# Source Camera Image Forensics

*A Thesis Submitted*

*in Partial Fulfilment of the Requirements*

*for the Degree of*

## DOCTOR OF PHILOSOPHY

*by*


**Kapil Rana**

**(2018CSZ0007)**




**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY ROPAR**


**April, 2024**

***Dedicated***

***To***

***My Family***

# Declaration of Originality

I hereby declare that the work which is being presented in the thesis entitled **Source Camera Image Forensics** has been solely authored by me. It presents the result of my own independent investigation/research conducted during the time period from January 2019 to April 2024 under the supervision of Dr. Puneet Goyal, Associate Professor, Department of Computer Science and Engineering, IIT Ropar. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted or accepted elsewhere, in part or in full, for the award of any degree, diploma, fellowship, associateship, or similar title of any university or institution. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgments, in line with established ethical norms and practices. I also declare that any idea/data/fact/source stated in my thesis has not been fabricated/ falsified/ misrepresented. All the principles of academic honesty and integrity have been followed. I fully understand that if the thesis is found to be unoriginal, fabricated, or plagiarized, the Institute reserves the right to withdraw the thesis from its archive and revoke the associated Degree conferred. Additionally, the Institute also reserves the right to appraise all concerned sections of society of the matter for their information and necessary action (if any). If accepted, I hereby consent for my thesis to be available online in the Institute's Open Access repository, inter-library loan, and the title & abstract to be made available to outside organizations.

Signature 

Name: Kapil Rana
Entry Number: 2018CSZ0007
Program: Ph.D.
Department: Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: 22-April-2024

# Acknowledgement

First and foremost, I express my gratitude to the **Almighty** for granting me the strength to embark on this journey and guiding me at every step. Without His blessings, this accomplishment would not have been possible.

I want to convey my sincere and profound thanks to my esteemed supervisor and mentor, **Dr. Puneet Goyal**, for his enduring blessings, invaluable guidance, unwavering support, well-wishes, and constant encouragement throughout this project. I am grateful to him for imparting numerous skills that I believe have greatly contributed to my journey and will play a significant role in shaping my future as a researcher. Words fall short in expressing my gratitude for his steadfast presence in both professional and personal challenges. The success of this thesis is undoubtedly attributed to his support and direction.

I extend my deep gratitude to **Prof. Rajeev Ahuja**, Director of the Indian Institute of Technology Ropar, and Dr. Sudarshan Iyengar, Head of the Department, CSE, along with Dr. Nitin Auluck and Dr. Somitra Sanadhya, former Heads of the Department, CSE. Their provision of facilities, assistance, and encouragement has been instrumental in conducting this research. Special thanks go to the members of my Doctoral Committee - Dr. Shashi Shekhar Jha, Dr. Mukesh Saini, Dr. Arun Kumar, and Dr. Nitin Auluck for their thorough reviews and valuable feedback. My gratitude also extends to the dedicated staff members of the CSE Department. I would like to thank Department of Science and Technology (DST) for providing the computational resources (DST/CSRI/2018/234).

I extend my deepest gratitude to **Prof. Gaurav Sharma**, Professor, University of Rochester, who has been an indispensable mentor and collaborator. His humility and readiness to assist at every turn have profoundly impacted my research experience. His encouragement and sage advice have been invaluable.

I am profoundly grateful to **Dr. Gurinder Singh**, not only for his invaluable collaboration and mentorship during my PhD journey but also for the brotherly bond we share. As an elder and a senior figure in my life, Dr. Singh's influence extends far beyond academic guidance; his life lessons and generous sharing of knowledge have significantly enriched both my personal development and the quality of my research. His role as a collaborator and akin to a second supervisor has been indispensable in shaping my thesis.

My heartfelt appreciation goes out to especially my IPSA lab colleagues Dr. Mandhatya Singh, Dr. Vishwas Rathi, Dr. Joohi Chauhan, Mr. Suhaib, Ms. Hadia and Mr. Joy for meaningful research discussions and collaborations. I also wish to thank Ms. Megha Mahobe and Mr. Charan for the initial discussions.

I want to thank my colleagues and friends at IIT Ropar - Dr. Ashwani Rana, Mr. Nikhil Reddy, Dr. Badri, Mr. Gulshan Sharma, Ms. Usma Bhatt, Mr. Waqar, Mr. Ashish, Mrs. Surbhi, Ms. Akansha, Mr. Sourabh Jaiswal, Mr. Prathamesh, Mr. Rahul Rai, Mr. Shivam, Mr. Ximi, Mr. Napinder, and Mr. Prabhu. Their companionship and support have been pivotal in ensuring a smooth and enjoyable PhD journey, filled with happy moments and lasting memories. I want to thank Dr. Divya Bansal and Dr. Manvjeet Kaur at Punjab Engineering College, Chandigarh, for providing me with computational

resources during the COVID-19 pandemic.

I am obliged to my most loving and wonderful parents **Sh. Laik Singh** and **Smt. Archana Devi** for their blessings, love, encouragement, and moral support in completing this task. Without their support, I couldn't think of going for a PhD. Their constant love and motivation helped me face difficult phases in the journey. I also wish to specially thank my brother Mr. Nishant Rana and bhabhi Mrs. Kiran Thakur for their constant support and my little niblings for their cheerful smile.

# Certificate

This is to certify that the thesis entitled **Source Camera Image Forensics**, submitted by **Kapil Rana (2018CSZ0007)** for the award of the degree of **Doctor of Philosophy** of Indian Institute of Technology Ropar, is a record of bonafide research work carried out under my guidance and supervision. To the best of my knowledge and belief, the work presented in this thesis is original and has not been submitted, either in part or full, for the award of any other degree, diploma, fellowship, associateship or similar title of any university or institution.

In my opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the Degree.

Signature of the Supervisor(s)

Dr. Puneet Goyal
Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: 22-April-2024

# Lay Summary

With the continuous advancement of imaging technology and the widespread use of smartphones, images have become a primary means of sharing information and capturing beautiful moments. However, with the widespread use of images, there are also increased concerns about the integrity of image information and the misuse of images, for example, for unauthorized information leakage, sharing of illicit photographs, and attacks on privacy. Understanding the origin of these images is crucial, especially in situations where people might misuse them for illegal activities or for leaking confidential information. Source Camera Image Forensics (SCIF) is a specialized field within digital image forensics that focuses on identifying the source camera associated with a given digital image. The Camera Model Identification (CMI) is the primary task in the field of SCIF which involves determining the camera model used to capture a particular image. This link between an image and the camera model can be pivotal, especially in investigative scenarios. This thesis aims to consider real-world aspects and associated challenges related to CMI and we aim to associate more accountability with image acquisition and image sharing. The key problems addressed in the thesis include:

- **Camera Model Identification:** The primary goal is to determine the specific make and model of the camera that captured an original image. Unlike other content-based image classification problems, the CMI is challenging as it involves analyzing the unique characteristics imprinted by different camera models during the image-capturing process and not much focus on image-content.

- **Camera model identification of social media processed images:** This task focuses on finding the source camera model for images that have been downloaded from popular social media platforms like Facebook, WhatsApp, or Instagram. Images shared on social media platforms undergo post-processing operations such as resizing and rescaling and this makes it very challenging to effectively perform CMI on social media processed images. We present generic and also social media specific CMI models in this thesis.

- **Camera identification of multispectral images:** Multispectral images capture a broader range of data, including wavelengths beyond the visible spectrum, and have more channel than the RGB images. There has been a rise in use of multispectral images for different applications, e.g. remote sensing, agriculture, and food quality inspection. We also explore the problem of camera identification of multispectral images. To the best of our knowledge, ours is the first work in direction.

- **New Dataset for CMI:** There are limited datasets available for CMI, with many captured in controlled settings. In response to this, we have developed a new dataset using contemporary smartphone cameras, considering various settings for increased diversity and better alignment with real-world scenarios. Our dataset

includes images captured by different users, featuring both similar and non-similar sets. In both sets, the content comprises a diverse variety of scenes.

# Abstract

With the advent of low-cost image acquisition devices, storage, and widespread network connectivity, digital images are being increasingly utilized for information capture and dissemination on social media platforms. However, with the widespread use of images there are also increased concerns about the integrity of image information and the misuse of images, for example, for unauthorized information leakage, sharing of illicit photographs, and attacks on privacy. Effective source camera model identification (CMI) techniques can play a crucial role in verifying the trustworthiness and integrity of the digital images and in investigating misuse by locating the source of images. In addition to forensic analysis and image tampering detection, the CMI techniques can also be used for intellectual property protection by identifying the source of copyrighted images, which can help prevent unauthorized use and distribution. This thesis delves into Source Camera Image Forensics (SCIF), a subfield of image source forensics that serves as a blind verification method for digital image authenticity and integrity. SCIF specifically aims to identify individual camera devices and models linked to images.

This thesis presents a detailed survey of existing methods for the SCIF. Additionally, it studies and improves the performance of CMI methods, presenting a basic framework for CMI. Within this framework, a dual-branch CNN method is proposed, incorporating improved methodologies for each stage. The first stage involves proposing a patch selection to extract important patches from the input image. In the second stage, high-pass filtering is applied to highlight artifacts related to the camera model, and this filtered image is passed to the second branch of the dual-branch CNN. In the third stage, ResNet is used to extract features from both RGB and high-pass filtered images. The proposed dual-branch CMI method demonstrates significant improvement compared to previous works, compared over multiple datasets.

Further, we consider more real-world aspects of images undergoing unknown processing for being shared over social media and making it very challenging to effectively perform CMI on these social media processed images. We present generic and also social media specific CMI models in this thesis. Identifying the Source Social Media Network (SSMN) of the image helps in channeling the image to the respective trained CMI model. So we propose a method, SNRCN2, for identifying the SSMN of digital images. SNRCN2 utilizes high-pass filtered images using steganalysis filters. The experimental results show the superior performance of SNRCN2. Furthermore, motivated by the fact that social media networks apply some Image Processing Operations (IPOs) during image upload, we propose a method, Multi-Scale Residual Deep CNN (MSRD-CNN), for detecting IPOs. The experimental results show that MSRD-CNN performs significantly better in classifying images post-processed with various operations. The thesis also acknowledges the growing applications of multispectral images, proposing a novel CMI method tailored for these images. A dual-branch network based on FractalNet rule is introduced, analyzing noise residuals from multispectral channels to classify camera models. Ours is also the first work related to camera identification of multispectral images.

Additionally, this thesis introduces a new dataset, IITRPR-CMI, designed to serve as a potential benchmark for evaluating CMI methods. This dataset comprises of a diverse set of images acquired using the contemporary smartphone cameras and features a unique train-test split based on content type and image acquisition methods for better alignment with the real-world scenarios.

**Keywords**: Camera model identification; High-pass filtering; Convolutional neural network (CNN); Source social media network; Multispectral images; Image forensics.

# List of Publications

**Journals**

1. Kapil Rana, Puneet Goyal, and Gaurav Sharma. Dual-branch convolutional neural network for robust camera model identification. Expert Systems with Applications, 238:121828, 2024.

2. Kapil Rana, Gurinder Singh, and Puneet Goyal. SNRCN2: Steganalysis noise residuals based CNN for source social network identification of digital images. Pattern Recognition Letters, 171:124–130, 2023.

3. Kapil Rana, Gurinder Singh, and Puneet Goyal. MSRD-CNN: Multi-scale residual deep CNN for general-purpose image manipulation detection. IEEE Access, 10:41267–41275, 2022.

**Conference Proceedings**

1. Kapil Rana, Vishwas Rathi, and Puneet Goyal. An effective CNN-based method for camera model identification in Privacy Preserving Settings. In 2023 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW) (pp. 1-5). IEEE.

**Under Review**

1. Kapil Rana, Vishwas Rathi, Joohi Chauhan, and Puneet Goyal. A Dual-Branch CNN for Multispectral Camera Device Identification. IEEE Transactions of Consumer Electronics (Under Review)

**Under Preparation**

1. Kapil Rana, Vishwas Rathi, and Puneet Goyal. A dataset for camera model identification. (Under Preparation)

**Publications not part of the thesis**

1. Kapil Rana, Aman Pandey, Parth Goyal, Gurinder Singh, and Puneet Goyal. A novel privacy protection approach with better human imperceptibility. Applied Intelligence, 1-11, 2023

2. Protyay Dey, Abhilasha S Jadhav, and Kapil Rana. An Effective CNN-based Approach for Synthetic Face Image Detection in Pre-Social and Post-Social Media Context. Presented in 8th International Conference on Computer Vision Image Processing (CVIP 2023), IIT Jammu, India

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AWGN | Adaptive White Gaussian Noise |
| AWB | Auto-White Balance |
| APMVC | Average Percentage of Majority class Votes for Correctly estimated images |
| CCD | Charge Coupled Device |
| CCL | Constrained Convolutional Layer |
| CDI | Camera Device Identification |
| CD-PRNU | Color-Decoupled Photo Response Non Uniformity |
| CE | Contrast Enhancement |
| CEAF | Contrast Enhancement Anti-Forensics |
| CFA | Color Filter Array |
| CMI | Camera Model Identification |
| CMOS | Complementary Metal Oxide Semiconductor |
| CNN | Convolutional Neural Network |
| DCT | Discrete Cosine Transform |
| DWT | Discrete Wavelet Transform |
| EM | Expectation-Maximization |
| GB | Gaussian Blurring |
| GIMD | General Purpose Image Manipulation Detection |
| HDR | High Dynamic Range |
| HPF | High-Pass Filter |
| ILA | Image-Level Accuracy |
| IPO | Image Processing Operation |
| JPEGAF | JPEG Anti-Forensics |
| LBP | Local Binary Pattern |
| MF | Median Filtering |
| MFAF | Median Filtering Anti-Forensics |
| OR | Original Image |
| PLA | Patch-Level Accuracy |
| PNRU | Photo Response Non Uniformity |
| PNU | Pixel Non-Uniformity |
| REN | Residual Extraction Network |
| ROC | Receiver Operating Charactersrtic |
| SCIF | Source Camera Image Forensics |
| SPN | Sensor Pattern Noise |
| SSMN | Source Social Media Network |
| SVM | Support Vector Machine |

# Chapter 1

# Introduction

The advancement of low-cost image acquisition devices, coupled with extensive storage and widespread network connectivity, has elevated images to an indispensable role in our daily lives. Images serve as a pivotal medium for information exchange, driving a range of applications such as telemedicine support, diagnosis, assistive technologies, entertainment, e-commerce, news media, and online education. With the convenience provided by smartphones and digital cameras, individuals can effortlessly capture and share images on popular social media platforms like Facebook and Instagram, contributing to the daily influx of billions of posted images. However, this surge in multimedia content on the web, facilitated by the pervasive use of these devices, also raises concerns about the integrity of image information and potential misuse. These concerns include unauthorized information leakage, sharing of illicit photographs, and privacy threats. It is essential to recognize the source of digital images and trace its history. Retrieving information about the history of an image is crucial in various investigations, encompassing forensic, criminal, security, privacy, and intellectual property inquiries. Knowing the camera used to capture the image and the processing it underwent is essential to verify its validity [1]. This information can assist in narrowing down potential suspects, thereby reducing their number and elucidating the authenticity of an image. The metadata, such as EXIF data within the image file, provide important information related to the camera model. However, it has limitations due to its susceptibility to modification or deletion. Consequently, relying on traces and inconsistencies within the image pixels emerges as a more reliable method, providing a higher level of trustworthiness to ascertain the authenticity of the images.

In recent years, the field of digital image forensics has seen considerable progress in addressing the issues of image authenticity and integrity. Image source forensics, a subfield of digital image forensics, focuses on determining the origin of a digital image. Figure 1.1 illustrates various sub-tasks integral to image source forensics, including camera model identification (CMI), camera device identification (CDI), recaptured image forensics, computer graphics image forensics, GAN-generated image detection, and source social network identification (SSNI). Each of these sub-tasks aims to uncover the source of the image. CMI and CDI fall under the category of Source Camera Image Forensics (SCIF), which is related to identifying the camera model and device of digital images, respectively. The remainder of this chapter is structured as follows: Section 1.1 provides an introduction to SCIF, and the objectives of this thesis and its organizational structure are detailed in Section 1.2

## 1.1 Source Camera Image Forensics

SCIF is a specialized field within digital image forensics that focuses on identifying the source camera associated with a given digital image. The primary objective of SCIF is to analyze the unique characteristics embedded in digital images during the image acquisition process, such as sensor noise patterns, lens distortions, and other artifacts specific to the camera. By scrutinizing these distinctive features, forensic analysts can ascertain crucial information about the camera used to capture the image. The following subsections provide an explanation of the image acquisition pipeline and details the primary task of camera model identification.

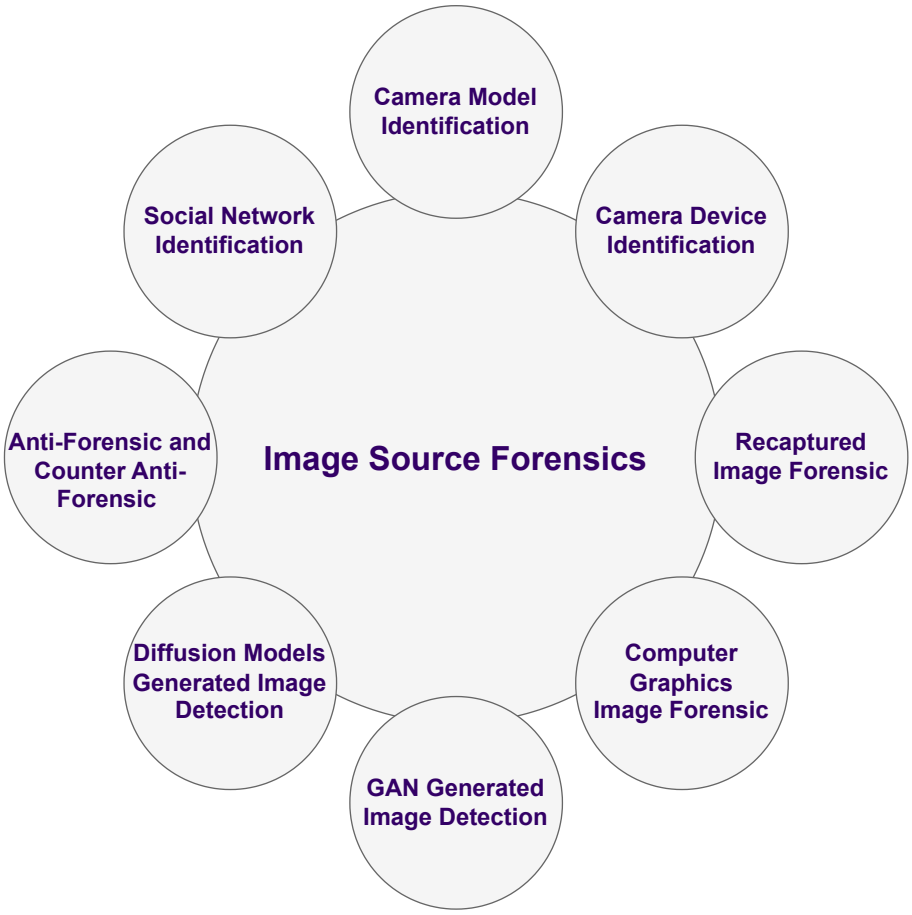



Figure 1.1: The overview of subtasks of image source forensics.

### 1.1.1 Image Acquisition Pipeline

The pipeline of image acquisition by a camera is not standard and can vary based on multiple factors such as the manufacturing company or the camera model. However, a general pipeline [2] of image acquisition can consist of a series of stages as shown in Figure 1.2.

The light passes through a lens and the lens is focused onto a sensor: Charge-Coupled Device (CCD) or Complementary Metal-Oxide Semiconductor (CMOS). The sensor

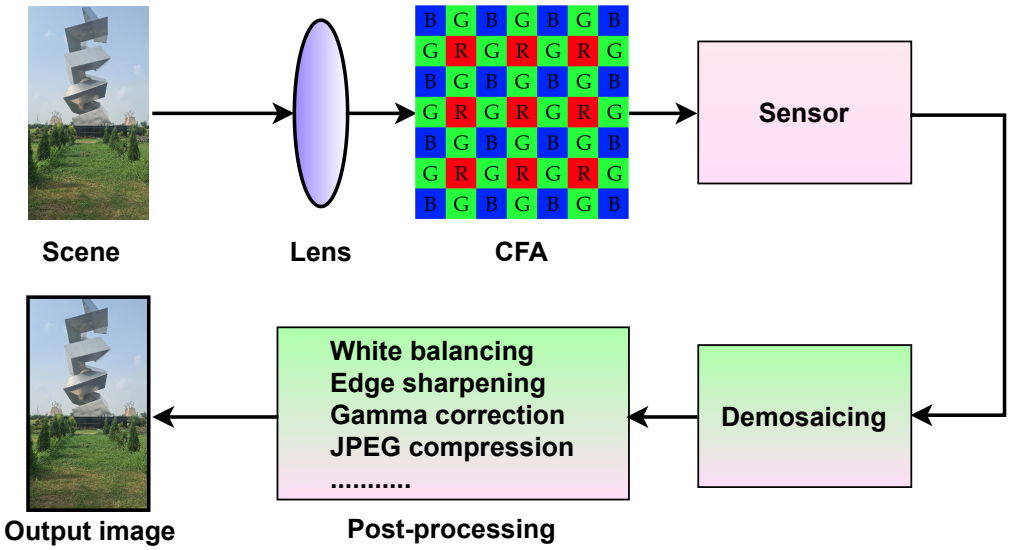
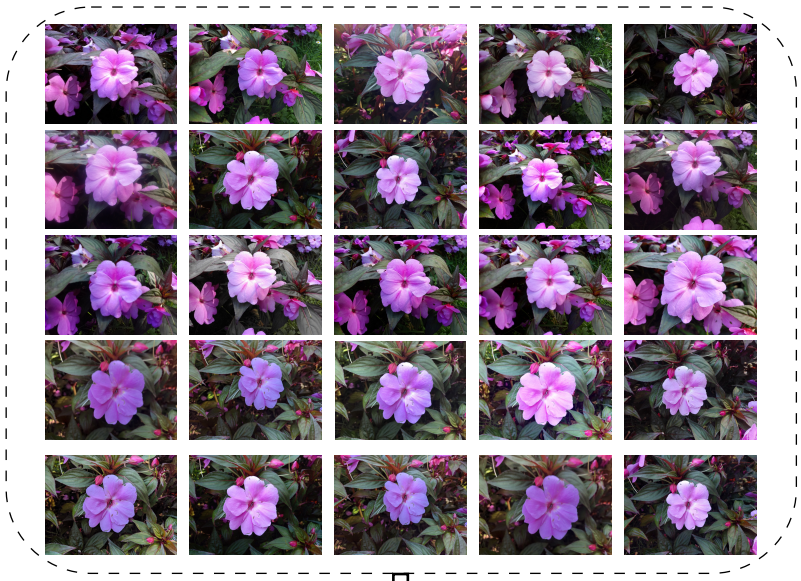



Figure 1.2: The stages of image acquisition and camera processing pipeline.

configuration comprises a matrix of diminutive elements arranged on a plane, where each element corresponds to a pixel. The voltage generated by each pixel is directly proportional to the brightness of the pixel. The use of a Color Filter Array (CFA) is a common method for capturing color images. This array of color filters is placed on the surface of the sensor, with each sensor element exposed to light within a narrow wavelength band corresponding to a specific color (Red, Green, or Blue). The intensity of green light is recorded for certain pixels, blue light for others, and red light for the remaining pixels. The assignment of colors to pixels is determined by the shape of the CFA, which varies depending on the manufacturer. Hereafter, the sensor's output comprises three partially sampled color layers, with only one color value recorded at each pixel location. The interpolation is employed to address missing color information, such as the absence of blue and red components for pixels that only received green light. This process, known as demosaicing, utilizes proprietary interpolation techniques to derive the missing color components from neighboring cells. Once the raw color image is acquired, a series of operations are usually performed in succession. Digital correction is used to fix optical distortions caused by lenses, such as barrel or pincushion distortion, which can leave forensic evidence. Additionally, white balancing and color correction are often done in a vendor-specific manner.

