 97.33 | 92.27 |
| Rise | 87.80 | 90.78 | 89.27 |
| Sierra | 99.11 | 94.78 | 96.90 |
| Sutro | 93.47 | 94.60 | 94.03 |
| Toaster | 95.14 | 98.85 | 96.81 |
| Valencia | 90.18 | 90.24 | 90.21 |
| Walden | 95.82 | 94.48 | 95.15 |
| XproII | 98.00 | 74.41 | 84.59 |

TABLE 8: Filter classification results on images from different number of filters by the proposed CNN-based classifier with transfer learning applied. The base model of the classifier is trained with images from 10 different filters.

| Number of filters | $\mathcal{P}$(%) | $\mathcal{R}$(%) | F1-measure(%) |
|---|---|---|---|
| 10 | 97.63 | 97.61 | 97.62 |
| 12 | 95.42 | 95.41 | 95.41 |
| 14 | 94.12 | 93.99 | 94.06 |
| 16 | 90.86 | 90.82 | 90.84 |
| 18 | 89.69 | 89.54 | 89.61 |

a disproportional change of number of filters from 10 to 18, the F1-measure remains at a reasonably high level, indicating that the network is able to extract generalized features for the filters by training on only a small number of filters.

### B. CLASSIFICATION REFINEMENT

In this section, we are going to test the performance of the proposed classification refinement method. We test the proposed method by performing clustering on image datasets of different sizes. First, we construct 5 image datasets with 900, 1350, 1800, 2250 and 2700 images, respectively. For each image dataset, we have equal number of images randomly chosen from 25 source devices and from 18 different filters. Thus, with each filter, each camera accounts for 2, 3, 4, 5 and 6 images for the above mentioned four datasets. We name the five datasets as $\mathscr{D}_2$, $\mathscr{D}_3$, $\mathscr{D}_4$, $\mathscr{D}_5$ and $\mathscr{D}_6$ for convenience.

As we have seen from Section V-A, the proposed CNN-based filter classifier may leave about $1.5\%$ of Group M filter applied images in $S_B^\dagger$. Thus, to ensure the misclassified images that have been processed by Group M filters would not contaminate the cluster centroids extracted after the clustering step and worsen the performance of the ensuing centroid attraction, the performance of the proposed filter classification refinement step can be critical. Figure 7 illustrates the performance of the proposed filter classifier and the classification refinement method over the test datasets. First, we notice as we have seen from Section V-A, the classifier's performance is satisfactory even for the biggest dataset, $\mathscr{D}_6$, with 2700 images in total and 1050 Group M filter applied images. Only 18 Group M filter applied images are misidentified and included in $S_B^\dagger$.

To apply the proposed classification refinement method, the pairwise correlation matrices and the SNN matrices for the datasets were computed. To compute the pairwise correlations, we use the green channel of the full-sized noise residuals from each image. For the computation of the SNN matrices, we compare the 20 nearest neighbours of each image between the image pairs. Following the method proposed in Section IV-B, the number of Group M filter applied images removed from $S_B^\dagger$ is plotted in yellow as shown in Fig. 7. The total number images in $S_M^{\dagger\dagger}$, which is the set of the images removed from $S_B\dagger$ by the refinement method, is plotted in red for each tested dataset. From Fig. 7, though as it has been discussed in IV-B, we can see clues indicating that the performance of the proposed refinement method is less effective when the number of Group M filter applied images are too small (e.g. $\mathscr{D}_2$) and Group M filter applied images become less sparse in $S_B^\dagger$ (e.g. $\mathscr{D}_6$), overall, it shows the proposed refinement method is effective in reducing the number of Group M filter applied images in $S_B^\dagger$. As a result, the subsequent clustering and centroids attraction steps from the proposed three-stage clustering framework can be less affected by the Group M filters.