Lastly, the JPEG standard is commonly used for lossy image compression, though the compression quality and implementation may differ between manufacturers. The advent of computational photography has changed the landscape of photography. Nowadays, devices come with custom features such as generating portrait images with a digitally blurred background, creating an artistic effect. Furthermore, many devices are able to capture High Dynamic Range (HDR) images by merging multiple exposures into one. Smartphones with multiple cameras utilize specific algorithms (that are generally not revealed to public) and produce the final images by combining the inputs, coming from



Input images

| | | | | |
|---|---|---|---|---|
| Motorola E3 | LG Optimus L50 | Wiko Lenny 2 | Apple iPhone 6 | LG G3 |
| LG G6 | Motorola Z2 Play | Motorola G8 Plus | Samsung Galaxy S4 mini | Samsung Galaxy J1 |
| Samsung Galaxy J3 | Samsung Galaxy Star | Sony Xperia E5 | Apple iPhone 3 | Samsung Galaxy A6 |
| Apple iPhone 7 | Samsung Galaxy S4 | Apple iPhone 8 Plus | Google Pixel 3 | Google Nexus 5 |
| BQ Aquaris X | Huawei P9 lite | Huawei P8 lite | Huawei p20 lite | Google Pixel XL |

**Camera models**

Figure 1.3: Illustration of CMI (mapping of input images to different camera models of Forchheim dataset).

multiple cameras. Additionally, vendors offer options for capturing photographs with special filter effects, which can enhance images in various artistic ways. All of these custom operations provide valuable traces for forensic analysis.

### 1.1.2 Camera Model Identification

The camera identification of the image aims to find the respective camera that is used to capture the image. The camera identification involves two subtasks: camera model identification (CMI) and camera device identification (CDI). Given the input image, the CDI aims to find the exact camera device that is used for image acquisition. Similarly, given the input image, the CMI aims to find the camera model of respective device that is used for image acquisition. At times, these both terminologies are used interchangeably where there is only one device with respect to each camera model [3]. In this thesis, the focus is on CMI of digital images.

CMI is a specialized field within digital image forensics that focuses on determining the specific model of the camera that was used to capture a given image. As depicted in Figure 1.3 for Forchheim dataset [4] camera models, CMI involves the process of

aligning digital images with their respective camera acquisition models. The goal is to establish a mapping that associates distinctive features within the images with the specific camera models responsible for their capture. The CMI process involves analyzing the traces and artifacts that are exclusive to the characteristics of the internal processing pipeline of the camera. These traces come from the processing algorithms used by the camera. For identification of camera models, several methods have been proposed in the literature and a comprehensive survey of CMI methods is included in Stamm et al. [5] and Yang et al.[1]. The approaches in initial years have relied more on handcrafted features and/or statistical measures, obtained by considering the lens characteristics such as radial distortion [6], lateral chromatic aberration [7] and the pattern of dust particles on the sensor [8]; the CFA interpolation method deployed by the camera [9]; RGB pairs correlation [10]; photo-response non-uniformity noise (PRNU) [11]; image texture features from well selected color models and color channels [12]; or the combination of multiple features [13]. Most of these handcrafted features based methods have relied on support vector machines (SVM) for classification. Researchers have also explored non-handcrafted feature methods using CNNs. CNN based CMI was first introduced by Bondi et al. [14] using a simple architecture comprising four convolutional layers followed by two fully connected layers. Features obtained from the trained CNN for $64 \times 64 \times 3$ input patches were utilized in linear binary SVM classifiers, trained using one-vs-one approach for final CMI classification. Chen et al. [15] experimented with various pre-trained CNNs such as ResNet [16], AlexNet [17], and GoogleNet [18] for CMI. Yao et al. [19] proposed a deeper CNN model, considering 11 convolutional layers. Freire-Obregón et al. [3] proposed a two convolutional layer based simple network architecture for camera identification and conducted experiments by considering three smartphone devices. Another category of CNN based CMI approaches is motivated by the observation that image contents may act as obfuscating noise for CMI. These methods rely on pre-processing input images to discount the impact of the image content by using operations such as highpass filtering [20], non-linear median filter residuals [21] and LBP filtering [22] before passing to CNNs. These filter choices were primarily inspired by their success in steganalysis. Yang et al. [23] proposed using learned convolutional filters for preprocessing in a multi-scale content-adaptive fusion residual network for CMI. In further work along this direction, [24] used a sequence of pre-processing blocks containing convolutional layers to suppress unnecessary content and to learn a better classifier. Liu et al. [25] also explored trainable modified Res2Net [26] module for pre-processing, in combination with VGG16 based CMI classifier. Bennabhaktula et al. [27] applied constrained convolutional layer for the preprocessing and seven layer CNN model for the classification. Fusion of multiple feature extractors has also been explored in other recent work on CMI. Rafi et al. [28] explored features extracted from different size patches ($64 \times 64$, $128 \times 128$, $256 \times 256$) using DenseNet201. These features obtained from three different sized patches are concatenated to provide final features for CMI. In the work [29], camera-specific features are extracted from input patches by fusing three branches, two of which utilize

convolutional pre-processing, and the third branch does not have any pre-processing. Although many approaches are proposed for the CMI problem, the majority of methods have performed evaluation with only one dataset, i.e., the Dresden dataset [30]. Few methods [24] have considered more than one dataset, including images from smartphone cameras. Furthermore, only a few methods [24, 19] have performed evaluation with post-processed images using operations such as JPEG compression, rescaling, and Gamma correction with known parameters. To the best of our knowledge, none of these methods evaluated performance under the increasingly important real-world scenario where CMI is performed on images shared on social media platforms. To be effective in this challenging settings, CMI methods must be robust and maintain high accuracy despite the unknown post-processing introduced by these platforms. Also, it is important to evaluate and characterize the robustness of CMI methods in cross-dataset settings, where the image datasets on which CMI is deployed are different from those on which it is trained.

## 1.2 Objectives and Organization of the Thesis

### 1.2.1 Objectives

The literature review in this domain of source camera image forensics reveals some of the existing challenges and gaps. Most of the existing works did not consider the smartphone cameras which are primarily the source now for image acquisition. The real-world aspect of images being shared on social media is also often ignored in the existing works. There is also no work in relation to source camera identification of multispectral images which are also getting prevalent in many applications. This thesis is, therefore, focused on addressing some of these challenges and developing effective methods for the identification of camera models of the images. Considering these different aspects, the following objectives are identified:

- To prepare, implement and analyze a method for CMI of the original images.
- To develop a method for source social media network (SSMN) identification of images and CMI of social media network downloaded images.
- To develop a method for CMI of multispectral images.
- To create a dataset for smartphone camera images for benchmarking CMI methods.

### 1.2.2 Organization of the Thesis

The outline of the thesis is as follows:

- Chapter 1: The first chapter of the thesis is dedicated to the introduction discussing the background and motivation of the research work and also the related definitions. A brief discussion about existing CMI frameworks is provided in this chapter. This chapter also provides the organization of the thesis.

- Chapter 2: This chapter comprehensively discusses the literature review of different existing CMI methods. We present in detail the different stages of the pipeline of CMI methods and how the approaches differ in performing the tasks in these different stages.

- Chapter 3: In this chapter, we prepare, implement and analyze a method for the identification of camera model of images. A novel method "dual-branch convolutional neural network for robust camera model identification" is discussed. This method addresses a favorable inductive bias in the convolutional neural network (CNN) for CMI task. The method is dual-branch model that incorporates the features from RGB image and high-pass filtered image to provide rich features for CMI. A thorough quantitative and qualitative evaluation on multiple camera model datasets and comparative evaluation with existing methods are discussed in this chapter.

- Chapter 4: In this chapter, we consider the more real-world aspects in relation to utility of camera model identification methods. We discuss the effect of social media network post-processing on CMI. We develop a method for SSMN identification of the images. We have assessed the effect of social media network on the performance of CMI. Also, we have explored the image processing operation detection on the images.

- Chapter 5: In this chapter, we have extended the CMI method on the multispectral images. We develop a method for CMI of multispectral images. To the best of our understanding, ours is the first work in this direction. We also summarize how a new dataset of multiple multispectral images is prepared and this newly created dataset is then used for performance evaluation.

- Chapter 6: In this chapter, we discuss the positives and limitations of the prior CMI datasets. We also describe the newly created benchmarking dataset comprising smartphone camera images for CMI. Initial results of different competing methods on this new challenging dataset are also presented.

- Chapter 7: In this chapter, we conclude the thesis by highlighting the work done during the PhD and we also present some promising future directions.

# Chapter 2

# Literature Review

This chapter presents a systematic review of camera identification methods for digital images. Original digital images mainly consists of RGB images captured using either DSLR or smartphone cameras. In this literature review, we have primary explored the methods for CMI for digital images. This chapter also provide a brief review of CDI methods which involves identification of individual device. The extensive insights and coverage offered in this chapter serve as a valuable reference for researchers and practitioners in the field of information forensics and security, fostering further advancements in this domain. The CMI leverages to address a range of problems, including safeguarding intellectual property rights, managing patent infringements, and authenticating acquisition sources. In the research part, special emphasis is placed on the each stage of the framework of CMI methods. We have mainly focused on the articles which discuss deep learning based approaches for CMI. Regarding the framework of CMI approaches, we have focused on each stage and show the different methods and strategy applied by different approaches in each stage. Also, we have mentioned different challenges associated with the each stage of the framework. We have explained the brief methodology of related works. The chapter provides a comprehensive overview of popular image datasets employed for evaluating the performance of deep learning based camera identification methods. The references to related methods, datasets are provided. The literature review is primarily divided in three different sections. Section 2.1 focuses on existing methods related to CDI, detailing their methodologies. Section 2.2 covers the handcrafted features based methods for CMI. Finally, Section 2.3 extensively reviews deep learning-based methods related to CMI.

## 2.1 Camera Device Identification

The CDI aims to identify the individual device/sensor which is used to capture the image. It can be achieved by analysing the features related to sensors. Researchers have explored different methods to extract sensor specific features. In this section we have presented the existing CDI methods based on sensor based features.

### 2.1.1 Sensor Pattern Noise based Methods

Sensor Pattern Noise (SPN) is a unique and inherent noise related to sensors that arises due to imperfections in the manufacturing process. It serve as fingerprint for individual sensors as it It manifests as a distinctive spatially varying pattern of noise across the sensor pixels. The SPN is also often called as Photo Response Non Uniformity (PRNU)

[11]. This PRNU is predominantly attributed to Pixel Non-Uniformity (PNU), which arises from variations in pixel sensitivity to light. Lukás et al. [11] have extracted the noise by applying a denoising filter as per the following equation

$$N = I - F(I), \tag{2.1}$$

where $I$ is the input image, $N$ is the noise residual and $F$ is denoising funtion or filter. The PNU noise is approximated by averaging the noise residuals from multiple images. Researcher have proposed methods by leveraging PRNU or PNU noise as reference pattern from the known sensors and correlating with questioned images.

Lukás et al. [11] have applied different denoising filters to extract noise residuals and used a wavelet-based denoising filter as it provided the best results. For each camera, noise fingerprint (reference pattern) is extracted by averaging the noise residuals from more than 50 images. The fingerprint is further used to find the correlation with noise residual of test images. The authors also investigated the robustness of PRNU for images post-processed using JPEG compression, gamma correction. In [31], authors presented a framework for CMI using the PRNU of the image. However, the PRNU computation in this framework is achieved through a refined maximum-likelihood estimator applied to a simplified model of sensor output. This estimator optimally utilizes the available data, demonstrating an advantage in terms of efficiency. Specifically, the number of images required to estimate the PRNU is notably smaller compared to the approach proposed by Lukas et al. in [11]. Jiang et al. [32] also utilized the SPN noise as per in the method [11, 31] for the user identification using their respective camera devices.

In the work [33], authors have proposed a method for CDI. Initially a A photon transfer curve is plotted as the noise curve using RAW photos. The camera gain histogram is generated based on the occurrences of various camera gain constants. Four distinctive features are extracted from the distribution by utilizing this histogram, which are then used for training and testing a SVM classifier.

Goljan et al. [34] investigated to SPN with respect cropped and scaled images. The results suggest that CDI can be done for images that have been linearly scaled down by a factor of 0.5 or more, or for images where 90% or more of the content has been cropped.

Rosenfeld et al. [35] performed number of experiments to assess the robustness of PRNU based CDI. Image is post-processed using multiple Image Processing Operations (IPOs) such as denoising, re-compression, and out-of-camera demosaicing.

Liu [36] et al. proposed a method to improve the PRNU based CDI. They observed that detecting the presence of PRNU in an image poses a considerable challenge due to its inherently weak signal. The recommendation is to extract the PRNU from the noise residual by isolating the significant regions with higher signal quality, while discarding regions heavily affected by irrelevant noises.

Li [37] investigated that the SPN noise extracted using the methods in [11, 31] can be impaired by the scene details. In the work [37], authors have proposed a methodology aimed at mitigating the impact of scene details on SPN. It is based on the hypothesis that

higher signal components within an SPN are indicative of reduced reliability, and as such, should be mitigated. An improved SPN can be achieved by applying weighting factors inversely proportional to the magnitude of the individual SPN components.

Li et al. [38] explored the color decoupling technique to enhance the performance of CDI by separately analyzing the PRNU contributions in different color channels. This approach may improve the robustness of camera identification, especially in scenarios where color information plays a crucial role. The PRNU is extracted by considering the presence of CFA interpolation noise. Authors proposed Color-Decoupled PRNU (CD-PRNU) extraction method, which can effectively mitigates the diffusion of the Color Filter Array (CFA) interpolation error from the synthetic color channels to the physical channels. The CD-PRNU extraction method has demonstrated significant improvements in performance.

Tomioks et al. [39] introduced a CDI method by leveraging clustered PNU noises. The method extracts robust features related to image sensor The clustering of PNU noises provides robust features related to image sensors,random noise, and scene content variations.

Gisolf et al [40] proposed a simplified First Step Total Variation (FSTV) algorithm designed for CDI. Authors observed that wavelet based denoising filter taked lot of time and not suitable for large number of images. Therefore, the primary objective was to create a faster algorithm, the results demonstrate that FSTV not only achieves reduced computation time compared to wavelets but also exhibits enhanced accuracy. The optimal outcomes are achieved through the combination of FSTV and Phase SPN. The utilization of FSTV yields a substantial reduction in calculation time, proving advantageous for handling extensive databases.

Kang et al. [41] presented a CDI method based on an eight-neighbor context-adaptive SPN predictor to enhance the receiver operating characteristic performance of CDI. The adaptability of the SPN predictor to various image edge regions allows for better suppression of the impact of image content, resulting in a more accurate SPN estimation. It particularly excels in withstanding mild JPEG compression, such as a quality factor of 90%, especially when maintaining a low false-positive rate. However, it is important to note that the proposed method requires a substantial number of original images, not less than 100, to create a camera fingerprint. The advantage of our method diminishes when the camera fingerprint is generated with fewer images.

Julliand et al. [42] provides an in-depth examination of diverse sources and models of noise in digital images. Authors investigated various noise alterations stemming from both the acquisition pipeline and post-processing stages. Authors observed the impact of alterations on the quality and intensity of noise, meticulously studying the specific effects of each modification. It is observed that, as a JPEG image is generated, even in the case of a high-quality rendition, the noise undergoes substantial transformation from its original state in the raw image.

Gupta et al. [43] observed that the PRNU extracted through existing methods retains high-frequency details (edge and textures) from the images. Authors propose a

pre-processing step to enhance the efficiency of widely accepted PRNU extraction methods. In the pre-processing step, pMihcak filter is used. Experimental results on Dresden [30] dataset demonstrate that the pMihcak filter effectively eliminates noise from given images without distorting their high-frequency details.

Valsesia et al. [44] proposed a method to address the storage and matching complexity challenges in camera fingerprint databases through the utilization of random projections. Authors demonstrated that random projections effectively preserve the database's geometry while significantly reducing the problem's dimension with minimal trade-offs. The theoretical analysis encompasses the use of real-valued and binary random measurements, considering detection and false alarm probabilities. Also, Random projections offer superior compression ratios and enhanced scalability.

Goljan et al. [45] explored the impact of JPEG compression on the performance of CDI using the sensor fingerprints. Authors found that JPEG compression amplifies the variance of the normalized correlation and the Peak to Correlation Energy (PCE). Consequently, adjustments to the decision threshold are necessary to maintain a prescribed false-alarm probability. Apart from image compression, authors also preformed experiments related to video compression and it is observed that in the case of MPEG-4, not only does the variance of the normalized correlation depend on compression quality, but there is also a positive bias that increases the normalized correlation.

In work proposed by Li et al. [46], principal component analysis is explored to provide a compact representation of SPN. Authors proposed a framework for denoising and compression large size SPN. Reducing the size can speed up the processing time. Also, authors presented a method for constructing a training set that minimizes the impact of interfering artifacts. This method plays a crucial role in training the SPN feature extractor. This make it robust to various unwanted noise sources. The combination of theoretical derivations and experimental results indicates that this framework can serve as a comprehensive post-processing framework for effective and efficient CDI.

Mieremet [47] presented a simple formula that allows for the a priori prediction of the standard deviation of the correlation value distribution of multiple PRNUs for mismatches. This formula serves as a decision rule in CDI, enabling a choice between conducting a thorough investigation, including reference recordings (a time-consuming process), or opting for a more efficient approach. In the context of common-source identification, this formula can be utilized to offer an informed estimate for the threshold value in the cluster algorithm, eliminating the need for arbitrary testing of a range of threshold values.

### 2.1.2 Auto-White Balance based Methods

Deng et al. [48] proposed a method for CDI using Auto-White Balance (AWB) residue pattern. They investigated method to approximate the AWB setting applied during image capture. This can be useful to recognizing the unique artifacts introduced by different cameras. Authors also investigated their method for CMI and it shows good performance.

### 2.1.3 Sensor Dust-based Methods

Dirik et al. [8] proposed a DSLR CDI method based on sensor dust. The persistent location and unique shapes of dust specks in front of the imaging sensor create a unique fingerprint for DSLR cameras. Despite of built-in dust removal mechanisms these cameras, these hardware-based solutions often fall short of their efficacy claims. Moreover, since dust spots are typically not overtly visible, users tend to overlook them. The effectiveness of the method is evaluated on a dataset exceeding 1000 images from various cameras. The proposed method exhibits robustness to post-processing operations such as JPEG compression and downsizing.

### 2.1.4 Pixel Pattern-based Methods

Geradts et al. [49] observed that the errors in the Charge Coupled Devices (CCDs) were visible and it is possible to utilize these error for identification of cameras. The patterns related to hot point defects, point defects dead pixels, pixel traps, and cluster defects is unique in different cameras. However, the errors in the expensive camera are not that visible and image compression algorithms can suppress or move the pixel defects.

### 2.1.5 Deep Learning based Methods

Numerous deep learning-based CDI methods have been introduced in the literature [3, 50, 51, 25, 29]. However, it is noteworthy that despite their claims, these methods primarily focus on identifying the camera model rather than the actual source camera of a given image. This observation is derived from their methodological considerations, which involve training and testing on various camera models of the Dresden dataset [30]. As a result, their effectiveness in pinpointing the specific source camera of an input image remains limited.

## 2.2 Conventional Camera Model Classification

The conventional CMI methods investigates different artifacts in the image acquisition pipeline, related to camera models. San Choi et al. [6, 52] proposed a method for CDI identification as discussed in section 2.1. However, the lens are specific to camera models. A manufacturer apply same configurations lens to all the devices of same camera model. So fingerprints related to lens abbreviations is related to camera model rather than the camera device. Also, authors performed experiments with single device per camera model. In their methodology, A vector of 36 features is extracted from the each input image. This vector contains two additional lens radial distortion parameters as compared to 34 features proposed in the CDI method [10]. Based on these 36 features from different cameras a classifier is trained for distinguishing between images originating from a specific cameras. Kharrazi et al. [10] explored 34 features related to CFA configuration, the demosaicing algorithm, the color processing transformations. These features includes Average pixel

value, RGB pairs correlation, Neighbor distribution Center of mass, RGB pairs energy ratio, and Wavelet domain statistics. This method also proposed for CDI, however CFA and other image processing operations are specific to camera model.

Bayram et al. [9] proposed a method for CDI of image based on color interpolation traces in the RGB color channels. Authors have used expectation-maximization (EM) algorithms to generate number of measures and further, a classifier is used to determine the reliability of selected measures for classification of different camera images.

The work in [6, 9, 10, 52] are proposed for CDI but all these methods have performed experiments with single device per camera model. The features utilized in their respective method are more specific to camera model rather than the individual camera device.

Deng et al. [48] proposed a method for CMI using Auto-White Balance (AWB) residue pattern. They investigated method to approximate the AWB setting applied during image capture. This can be useful to recognizing the unique artifacts introduced by different cameras models. Authors investigated their methodology for both CDI and CMI. Experimental results shows good performance with respect to CMI.

San Choi et al. [53] explored statistics related to JPEG compression left on the images for CMI. Authors suggests employing the bit per pixel and the percentage of non-zero integers in each Discrete Cosine Transform (DCT) coefficient to represent the trade-off between quality and size in a digital camera. A CMI classifier is constructed using these features to assess their effectiveness.

Swaminathan et al. [54] examined the problem of component forensics and proposed the methods to find algorithms and settings used by camera during image acquisition. They have proposed a method to estimate the CFA pattern and interpolation kernel.

In the work [55], Filler et al. explored the PRNU for CMI. The PRNU is used primary for CDI. However, authors demonstrated that the same PRNU fingerprint can be used for CMI. Authors observed that fingerprints derived from images in the TIFF/JPEG format encompass local structure resulting from diverse in-camera processing. This structure can be identified by extracting a set of numerical features from the fingerprints and subjecting them to classification using pattern classification methods. The experimental results show the accuracy over 90%.

Swaminathan et al. [54] explored the distinct intrinsic traces on digital images related to camera model. The proposed model is based on the assumption that in-camera and post-camera image processing operations laves some intrinsic fingerprint artifacts on the final images which can be distinctive in between camera models. related to fingerprint. Authors have analysed the direct camera output and determine its component parameters along with intrinsic fingerprints. Any subsequent post-camera processing is treated as a manipulation filter. Authors ascertain the coefficients of its linear shift-invariant approximation through blind deconvolution. The integrity of the provided image is validated by the close resemblance between the estimated coefficients and the reference pattern.

Cao et al. [2] presented a framework to detect the demosaicing regularities in between

images. Authors have proposed an EM reverse classification algorithm. The proposed method investigated the inherent disparities in color filtering and demosaicing algorithms, achieving accurate detection through a meticulous reverse classification method coupled with partial derivative correlation models. The effectiveness of the reverse classification is demonstrated by employing an EMRC algorithm, particularly adept at resolving ambiguities in demosaiced axes. Utilizing partial derivative correlation models enables our method to efficiently discern both cross- and intra-channel correlations resulting from demosaicing. Identifying the correct demosaicing algorithms helps in the identification of camera model as demosaicing algorithm in these model is different. Experiments on 14 camera models have shown accuracy of 97.5%. Authors have proposed an expectation-maximization reverse classification algorithm

Goljan et al. [56] have performed large scale experiment consisting of images from 6896 individual cameras from 150 models. All the images are downloaded from Flickr and verified using the EXIF file. The camera identification is performed on the basis of SPN [11]. It is observed that error rates remain consistent among cameras of the same model, suggesting the efficacy of existing methods designed to eliminate non-unique systematic artifacts from fingerprints. Also, the primary factor leading to missed detection is the quality of the images utilized for fingerprint estimation.

Gloe et al. [57] presented a method for efficiently estimating lateral chromatic aberration (LCA). In their work, total 82 features consists of color features, image quality matrices, and wavelet featurs. Experiments on image source identification suggest the significant use of LCA for CMI in small sets. However, authors have mentioned the limitation of method on distinguishing large number of camera models.

Kirchner et al. [58] proposed a method for determining CFA pattern in demosaiced digital images. Understanding the pre-processing history and local inter-pixel correlation pattern structure can prove valuable in camera identification and digital watermarking.

Xu et al. [59] have utilized the 354 features related to gray-scale local binary pattern for CMI. Total 59 local binary patterns from the spatial domain of the red and green color channels are extracted, along with their corresponding prediction-error arrays and the 1st-level diagonal wavelet sub-bands of each image. All these features are extracted by considering 8-neighbor binary co-occurrence.

Thai et al. [60] have related the CMI problem with hypothesis testing theory. The approach in the work is based on heteroscedastic noise model which leveraging two parameters (a, b) as distinctive fingerprints for CMI. It comprehensively addresses all the noises affecting the raw image at the sensor output. The primary advantage of this approach lies in the development of two generalized likelihood ratio tests with analytically quantifiable performance, ensuring a specified false alarm rate. However, a notable limitation is the emphasis on raw images, which may not always be available in practical scenarios.

Milani et al. [61] investigated the traces left by color demosaicing algorithm. The method entails identifying the demosaicing algorithm by analyzing the interpolated image using

eigenalgorithms.

Chen et al. [62] stated that components related to image acquisition pipeline are complex and nonlinear. Also, it is hard to define a model based on component parameters. Authors investigate the rich models originally defined for steganalysis [63]. The approach in [63] relies on analyzing the prediction errors in stego images with concealed information added through steganography. Chen et al. [62] defined a rich model of demosaicing algorithm used in a camera by generating a diverse set of sub-models.

Marra et al. [64] utilized the features related to co-occurrence matrices of selected neighbors. The feature extraction involves three step: The residuals are extracted using high-pass filtering, the residuals undergo quantization and truncation processes, finally, the method includes the computation of the histogram of co-occurrences. The final 338 features are passed to SVM classifier for classification.

Celiktutan et al. [65] have explored three set of forensics features for CMI: binary similarity measures, image-quality measures, and higher order wavelet statistics. These features are further used for training SVM based classifier.

Tuama et al. [13] utiized features related to CFA interpolation, co-occurrences matrix, and conditional probability statistics. The high-order statistics provided by these features leverages the CMI performance. Feature are passed to train a SVM classifier. The method achieves accuracy of 98.75% on 14 camera models from Dresden dataset [30].

Thai et al. [66] proposed a statistical test for the CMI. Similar to the heteroscedastic noise model in the work [60], authors defined method on generalized noise model. Considering both the linear connection between the expectation and variance of a RAW pixel and incorporating the non-linear impact of gamma correction, the generalized noise model provides a more precise characterization of a natural image in TIFF or JPEG format.

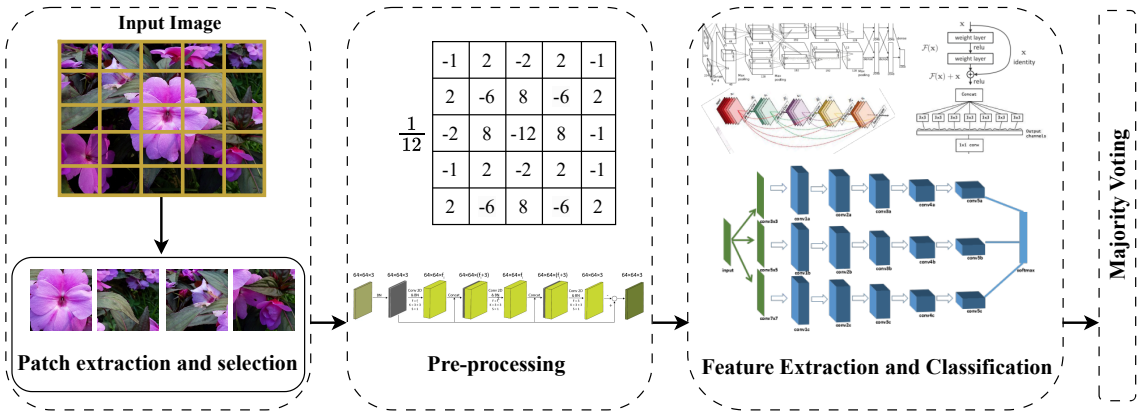



Figure 2.1: The framework of the deep learning-based methods for CMI.

## 2.3 Deep Learning based Camera Model Identification

In this section, we provide a comprehensive overview of each stage in CMI methods. Initially, we explain the framework of the CMI framework, reviewing each method within

the corresponding stages of the framework. Further, we present the major datasets used for evaluating CMI methods, along with their respective dataset splitting strategies. Subsequently, we provide a brief explanation of the methodology employed by each method.

### 2.3.1 Patch Extraction and Selection

The basic framework of CMI methods is illustrated in Figure 2.1. The initial stage encompasses patch extraction and selection from each image. Subsequently, a pre-processing stage is employed to enhance features. Following this, the patches undergo processing by a deep learning-based model for feature extraction and subsequent classification.

The input size of the image is very important aspect for the deep learning based models. The number of parameters of a CNN model is directly proportional to input image size. Numerous computer vision application applies resizing to input image prior to passing it to CNN model. The resizing of image is suitable for object recognition or ImageNet dataset image classification. However, for the forensics application such as CMI, CDI, or image manipulation detection [67, 68], there is information loss due to resizing and the artifacts related to fingerprints and manipulation lost from the image during the resizing operation. Due to limitation of resizing, majority of CMI methods applies patch extraction strategy to extract fixed sized patches from the image. The label of each patch is assigned as same as label of the input image. Extracting patches from images also provide the larger training dataset. The final prediction label of input image is estimated using the majority voting over the prediction of CNN for patches per input image. The majority is also very important aspect as it provide better accuracy when majority of patches are correctly classified.

The patch selection strategy defines the three major aspects: how many patches to be selected? 2. What is the size of the patches? 3. what kind of patches to be selected? Bondi et al. [69] defines the first strategy for the patch selection. They selected 32 quality non-overlapping patches of size $64 \times 64 \times 3$. The idea of their patch selection is providing large training data to smaller and lighter CNN architectures. According to Bondi et al. [69] patch selection strategy, some patches from the input image are not important as all patches may not contains statistical information. Therefore, the saturated patches should not be considered. A quality factor ($Q$) [69] is defined to extract only 32 most quality patches from image.

$$Q\left(P_k\right) = \frac{1}{3} \sum_{c \in [R,G,B]} \left[\alpha \cdot \beta \left(\mu_{\mathbf{c}} - \mu_{\mathbf{c}}^2\right) + (1-\alpha)\left(1 - e^{\gamma \sigma_{\mathbf{c}}}\right)\right],$$

where $\alpha$, $\beta$ and $\gamma$ are empirically set constants. $\mu_c$ and $\sigma_c$ are the mean and standard deviation of respective $c \in [R, G, B]$ channel. Methods in [25, 28, 70] extracts different number of patches of different sizes based on ($Q$). Moreover, alternative strategies for patch extraction have been explored by Mayer et al. [71] and Bennabhaktula et al. [27], wherein

patches are extracted based on entropy and standard deviation, respectively. Instead of parameter-based patch extraction, certain methods adopt simpler strategies by extracting all patches from the entire image. Methods in [20, 22, 72, 29] encompass all patches of specific dimensions across the entire image without imposing any conditional constraints. Some methods in [3] have extracted fixed number of patches from the image. However, few methods [19] considers some portion of image to extract the patches from training and inference. This approach is adopted due to the potential consequence of generating an excessively large training dataset when extracting all patches from all images. A summary of strategies of patch extraction applies in existing methods is presented in Table 2.1. where $\alpha$, $\beta$ and $\gamma$ are empirically set constants. $\mu_c$ and $\sigma_c$ are the mean and standard deviation of respective $c \in [R, G, B]$ channel. Methods in [25, 28, 70] extracts different number of patches of different sizes based on $(Q)$. Moreover, alternative strategies for patch extraction have been explored by Mayer et al. [71] and Bennabhaktula et al. [27], wherein patches are extracted based on entropy and standard deviation, respectively. Instead of parameter-based patch extraction, certain methods adopt simpler strategies by extracting all patches from the entire image. Methods in [29, 20, 22, 72] encompass all patches of specific dimensions across the entire image without imposing any conditional constraints. Some methods in [3] have extracted fixed number of patches from the image. However, few methods [19] considers some portion of image to extract the patches from training and inference. This approach is adopted due to the potential consequence of generating an excessively large training dataset when extracting all patches from all images. A summary of strategies of patch extraction applies in existing methods is presented in Table 2.1.

### 2.3.2 Preprocessing

In the context of deep learning-based problems in computer vision, it is a prevailing practice to employ RGB images as the input data for neural network architectures. This selection is substantiated by benchmark datasets like ImageNet [17], which find application in object or animal recognition tasks founded on the principles of human visual perception. Nonetheless, this foundational proposition may not uniformly translate to the image forensics tasks such as CMI, source social media network detection [81] and image manipulation detection [67, 68]. In the image forensics tasks, the main focus is the identification and localization of artifacts intrinsic to an image, rather than on the content present in the image. Hence, the significance of the preprocessing layer becomes evident, as it is regarded as pivotal for mitigating the impact of scene content that obfuscates the camera model fingerprints to a substantial extent. The summary of different methods pre-processing is mentioned in Table 2.2.

The initial works of deep learning based CMI methods try to extract the artifacts directly from the RGB image. All the methodologies outlined in [3, 14, 15, 69, 28, 70, 71, 19, 75, 76, 79, 82] apply CNN-based model directly on input RGB image. The primary distinction among these methodologies resides in the selection of the CNN model utilized for feature extraction and subsequent classification. It is assumed that these CNN models possess

Table 2.1: Summary of different patch extraction strategies used by different CMI methods. — implies that the method have not mentioned theirs strategy.

| Year | Method | Patches per image | Size of patch | Patch exctraction critieria |
|---|---|---|---|---|
| 2016 | Tuama et al. [20] | All patches | $256 \times 256$ | Select all patches. |
| 2016 | Bondi et al. [14] | 32 | $64 \times 64$ | Non-saturated patches based on quality factor $Q$ [69]. |
| 2017 | Bayar et al. [73] | 25 | $256 \times 256 \times 1$ | Patches from centre of green channel of image. |
| 2017 | Bayar et al. [21] | 36 | $256 \times 256 \times 1$ | Patches from centre of green channel of image. |
| 2017 | Bondi et al. [69] | 32 | $64 \times 64$ | Non-saturated patches based on quality factor $Q$ [69]. |
| 2017 | Chen et al. [15] | NA | NA | Resize the image to $256 \times 256$ (no patch extraction). |
| 2018 | Mayer et al. [74] | $\sim 165$ | $256 \times 256 \times 1$ | Patches from random position at green channel. |
| 2018 | Wang et al. [22] | All patches | $256 \times 256$ | Select all patches |
| 2018 | Yao et al. [19] | 256 | $64 \times 64$ | 75% of the image (from the centre of image) |
| 2018 | Güera at al. [75] | 300 | $64 \times 64$ | From the center of image |
| 2018 | Kuzin et al. [76] | 1 | $960 \times 960$ | Randomly cropped patch |
| 2018 | Ferreira at al. [70] | 32 | $229 \times 229$ | Non-saturated patches based on quality factor $Q$ [69]. |
| 2019 | Cozzolino et al. [77] | 1 | $1024 \times 1024$, $128 \times 128$ | Center crop |
| 2019 | Zou et al. [78] | 64 | $64 \times 64$ | Randomly selected patches |
| 2019 | Yang et al. [50] | All patches | $64 \times 64$ | Select all patches. |
| 2019 | Ding et al. [51] | — | $48 \times 48$ | Randomly cropped patch. |
| 2019 | Rafi et al. [28] | 20 | $256 \times 256$ | Non-saturated patches based on quality factor $Q$ [69]. |
| 2019 | Freire-Obregón et al. [3] | 256 | $32 \times 32$ | — |
| 2019 | Banna et al. [79] | — | $224 \times 224$ | — |
| 2019 | Mayer et al. [71] | All patches | $128 \times 128$, $256 \times 256$ | Based on Entropy $E$. |
| 2020 | Kang et al. [72] | All patches | $256 \times 256$ | All patches |
| 2021 | Liu et al. [25] | 128 | $64 \times 64$ | 64 based on $Q$ [69] and 64 using K mean clustering. |
| 2021 | Rafi et al. [24] | 20 | $256 \times 256$ | Non-saturated patches based on quality factor $Q$ [69]. |
| 2021 | You et al. [29] | All patches | $64 \times 64$ | Select all patches |
| 2021 | Laio et al. [80] | 5 | $227 \times 227$ | Randomly selected |
| 2022 | Bennabhaktula et al. [27] | 200, 400 | $128 \times 128$ | Homogeneous patches based on standard deviation. |
| 2023 | Sychandran et al. [80] | 256 | $32 \times 32$ | — |

the capability to discern and extract features specific to the camera model from the input RGB image. All these methods that involve directly feeding RGB input images into CNN models exhibit descent performance, demonstrate the efficacy of CNNs in extracting camera model-specific features. Also, a certain opacity remains in terms of the features that CNN models learn directly from the RGB image.

However, recent advancements in research have directed their focus towards the adoption of specific preprocessing operations/layer applied to input image prior to CNN-based classifiers. This approach holds significance due to the primary objective of preprocessing, which is to attenuate image content and guide the CNN classifier to prioritize non-content-related aspects, such as CMI feature rich image noise. The deliberate inclusion of image noise within the preprocessing phase amplifies the prominence of fingerprint-related attributes, thereby elevating the overall efficacy of CNN-based classifiers. For the extraction of CMI rich noise image, multiple method applies high-pass filtering layer. Tuama et al. [20] and Laio et al. [80] applies a single high-pass filter (HPF) [83] for noise extraction. Wang et al. [22] utilized the local binary patterns (LBP) for the CMI. Ding et al. [51] and Kang et al. [72] applies Gaussian filters to extract noise residuals from the input image. One branch of dual-branch CNN model in [21] apply median filtering [84] to suppress content. Instead of applying fixed HPFs, some method applies a dynamic high-pass filtering. The dynamic HPF can be a custom convolutional

layer or CNN-based network. Although it is similar as CNN based feature extracting layer, but a prerequisite here is that dimension of output of dynamic high-pass filtering is same as dimension of input RGB image. Methods detailed in [71, 27, 21, 85] applies constrained convolutional layer (CCL) [85] for enhancing camera model specific features. Both CCL and HPFs employ linear transformations to enhance pertinent features while simultaneously diminishing the impact of image content, resulting in enhanced capabilities of CNNs for extracting camera model specific features. In a distinct approach, Yang el al. [50] deploy three independent convolutional layers, each with varied sizes of learnable filters. Methods in [25, 78, 24] applies a CNN-based residual network for extract for CMI features enhanced noise image. Liu et al. applies a modified layer of Res2Net model [26] for residual extraction. Cozzolino et al. [77] applies a CNN-based network to extract camera model specific fingerprint (Noiseprint) which is distinctive across different camera models. In contrast to residual extraction, the method in [29] applies three branch network, wherein two branches deploy convolutional blocks with differing kernel sizes to extract multi-scale content features. Among all these methods in [50, 51, 29, 21] are multi-branch methods where each branch exhibiting distinct pre-processing.

Table 2.2: Summary of pre-processing used for CMI methods.

| Year | Method | Input image to CNN | Pre-processing operation |
|---|---|---|---|
| 2016 | Tuama et al. [20] | High-pass filtered image | Denoising high-pass filter |
| 2016 | Bondi et al. [14] | RGB image | — |
| 2017 | Bayar et al. [73] | Dynamic high-pass filtered image | CCL [86] |
| 2017 | Bayar et al. [21] | Dynamic high-pass filtered image | Median Filtering and CCL [86] |
| 2017 | Bondi et al. [69] | RGB image | — |
| 2017 | Chen et al. [15] | RGB image | — |
| 2018 | Mayer et al. [74] | Dynamic high-pass filtered image | CCL [86] |
| 2018 | Wang et al. [22] | LBP of image | LBP [59] |
| 2018 | Yao et al. [19] | RGB image | — |
| 2018 | Stamm et al. [82] | RGB image | — |
| 2018 | Güera at al. [75] | RGB image | — |
| 2018 | Kuzin et al. [76] | RGB image | — |
| 2018 | Ferreira at al. [70] | RGB image | — |
| 2019 | Cozzolino et al. [77] | CNN based Noiseprint [77] | CNN network |
| 2019 | Zou et al. [78] | CNN based residual image | CNN network |
| 2019 | Yang et al. [50] | Dynamic high-pass filtered image | Fusion based CNN network |
| 2019 | Ding et al. [51] | RGB and High-pass filtered images | Gaussain filters based residuals |
| 2019 | Rafi et al. [28] | RGB image | — |
| 2019 | Freire-Obregón et al. [3] | RGB image | — |
| 2019 | Banna et al. [79] | RGB image | — |
| 2019 | Mayer et al. [71] | RGB image | — |
| 2020 | Kang et al. [72] | High-pass filtered image | Conditional Gaussian filtering |
| 2021 | Liu et al. [25] | CNN based residual image | CNN network |
| 2021 | Rafi et al. [24] | CNN based residual image | CNN network |
| 2021 | You et al. [29] | CNN based image | CNN network |
| 2021 | Liao et al. [80] | High-pass filtered image | HPF |
| 2022 | Bennabhaktula et al. [27] | Dynamic High-pass filtered image | CCL [85] |

Table 2.3: Summary of feature extraction networks used for CMI methods.

| Year | Method | Feature Extractor | Feature Extractor Details |
|---|---|---|---|
| 2016 | Tuama et al. [20] | Single branch CNN | Convolutional layers |
| 2016 | Bondi et al. [14] | Single branch CNN | Convolutional layers |
| 2017 | Bayar et al. [73] | Single branch CNN | Convolutional layers |
| 2017 | Bayar et al. [21] | Dual-branch CNN | Convolutional layers |
| 2017 | Bondi et al. [69] | Single branch CNN | Convolutional layers |
| 2017 | Chen et al. [15] | Single branch CNN | ResNet34 [16] |
| 2018 | Mayer et al. [74] | Single branch CNN | Convolutional layers |
| 2018 | Wang et al. [22] | Single branch CNN | Convolutional layers |
| 2018 | Yao et al. [19] | Single branch CNN | Convolutional layers |
| 2018 | Güera at al. [75] | Single branch CNN | Convolutional layers |
| 2018 | Kuzin et al. [76] | Single branch CNN | DenseNet161 [87] |
| 2019 | Ferreira at al. [70] | Dual-branch CNN | Inception-ResNet [88] and XceptionNet [89] |
| 2019 | Cozzolino et al. [77] | Dual-branch CNN | Convolutional layers and Siamese Network [90] |
| 2019 | Zou et al. [78] | Single branch CNN | Modified SqueezeNet [91] |
| 2019 | Yang et al. [50] | Three-branch CNN | Residual blocks [16] |
| 2019 | Ding et al. [51] | Four-branch CNN | Residual blocks [16] |
| 2019 | Rafi et al. [28] | Three-branch CNN | DenseNet201 [87] |
| 2019 | Freire-Obregón et al. [3] | Single branch CNN | Convolutional layers |
| 2019 | Banna et al. [79] | Single branch CNN | MobileNet [92] |
| 2019 | Mayer et al. [71] | Dual-branch CNN | MISLNet [85] |
| 2020 | Kang et al. [72] | Single branch CNN | Convolutional layers |
| 2021 | Liu et al. [25] | Single branch CNN | VGG16 [93] |
| 2021 | Rafi et al. [24] | Single branch CNN | Convolutional layers |
| 2021 | You et al. [29] | Three-branch CNN | Convolutional layers and SE Block [94] |
| 2021 | Liao et al. [80] | Three-branch CNN | Residual blocks [16] |
| 2022 | Bennabhaktula et al. [27] | Single branch CNN | MISLNet [85] |

### 2.3.3 Feature Extraction and Classification

One of the key aspect of the deep learning based methods is to automatically extracts features from the images based on the error function between the estimated values and original target values. The CNN is widely used as feature extractor from images in the computer vision problems. Similar to vision problems, forensic community also employs the CNN for extracting feature directly from the given input image or the preprocessed input image. Most methods pass input to CNN based feature extractor after applying pre-processing as we discussed in section 2.3.2. As there is no universal architecture related to CNN based feature extractor, researchers proposed different models based on their respective hypothesis. Most methods utilize simpler one-branch models by varying the depth of the network and different parameters [3, 14, 69, 22, 72, 74, 75, 77, 24, 86]. Few methods [15, 25, 70, 76, 24] employ standard CNN models which are originally proposed for image classification tasks. Researchers also employed methods which are used in other forensics task such as image manipulation. Method in [71, 27] have utilized the MISLNet which is originally proposed for general proposed image manipulation (GIMD). Apart from one-branch models, researcher also explored multi-branch model based on the hypothesis that different pre-processed images may provide better camera specific features. The multi-branch model refers to a model wherein different types of inputs are passed to the CNN based feature extractor. Methods in [70, 71, 21] and [50, 25, 29, 28] utilize two-branch and three-branch feature extractor. Ding et al. [51] employs four-branch

network for feature extraction. The summary of different classifiers used in prior works is illustrated in Table 2.3.

The feature extracted from deep learning based feature extractor are passed to classifier to further classification. Initial methods methods [14, 85] utilized the support vector machine (SVM) to train a classifier. Training CNN and SVM together is two phase process. Firstly, CNN is trained using the training images. Secondly the trained feature extractor is used to provide the high-level features to train the SVM. Recent methods have applied fully-connected network and further to softmax layer. This softmax layer provides the probability distribution over all the camera models. The model with highest probability will be estimated camera model.

Table 2.4: Summary of datasets used for CMI methods.

| **Dataset** | **Total Images** | **Total original images** | **Total camera models** | **Total camera devices** |
|---|---|---|---|---|
| Dresden [30] | 14999 | 14999 | 25 | 73 |
| VISION [95] | 34,427 | 11,732 | 30 | 35 |
| Socrates [96] | 9721 | 9721 | 65 | 103 |
| Forchheim [4] | 23106 | 3851 | 25 | 27 |
| IEEE SP Cup [82] | 2750 | 2750 | 10 | 10 |

### 2.3.4 Datasets

The CMI requires model to learn from the dataset and evaluate their performance based on the learned information. Different camera identification dataset have been used for CDI, CMI. The CDI datasets provide images from multiple camera devices, wheres the CMI datasets provide images from multiple camera models. In the context of the CDI dataset, the count of distinct camera models is always less than or equal to the number of devices. In contrast, the desired aspect of CMI dataset is there should be at least two device per camera model. This ensures a fair evaluation by utilizing training images from one device and test images from a another device. The datasets are classified into two categories depending on the acquisition device type.

Furthermore, the datasets are systematically categorized into two classes based on the type of acquisition device. This categorization aids in organizing and analyzing data, facilitating a comprehensive exploration of model behavior across different device types within the CDI and CMI domains.

- Non-smartphone cameras datasets: This category of datasets contained images acquired from the non smartphone camera equipped with large size sensors. Due to large sensor size of these cameras, more light can be captured and more detail of the scene can be captures. Further, it provide interchangeable lenses for different view preferences and lenses also have different aperture.

- Smartphone cameras datasets: This category comprises images acquired from smartphone cameras. The sensor size of smartphone cameras is notably smaller

compared to other digital cameras. Smartphones typically have a fixed number of lenses, usually around three, and most of them feature a fixed aperture size. Given the widespread use of smartphone cameras among the general population, the CMI with respect to smartphone camera models are of utmost importance. Several CMI datasets have been released, containing images captured with smartphone cameras. The details of these datasets are presented in Table 2.4. Notably, there isn't a single primary or major CMI dataset encompassing images from smartphone cameras. This absence stems from the fact that these datasets have emerged at different points in time and are tied to the camera models available when they were introduced. The smartphone technology landscape is dynamic, with new smartphone cameras entering the market daily. Nonetheless, these datasets offer a platform for comparative analysis of CMI methods and CMI methods can subsequently be adapted and retrained to accommodate new datasets from evolving smartphone cameras.