Another aspect worth mentioning is that though the proposed classification refinement step may also remove some Group B filter applied images from $S_B^\dagger$, it is not a serious problem. First, the number of images being removed is small comparing to the total number of Group B filter applied images to be clustered (e.g. 8 images falsely removed from 1650 Group B filter applied images in $\mathscr{D}_6$). More importantly, by applying the proposed refinement step, the centroids extracted from the clusters can be less contaminated by the Group M filters, which makes them more representative for the source device each cluster accounts for. With each centroids better representing the source devices in the test dataset, the wrongly removed Group B filter applied images can have greater chance being attracted to the right cluster during the centroids attraction step. Overall, by testing over different datasets, the effectiveness of the proposed classification refinement step is proved.

### C. SOURCE-ORIENTED CLUSTERING OF INSTAGRAM IMAGES

After testing the effectiveness of the proposed CNN-based image filter classifier and the classification refinement method, we test the overall performance of the proposed three-stage clustering framework with the five datasets mentioned above. We use the SOC method in [21] to perform the clustering step as described in Section IV. The centroids for each cluster are calculated by averaging the noise residuals of

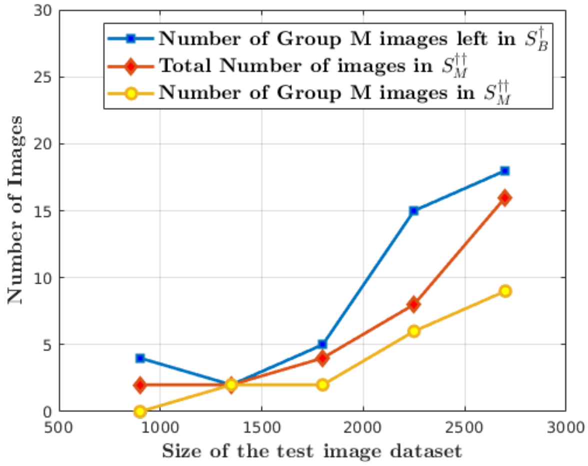


FIGURE 7: The performance of the proposed CNN-based filter classifier and the classification refinement method tested on image datasets of different sizes.

TABLE 9: The performance of the proposed three-step clustering framework on 5 Instagram image dataset of different sizes. The figures in the table are presented in percentage.

| *Dataset* | No. of Images | $\mathcal{P}$ (%) | $\mathcal{R}$ (%) | F1-measure (%) |
|---|---|---|---|---|
| $\mathscr{D}_2$ | 900 | 87.39 | 87.41 | 87.40 |
| $\mathscr{D}_3$ | 1350 | 93.73 | 83.75 | 88.46 |
| $\mathscr{D}_4$ | 1800 | 95.72 | 85.52 | 90.33 |
| $\mathscr{D}_5$ | 2250 | 94.57 | 76.28 | 84.45 |
| $\mathscr{D}_6$ | 2700 | 95.49 | 77.02 | 85.26 |

the images in the cluster. Table 9 shows the precision, recall and F1 measure for the proposed framework on the five test sets, the same ones as in Section V-B. Though the performance varies slightly across different datasets, the framework is able to obtain F1 measures over $80\%$ for all of the five test sets. The consistently high F1-measures show the effectiveness of the proposed framework. Comparing the performance of the proposed framework over $\mathscr{D}_4$ in Table 9 with the results from Section III-B, which was obtained on the same set of images, by applying the same clustering method proposed by [21] without using the three-step clustering framework, an overall improvement in both precision and recall rate can be observed. Thus, despite the Group M filters may contaminate the SPNs embedded in the images, the proposed three-step clustering framework provides a practical solution to perform SPN-based SOC on Instagram images.