One significant aspect related to datasets involves the uploading and downloading of post-processed images from social media platforms. The majority of information is disseminated through images shared on platforms like Facebook and WhatsApp. Almost all images are captured using smartphone cameras, given their compatibility with internet browsers and mobile applications. Therefore, assessing the robustness of CMI methods across social media platforms holds great importance. One major aspects related to datasets is the post-processed images uploaded and downloaded from social media platforms. Most of the information is circulated thought the images shared over social media platforms such as Facebook, WhatsApp. Majority of these images are captures using smartphone cameras as it support Internet browsers and mobile applications. So, it is very important to evaluate the robustness of CMI methods over social media platforms. Few datasets include a social media version of each image, corresponding to the original image captured using smartphone cameras. These datasets are shown in the Table 2.4. The post-processing version of images provide significant robustness of the CMI method. In IEEE SP Cup [82], test dataset contains images from seconds device with respect to camera model and these images are post-processed using JPEG compressing, resizing and gamma correction. These post-processing operation further provide a evaluation of the robustness of CMI methods. Few method have created copies of post-processing images by applying some image processing operations (IPOs). The most common image IPOs are JPEG compression, Gamma correction, resempling, resizing etc as shown in Table 2.5. Here are a few examples of IPOs with different parameters: JPEG compression is applied with quality factors of 70, 80, and 90; gamma correction is applied with values of 0.8 and 1.2; and contrast enhancement is performed with factors of 0.6, 0.8, 1.2, and 1.4.
Within the realm of CMI , the availability of multiple datasets is evident. However, a notable absence exists, characterized by the absence of a benchmarking CMI dataset that provides a predefined splitting into training and test sets. Various methods have employed different approaches when comes to splitting the dataset. The splitting of datasets are

very important as it provide a fair comparative evaluation of all different CMI methods. One of the significant split is stated by Bondi et al. [14], in which the dataset split is performed on the basis of device identifier. The training and test dataset are disjoint in terms of image acquisition device. It is worth noting that the Dresden dataset provides images from only 18 camera models that have two or more associated devices and does not include any smartphone images. In the Table we provide a summary of different methods dataset split strategy.

Table 2.5: Summary of dataset splits used for CMI methods.

| Year | Method | Datasets | Dataset splits | Folds | Post-processing |
|---|---|---|---|---|---|
| 2016 | Tuama et al. [20] | Dresden + Custom | (8:0:2) | 5 | — |
| 2016 | Bondi et al. [14] | Dresden (18) | Split dataset based on scene and device id | — | — |
| 2017 | Bayar et al. [73] | Custom | (8:0:2) | — | — |
| 2017 | Bayar et al. [21] | Dresden | (4:0:1) | — | Resampling + JPEG) |
| 2017 | Bondi et al. [69] | Dresden | Split dataset based on scene and device id | — | — |
| 2017 | Chen et al. [15] | Dresden | (7:0:3) | — | — |
| 2018 | Mayer et al. [74] | Dresden + Custom | Split dataset based on scene and device id | — | — |
| 2018 | Wang et al. [22] | Dresden | (8:0:2) | — | — |
| 2018 | Yao et al. [19] | Dresden | (3:0:2) | — | — |
| 2018 | Güera at al. [75] | Dresden | Split dataset based on scene and device id | — | — |
| 2018 | Kuzin et al. [76] | SP Cup + Custom | — | — | — |
| 2018 | Ferreira at al. [70] | SP Cup + Custom | — | 2 | Gamma correction (Resizing+ JPEG) |
| 2019 | Cozzolino et al. [77] | Dresden (3) | — | — | — |
| 2019 | Zou et al. [78] | Dresden (18) | — | — | — |
| 2019 | Yang et al. [50] | Dresden (23), Custom, SP Cup (3) | — | — | — |
| 2019 | Ding et al. [51] | Dresden (27) + Custom | — | — | Average blur Motion blur Bilateral blur Median blur) Compression |
| 2019 | Rafi et al. [28] | Dresden (27) + SP Cup (10) | — | — | Gamma correction (Resizing + JPEG) |
| 2019 | Freire-Obregón et al. [3] | MICHE-I (3) | — | — | — |
| 2019 | Banna et al. [79] | Custom | (9:0:1) | — | — |
| 2020 | Mayer et al. [71] | Dresden (26) + Custom $_{\text{tabular}}$ | — | — | |
| 2020 | Kang et al. [72] | Dresden (27) | (4:0:1) | 5 | — |
| 2021 | Liu et al. [25] | Dresden (18) | (7938, 1353, 540) (# images) | — | — |
| 2021 | Rafi et al. [24] | Dresden (27) SP Cup (10) | (7938, 1353, 540) (# images) | — | — |
| 2021 | You et al. [29] | Dresden (23) | (4:1:1) | — | — |
| 2021 | Laio et al. [80] | Dresden (17) | (4:0:1) | — | Contrast enhancement Resizing median filtering JPEG |
| 2022 | Bennabhaktula et al. [27] | Dresden | Split dataset based on device id | 5 | — |

### 2.3.5 Methodologies

The first work related to employing deep learning based CMI is proposed by Bondi et al. [14]. In their work, a RGB image patch is passed to four layer CNN network for

feature extraction. The extracted features from the trained CNN are subsequently fed into a SVM for further classification. The overall process is executed in two phase: the initial phase involves the training of the CNN model, followed by the subsequent training of the SVM classifier. The training of the CNN is based on the cross-entropy loss. Most of the CNN based CMI method have applied cross-entropy loss for the error computation. Experimental outcomes shows the patch-level accuracy (PLA) of 93% on 18 camera models of Dresden dataset [30].

Tuama et al. [20] have apply similar 3 layer CNN model as in [14]. However, they incorporated a high-pass filter [83] to suppress content and accentuating the artifacts related to camera model. The results shows the PLA of 91.9% on 33 camera models.

Bayar et al. [73] have discussed that the static high-pass filter for content suppression is not adaptive. The authors have addressed this issue by incorporating the CCL [86] which jointly learn the filter values based on the training of the CNN based CMI. Also, instead of utilizing all the channels of RGB image, the method takes only green channel as the input.

Bayar et al. [21] have expanded upon their prior work in [73] by enhancing the methodology through augmentation of the CCL layer with the median filtering layer. Total four convolutional layers are used to extract the features from pre-processed image. The robustness of proposed approach is evaluated on resampling and JPEG compression operations. Experimental results shows the PLA of 98.58% on 26 camera models of Dresden dataset. It is noted that authors have considered Nikon D70 and Nikon D70s as separate camera model in Dresden dataset.

Inspired by [14], Bondi et al. [69] investigated the different CNN based models for learning discriminant features from RGB image. They have investigated models incorporating four, six, eight, and ten convolutional layers. Authors also performed experiments via varying size of training data, correct splits of data into training, validation, and test sets. Experimental results shows the PLA of 93.93% on 18 camera models of Dresden dataset.

The work proposed by Chen et al. [15] is the first work related to apply transfer learning approach in CMI. This entails leveraging a CNN model initially designed for a distinct classification problem and subsequently fine-tuning it for the CMI problem. In this work, the ResNet34 [16] has been utilized for CMI. The experiments were conducted on a dataset comprising 27 camera models from the Dresden dataset and the model achieved image-level accuracy (ILA) of 94.73%. It must be noted that the images are resized to $256 \times 256 \times 3$ to meet the model requirement.

Similar to the work in [73], Mayer et al. [74] employed a four-layer CNN incorporating CCL and utilizing only the green channel of input images. However, the authors adopted a dual-network approach, training two separate networks to discern the similarity between images of two known camera models and a test image during the inference phase. The proposed method achieved the ILA of 95.8% on 10 camera model, when both the camera models are known.

Wang et al. [22] incorporated a three-layer CNN network with LBP of the input image.

it is observed that LBP coding operation enhances the overall performance and also highlights the effectiveness of a well-designed CNN structure with suitable preprocessing. The proposed method shows the PLA of 97.41% on 14 camera models of Dresden dataset.

Yao et al. [19] explored the deep CNN network with thirteen convolutional layers. It is emphasized that the deeper networks provide better high-level features. The proposed method achieve PLA of more than 93% and ILA of more than 98% on 25 camera models of Dresden dataset. The authors have performed experiments to check to robustness of proposed method under JPEG compression, adding Gaussian-distributed noise, and rescaling. The ILA on JPEG compressed and noise added images are more then 80%. However, the performance on re-scaled images are not good.

Similar to prior CNN-based works, Güera at al. [75] utilized a four layer CNN model without any preprocessing for CMI. However, the authors introduced a novel approach by proposing the estimation of patch reliability to identify the patches that not important for the training of CNNs. The utilization of this reliable patch selection method with the four-layer CNN model resulted in achieving a maximum ILA of 95%.

Inspired by the transfer learning methodologies, Kuzin et al. [76] applied a DenseNet161 [87] for CMI. Notably, they actively participated in the IEEE SP Cup 2018 on the Kaggle platform [82] and secured the second position. Throughout the training of DenseNet161, the authors implemented multiple augmentations, including rotation, JPEG compression, and Gamma transformations to enhance the robustness of model. The model achieved the ILA of 98.79% on 10 camera models of SP cup dataset.

Ferreira et al. [70] have propsped a dual-branch model which consists for two different models. One branch consists of Inception-ResNet [88] based network and another branch consist of Xception [89] based network. Each branch network independently outputs a high-level feature vector of size 256. The combined features from two baranches of 512 size is passed to fully-connected layer for further classification. The experiments has been performed in IEEE SP Cup [82] dataset.

Similar to PRNU fingerprint for the CDI, Cozzolino et al. [77] proposed a CNN based fingerprint named “Noiseprint” for distinguishing between different camera models. This fingerprinting approach also used for detecting manipulated images. In their work, The methodology employs CNNs to effectively extract and analyze distinctive noise features, aiming to establish a robust and distinct fingerprint for each camera model. Through the training of a Siamese-based model [90] and a CNN-based denoiser on a diverse dataset, the research showcases the potential of noiseprint in achieving accurate and reliable CMI. The emphasis on noise patterns as identifiable signatures signifies a valuable advancement in the field of digital image forensics.

Zou et al. [78] employed a residual network to extract the noise residuals from the input image. The essence of extracting noise lies in the ability of network to maintain the input image dimensions unchanged, and the learned Residual Extraction Network (REN) facilitates the generation of content-suppressed images. The noise residuals obtained from REN are then fed into a CNN network inspired by SqueezeNet [91]. The modified

SqueezeNet consists of eight fire modeules. The modified SqueezeNet comprises eight fire modules. Each fire module consisting of a 1×1 squeeze convolutional layer, a 1×1 expand convolutional layer, and a 3×3 expand convolutional layer. The squeeze convolutional layer plays a crucial role in diminishing the number of input channels for subsequent convolutional layers, thereby reducing the overall parameter count. The features from two different expand convolutional layer are connected through the concatenation layer. The proposed method achieved ILA of 91.29% on 18 camera models of Dresden dataset.

Yang et al. [50] have employed fusion networks for CMI. A fusion network is consist of three parallel network consists of same architecture with different parameters (kernel size). The network consists of three residual blocks and a convolutional layer. Initially, a fusion network is trained comprehensively on all patches and subsequently saved for transfer learning. The saved model serves as the initialization for three distinct fusion networks. These three networks are individually trained on three different sets of image patches categorized as saturated, smooth, and others based on mean and variance. During the inference of a test image patch, the model selection is determined by the type of patch, and the output from the chosen model is defined as the final estimation of the image patch.

Ding et al. [51] have employed fusion network for feature extraction and classification. In their methodology, three version of high-pass filtering is applied in three distinct branches. The first branch utilized a $3 \times 3$ Gaussian filter to extract noise. The second branch processed the denoised image from the first branch and applied a $5 \times 5$ Gaussian filter. The output from the second branch was then further processed with a $7 \times 7$ Gaussian filter. This approach resulted in three different versions of high-pass filtered images, which were then passed through three separate residual blocks to obtain low-level features. The outputs of three residuals blocks are concatenated with low-level features of RGB input image outputs from another residual block. The concatenated features were subsequently fed into a CNN-based network consisting of multiple residual blocks. The proposed method achieved ILA of 98.8% on Dresden dataset. In the experiments, the robustness of proposed is evaluated on different post-processed images e.g. blurring, compression, and contrast enhancement.

Rafi et al. [28] have applied three DenseNet201 [87] models to extract features from three different reshaped version of input image. All these three are applied in parallel to extract high-level features of size 1920 from the input. The concatenated features are further passed CNN based model incorporating squeeze and excitation blocks [94] along with convolutional layers. Each DenseNet201 has been initialized with DenseNet201 trained on all patches of size $256 \times 256$ from all images. Further, three different DenseNet201 is trained with $1 \times 256 \times 256$, $4 \times 128 \times 128$, and $16 \times 64 \times 64$ size reshaped input image of size $256 \times 256$. The training dataset is augmented with the JPEG compression, resizing, and gamma correction. The proposed method achieved accuracy of 98.37% on test set of IEEE SP Cup [82] dataset.

Freire-Obregón et al. [3] have employed a simple CNN model consisting of two convolutional layers. The proposed method have achieved ILA of 92.3% on MICHE-I

dataset. Different experiments have been perfom to evaluate different hyperparameters such as dropout, activation function, and topology of the model.

Banna et al. [79] have applied the transfer learning methodology and employed MobileNet [92] for feature extraction. The features extracted from MobileNet are subsequently fed into three distinct statistical classifiers: SVM, random forest, and logistic regression. The methods employing SVM, random forest, and logistic regression achieved ILA of 98.82%, 97.16%, and 98.54% on IEEE SP Cup dataset, respectively.

Kang et al. [72] utilized a CNN model consisting of three convolutional layer for feature extraction from reduced edge image patches. It is observed that image patches containing strong edges tend to be scene-specific and may not encapsulate crucial artifacts distinctive to the camera model. Therefore, it is better to apply smoothing filter to reduce the affect of strong edge on CNN based CMI training. The proposed method achieved ILA of 95% on 27 camera models of Dresden dataset.

Mayer et al. [71] introduced a dual-branch similarity network designed to ascertain the similarity of forensic features between two input instances. The input features to similarity network are output of two distinct feature extractors with similar architectures. Notably, features originating from the same camera model exhibit higher similarity compared to those from different models. The feature extractor is inspired from the MISLNet [85]. The MISLNet consists of CCL layer to extract the noise residuals from the input image. These noise residuals are subsequently passed to four convolutional layer for feature extraction. The training of proposed method is executed in two phase. Initially, the feature extractor is trained separately, and the same feature extractor is employed in both branches. Subsequently, the similarity network is trained using features extracted from the trained feature extractors. The proposed method achieved ILA of 94% on 25 camera models of Dresden dataset.

Similar to work by Zou et al. [78], Liu et al. [25] have incorporated dynamic extraction of noise residuals. The noise residuals are extracted by applying proposed Res2Net [26] based residual prediction module. The noise residuals outputs from residual prediction module are passed to VGG16 CNN model for further feature extraction and classification. The distinctive aspect of their approach lies in the residual prediction module which effectively and efficiently suppress the content information and provide artifacts rich noise residuals. The proposed method achieved ILA of 92.62% on 18 camera models of Dresden dataset.

Rafi et al. [24] employed CNN based RemNet network for the extraction of noise residuals and subsequent feature extraction. Inspired from highway networks [97], the authors designed a remnant block comprising a convolutional layer, batch normalization, and ReLU activation. The dimension of output and input of remnant block is same. The RemNet network is constructed with three remnant blocks arranged sequentially, followed by a CNN-based classifier. he noise residuals derived from the remnant blocks are directed to the classifier network for feature extraction and classification. The RemNet method achieve ILA of 97.59% and 95.11% on Dresden and IEEE SP Cup dataset, respectively. In addition to evaluating performance on original images, the robustness of RemNet was

assessed on post-processed images subjected to various IPOs such as JPEG compression, gamma correction, and rescaling.

You et al. [29] investigated the multiscale convolutional layers for the extraction of low-level features. The proposed method is three branch CNN model. The first branch processes RGB images as input without applying any operations. The second branch comprises two convolutional layers with sequentially sized kernels of $3 \times 3$ and $5 \times 5$. Conversely, the third branch reverses the order of convolutional layers. Additionally, both the second and third branches incorporate a skip connection at the second convolutional layer. The outputs from all preprocessing steps are channeled into three distinct CNN-based SE-SCINet models with same architecture. The SE-CDINet consist of multiple convolutional layers and Squeeze and Excitation block. The proposed method achieved ILA of 98.51% on 23 camera models of Dresden dataset.

Liao et al. [80] also explored the multiscale convolutional layer for extracting features from the input image. However, their approach differs from conventional RGB image feature extraction. Instead, features are extracted from high-pass filtered images, employing a High-Pass Filter (HPF) inspired by prior work [83]. This output noise residuals are passed to three convolutional layer with different kernel sizes. The outputs from these convolutional layers are concatenated and subsequently fed into a CNN-based network comprising four residual blocks. The proposed method achieves accuracy of 98.21% on 27 camera models of Dresden dataset. Apart from experimentation with original images, the experiments were performed on images post-processed with multiple IPOs such as contrast-enhancement, JPEG compression, median filtering, and resizing.

Bennabhaktula et al. [27] employed the MISLNet [85] for feature extraction and further classification. The only difference from MISLNet is the activation. The author have used ReLU activation function instead of TanH activation function. The proposed method have achieved ILA of 99.01% on 18 camera models of Dresden dataset.

## 2.4 Limitations of Prior Works

The majority of existing methods predominantly focus on extracting features primary from RGB images or high-pass filtered images. Some methods, such as Ding et al. [51], have explored fusion networks by incorporating both RGB images and multiple noise residuals. The exploration of robustness is crucial, necessitating the consideration of diverse feature images. Another noteworthy observation is that the majority of methods have conducted experiments on a single dataset. Only few methods [51, 28, 24] have extended their evaluations to encompass more than one dataset. Evaluating methods across multiple datasets is pivotal, highlighting another important aspect on the generalization capabilities of the proposed methods. Furthermore, assessing methods in a cross-dataset setting, where the model is tested on images from a distinct dataset, remains largely unexplored. This aspect is pivotal for understanding the broader generalization capacity of the method. An additional aspect worth noting is that prior works often prioritize their respective PSS,

as evident in studies like [14, 27]. The focus tends to shift more towards the PSS rather than the model itself. Furthermore, in comparative analyses, authors frequently overlook the importance of a fair comparison by not employing the same PSS. However, achieving this can be experimentally challenging, particularly considering the large number of image patches generated after patch extraction.

## 2.5 Summary

In this chapter, we have investigated the problem of CDI and CMI. The primary focus is on the deep learning based method for the CMI. We formulate a pipeline consisting of different stages of the deep learning based CMI method. The pipeline includes patch extraction and selection, preprocessing, and feature extraction and classification. We comprehensively provide the methodology of each method with respect to each stage. This provides a vision for the limitation of prior CMI methods and further improvements. We have also briefly explained the methodology of each method.

# Chapter 3

# Dual-branch Convolutional Neural Network for Camera Model Identification of Images

## 3.1 Introduction

In this chapter, we propose a dual-branch convolutional neural network (CNN) for camera model identification (CMI), where one branch directly uses the three-channel RGB image and the other uses a noise image obtained via high-pass filtering. For scalability, the method operates on cropped image patches and majority voting is used for image-level CMI. We conducted extensive experiments to evaluate the proposed method on multiple datasets and compare its performance against prior methods. For quantifying CMI accuracy, we use existing PLA and ILA metrics and also a new metric that we propose for assessing the robustness of image-level camera model estimates. Importantly, our evaluations and performance comparisons include cross-dataset scenarios where the evaluation is performed on a dataset different from and not necessarily represented by the training dataset. The significant improvements over prior methods that have used a single RGB or noise branch support our hypothesis that the proposed dual-branch architecture provides a convenient mechanism to introduce a favorable inductive bias in CNN architectures for CMI.

Traditional CMI methods that operate by examining the header (meta-data information) of the image file are not as applicable now-a-days because meta-data information can be altered or lost when images are shared on social media [5]. Therefore, researchers have been exploring image processing and deep learning based CMI methods considering the intrinsic traces left in the image during the image-acquisition process because of specific modules (software/hardware) in the imaging pipeline [98, 99]. For example, there are traces left by the camera lens, the Color Filter Array (CFA) pattern, and/or the camera sensor(s). These camera specific fingerprints are not visually observable, but are present and can be effectively utilized for CMI. Existing literature has primarily explored CMI for digital cameras, with only limited prior work addressing smartphone devices that are increasingly the dominant source of captured and shared images. There are many variations in sensor size, lens characteristics, image resolution and image processing techniques used, across different camera models, which makes it challenging to identify the camera model from the

captured image. This work aims to address these challenges by developing a deep learning based CMI method that works effectively across multiple datasets. So, we present a novel CMI method that utilizes a dual-branch architecture with two ResNet [16] CNN branches, with one operating on the RGB color image and the other on a pre-processed high-pass filtered image. Unlike most of existing works that are single branch models and apply a preprocessing layer for the extraction of camera model features, our dual-branch architecture is predicated on the hypothesis that the RGB images and corresponding high-pass filtered (noise) images carry complementary fingerprints indicative of the camera model, which the dual branches can effectively exploit and combine for CMI. In this sense, our method can also be seen as a convenient mechanism for introducing an advantageous inductive bias [100] in CNN architectures for CMI. Compared to single branch models, the inclusion of an additional branch generally increases the number of learning parameters and thus, the computational cost. However, the proposed high-pass filtered images based additional branch aims to extract better distinctive features that are specific to camera models. Extensive experiments on multiple datasets also demonstrate that the proposed dual-branch method provides superior accuracy compared to prior methods. Also, while the training cost is higher for the dual branch model, the cost of inference during use of the trained model is orders of magnitude lower than the cost of training and, for the dual branch model, will only be about two times that of a single branch model. For large scale deployments of forensic and related techniques, even relatively small increases in CMI accuracy are well-worth such modest increases in computation. This is because of the critical legal evidence CMI can provide, and also because, in many applications, CMI occurs at the front-end and errors in CMI can also adversely impact the accuracy of downstream tasks such as image authentication, image retrieval, and camera device identification [25]. Extensive experiments on multiple datasets demonstrate that the proposed method performs better than prior methods.

The rest of the chapter is organized as follows. Section 3.2 explains our proposed dual-branch CNN based CMI framework. Experiments and results including the comparison with alternative methods are presented in Section 3.3. Finally, we summaries the chapter in Section 3.4.

## 3.2 Proposed Camera Model Identification framework

The proposed framework for CMI comprises of two stages after patch extraction: a dual-branch feature extraction CNN and a classification network, as illustrated in Figure 3.1. To ensure scalability of the method to large and varying size images, we adopt the established paradigm of first extracting informative patches from the image, which then serve as inputs to dual-branch CNN that extracts features from which initially patch-level and then overall image-level estimates of the camera model are obtained. The patches are obtained by cropping with no resizing because resizing would affect the intrinsic correlations and noise features of the input image which are critical for identifying camera

model fingerprints and training the deep learning model. In the proposed dual-branch CNN, the first (top) CNN branch operates directly on the camera-captured RGB image (patch) $X$ and the second (bottom) CNN branch operates on a high-pass filtered version of the RGB image. Using two branches increases the computational cost in comparison to that of similar backbone based single-branch models, but we aim to improve the accuracy and robustness by extracting more distinctive features related to CMI. The branch operating on the RGB image is better suited to learning color related features, whereas the high-pass filtering de-emphasizes image spatial content in favor of camera model specific spatial features. Thus, we hypothesize that the proposed dual-branch architecture provides a convenient method for introducing a favorable inductive bias [100] in CNN architectures for CMI, which have previously been used with either the RGB branch alone, or a noise branch alone. The high-level features obtained from both branches are fused via concatenation to obtain the final camera model related features. In the second stage, a camera model estimate is obtained for each patch using a two-layer fully-connected classification network. Finally, the majority voting is applied on patch-level estimates to obtain the camera model estimate at the overall image-level.

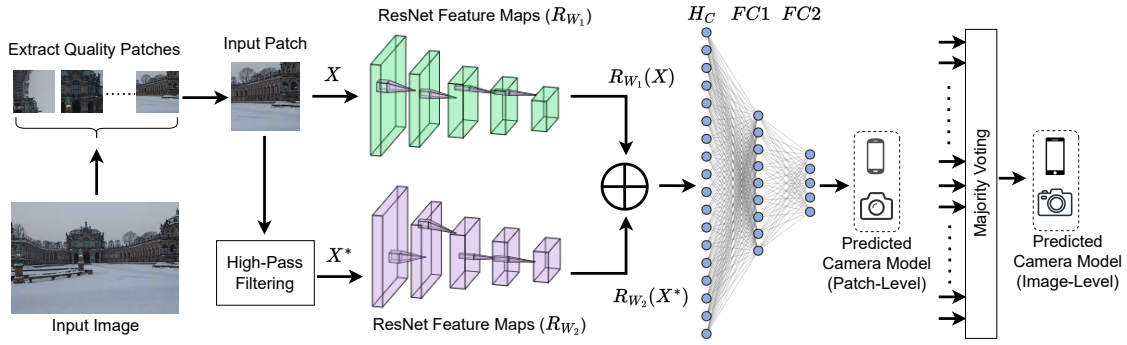



Figure 3.1: Framework of the proposed dual-branch CNN for CMI.

Table 3.1: Architecture of the proposed method

| **Network modules** | **Layers** | **Input size** | **Output size** | **Kernels** |
|---|---|---|---|---|
| RGB Feature Extractor ($R_{W1}$) | Conv1 ($B_1$) | $64 \times 64 \times 3$ | $32 \times 32 \times 64$ | $7 \times 7\,(64)$ |
| | Conv2_x ($B_2$) | $32 \times 32 \times 64$ | $16 \times 16 \times 256$ | $[1 \times 1\,(64), 3 \times 3\,(64), 1 \times 1\,(256)] \times 3$ |
| | Conv3_x ($B_3$) | $16 \times 16 \times 256$ | $8 \times 8 \times 512$ | $[1 \times 1\,(128), 3 \times 3\,(128), 1 \times 1\,(512)] \times 4$ |
| | Conv4_x ($B_4$) | $8 \times 8 \times 512$ | $4 \times 4 \times 1024$ | $[1 \times 1\,(256), 3 \times 3\,(256), 1 \times 1\,(1024)] \times 6$ |
| | Conv5_x ($B_5$) | $4 \times 4 \times 1024$ | $2 \times 2 \times 2048$ | $[1 \times 1\,(512), 3 \times 3\,(512), 1 \times 1\,(2048)] \times 3$ |
| | Average pooling | $2 \times 2 \times 2048$ | $1 \times 1 \times 2048$ | – – – – – – – |
| | High-Pass Filtering | $64 \times 64 \times 3$ | $64 \times 64 \times 3$ | $3 \times 3, 3 \times 3, 5 \times 5$ |
| Noise Feature Extractor ($R_{W2}$) | Layers are similar as in $R_{W1}$ | $64 \times 64 \times 3$ | $1 \times 1 \times 2048$ | – – – – – – – |
| Fusion Network | FC1 | $(2048 + 2048)$ | 2048 | – – – – – – – |
| | FC2 | 2048 | $K$ classes | – – – – – – – |

### 3.2.1 RGB Image Feature Extraction Branch

For the RGB feature extraction branch, we utilize the residual neural networks (ResNets) [16] as the ResNets provide residual connections straight to the earlier layers and negate the challenge of vanishing gradients, prevalent in many deep neural networks. The ResNet model comprises of five residual blocks, each of which is having several convolution layers, batch normalization, ReLU activation function and one skip connection. More specifically, we use ResNet50 for extracting feature maps from input RGB patch in the top branch of the proposed method. For effective learning, the ResNet50 model, pretrained on ImageNet, is fine-tuned during the training process. The architectural details with parameters of RGB feature extractor branch is provided in Table 3.1. Representing the function depicted by RGB feature extractor as $R_{W_1}(\cdot)$, where $W_1$ the set of weights of the trained RGB feature extractor, the output feature maps for the branch can be written as:

$$R_{W_1}(X) = (B_5(B_4(B_3(B_2(B_1(X)))))), \tag{3.1}$$

where $B_1, B_2, B_3, B_4$, and $B_5$ represent the residual blocks of RGB feature extractor.

$$F_1 = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_2 = \frac{1}{4}\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & 1 \end{bmatrix}$$

$$F_3 = \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -1 \\ -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \end{bmatrix}$$

Figure 3.2: Three high-pass filters ($F_1$, $F_2$, and $F_3$ ) used for extracting noise image from RGB image.

### 3.2.2 Noise Image Feature Extraction Branch

The noise image feature extraction branch is used in parallel with the RGB image feature extraction branch for extracting the enhanced camera model specific features by suppressing the image content information. The noise image has three channels corresponding to the three high-pass filters $F_1$, $F_2$, and $F_3$ of varied scales as shown in Figure 3.2. The motivation for selecting these filters is their proven better performance and wider adoption in steganalysis [63, 101, 102]. The filter $F_3$ has also been considered earlier for CMI [20]. The illustration of each high-pass filter output on two RGB images

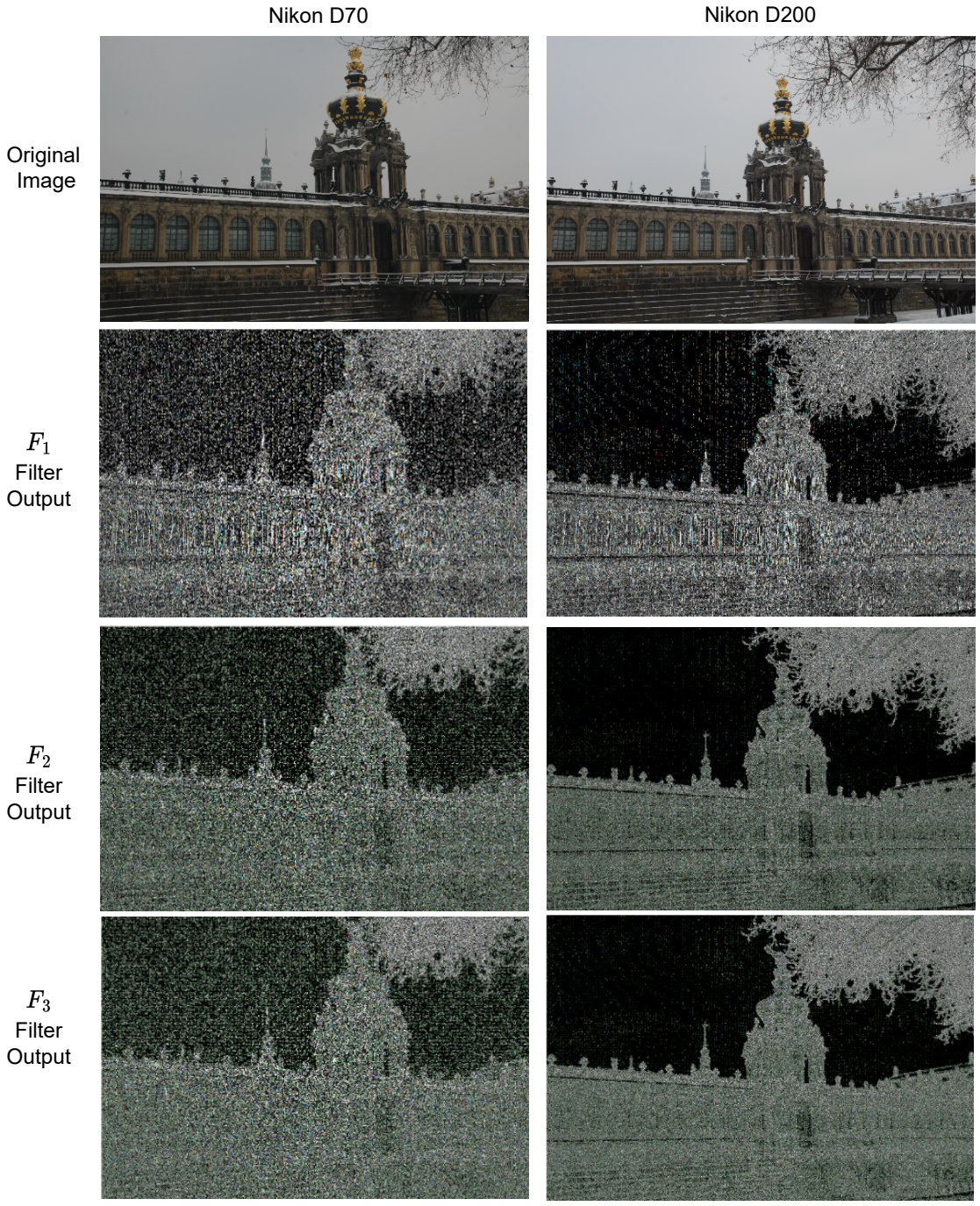



Figure 3.3: Illustration of output noise images ($X^*$). RGB image of original camera image convolved with three high-pass filters $F_1$, $F_2$, $F_3$ (top to bottom). The original images are from the Dresden dataset with almost identical scene content but captured with two different cameras, Nikon 70 and Nikon D200.

of similar content from different camera models is shown in the Figure 3.3 and we can observe that the difference in the respective images of the considered RGB images is quite visible. Each channel of the three-channel noise image ($X^*$), which serves as input to ResNet50 based feature extractor in this branch, is obtained from the input RGB image by performing the high-pass filtering using each of these above-specified filters individually. This whole procedure to generate $X^*$ is described in Algorithm 1, where $Z$ defines the three channel high-pass filter, derived by triplicating the same filter for convolution with the RGB input image and $\circledast$ represents a convolutional operator.

**Algorithm 1** High-Pass Filtering ($HPF$)

**Require:** $X$, $F_1$, $F_2$, $F_3$
**Ensure:** $X^*$
1: **for** $k \leftarrow 1$ to 3 **do**
2: **for** $l \leftarrow 1$ to 3 **do**
3: $Z(:,:,l) = F_k$
4: **end for**
5: $X^*(:,:,k) = X \circledast Z$
6: **end for**

The generated noise image ($X^*$) is given as input to ResNet50 of noise branch to extract high-level features. The high-pass filters used can help extract multi-scale noise features for better CMI performance. Although both branches have same architecture as shown in Table 3.1, but ResNet50 employed in the RGB image feature extraction branch does not share any weights with ResNet50 of the noise image feature extraction branch. The ResNet50s used in both the branches are pretrained on ImageNet [103] which are further fine-tuned simultaneously during the training process. The overall operation of the noise branch can be formulated in Eqs. 3.2 and 3.3. Considering $W_2$ as a set of weights of the trained ResNet50 based feature extractor $R_{W_2}(\cdot)$ of the noise branch, the output of this branch $R_{W_2}(X^*)$ can then be formulated as:

$$R_{W_2}(X^*) = (B_5(B_4(B_3(B_2(B_1(X^*)))))), \tag{3.2}$$

where,

$$X^* = HPF(X, F_1, F_2, F_3). \tag{3.3}$$

### 3.2.3 Fusion Network

The extracted features from RGB image feature extraction branch ($R_{W_1}(X)$) and noise image feature extraction branch $R_{W_2}(X^*)$ are fused together via concatenation ($\oplus$) in a vector as:

$$H_C = R_{W_1}(X) \oplus R_{W_2}(X^*), \tag{3.4}$$

which represents the combined high-level features for CMI. These final features $H_C$ are further provided to the fully-connected classification neural network for classification of camera models. This classification network consists of two fully-connected layers i.e. $FC1$

and $FC2$. The $FC1$ and $FC2$ consists of 2048 and $K$ nodes respectively, where $K$ is total number of camera models. The fused features $H_C$ are input to $FC1$ and the output of the $FC1$ layer forms the input for the final layer $FC2$. The output of a node in $FC1$ and $FC2$ is a linear combination of input features from the previous layer. The output of last layer is input to a softmax layer that outputs an estimated probability distribution over the camera models. The node with highest probability is labeled as the estimated camera model. The loss function used in the proposed method is the cross-entropy loss,

$$L = -\log \frac{\exp(o_i)}{\sum_{j=1}^{K} \exp(o_j)}, \tag{3.5}$$

where $o_i$ is the output of the final layer's $i^{th}$ node, corresponding to the true camera model and $\exp(\cdot)$ is the exponential function.

## 3.3 Experiments and Results

We evaluated the performance of the proposed method on multiple datasets. In the following, we first describe the experimental setup, the training and testing strategies, and the four datasets used in the experiments. Then, we discuss the evaluation metrics we use, which include two commonly used prior evaluation metrics for CMI and a new metric that we motivate and propose, which advantageously also quantifies the robustness of the image-level classification. We also propose a new evaluation metric for potentially deeper considerations. Lastly, we present extensive results on multiple datasets and a detailed comparison and discussion, also exploring alternative settings for the proposed method.

### 3.3.1 Experimental Setup

As indicated in Section 3.2, for scalability, multiple crops from an original image, with no resizing or other pre-processing are used as inputs for the proposed method. Specifically, we used $64 \times 64 \times 3$ patches. Images acquired now-a-days via smartphone or other digital cameras, and also the majority of images in the considered datasets, are larger than $1024 \times 1024 \times 3$ pixels and yield 256 or more patches. The patches with saturated regions typically do not provide discerning information for CMI. Therefore, we extract 256 quality patches per image using the [14] patch quality measure

$$Q(\mathcal{P}) = \frac{1}{3} \sum_{c \in [R,G,B]} \left[ \alpha \cdot \beta \cdot \left( \mu_c - \mu_c^2 \right) + (1 - \alpha) \cdot (1 - \exp(\gamma \sigma_c)) \right], \tag{3.6}$$

where $\alpha$, $\beta$ and $\gamma$ are empirically set constants with values 0.7, 4 and $\ln(0.01)$ respectively. $\mu_c$ and $\sigma_c$ are the mean and standard deviation of respective channel $c \in [R, G, B]$.
Prior to patch selection, images in a dataset are apportioned in a 80:20 ratio into training and test subsets. This ensures that the training and test data do not share patches in common that are derived from the same image. All the experiments are performed on an 2.40 GHz 2X Intel Xeon Silver 4210R system with 128 GB RAM and equipped with two

Nvidia V100 GPUs each having 32 GB memory. The proposed and the other competing models considered in the experiments were implemented in PyTorch (version 1.8.1) with Torchvision (version: 0.9.1). All other required code was written in Python. For training the deep neural network models, we used the Adam optimizer [104] with a learning rate of 0.0001 and default settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$, zero weight decay, and a batch size of 128. All the considered models were trained for 100 epochs and for each model, it was observed that the training loss was converging by the 100 epochs or earlier.

### 3.3.2 Datasets Used

We used four datasets in our work: the Dresden [30] dataset which has been extensively utilized in prior camera forensics work and three public smartphone camera datasets [4, 82, 96]. The smartphone camera datasets are particularly relevant given that the overwhelming majority of captured images nowadays come from smartphones. We summarize the relevant characteristics and considerations for each of these datasets as follows:

- The Dresden dataset [30] contains more than 16000 images acquired using 73 digital cameras representing 25 different camera models. The images feature diverse lighting conditions and image content (indoor/outdoor, public places, trees etc). To more closely reflect real-world situations, the images were captured with varied camera settings (e.g. focal length, flash on/off). As in prior works [14, 24, 25], for our experiments, we only include around 15000 images captured with the 18 camera models for which the dataset contains multiple cameras and exclude images from camera models that were represented only by one single camera device.

- The SOurce Camera REcognition on Smartphones (SOCRatES) dataset provided by [96] contains images from 65 camera models. The dataset includes the largest number of smartphone models in existing and publicly available source camera identification datasets. The dataset is also more diverse and representative of real-world situations because the images were collected by individual smartphone owners themselves.

- The Forchheim dataset [4] provides around 4000 images captured using 25 different smartphone camera models, with varied scene content and different capture conditions: indoor/outdoor, day/night, and close-up/distant. By providing images of the same scene captured with the different devices, the dataset helps minimize the role of image content as an obfuscating factor in CMI. Additionally, the dataset also contains versions of the images post-processed using five different popular social media platforms: WhatsApp, Facebook, Instagram, Telegram, and Twitter. CMI on the post processed images represents more realistic application scenarios because the social media platforms are commonly used to share images.

- The SP Cup dataset [82] is provided by the IEEE Signal Processing society. It contains a total of 2750 images captured using 10 different smartphone camera

devices and each of these devices are of different smartphone camera model.

The numbers of images, camera models, and individual (camera) devices in the datasets used for the experiments are summarized in Table 3.2.

Table 3.2: Details of datasets used in experiments

| **Dataset** | **Total images** | **Total camera models** | **Total devices** |
|---|---|---|---|
| Dresden [30] | 14999 | 18 | 66 |
| Socrates [96] | 9721 | 65 | 103 |
| Forchheim [4] | 3851 | 25 | 27 |
| IEEE SP Cup [82] | 2750 | 10 | 10 |

### 3.3.3 Evaluation Metrics

The majority of the prior works have used the accuracy metric or image-level accuracy (ILA) to evaluate the performance of the CMI methods. Also, some of the CMI methods [14, 25, 27] based on extracting patches from the test images followed by majority voting for the estimation of camera model used patch-level accuracy (PLA) for analysis. We consider both of these evaluation metrics in our analysis and we also propose a new evaluation metric: Average Percentage of Majority class Votes for Correctly estimated images (APMVC) that, advantageously, also characterizes the robustness of the image-level camera model estimate provided by the model. Let $N$ and $C$ denote the total number of images and camera models in the dataset, $X_i, Y_i, \hat{Y}_i$ denote the selected $i^{th}$ image, its true camera model, and estimated camera model, respectively, and let $P_i$ denote the total number of patches from the $i^{th}$ image. $y_{ij}$ and $\hat{y}_{ij}$ denote the true camera model, and estimated camera model respectively, for the $j^{th}$ patch of $i^{th}$ image. The PLA, ILA, and APMVC metrics are then given by

$$\text{PLA} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{P_i}\mathbb{I}\left(\hat{y}_{ij} = y_{ij}\right)}{\sum_{i=1}^{N} P_i}, \tag{3.7}$$

$$\text{ILA} = \frac{\sum_{i=1}^{N}\mathbb{I}\left(\hat{Y}_i = Y_i\right)}{N}, \tag{3.8}$$

$$\text{APMVC} = \frac{\sum_{i=1}^{N}\left(\left(\sum_{j=1}^{P_i}\mathbb{I}\left(\hat{y}_{ij} = y_{ij}, \hat{Y}_i = Y_i\right)\right)/P_i\right)}{\sum_{i=1}^{N}\left(\mathbb{I}\left(\hat{Y}_i = Y_i\right)\right)}, \tag{3.9}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\hat{Y}_i$ corresponds to camera model with highest number of votes as per the estimations of the patches of image $X_i$, $\hat{Y}_i$ is estimated as

$$\hat{Y}_i = \arg\max_{l}\left(\sum_{j=1}^{P_i}\mathbb{I}\left(\hat{y}_{ij} = l\right)\right). \tag{3.10}$$

### 3.3.4 Results and Discussion

The performance of the proposed dual-branch CMI method is evaluated in detail considering several different scenarios, including varying patch selection strategies and the cross-dataset settings. We also highlight the benefit of the proposed dual-branch method by comparing against single branch alternatives. The competing CMI methods considered for performance comparison are [3], [14], [15], [25], [29], [27], [19], and [24]. Out of these methods, the CMI methods [25, 29, 27, 24] are pre-processing based CNNs and [3, 14, 15, 19] are CNN based methods that do not use a pre-processing stage.

Table 3.3: Results on All Datasets considering 256 maximum quality patches of size $64 \times 64$ per image.

| **Dataset** | **Dresden** | | | **Socrates** | | |
|---|---|---|---|---|---|---|
| **Method** | **PLA** | **ILA** | **APMVC** | **PLA** | **ILA** | **APMVC** |
| Bondi et al. [14] | 90.93 | 96.73 | 93.27 | 63.09 | 79.36 | 76.08 |
| Chen et al. [15] | 99.07 | 99.90 | 99.14 | 94.19 | 98.04 | 95.73 |
| Yao et al. [19] | 90.63 | 99.60 | 90.88 | 65.94 | 83.57 | 77.38 |
| Freire-Obregon et al. [3] | 93.17 | 98.30 | 94.24 | 84.73 | 93.68 | 89.21 |
| You et al. [29] | 98.00 | 99.86 | 98.07 | 92.33 | 98.20 | 93.61 |
| Liu et al. [25] | 97.53 | 98.50 | 98.51 | 93.88 | 97.74 | 95.46 |
| Rafi et al. [24] | 98.81 | 99.93 | 98.85 | 96.06 | 98.51 | 97.25 |
| Bennabhaktula et al. [27] | 98.52 | 99.90 | 98.58 | 91.59 | 97.68 | 93.18 |
| **Proposed CMI method** | **99.19** | 99.90 | **99.25** | **96.58** | **98.66** | **97.63** |
| | | | | | | |
| **Dataset** | **Forchheim** | | | **IEEE SP Cup** | | |
| **Method** | **PLA** | **ILA** | **APMVC** | **PLA** | **ILA** | **APMVC** |
| Bondi et al. [14] | 74.03 | 93.61 | 77.41 | 90.25 | 98.54 | 91.22 |
| Chen et al. [15] | 91.62 | 99.10 | 92.08 | 96.93 | 100 | 96.93 |
| Yao et al. [19] | 74.46 | 96.93 | 76.09 | 90.78 | 99.45 | 91.11 |
| Freire-Obregon et al. [3] | 84.71 | 98.97 | 85.29 | 94.83 | 99.81 | 94.97 |
| You et al. [29] | 89.66 | 99.61 | 89.89 | 96.99 | 100 | 96.99 |
| Liu et al. [25] | 96.27 | 99.48 | 96.56 | 98.77 | 99.81 | 98.89 |
| Rafi et al. [24] | 97.11 | 99.87 | 97.18 | 99.27 | 100 | 99.27 |
| Bennabhaktula et al. [27] | 91.78 | 99.74 | 91.97 | 97.2 | 99.63 | 97.44 |
| **Proposed CMI method** | **97.59** | **100** | **97.59** | **99.33** | 99.81 | **99.47** |

#### Comparison considering varying Patch Selection Strategies

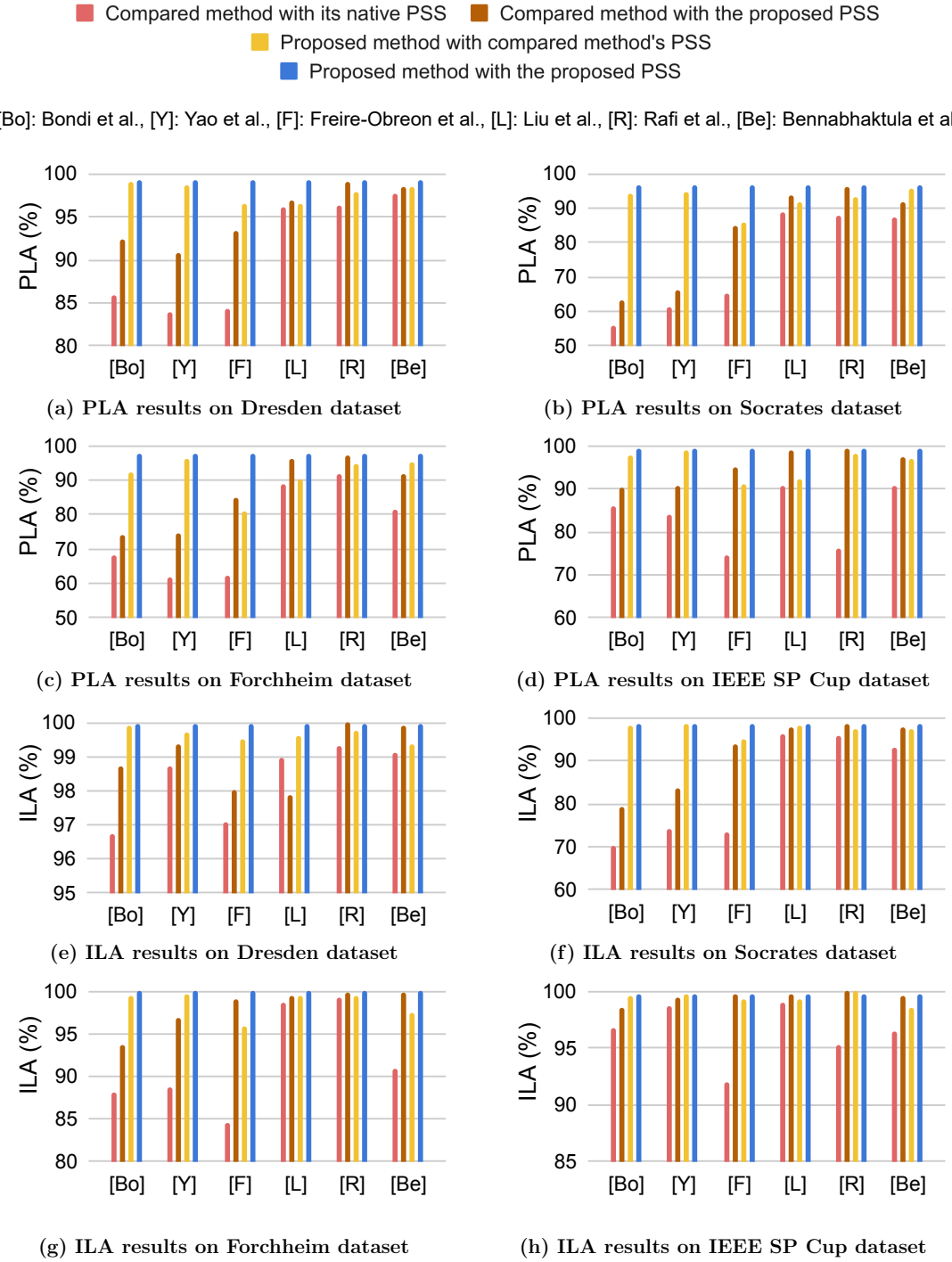
The CMI methods considered extract and select the patches from the input image via a patch selection strategies (PSS) before feeding to the CNN network except the method [15] which resizes the input image to a fixed size image. However, the patch selection strategies used in different CMI methods are generally not the same and do play a significant role in the performance of the model, as also discussed later in this section. So for fair comparison, we first compare the proposed method with other CMI methods using the same PSS as used in the proposed method and then followed by a thorough comparison with different methods. We further evaluate the individual methods using their native PSS.