## VI. CONCLUSION

With built-in image editing tools like 'filters' on Instagram becoming a common practice on SNSs, these tools ultimately pose new challenges to sensor pattern noise (SPN) based forensic investigations. In this work, using Instagram filter as an example, we took a close look at the impact of these image editing tools on SPN-based source camera identification (SCI) and source-oriented clustering (SOC). We discovered that though SPN-based SCI remains effective for filtered images on Instagram when quality reference SPNs are available, the artifacts introduced by certain Instagram filters can severely affect the performance of SPN-based SOC as there is no reference SPN. To address this problem, we proposed a three-step clustering framework. As a main component of the framework, a CNN-based filter-oriented image classifier is proposed and it achieves an overall $93.52\%$ precision in identifying the filters applied to images. We have also shown that the proposed CNN architecture generalizes well on new cameras and image filters. With the success of the filter-oriented image classifier, the proposed three-step clustering framework achieves an F1-measure of $90.33\%$ in SOC, which is a significant improvement compared to the F1-measure $47.74\%$ obtained by directly applying existing clustering methods on Instagram images. Thus, the framework provides a practical solution for the provenance analysis of user-edited images on SNSs. For future work, we will develop methods to deal with cross-platform in-app editing tools to cluster images from multiple SNSs.

## REFERENCES

[1] K. San Choi, E. Y. Lam, and K. K. Wong, "Source camera identification using footprints from lens aberration," in *Digital Photography II*, vol. 6069. International Society for Optics and Photonics, 2006, p. 60690J.

[2] L. T. Van, S. Emmanuel, and M. S. Kankanhalli, "Identifying source cell phone using chromatic aberration," in *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 2007, pp. 883–886.

[3] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, 2005, pp. III–69.

[4] M. J. Sorrell, "Digital Cannera Source Identification Through JPEG," *Multimedia forensics and security*, p. 291, 2008.

[5] E. J. Alles, Z. J. Geradts, and C. J. Veenman, "Source camera identification for heavily jpeg compressed low resolution still images," *Journal of forensic Science*, vol. 54, no. 3, pp. 628–638, 2009.

[6] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.

[7] O. Mayer and M. C. Stamm, "Forensic similarity for digital images," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1331–1346, 2020.

[8] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[9] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 5, no. 2, pp. 280–287, 2010.

[10] X. Kang, Y. Li, Z. Qu, and J. Huang, "Enhancing source camera identification performance with a camera reference phase sensor pattern noise," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 7, no. 2, pp. 393–402, 2012.

[11] X. Lin and C.-T. Li, "Preprocessing Reference Sensor Pattern Noise via Spectrum Equalization," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 11, no. 1, pp. 126–140, 2016.

[12] X. Lin and C.-T. Li, "Enhancing sensor pattern noise via filtering distortion removal," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 381–385, Mar. 2016.

[13] D. Cozzolino, F. Marra, D. Gragnaniello, G. Poggi, and L. Verdoliva, "Combining prnu and noiseprint for robust and efficient device source identification," *EURASIP Journal on Information Security*, vol. 2020, no. 1, pp. 1–12, 2020.

[14] G. J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 532–534, 2007.

[15] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proceedings of IEEE International Workshop on Information Forensics and Security*, Dec 2010, pp. 1–5.

[16] B. Liu, H. Lee, Y. Hu, and C. Choi, "On classification of source cameras: A graph based approach," in *2010 IEEE International Workshop on Information Forensics and Security*, Dec 2010, pp. 1–5.

[17] I. Amerini, R. Caldelli, P. Crescenzi, A. Del Mastio, and A. Marino, "Blind image clustering based on the normalized cuts criterion for camera identification," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 831–843, 2014.

[18] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind prnu-based image clustering for source identification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2197–2211, Sep. 2017.

[19] X. Lin and C.-T. Li, "Large-scale image clustering based on camera fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 793–808, Apr. 2017.

[20] Q. Phan, G. Boato, and F. G. B. De Natale, "Accurate and scalable image clustering based on sparse representation of camera fingerprint," *IEEE Transactions on Information Forensics and Security*, 2018.