Table 3.4: Different methods patch selection criteria

| **Method** | **Number of patches per image** | **Patch size** | **Patch selection strategy** |
|---|---|---|---|
| Bondi et al. [14] | 32 | 64×64 | Extract patches based on value of quality measure (Q) |
| Yao et al. [19] | 256 | 64×64 | Extract patches from central 75% of the image. |
| Freire-Obregon et al. [3] | 256 | 32×32 | Not mentioned the exact criteria. Assuming patches from center. |
| Liu et al. [25] | 128 | 64×64 | 64 patches based on quality measure (Q). + 64 patches using K-means clustering and nearest patches. |
| Rafi et al. [24] | 20 | 256×256 | Extract patches based on quality measure (Q). |
| Bennabhaktula et al. [27] | 400 | 128×128 | Extract homogeneous patches based on standard deviation. |
| **Proposed** | **256** | **64 × 64** | **Extract patches based on value of quality measure (Q)** |

Table 3.3 shows the performance of different methods using proposed PSS on the four different datasets. In terms of PLA, the proposed method consistently shows better performance than all other competing CMI methods on all the datasets. In terms of ILA, the performance of 99.93% is achieved by [24] on Dresden dataset and perfect performance of 100% is achieved by three methods [15, 29, 24] including [24] on the IEEE SP dataset. The proposed method performs marginally worse, by around 0.05% , as it mis-classifies one additional image in each of these two datasets. Also, the proposed method provides the best ILA performance of 98.66% and 100% on the larger smartphone camera based datasets i.e. Socrates and Forchheim, respectively. This is significant as these two datasets have larger number of classes (65 and 25) and the number of training images per class was smaller in comparison to that of other two datasets. In terms of APMVC also, the proposed method provides the best performance in comparison to all other methods on all the datasets. Further, in some cases when the ILA performance is identical, the APMVC performance of the proposed method is better. For example on the Dresden dataset, the APMVC of the proposed method is better than that of [15] and on IEEE SP dataset, APMVC of the proposed method is again better than that of [3] and [25]. Overall the performance of many other methods in this setting was quite competing, but it may be noted that we used the proposed PSS here.

Table 3.4 shows the PSS strategies of different methods including the proposed PSS. Two methods [15] and [29] are not considered here as there was only resizing in [15] with no patch selection and the overall PSS is not clearly specified in [29]. In this detailed comparison with the other methods [24, 14, 3, 25, 19], we consider four different experiment settings: (i) compared method with its native PSS, (ii) compared method with the proposed PSS, (iii) proposed method with compared method's PSS and (iv) proposed method with the proposed PSS. The plots in Figure 3.4 (a-h) compare the




(a) PLA results on Dresden dataset

(b) PLA results on Socrates dataset

(c) PLA results on Forchheim dataset

(d) PLA results on IEEE SP Cup dataset

(e) ILA results on Dresden dataset

(f) ILA results on Socrates dataset

(g) ILA results on Forchheim dataset

(h) ILA results on IEEE SP Cup dataset

Figure 3.4: Comparison of classification accuracy for the proposed method vs alternative methods with different patch selection strategies (PSSs). Sub-figures (a)-(d) show the patch-level accuracy (PLA) and (e)-(h) show the image-level accuracy (ILA) for the Dresden, SOCRATES, Forchheim, and IEEE SP Cup datasets, respectively.

PLA and ILA metrics for all these 4 settings on all four datasets. The numerical related to comparison, over all datasets, of the proposed and alternative methods, with both adopting the native PSS of the alternative method is shown in Table. 3.5. In all the cases, both PLA and ILA of other methods improved upon using the proposed PSS in comparison to their native PSS and the proposed method with another method's PSS also consistently provided better performance than the method using its native PSS. The robustness of the proposed method is also further illustrated here; compared to other methods, the proposed method suffers from a smaller drop in ILA and PLA performance metrics when using another method's PSS instead of the proposed PSS. Comparisons of the proposed method's performance with that of other methods using their native PSS, highlight the superior performance of the proposed method. Compared to [24], the proposed method provides the PLA improvements of 1.56%, 6.27%, 4.64% and 28.79% and the ILA improvements of 0.48%, 1.61%, 0.77% and 4.96% on Dresden, Socrates, Forchheim and IEEE SP cup datasets, respectively. Therefore, the proposed method achieves an the overall ILA and PLA improvement of 1.96% and 10.31%, respectively, compared with the [24] method. The overall improvements in ILA and PLA in comparison to other methods [14, 3, 19] are even more than 11% and 24%, respectively.

### Cross-Dataset Evaluation

To further evaluate the robustness of proposed method, we experimented with cross-dataset setting which includes images of 6 common smartphones (Apple iPhone 6s, Samsung Galaxy S4 mini, Apple iPhone 7, Samsung Galaxy S4, Google Nexus 5, Huawei P8 lite) of Forchheim and Socrates test set. These camera models are common in both the datasets. In this experiment, we selected images from one dataset smartphones for training the model and tested on images of another dataset. The results are shown in Table 3.6. In the results, we observed that the proposed method performs best in terms of PLA and ILA among all other methods. The proposed method provides the improvement of 1.84% and 5.17% in terms of PLA and ILA, respectively as compared to second best method [25] when all methods are trained on Socrates images and tested on Forchheim images. Also, the proposed method provides the improvement of 1.93% and 3.52% in terms of PLA and ILA, respectively as compared to second best method [25] when all methods are trained on Forchheim images and tested on Socrates images.

### Effectiveness of Dual-Branch Fusion Method

We explore and discuss the effectiveness of the dual-branch fusion method used in the proposed method by comparing it with single-branch architectures (either RGB or noise image) obtained by dropping one of the branches. We also vary the number of patches extracted per image. For these comparison experiments, we use the Forchheim dataset because it provides similar image content across all camera models. The plots in Figure 3.5 present the comparative results in terms of PLA, ILA, and APMVC and the numerical values corresponding to the plots are shown in Table 3.7. We also computed the inference

Table 3.5: Comparison, over all datasets, of the proposed and alternative methods, with both adopting the native PSS of the alternative method

| **Dataset** | **Dresden** | | | **Socrates** | | |
|---|---|---|---|---|---|---|
| **Method** | **PLA** | **ILA** | **APMVC** | **PLA** | **ILA** | **APMVC** |
| Bondi et al. [14] | 85.94 | 96.72 | 93.53 | 56.02 | 70.27 | 74.75 |
| **Proposed CMI method** | **99.02** | **99.91** | **99.08** | **94.31** | **97.99** | **95.83** |
| Yao et al. [19] | 84.00 | 98.72 | 92.78 | 61.30 | 74.07 | 80.32 |
| **Proposed CMI method** | **98.64** | **99.73** | **98.81** | **94.79** | **98.35** | **96.10** |
| Freire-Obregón et al. [3] | 84.36 | 97.07 | 86.36 | 65.11 | 73.51 | 85.46 |
| **Proposed CMI method** | **96.57** | **99.51** | **96.87** | **85.89** | **95.02** | **89.43** |
| Liu et al. [25] | 96.17 | 98.98 | 96.85 | 88.59 | 96.25 | 91.20 |
| **Proposed CMI method** | **96.45** | **99.60** | **96.71** | **91.56** | **97.94** | **92.96** |
| Rafi et al. [24] | 96.29 | 99.29 | 96.84 | 87.58 | 95.89 | 90.36 |
| **Proposed CMI method** | **97.79** | **99.77** | **97.93** | **93.07** | **97.43** | **94.86** |
| Bennabhaktula et al. [27] | 97.71 | 99.10 | 98.44 | 87.50 | 93.06 | 93.07 |
| **Proposed CMI method** | **98.49** | **99.36** | **98.95** | **95.77** | **97.27** | **98.00** |
| | | | | | | |
| **Dataset** | **Forchheim** | | | **IEEE SP Cup** | | |
| **Method** | **PLA** | **ILA** | **APMVC** | **PLA** | **ILA** | **APMVC** |
| Bondi et al. [14] | 68.15 | 88.12 | 74.67 | 85.94 | 96.72 | 87.87 |
| **Proposed CMI method** | **92.28** | **99.36** | **92.76** | **97.73** | **99.63** | **98.02** |
| Yao et al. [19] | 61.76 | 88.63 | 67.51 | 84.00 | 98.72 | 84.79 |
| **Proposed CMI method** | **95.97** | **99.61** | **96.25** | **98.76** | **99.81** | **98.87** |
| Freire-Obregón et al. [3] | 62.30 | 84.54 | 70.49 | 74.58 | 92.00 | 79.73 |
| **Proposed CMI methodd** | **80.92** | **95.91** | **83.16** | **91.17** | **99.27** | **91.68** |
| Liu et al. [25] | 88.87 | 98.59 | 89.62 | 90.77 | 99.09 | 91.33 |
| **Proposed CMI method** | **90.02** | **99.48** | **90.25** | **92.25** | **99.27** | **92.73** |
| Rafi et al. [24] | 90.38 | 98.72 | 90.86 | 76.03 | 95.27 | 78.29 |
| **Proposed CMI method** | **94.57** | **99.48** | **94.81** | **97.92** | **100** | **97.92** |
| Bennabhaktula et al. [27] | 81.46 | 90.80 | 88.03 | 90.60 | 96.54 | 92.97 |
| **Proposed CMI method** | **95.21** | **97.44** | **97.28** | **96.72** | **98.54** | **97.75** |

Table 3.6: Results in cross-dataset settings

| **Dataset** | **Train dataset: Socrates, Test dataset: Forchheim** | | **Train dataset: Forchheim, Test dataset: Socrates** | |
|---|---|---|---|---|
| **Method** | **PLA** | **ILA** | **PLA** | **ILA** |
| Bondi el al. [14] | 47.87 | 58.04 | 56.41 | 62.32 |
| Chen el al. [15] | 57.44 | 67.24 | 55.77 | 64.78 |
| Yao el al. [14] | 19.20 | 21.26 | 32.19 | 38.02 |
| Freire-Obregon el al. [3] | 48.36 | 58.62 | 52.88 | 60.21 |
| You el al. [29] | 58.39 | 66.66 | 62.26 | 68.30 |
| Liu el al. [25] | 70.99 | 79.88 | 67.40 | 72.53 |
| Rafi el al. [24] | 61.76 | 68.39 | 65.89 | 75.00 |
| Bennabhaktula el al. [27] | 49.36 | 46.71 | 46.71 | 51.06 |
| **Proposed CMI method** | **72.83** | **85.05** | **69.33** | **78.52** |

time for all patches in all three cases and as anticipated, the inference time for each patch in the dual-branch method is around 1.84 and 1.83 times that of RGB branch and noise

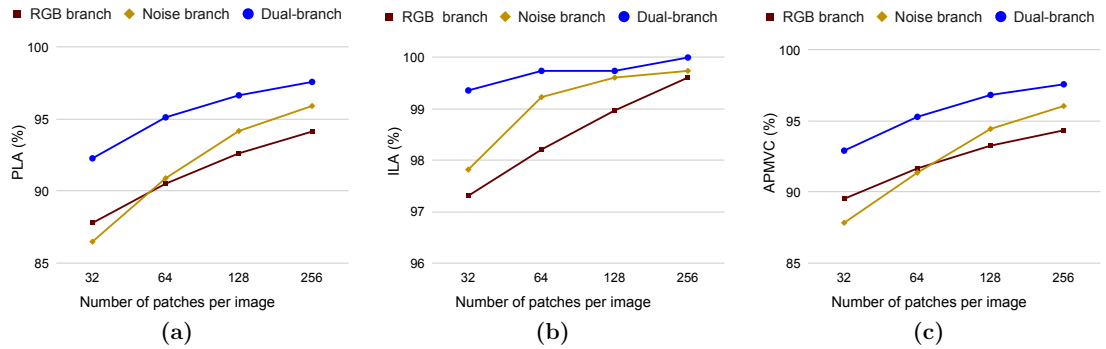



Figure 3.5: Comparison of results using only RGB branch, only noise branch and with dual-branch (Fusion) with choosing different patches per image on Forchheim dataset. PLA, ILA, APMVC (left to right)

branch, respectively. However, it can be noted from Figure 3.5 that irrespective of the number of patches used in a method, the dual-branch method performs consistently better than either the RGB branch or the noise branch alone on all three evaluation metrics considered. It can also be observed that increasing the number of patches improved the performance in all cases on all three evaluation metrics considered. These results support our hypothesis that the proposed dual-branch architecture provides a convenient mechanism for introducing an effective inductive bias for CMI.

Table 3.7: Performance on the Forchheim dataset for the proposed dual-branch approach and approaches using only one of the two branches (RGB or noise), considering different number of patches per image.

| **No. of patches per image** | **Evaluation metrics** | **RGB branch only** | **Noise branch only** | **Dual-branch** |
|---|---|---|---|---|
| 32 | PLA | 87.79 | 86.49 | **92.28** |
| | ILA | 97.31 | 97.82 | **99.36** |
| | APMVC | 89.53 | 87.84 | **92.92** |
| 64 | PLA | 90.52 | 90.90 | **95.13** |
| | ILA | 98.21 | 99.23 | **99.74** |
| | APMVC | 91.67 | 91.37 | **95.30** |
| 128 | PLA | 92.63 | 94.18 | **96.66** |
| | ILA | 98.97 | 99.61 | **99.74** |
| | APMVC | 93.27 | 94.45 | **96.84** |
| 256 | PLA | 94.14 | 95.92 | **97.59** |
| | ILA | 99.61 | 99.74 | **100** |
| | APMVC | 94.35 | 96.06 | **97.59** |

### Ablation Study with Different CNN based Models

In this section, we present results from an ablation study that explores differ-ent ResNet (He et al., 2016) methods as the feature extractor for both branches in the proposed method. Table S.3 summarizes the results for the proposed method on the Forchheim dataset with four different ResNet methods used for feature extraction: ResNet34,

ResNet50, ResNet101 and ResNet152. It can be seen that ResNet50 provides the highest PLA and APMVC. ResNet50, ResNet101, and ResNet152 provide 100% ILA. However, the number of param- eters in ResNet50 is significantly smaller than in ResNet101 and ResNet152. In addition to ResNet models, we experimented with VGG16, VGG19, DenseNet121, and EfficentNetB0 CNN models as the feature extractor.

The results with these models are shown in Table 3.8. All of these models exhibit excellent performance on the Forchheim dataset, achieving 100% ILA. ResNet50 provides slightly better PLA. ResNet50 is selected as the primary feature extractor for the proposed technique. With the exception of DenseNet121, the dual-branch framework achieves 100% ILA for each of these CNN models as the feature extractor, highlighting the effectiveness of the overall proposed architecture and the high-pass filtering employed.

Table 3.8: PLA, ILA, and APMVC on the Forchheim dataset for the ablation study with different CNN based models used as the feature extractor for the proposed method.

| **Model** | **PLA** | **ILA** | **APMVC** |
|---|---|---|---|
| VGG16 [93] | 97.16 | **100** | 97.16 |
| VGG19 [93] | 97.22 | **100** | 97.22 |
| DenseNet121 [87] | 97.44 | 99.74 | 97.33 |
| EfficientNetB0 [105] | 97.51 | **100** | 97.51 |
| ResNet34 [16] | 97.10 | 99.87 | 97.08 |
| **Proposed (ResNet50)** | **97.59** | **100** | **97.53** |
| ResNet101 [16] | 97.11 | **100** | 96.99 |
| ResNet152 [16] | 97.44 | **100** | 97.33 |

## 3.4 Summary

In this chapter, we addressed the challenge of CMI in digital images. The dual-branch CNN-based framework introduced in this chapter presents a novel, efficient, and robust solution for identifying the camera model used in capturing an image. Comparative to previous methods, our proposed approach demonstrates substantial enhancements in CMI accuracy. In cross-dataset scenarios, where evaluation images not only differ from the training set but are sourced from an entirely distinct dataset, our method achieves improvements in PLA ranging from 1.8% to 1.9%, and ILA between 3.5% and 5.2%. Additionally, our method enhances CMI robustness, measured by a new metric, APMVC, introduced specifically for this purpose. However, it's worth noting that the presented method, along with many explored in prior work, operates on original images without accounting for post-processing operations that may occur when images are shared on social media platforms.

# Chapter 4

# Source Social Media Platform Identification of Images and Camera Model Identification of Social Media Post-processed Images

## 4.1 Introduction

The focus of CMI has traditionally been on discerning the specific camera model associated with a given digital image. However, with the pervasive role of social media platforms in image sharing, there is an increasing need to extend the scope of CMI to include the identification of camera models for post-processed images disseminated through these platforms. The recognition of the Source Social Media Network (SSMN) for digital images has emerged as a critical concern within the image forensic scientific community. Consequently, evaluating the robustness of CMI methods, particularly on social media post-processed images, has become imperative. This study delves into the challenge of identifying the SSMN before determining the camera model of an image. This unique approach facilitates the assessment of CMI method robustness in two distinct scenarios: firstly, by identifying the source social media platform of an image and subsequently applying the CMI method; and secondly, by evaluating the robustness of the CMI method without prior knowledge of the SSMN. Additionally, it is noteworthy that social media platforms often employ Image Post-Processing Operations (IPOs) when images are uploaded. This complicates the accurate detection of IPO on a digital image. Identifying Image Post-Processing Operations (IPOs) is commonly referred to as General Purpose Image Manipulation Detection (GIMD) [106].

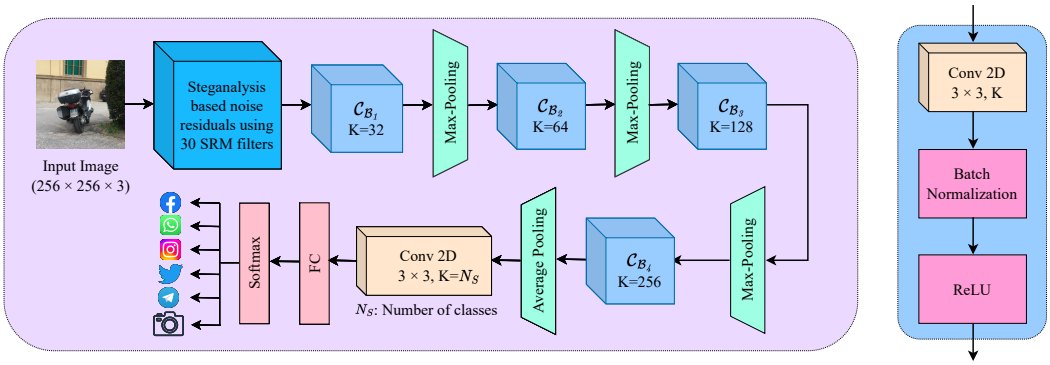
In recent years, with the technological advancement of smartphones and social media platforms, a large number of images are being shared over the internet on a daily basis. In this social network ecosystem, digital images have become a major source of real-time information. Moreover, digital images play a critical role as a piece of evidence in the judicial courts. However, this ecosystem also provides an accessible channel for proliferating illegal activities such as spreading fake news, hurting religious

sentiments, violence and terrorism provocation, and defamation activities. Therefore, it is imperative to find the provenance of the digital images [99, 107]. Identification of the social networks such as Facebook, WhatsApp, Instagram, etc., used for sharing digital images is a critical task in the area of source image forensics. It is observed that these social media platforms introduced specific artifacts in the images during the process of uploading and downloading. The extraction of these artifacts can be used as a signature to identify the social media platforms. The aim of the social network identification of an image is to find the social network from which the image is downloaded and saved in the media device. This forensic analysis helps in combating cybercrime by locating the source/origin of the images.

Most of the existing SSMN identification methods focused on the analysis of traces/artifacts introduced by the post-processing operations of social media networks. Some works were dedicated to the detection of particular post-processing operations [85, 108]. However, SSMN identification is a challenging task because the parameters of the post-processing operations used by social media networks are not available publicly. Also, different versions of a social network may use different post-processing operations with different parameters. Therefore, researchers explored data-driven approaches to find the related artifacts in the images. CNNs are widely used as feature extractor in the data-driven approaches for image forensic problems [14, 86, 109, 110]. Most existing works on SSMN identification transform the given image into different domains to highlight the artifacts and then pass the transformed image to a CNN classifier. In [111], histogram of the Discrete Cosine Transform (DCT) coefficients of the input image is passed to a CNN based classifier for social network detection. Researchers also explored the artifacts left in the Photo Response Non-Uniformity (PRNU) of the image for SSMN classification. The PRNU is the major component of sensor pattern noise which is generally used to identify the source digital camera that captured the image [11, 31]. It is observed in [112] that different social networks may introduce unique artifacts in the PRNU. The method proposed in [112] used a Wiener filter to extract the PRNU noise of the image and then these noise features are passed to CNN based classifier. Amerini et al. [113] proposed a dual-channel CNN for SSMN identification by exploring two different transformations of the given input image. The input to the first CNN branch is the 909 elements vector which is extracted from 101 histogram bins of first 9 DCT frequencies in zigzag order. The input to the second CNN is the PRNU noise of the image. It is observed that the PRNU of an image can be diminished by scene details [37] or ISO settings [114]. Therefore, PRNU based method works when the strong PRNU is present in the image [115]. In the recent work [115], researchers utilized the transfer learning approach for extracting features and classification. The images are first transformed into DCT/Discrete Wavelet Transform (DWT) domain, as DCT and DWT are the basis of JPEG [116] and JPEG2000 [117] compression formats, respectively. Afterwards, the transformed image is passed to VGG16 [93] for high-level features extraction and classification. In this work [115], the input image is resized to $224 \times 224$ to meet the VGG16 model requirement, which may cause

information loss. In [118], researchers applied a three-branch CNN model (MSF-CNN) on the DWT transformed image for extracting multi-scale content insensitive features for the identification of SSMN. It is observed that most social media networks use JPEG format for image compression [119]. However, social networks may also apply rescaling or other unknown post-processing operations apart from image compression.

With the precedence of deep Convolutional Neural Networks (CNNs) in the domain of image forensics, we propose a novel Steganalysis Noise Residuals based CNN (SNRCN2) to extract pertinent features related to social networks for SSMN identification of an image. We consider the suppression of image content information for developing better SSMN identification method. We employ the steganalysis-based high-level SRM filters to suppress the image content information and extract the noise residuals from an image which are then passed to a CNN for SSMN identification. We perform a set of ablation studies to select the optimal parameters for designing the CNN. The rest of the chapter is organized as follows. Section 4.2 explains a detailed description of the proposed SNRCN2. Section 4.3 includes various experimental results including the comparative analysis. Further in the section 4.3.5, we have evaluate the CMI method robustness on social media network post-processed images. Furthermore, Section 4.4 extends the scope by incorporating SSMN identification into the broader context of GIMD.




(a) Overall architecture, $\mathcal{C}_{\mathcal{B}_1}$, $\mathcal{C}_{\mathcal{B}_2}$, $\mathcal{C}_{\mathcal{B}_3}$, and $\mathcal{C}_{\mathcal{B}_4}$ denotes four different CBR blocks, K denotes number of filters

(b) CBR block ($\mathcal{C}_{\mathcal{B}}$)

Figure 4.1: The architecture of SNRCN2.

## 4.2 Proposed Method

This section describes our SNRCN2 method for the SSMN identification of a given input image. The proposed architecture is provided in Figure 4.1. The proposed method consists of following steps: extraction of steganalysis based noise residuals using 30 SRM filters from the given input image for content information suppression, extraction of high-level hierarchical features related to social network platforms from noise residuals using a robust CNN, and finally the SSMN classification based on these high-level features. These steps are further described in more detail in following two subsections.

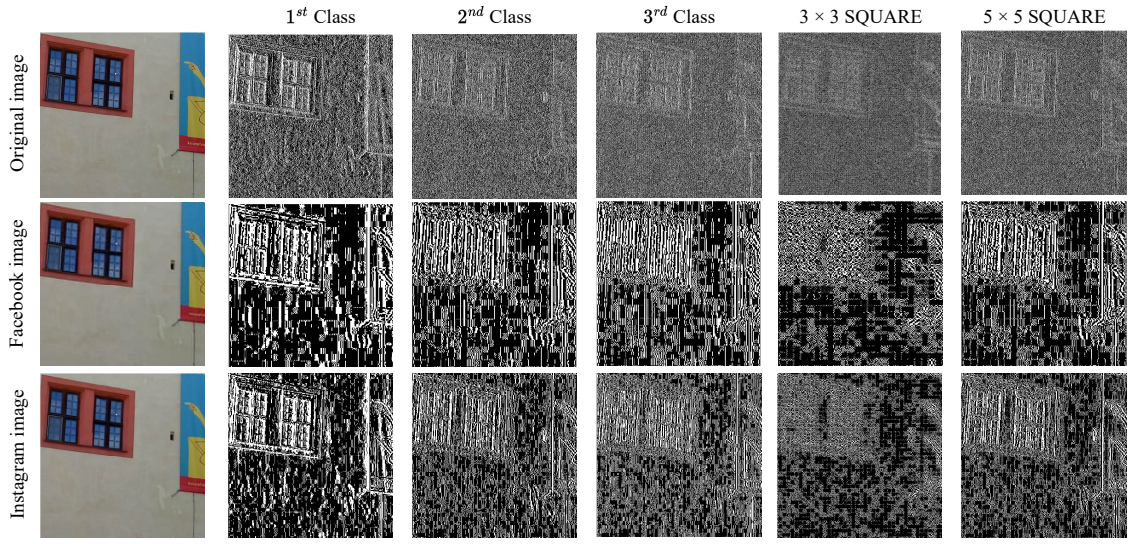



Figure 4.2: Illustration of noise residuals for Original, Facebook, and Instagram image corresponding to the different classes of SRM filters.

### 4.2.1 Steganalysis based Noise Residuals

Most of the solutions for image forensic problems such as camera model identification and image manipulation detection, rely on suppressing the content information of an input image [5, 99]. It is useful to highlight the artifacts in the underlying noise fingerprints. Also, extracting features from RGB image using CNNs tends to learn content-dependent information which leads to the consideration of undesired features that reduce the performance of forensic models. Inspired from [63], we deploy a high-pass layer in the first stage for the extraction of noise residuals. In this layer, the input RGB image is convolved with well-known 30 SRM fixed high-pass kernels defined in [63]. Initially, all of these kernels are used in steganalysis-based methods. It is mentioned in [63] that noise feature maps obtained using all of these kernels may be useful to recover the processing history of an image. These 30 SRM filters correspond to 7 different residual classes include 8, 4, 8, 1, 1, 4, and 4 filters in class $1^{st}$, $2^{nd}$, $3^{rd}$, 3×3 SQUARE, 5×5 SQUARE, 3×3 EDGE, and 5×5 EDGE, respectively. The maximum filter size of these SRM filters is 5×5. Therefore, we set the filter size of 5×5 for the first HPL layer. All the SRM filters are converted to the same size i.e. 5×5 using zeros padding. Moreover, this layer acts as a regularization term in deep learning to ease network convergence by reducing the feasible parameter space. To perform convolution operation with a three-channel RGB image, each single-channel kernel is converted to a three-channel kernel by duplicating the values. For each kernel, the convolution operation outputs a single-channel noise feature map. Therefore, this high-pass layer provides 30 noise feature maps. We also provide the visualization of obtained noise residuals corresponding to the $1^{st}$ (first filter), $2^{nd}$ (first filter), $3^{rd}$ (first filter), 3×3 SQUARE, and 5×5 SQUARE class filters as shown in Figure 4.2. Note that we have not shown all the 30 noise residuals including noise residuals corresponding to 3×3 EDGE and 5×5 EDGE classes due to the space constraint. The

convolution of an image $I$ with the kernel $W_j$ can be formulated as:

$$F_j = I \circledast W_j \qquad j \in \{1,\ 2,\ ..., 30\}, \tag{4.1}$$

where, $F_j$ and $W_j$ represent the $j^{th}$ noise feature map and $j^{th}$ SRM kernel, respectively. $W_j$ can be formally defined as:

$$W_j = [W_j^1,\ W_j^2,\ W_j^3], \tag{4.2}$$

where, $W_j^1$, $W_j^2$ and $W_j^3$ represent single-channel kernels with same values. Consider the input image of size $M \times N \times 3$, the output noise residuals ($I'$) after convolution operation of high-pass layer can be formulated as:

$$I' = [F_1,\ F_2,\ ...,\ F_j,\ ..., F_{29},\ F_{30}], \tag{4.3}$$

where, the size of $I'$ is $M \times N \times 30$.

### 4.2.2 CNN for High-level Features Extraction and Classification

The noise residuals obtained from the first stage are passed to CNN-based classifier for the identification of SSMN. The aim of this classifier is to extract high-level features related to social media networks and generate the corresponding class probabilities. The CNN architecture as shown in Figure 4.1(a) consists of four CBR blocks. Each CBR block ($\mathcal{C}_\mathcal{B}$) is a sequential network consisting of three layers i.e., convolution, batch normalization, and ReLU activation function layers as shown in Figure 4.1(b). The convolution layer of each CBR block is responsible for extracting relevant features for SSMN identification. Afterwards, batch normalization is performed to standardize the obtained feature maps distribution which results in the reduction of generalization error and fast training. Lastly, the purpose of the ReLU layer is to introduce non-linearity to our model and resolve the issue of vanishing gradients. All CBRs are connected to each other in a sequential manner, with the max-pooling layer as the intermediate layer. The purpose of the max-pooling layer is to down-sample the input along the spatial dimension and pass the dominating features to the next CBR block. The proposed CNN is able to learn relevant features with low computation cost. The output of the last CBR is passed to the average pooling layer with a kernel size of $4 \times 4$ and the resultant output is further provided to a convolution layer ($C_L$) to extract high-level features. The number of kernels in this convolutional layer is the same as the total number of classes ($N_S$), i.e. one more than the number of social media networks. These high-level features become the input of a fully-connected layer (FC) having neurons equal to the $N_s$. The number of kernels in the four CBRs i.e., $\mathcal{C}_{B_1}$, $\mathcal{C}_{B_2}$, $\mathcal{C}_{B_3}$, and $\mathcal{C}_{B_4}$ are 32, 64, 128, and 256 respectively, denoted by $K$. The high-level features ($H$) obtained from input noise residuals ($I'$) are passed to FC layer and can be formulated as,

$$H = C_L(P_A(\mathcal{C}_{B_4}(P_M(\mathcal{C}_{B_3}(P_M(\mathcal{C}_{B_2}(P_M(\mathcal{C}_{B_1}(I')) \tag{4.4}$$

where $P_A$ and $P_M$ denote the average and max pooling layers, respectively. At last, the softmax layer provides the probabilities related to the social media networks and classification is done using the highest probability value. The training loss $\mathcal{L}$ of SNRCN2 is carried out by computing cross-entropy loss between the target and the estimated output of the proposed method. The loss function is formulated as:

$$\mathcal{L}\left(f\left(H;W\right);\{y_1,y_2,...,y_{N_s}\}\right) = -\sum_{k=1}^{N_s} \log\left(\hat{y}_k\right).y_k \tag{4.5}$$

where, $\hat{y}_k$ is the output of softmax function $f$ corresponding to $k^{th}$ class and $y_k$ is the true binary label of $k^{th}$ class with value 1 only for the correct class. The function $f$ is applied on linear transformation of the high-level features $H$, with weight matrices $W$ of FC layer, corresponding to the input image.

## 4.3 Experimental Results and Discussion

In this section, we initially provide the details related to the datasets and settings used in our experiments. Then, we present the results obtained by our proposed model and other existing methods for comparative analysis, results of the ablation studies, and finally the discussion related to our approach.

### 4.3.1 Dataset Details

We evaluate our SNRCN2 on two major publicly available datasets i.e., VISION [95] and Forchheim [4]. The criteria for selecting these two datasets is based on the availability, the number of image acquisition devices and the number of images. The objective is to select the datasets having a maximum number of acquisition devices and a large number of images per social media network class.

The VISION dataset contains around 34427 RGB images in JPEG format. Out of these, 11732 are the original images and the remaining 22695 are images from social media networks. The 11732 images have been captured using 35 different smartphone cameras. These smartphone devices belong to 11 widely used camera manufacturing brands. Out of 11,732 original non-processed images, 7565 have been uploaded and downloaded from two majorly used social networks i.e., Facebook and WhatsApp. Facebook provides two quality versions i.e., low and high. Therefore, there are three copies of the original image i.e., two belong to Facebook and one to WhatsApp. So, the total number of social media network images are 22695 i.e. $(3 \times 7565)$. We considered all images of the VISION dataset for the experiments.

The Forchheim dataset contains 23106 images captured from 27 smartphone cameras. It contains 3851 original non-processed images along with five copies of social network platforms. Each image has a copy of Facebook, Instagram, Telegram, Twitter and WhatsApp social networks. So, the total number of social media network images are

19255 i.e. $(5 \times 3851)$ and the images in each social media network of this dataset share the same content, making it suitable for tracing forensic fingerprints.

### 4.3.2 Experimental Settings

All the experiments have been performed with a system consisting of Tesla V100 GPU of 32GB memory and 3.20 GHz Intel Core i7-8700 CPU with 64GB RAM. Each dataset has been divided into three sets i.e., training, validation and testing set in the proportion of 80%, 10%, and 10%, respectively. For each image in the considered datasets, we extract a maximum of 64 non-overlapping patches of size $256 \times 256 \times 3$ from the centre portion of the image. The label of each patch is the same as of the image label. Further, the model output for all patches are combined by majority voting to obtain the final estimation of the input image. This also helps in achieving better accuracy at image level when the majority of the patches are correctly classified.
We implement the SNRCN2 in PyTorch 1.8.0 framework. We apply mini-batch stochastic gradient descent with a batch size of 64 to train the SNRCN2. We used Adam optimizer with an initial learning rate of 0.0001. All the considered methods were trained for a maximum of 100 epochs. It was observed that all the methods were converging by 100 epochs. During the training, we observed validation loss for the convergence and picked the model with maximum validation accuracy for testing.

### 4.3.3 Results and Analysis

We conduct a set of experiments to evaluate the performance of SNRCN2 along with a comparative analysis with recent existing techniques including PRNU based CNN [112], DCT based VGG16 network [115], DWT based VGG16 network [115], and MSF-CNN [118]. To the best of our knowledge, MSF-CNN [118] is the most recent method for SSMN identification. For performance comparison and evaluation, we consider primarily two evaluation metrics: PLA and ILA.

#### Results on the VISION and Forchheim Datasets

Table 4.1 presents the comparative analysis of different SSMN identification methods on two different datasets. It is observed that SNRCN2 provides the highest image-level accuracies of 99.53% and 100% on the VISION and Forchheim datasets, respectively. Our method provides an image-level accuracy improvement of 1.07% and 1.60% on VISION and Forchheim dataset, respectively as compared to the second-best method [112]. Moreover, SNRCN2 outperforms the existing methods by providing better patch-level accuracies of 99.84% and 99.81% on the VISION and Forchheim datasets, respectively. Note that the approaches [115] and [118] resized the input image before passing it to CNN instead of extracting patches. Therefore, patch-level accuracy is not applicable for these two methods. It is also observed that the image-level accuracy on Forchheim dataset for all the considered existing techniques is significantly less as compared to the VISION

dataset. But, our method provides high image-level accuracy on both the datasets, thereby confirming its better generalization ability.

Table 4.1: Comparative analysis of different methods on VISION and Forchheim datasets.

| **Dataset** | **VISION** | | **Forchheim** | |
|---|---|---|---|---|
| **Method** | **PLA** | **ILA** | **PLA** | **ILA** |
| PRNU+CNN | 97.65 | 98.46 | 92.39 | 98.40 |
| DCT+VGG16 | — | 98.23 | — | 90.22 |
| DWT+VGG16 | — | 95.66 | — | 94.50 |
| MSF-CNN | — | 96.66 | — | 91.34 |
| **SNRCN2** | **99.84** | **99.53** | **99.81** | **100** |

Table 4.2: Image-level accuracy results of different methods for each class on VISION and Forchheim datasets.

| **Dataset** | **VISION** | | |
|---|---|---|---|
| **Method** | **Original** | **Facebook** | **WhatsApp** |
| PRNU+CNN [112] | 97.44 | 99.60 | 97.75 |
| DCT+VGG16 [115] | 99.14 | 98.41 | 96.43 |
| DWT+VGG16 [115] | 99.40 | 99.07 | 96.69 |
| MSF-CNN [118] | 98.47 | 97.22 | 92.73 |
| **SNRCN2** | **99.57** | **100** | **98.55** |

Table 4.3: Image-level accuracy results of different methods for each class on Forchheim dataset.

| **Dataset** | **Forchheim** | | | | | |
|---|---|---|---|---|---|---|
| **Method** | **Original** | **Facebook** | **WhatsApp** | **Instagram** | **Telegram** | **Twitter** |
| PRNU+CNN [112] | 100 | 97.14 | 96.62 | 100 | 97.92 | 98.70 |
| DCT+VGG16 [115] | 98.44 | 86.49 | 90.90 | 82.85 | 86.49 | 96.10 |
| DWT+VGG16 [115] | 99.44 | 99.22 | 92.72 | 90.66 | 88.31 | 96.36 |
| MSF-CNN [118] | 98.18 | 94.81 | 88.57 | 90.65 | 88.57 | 87.27 |
| **SNRCN2** | **100** | **100** | **100** | **100** | **100** | **100** |

We further provide the results based on each social media network class for both the VISION and Forchheim datasets as shown in Table 4.2 and 4.3. Note that the VISION dataset has three classes while the Forchheim dataset includes six. Table 4.2 and 4.3 show that proposed model outperforms the existing methods by providing highest image-level accuracies i.e., 99.57%, 100%, and 98.55% for Original, Facebook, and WhatsApp class, respectively on VISION dataset. It is observed that our approach provides an image-level accuracy improvement of 1.10%, 2.78%, and 5.82% as compared to the recent MSF-CNN [118] method. Also, our model attains a perfect image-level accuracy of 100% for the Facebook social media network class. Similarly, Table 4.2 and 4.3 show that our model achieves perfect image-level accuracy of 100% for all the social media network classes of the Forchheim dataset. The results on the Forchheim dataset is very significant due to the reason that each social media network has the same number of images with similar content. This implies that SNRCN2 emphasizes unique artifacts left by particular social

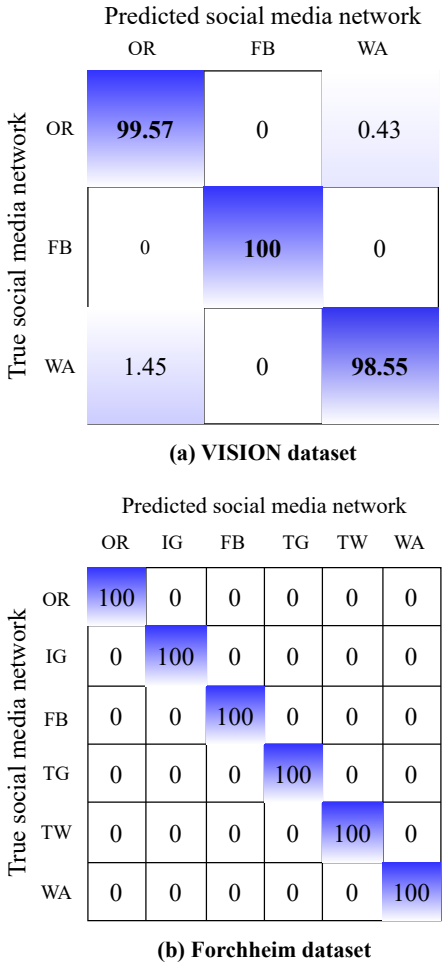



Figure 4.3: Confusion matrix of SNRCN2 on different datasets. Social media networks are abbreviated as, Facebook: FB, Instagram: IG, Original: OR, Telegram: TG, Twitter: TW, WhatsApp: WA.

media networks.

The performance of the proposed model is also confirmed from the confusion matrices based on VISION and Forchheim datasets as shown in Figure 4.3. It is observed from Figure 4.3(a) that only 5 original and 11 WhatsApp images are wrongly classified in WhatsApp and original class, respectively on the VISION dataset. Importantly, all the social media network images of the Forchheim dataset are correctly classified as shown in Figure 4.3(b).

Table 4.4: Image-level accuracy results of different methods on combined dataset.

| **Method** | **Original** | **Facebook** | **WhatsApp** | **Overall** |
|---|---|---|---|---|
| PRNU+CNN [112] | 97.90 | 99.80 | 86.60 | 95.61 |
| DCT+VGG16 [115] | 99.06 | 94.05 | 82.82 | 92.72 |
| DWT+VGG16 [115] | 98.71 | 96.73 | 95.63 | 97.10 |
| MSF-CNN [118] | 96.86 | 96.14 | 83.84 | 92.28 |
| **SNRCN2** | **99.19** | **99.41** | **98.54** | **99.10** |

**Results on the Combined Dataset**

The VISION and Forchheim datasets include original images as well as images downloaded from two common social media networks i.e., Facebook and WhatsApp. As both datasets are released at different times, it is quite possible that the post-processing operations/parameters applied on the images by these social media networks are different. To further evaluate the robustness of the proposed model, we created a new dataset that contains images of common social media networks of VISION and Forchheim datasets along with original images. This results in a dataset having an almost double number of VISION dataset images as compared to the Forchheim dataset images. Therefore, to maintain the equivalent number of images from both datasets, we considered the images related to the first 15 devices out of 35 devices from the VISION Dataset. The final combined dataset contains 14423 and 11553 images of the VISION and Forchheim dataset, respectively. It is observed from Table 4.4 that SNRCN2 achieves significant improvement with an overall accuracy of 99.10% as compared to the second-best method Manisha et al. [115] with an accuracy of 97.10% with DWT input. These results on the combined dataset are quite important because we can observe a significant difference in the accuracy values obtained by different methods as shown in Table 4.4, thereby confirming the proposed model robustness.

Table 4.5: Performance of proposed model on Forchheim dataset considering different batch sizes, learning rates and patch sizes.

| | | **PLA** | **ILA** |
|---|---|---|---|
| Batch size | 8 | 99.56 | 100 |
| | 16 | 99.62 | 100 |
| | 32 | 99.72 | 100 |
| | **64** | **99.81** | **100** |
| Learning rate | 0.1 | 40.81 | 16.66 |
| | 0.01 | 99.31 | 99.69 |
| | 0.001 | 99.73 | 99.95 |
| | **0.0001** | **99.81** | **100** |
| | 0.00001 | 99.41 | 99.82 |
| Patch size | 64 × 64 | 98.21 | 99.74 |
| | **256 × 256** | **99.81** | **100** |

### 4.3.4 Ablation Studies

In this section, we present the ablation studies related to the model training hyper-parameters and design choices on the Forchheim dataset. To obtain the optimal hyper-parameter values for our network, we train our SNRCN2 by considering different values of the most significant hyper-parameters i.e., batch size and learning rate. We have also evaluated our model by extracting patches of size $64 \times 64$ similar to [112] method and also the patches of size $256 \times 256$. The results of these methods are presented in Table 4.5. Based on these results, we select the batch size of 64, the learning rate of 0.0001

and image patch size of $256 \times 256$, as these provide the best values for both patch-level accuracy (99.81%) and image-level accuracy (100%).

We also perform an ablation study by considering varying number of consecutive CBR blocks in between the pooling layers of our proposed network, as shown in Figure 4.1(a). The image-level accuracies with two and three consecutive CBR blocks in between pooling layers is the same as in the case of one CBR block i.e. 100%. However, there is noted a slight improvement of around 0.1% in patch-level accuracy in these cases but with a significant increase in the computation time. So, we do not include more than one CBR block in between the pooling layers in proposed SNRCN2. Lastly, we also compare with another network setting by having a total of five CBR blocks and five pooling layers. In this case of five CBR blocks, there is a miss-classification of one image and the performance of the model marginally decreased by 0.01% and 0.06% in terms of patch-level and image-level accuracy, respectively.

Table 4.6: Results on Forchheim social media platform based images when trained on augmented dataset

| **Social media** | **Facebook** | | **WhatsApp** | | **Instagram** | |
|---|---|---|---|---|---|---|
| **Method** | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** |
| Chen et al. [15] | 40.29 | 69.60 | 54.91 | 81.22 | 48.06 | 76.62 |
| Yao et al. [19] | 25.33 | 49.80 | 33.52 | 58.10 | 31.02 | 57.98 |
| Freire-Obregon et al. [3] | 25.50 | 38.44 | 34.96 | 53.38 | 31.53 | 49.04 |
| You at al. [29] | 31.10 | 62.06 | 49.85 | 77.77 | 38.52 | 67.81 |
| Liu at al. [25] | 35.10 | 66.92 | 55.25 | 80.72 | 45.45 | 74.45 |
| Rafi et al. [24] | 34.81 | 66.53 | 53.89 | 81.22 | 42.53 | 72.73 |
| Bennabhaktula et al. [27] | 91.53 | 60.15 | 48.51 | 77.29 | 39.54 | 69.98 |
| **Our CMI method** | **43.38** | **70.90** | **61.41** | **85.52** | **51.86** | **78.92** |
| | | | | | | |
| **Social media** | **Telegram** | | **Twitter** | | **Average (SM)** | |
| **Method** | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** |
| Chen et al. [15] | 54.24 | 81.35 | 64.81 | 88.76 | 52.46 | 79.51 |
| Yao et al. [19] | 33.32 | 62.32 | 39.21 | 67.17 | 32.48 | 59.07 |
| Freire-Obregon et al. [3] | 35.21 | 54.53 | 40.15 | 61.94 | 33.47 | 51.47 |
| You at al. [29] | 49.03 | 81.35 | 57.44 | 84.29 | 45.19 | 74.66 |
| Liu at al. [25] | 54.86 | 84.29 | 67.90 | 90.80 | 51.73 | 79.44 |
| Rafi et al. [24] | 55.97 | 86.71 | 67.63 | 90.54 | 50.97 | 79.55 |
| Bennabhaktula et al. [27] | 44.46 | 77.77 | 54.23 | 83.26 | 43.65 | 73.71 |
| **Our CMI method** | **62.38** | **87.22** | **74.07** | **92.46** | **58.62** | **83.20** |

### 4.3.5 Robustness of CMI Mthods against Real-World Social Media Network Post-processed Images

Social media platforms have become the prominent medium for sharing the images. Therefore, the forensic query regarding identifying the source camera model is very likely to be applied for image(s) coming via one of these platforms. All these platforms apply some post-processing operations, such as JPEG compression and rescaling, and the parameters

used in such operations are generally not known. Therefore, it is important to examine the robustness of the proposed approach against real-world post processed images. We perform this evaluation using the Forchheim dataset that consists of original images acquired using some smartphones and also the social media platforms based post-processed versions of each image. The reason of choosing this Forchheim dataset is that it provides the images from five social media network and number of images (3851) related to each social media platform and original images are equivalent. For this evaluation, the training of each network model considered is done using the augmented dataset that consists of 80% of the original images of Forchheim dataset and 5 different copies of each of these images acquired using highly popular social media platforms: Facebook, Whatsapp, Instagram, Telegram, and Twitter. After training, the evaluation on the similarly enhanced test set is performed, the results of which are presented in Table 4.6 for the social media platforms based images. As per the results, the best robustness is achieved by the proposed method for each of the five different platforms, with overall average PLA and ILA of 58.62% and 83.20%, respectively. On Twitter, the proposed method achieves highest ILA of 92.46% among all social media platforms. On Whatsapp and Telegram also, it is more than 85%, whereas, on Instagram and Facebook, it is somewhat lower i.e. 78.92% and 71.9% respectively. However, these all are better than that of any other CMI method considered. The substantial ILA improvements of 7.67%, 10.95%, 7.91%, 11.45%, and 9.09% are achieved on Facebook, Whatsapp, Instagram, Telegram, and Twitter, respectively, in comparison to the second-best performing methods. In terms of overall ILA, the methods: [24], [15] and [25] performed almost similarly i.e. around 79.5%, and they can be considered then second-best performing methods, as the proposed method performs considerably better i.e. 83.2%, as shown in Table 4.6. Further, we evaluate the efficacy of directing images to models specifically trained on individual social media network images. We trained five separate models, each corresponding to one of the five social media networks included in the Forchheim dataset. These models were then compared with a model trained on an augmented dataset. The results of this comparative evaluation are presented in Table 4.7. It has been observed that there is an improvement in ILA across all the social media network images. This finding underscores the potential benefits of specialized model training for CMI of image in the context of different social media platforms.