[21] C.-T. Li and X. Lin, "A fast source-oriented image clustering method for digital forensics," *EURASIP Journal on Image and Video Processing: Special Issues on Image and Video Forensics for Social Media analysis*, vol. 1, pp. 69–84, Oct. 2017.

[22] X. Lin and C.-T. Li, "Rotation-invariant binary representation of sensor pattern noise for source-oriented image and video clustering," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov 2018.

[23] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Inforamtion Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[24] Y. Quan and C.-T. Li, "On addressing the impact of iso speed upon prnu and forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 190–202, 2020.

[25] Y. Quan, X. Lin, and C.-T. Li, "Provenance analysis for instagram photos," in *Australasian Conference on Data Mining*. Springer, 2018, pp. 372–383.

[26] L. J. G. Villalba, A. L. S. Orozco, and J. R. Corripio, "Smartphone image clustering," *Expert System with Applications*, vol. 42, no. 4, pp. 1927–1940, 2015.

[27] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media forensics and security*, vol. 7254. International Society for Optics and Photonics, 2009, p. 72540I.

[28] R. Satta and P. Stirparo, "On the usage of sensor pattern noise for picture-to-identity linking through social network accounts," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, Jan 2014, pp. 5–11.

[29] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, June 2017.

[30] I. Amerini, T. Uricchio, and R. Caldelli, "Tracing images back to their social network of origin: A cnn-based approach," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[31] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, Oct 2017.

[32] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

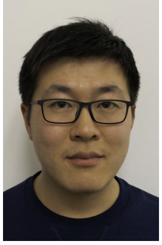
[33] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on computers*, vol. 100, no. 11, pp. 1025–1034, 1973.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.

[35] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv*, 2015.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, 2015.

[37] T. Gloe and R. Böhme, "The 'Dresden Image Database' for Benchmarking Digital Image Forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.



YIJUN QUAN received the BA degree in Natural Science from Trinity College, University of Cambridge, UK, in 2015, the MSc degree in Computer Science from University of Warwick, UK, in 2016. He is currently a Ph.D. Candidate at University of Warwick. He was a visiting scholar to South China University of Technology (SCUT) under Marie Sklodowska-Curie fellowship in 2018. His research interests include multimedia forensics and security, machine learning, image processing and computational photography.

XUFENG LIN received the B.E. degree in electronic and information engineering from the Hefei University of Technology, Hefei, China, in 2009, the M.E. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2012, and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2017. He is currently a Research Fellow with the School of Information and Technology, Deakin University, Australia. His research interests include digital forensics, multimedia security, machine learning, and data mining. and Technology, Deakin University, Australia. His research interests include digital forensics, multimedia security, machine learning, and data mining.

CHANG-TSUN LI received the BSc degree in electrical engineering from National Defence University (NDU), Taiwan, in 1987, the MSc degree in computer science from U.S. Naval Postgraduate School, USA, in 1992, and the PhD degree in computer science from the University of Warwick, UK, in 1998. He was an associate professor of the Department of Electrical Engineering at NDU during 1998-2002 and a visiting professor of the Department of Computer Science at U.S. Naval Postgraduate School in the second half of 2001. He was a professor of the Department of Computer Science at the University of Warwick (UK) until January 2017 and a professor of Charles Sturt University (Australia) from January 2017 to February 2019. He is currently a professor of the School of Information Technology at Deakin University, Australia. His research interests include multimedia forensics and security, biometrics, data mining, machine learning, data analytics, computer vision, image processing, pattern recognition, bioinformatics, and content-based image retrieval. The outcomes of his multimedia forensics research have been translated into award-winning commercial products protected by a series of international patents and have been used by a number of police forces and courts of law around the world. He is currently the EURASIP Journal of Image and Video Processing (JIVP) and Associate Editor of IET Biometrics. He involved in the organisation of many international conferences and workshops and also served as member of the international program committees for several international conferences. He is also actively contributing keynote speeches and talks at various international events.

• • •