Table 4.7: Results on Forchheim social media platform based images when trained on each social media images dataset

| **Test on social media** | **Facebook** | | **WhatsApp** | | **Instagram** | | **Telegram** | | **Twitter** | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Trained on social media** | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** |
| Facebook | 40.64 | **72.66** | 25.66 | 39.46 | 5.73 | 4.56 | 26.25 | 40.74 | 15.64 | 16.68 |
| WhatsApp | 21.95 | 36.27 | **61.86** | **86.84** | 7.66 | 8.17 | 18.48 | 21.20 | 20.60 | 22.34 |
| Instagram | 3.98 | 3.70 | 4.70 | 4.46 | 50.50 | **81.73** | 5.22 | 4.72 | 6.96 | 6.62 |
| Telegram | 14.01 | 14.81 | 22.75 | 23.64 | 5.21 | 5.23 | **64.78** | **88.76** | 27.09 | 35.12 |
| Twitter | 8.14 | 7.79 | 19.13 | 21.22 | 11.11 | 14.17 | 21.72 | 26.55 | **75.55** | **92.97** |

### 4.3.6 Discussion and Limitations

In our work, all the considered methods are trained for 100 epochs. We observe the validation loss and find that all the methods are converging by 100 epochs or earlier. Our method is converging by around 50 epochs. For testing purposes, we pick the model weights providing maximum validation accuracy. Also, we have performed a set of ablation studies by considering the Forchheim dataset to select the appropriate network hyper-parameters such as batch size and learning rate. The values chosen are the ones that provide the best results for SSMN identification. We also consider enhancing the total number of CBR blocks from 4 to 5, but that results in decrease in performance, potentially due to low resolution of high-level features obtained by the last convolution layer in case of 5 CBR block., So, we finally choose to have only four CBR blocks in our proposed model.
The existing methods consider different strategies while feeding the input image to their models for training/testing. Caldelli et al. [112] considered the input image to be the $64 \times 64$ size patches extracted from the PRNU of the given image. Manisha et al. [115, 118] considered the input image to be the resized version of the preprocessed image (DCT/DWT) in their works. In our work, we initially extract the maximum of 64 non-overlapping patches of size $256 \times 256$. We prefer patch-extraction instead of resizing as partitioning the image into patches does not affect the noise fingerprints of the image. Also, the partitioning of an image into patches increases the total number of images for the training. Unlike approach [112], we consider larger patch size to provide larger receptive field to our model for better feature extraction and better performance, as evident also from the results presented in Table 4.5. Inference time of method [112] is least, likely because of using smaller sizes patches, and the inference times of other methods [115, 118] are much higher than that of our method. Overall the experimental results and comparative analysis across two different datasets and the combined dataset demonstrate the effectiveness and robustness of our method for SSMN identification tasks. However, it may be noted that the proposed method gives more emphasis on the detection of artifacts left by the most recent social media platform and this may be a challenge when the images are shared multiple times via different social media networks. Further, the datasets used considered maximally 30 smartphone cameras and most of the images in these are natural scene images. The performance on more diverse and real-world scenarios is yet to be examined. We also plan to investigate the robustness against adversarial attacks in future.

## 4.4 MSRD-CNN: Multi-Scale Residual Deep CNN for General-purpose Image Manipulation Detection

### 4.4.1 Introduction

The digital information can be shared in the form of audio, image, and video using various social media platforms such as Facebook, Instagram, Snapchat, etc. The advent of powerful editing software results in a significant increase in the number of tampered

images on social media related to political, individual attacks, publicity, etc. Therefore, the authenticity of digital images is very crucial. Moreover, the investigation of digital images can play important role in many fields related to medical, news media, scientific exploration, law and crime [120, 121, 98]. Thus, it is a concern of great importance in multimedia forensics. The detection of different image processing operations has a great relevance to the forensic community due to the fact that these operations may be used by the counterfeiter in the creation of an image forgery. It is perceived that different image processing operations embed special artifacts or footprints in the processed image. Several forensic algorithms have been designed to detect the particular image processing operation by analyzing the corresponding artifacts. Some image processing operations considered are resampling [122, 123, 124, 125, 126], JPEG compression [127, 128, 129, 130], median filtering [131, 84, 132, 133], contrast enhancement [134, 135, 136, 137], etc. Also, many anti-forensic approaches related to different image processing operations such as JPEG compression [138], [139], median filtering [140], and contrast enhancement [141] have also been proposed to mislead the forensic techniques by concealing the footprints of corresponding image processing operations. The researchers have also developed general-purpose image manipulation detection schemes to detect different image processing operations [86, 142, 143, 85, 23, 106]. Moreover, it is observed that recent works on multi-purpose image tampering detection are based on deep learning techniques, for instance, Convolutional Neural Networks (CNNs). These CNNs have demonstrated the ability to automatically learn the image manipulation features from data. A novel constrained convolutional layer based CNN is proposed in [86] to detect the multiple image processing operations by suppressing the image content information and the authors further optimized their constrained neural network in [85] for better performance. In [142], a densely connected CNN based on isotropic constraint is proposed for general-purpose image forensics by considering the anti-forensic attacks. The isotropic convolutional layer works as a high-pass filter to highlight the image processing operations artifacts by suppressing the image content information. Moreover, an image manipulation detection approach built upon [86] and combined with a deep Siamese CNN network is presented in [143]. However, their work was not to identify the specific image manipulation but to classify the input patch pair (two images) whether they are identically processed or not. In [23], Xception architecture is employed to classify multiple image processing operations by considering small-sized images. Most of the existing general-purpose forensic techniques can be easily circumvented by using some anti-forensic attacks. Recently, a universal image manipulation detection approach based on densely-connected CNN is proposed in [106] and it has also considered most of the image processing operations including various anti-forensic techniques for evaluation. However, the proposed CNN is significantly different from the existing approach [106] in terms of network architecture as well as used image manipulation datasets.

Overall, designing a unified forensic scheme capable of detecting different image manipulations under different attacks is still a challenging task for the researchers. Also,

to the best of our knowledge, the existing works have not performed any cross dataset testing to evaluate the generalization of their models. In this section, we present a novel and effective image manipulation detection network capable of detecting multiple editing operations including anti-forensic methods. Our network comprises of three stages: pre-processing, hierarchical high-level feature extraction, and classification. Firstly inspired by Res2Net [26], a multi-scale residual module is employed in pre-processing stage to extract the prediction error or noise features adaptively. Further, the obtained noise features are processed by using FEBs to extract the high-level image manipulation features. We have considered several image processing operations including anti-forensic schemes and with arbitrary parameters to evaluate our network. The remaining part of the section includes a detailed description of the proposed network in Section 4.4.2 and the experiment results are discussed in Section 4.4.3.

### 4.4.2 Proposed MSRD-CNN Architecture

In this section, we propose a novel MSRD-CNN architecture capable of detecting the traces of multiple image processing operations and anti-forensic techniques. The architecture of MSRD-CNN, as shown in Figure 4.4, includes three different stages i.e., extraction of noise features using a multi-scale residual module, feature extraction network to extract high-level features related to image tampering artifacts, and classification.

#### Multi-Scale Residual Module

Most of the image manipulation detection schemes use the idea of suppressing the content information of an input image to highlight the image manipulation artifacts. Compared to applying fixed filters to the input image prior to CNN for the extraction of prediction error features, it is preferred to employ a trainable filtering scheme for pre-processing to potentially learn more appropriate image manipulation features adaptively for image forensic tasks. In our approach, we use a data-driven pre-processing scheme that consists of a two-layer CNN and a multi-scale residual module. Each convolution layer in the two-layer CNN contains 64 filters of $3 \times 3$ followed by batch normalization and the ReLU layer. This two-layer CNN is employed to obtain better input features for the multi-scale residual module. Let us denote the functions of these two convolution layers by $C_1(\cdot)$ and $C_2(\cdot)$, respectively. For a given input image $I$ of size $256 \times 256$, the output of this two-layer CNN is formulated as:

$$I_{C_1C_2} = C_2(C_1(I)), \tag{4.6}$$

This output $I_{C_1C_2}$, having size of $256 \times 256 \times 64$, is then passed to the multi-scale residual module which is inspired from Res2Net [26] and designed to learn the suitable noise features. The proposed multi-scale residual module explores the multi-scale feature representation by dividing the input features of size $256 \times 256 \times 64$ along the channel axis, which results in four different groups of size $256 \times 256 \times 16$. These groups are then interconnected in a hierarchical residual-like style as shown Figure 4.4(b). Each group is

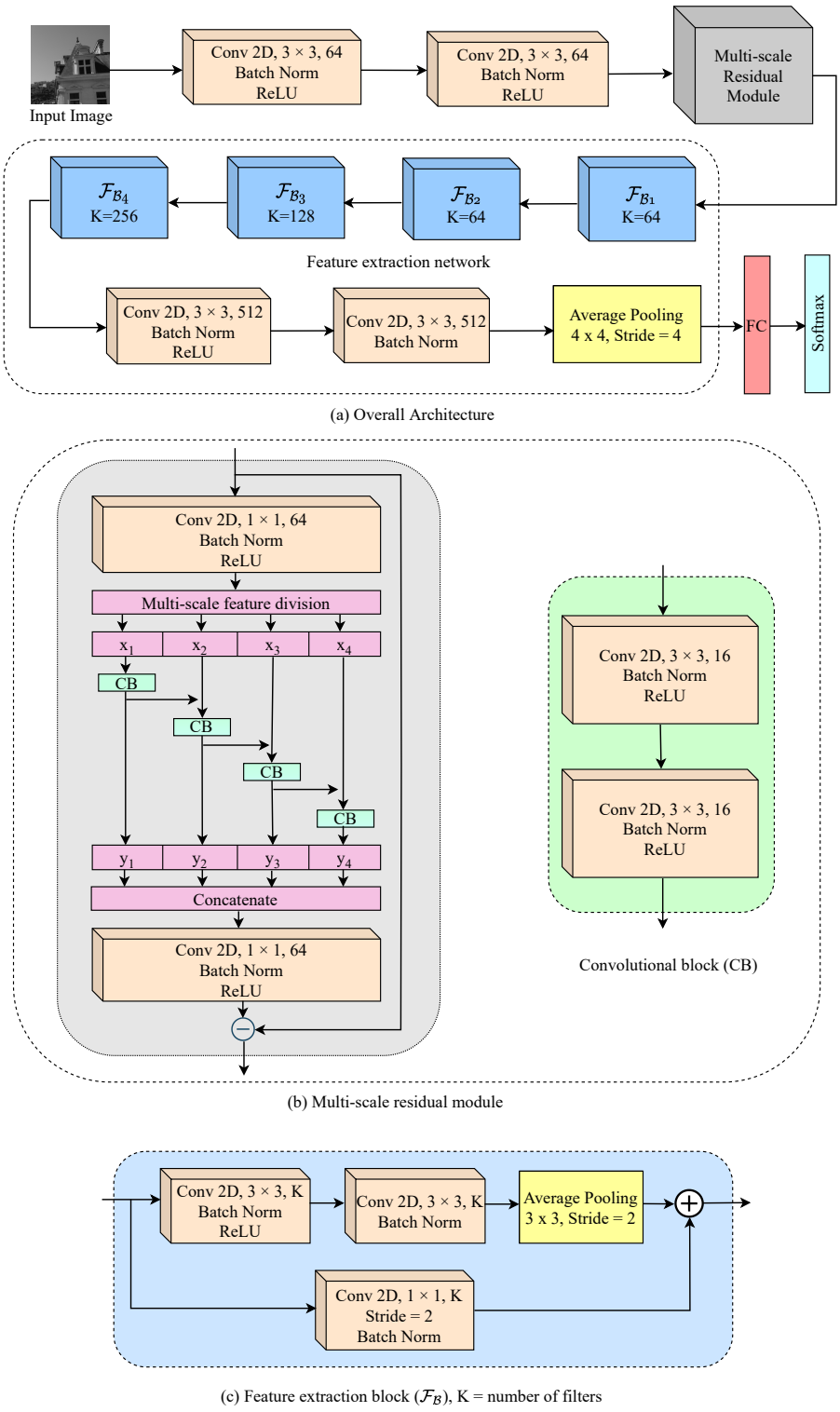



Figure 4.4: The architecture of MSRD-CNN

further processed by a Convolutional Block (CB) having two convolution layers with 16 filters of $3 \times 3$ followed by batch normalization and ReLU layers. The output feature maps of the first CB is added to the second group before passing to the second CB as shown in Figure 4.4(b). Let $x_i$ represents the feature maps of $i^{th}$ group, where $i \in \{1, 2, 3, 4\}$, and $H_i(\cdot)$ is the function performed by the convolutional block of $i^{th}$ group. The output of

$H_i(\cdot)$ which is $y_i$ will be added to $x_{i+1}$ group and passed to $(i+1)^{th}$ convolutional block $(H_{i+1})$ as provided in Eq. 4.7.

$$y_i = \begin{cases} H_i\left(x_i\right) & i = 1 \\ H_i\left(x_i + y_{i-1}\right) & i = 2, 3, 4 \end{cases} \tag{4.7}$$

The outputs of all the convolutional blocks are concatenated and passed to a convolution layer having 64 filters of size $1 \times 1$. The output of this convolutional layer is subtracted from the input of the multi-scale residual module to obtain the final noise features as:

$$I_{MSRM} = MSRM(I_{C_1C_2}) - I_{C_1C_2} \tag{4.8}$$

where, $MSRM(\cdot)$ denotes the function performed by the multi-scale residual module. The feature extraction blocks further process these noise features to extract the high-level image manipulation features. Note that the features size i.e., height and width remains same during the pre-processing stage except the channel size.

**Feature Extraction Network**

The noise features obtained from the multi-scale residual module are passed to the feature extraction network to extract the high-level image manipulation features. This feature extraction network has four FEBs and each FEB $(\mathcal{F}_{\mathcal{B}})$ is based on a residual skip connection containing two regular convolution layers of size $3 \times 3$ and a $1 \times 1$ convolution layer. The input of a FEB is added to the output of the second convolution layer followed by the average pooling operation as shown in Figure 4.4(c). Note that we have not used the pooling layer in the multi-scale residual module of pre-processing stage because pooling layer strengths the image content and reduces noise signal by averaging. The purpose of the pooling layer is to down-sample the features for learning high-level image manipulation features. Number of filters in the four FEBs i.e $\mathcal{F}_{\mathcal{B}1}, \mathcal{F}_{\mathcal{B}2}, \mathcal{F}_{\mathcal{B}3}$, and $\mathcal{F}_{\mathcal{B}4}$ are 64, 64, 128 and 256 respectively. The resultant features obtained from this feature extraction network can be formulated as:

$$I'_{\mathcal{F}_{\mathcal{B}}} = \mathcal{F}_{\mathcal{B}4}(\mathcal{F}_{\mathcal{B}3}(\mathcal{F}_{\mathcal{B}2}(\mathcal{F}_{\mathcal{B}1}(I_{MSRM})))) \tag{4.9}$$

The output of this feature extraction network i.e. $I'_{\mathcal{F}_{\mathcal{B}}}$ is further processed by two convolution layers each having 64 filters of size $3 \times 3$ to obtain the more relevant image manipulation features. First convolution layer is followed by batch normalization and ReLU and the second convolutional layer is followed by batch normalization. Afterward, the average pooling layer with filter size $4 \times 4$ and stride 4 is applied to reduce the feature dimension.

Lastly, the global features obtained after the average pooling layer is fed to a fully-connected (FC) layer with 11 neurons corresponding to image processing operations used for classification. We use the softmax function to get the probability of predicted classes and the cross-entropy function to calculate the overall network loss.

### 4.4.3 Experimental Results

We conducted extensive experiments to evaluate the performance of the proposed model in the detection of multiple image processing operations and various anti-forensic attacks. Firstly, to confirm the multi-purpose nature of our MSRD-CNN, we considered 10 image processing operations along with corresponding parameters listed in Table 4.8. The image processing parameters are selected randomly to create more challenging image manipulation datasets. For instance, in JPEG compression, we compress the original images by randomly selecting the Quality Factor (QF) ranging from 60 to 90.

Table 4.8: Different image processing operations used for the generation of manipulation datasets with arbitrary parameters.

| **Image editing operations** | **Parameters** |
|---|---|
| JPEG compression (JPEG) | $QF = 60, 61, 62, \ldots, 90$ |
| Gaussian Blurring (GB) | $\sigma = 0.7, 0.9, 1.1, 1.3$ |
| Adaptive White Gaussian Noise (AWGN) | $\sigma = 1.4, 1.6, 1.8, 2$ |
| Resampling (RS) using bilinear interpolation | Scaling $= 1.2, 1.4, 1.6, 1.8, 2$ |
| Median Filtering (MF) | Kernel $= 3, 5, 7, 9$ |
| Contrast Enhancement (CE) | $\gamma = 0.6, 0.8, 1.2, 1.4$ |
| JPEG anti-forensics (JPEGAF) [138] | $QF = 60, 61, 62, \ldots, 90$ |
| JPEG anti-forensics (JPEGAF) [139] | $QF = 60, 61, 62, \ldots, 90$ |
| Median filtering anti-forensics (MFAF) [140] | Kernel $= 3, 5, 7, 9$ |
| Contrast enhancement anti-forensics (CEAF) [141] | $\gamma = 0.6, 0.8, 1.2, 1.4$ |

[144] and Dresden image dataset [30] for the evaluation of different image tampering detection approaches. The standard BOSSBase dataset comprises of 10,000 grayscale images of resolution 512×512 in PGM format. We have transformed these PGM images into PNG format for evaluation purposes. The standard Dresden dataset contains 3008×2000 size 1491 raw images in NEF format. We converted these raw images into PNG format for evaluation. Our model is implemented by using PyTorch 1.8 deep learning framework and all the experiments are performed using Tesla V100 GPU with 32GB RAM. We compared our network with recent multi-purpose image tampering detection methods [142, 85, 23, 106] in terms of detection accuracy. We also assessed our model's robustness and generalization by performing cross-dataset testing. The experimental results exhibit the efficacy of the proposed model in comparison to the existing image manipulation detection methods. All the relevant codes are available on request for reproducibility and research advancement.

#### Multiple Image Manipulation Detection

In this subsection, we evaluate our MSRD-CNN performance in the detection of multiple image processing operations including anti-forensic techniques using BOSSBase and Dresden datasets. We created one original image (OR) and 10 tampered image datasets using the image processing operations as listed in Table 1 by considering 4,167 and 1,333 images sequentially from the BOSSBase dataset for training and testing, respectively.

We extracted 4 patches of size 256×256 from each of these images, which results in 16,668 training and 5,332 testing images for each of the image processing operations. Therefore, we obtained a dataset having 2,42,000 grayscale images. We used 1,83,348 images (including 16,668 original images) for training, and remaining 58,652 images (including 5,332 original images) for testing purposes. Note that we follow the strategy used by the existing works [85] to create image manipulation datasets corresponding to different image manipulation operations to make the comparison feasible. Therefore, we have used only 4167 and 1333 images from the BOSSBase dataset for training and testing, respectively. This may also be noted that the complete BOSSBase dataset images are not used in consideration to the limited computational facilities availability, as we are considering 10 image manipulation methods including anti-forensic approaches which are highly compute-intensive and time-consuming.

We also evaluated our network ability using 881 images from the Dresden dataset. We follow the same strategy as used for the BOSSBase dataset in preparing image manipulation datasets using different image processing operations. We considered 667 images for training and 214 images for testing the considered neural networks. All of these images are cropped from the center to obtain a sub-image region of size $1280 \times 1280$. Afterward, each sub-image region is processed to extract 25 patches of size $256 \times 256$ and then converted into grayscale format. Therefore, we obtained 16,668 (approx.) images for training and 5,332 (approx.) images for testing corresponding to image processing operations provided in Table 1. The training of our network is performed by using the Adam optimizer with a learning rate of 0.001 and we trained our network for 100 epochs in each experiment.

We evaluated confusion matrices for our model based on multiple image processing operations for BOSSBase and Dresden datasets as shown in Tables 2 and 3. Our MSRD-CNN provides average accuracies of 97.07% and 97.48% for BOSSBase and Dresden datasets, respectively, when evaluated on multiple image processing operations. Table 2 reveals that the proposed network gives an accuracy of greater than 97% for each image processing operation except for the original and CE images on the BOSSBase dataset. The accuracy of original and contrast-enhanced images is 87.92% and 90.15%, respectively for the BOSSBase dataset. Table 3 demonstrates that our proposed approach identifies each image processing operation with an accuracy of greater than 97% except for the original and contrast-enhanced images with 92.22% and 85.03% respectively on the Dresden dataset. Moreover, the robustness of our model is confirmed by the fact that it provides high accuracies against different anti-forensic approaches on both the datasets.

We also conducted an experiment by combining both the training sets of BOSSBase and Dresden datasets. It is observed that combining both the training datasets increases the model accuracy further, likely because of the increase of training dataset size and/or more diversity. The testing accuracy increases from 97.07% to 97.38% on the BOSSBase test dataset. Similarly, model testing accuracy increases from 97.48% to 98.11% on the Dresden test dataset. However, the training time increases significantly due to the large training

data.

Table 4.9: Performance comparison of different multi-purpose forensic schemes on BOSSBase dataset by considering multiple image processing operations. (OR: original images, JPEG: JPEG compression, GB: Gaussian blurring, AWGN: adaptive white Gaussian noise, RS: resampling using bilinear interpolation, MF: median filtering, CE: contrast enhancement, JPEGAF: JPEG anti-forensics, MFAF: median filtering anti-forensics, CFAF: contrast enhancement anti-forensic.)

| | **BOSSBase Dataset** | | | | |
|---|---|---|---|---|---|
| | Chen [142] | Bayar [85] | Yang [23] | Singh [106] | **Ours** |
| OR | 23.65 | 54.86 | 79.76 | 82.50 | **87.92** |
| JPEG | 96.66 | 99.72 | 99.83 | 99.76 | **99.89** |
| GB | 98.52 | 99.36 | **99.93** | 99.61 | 99.70 |
| AWGN | 80.65 | 93.12 | 98.54 | 98.33 | **99.34** |
| RS | 62.04 | 90.72 | 97.51 | 96.31 | **97.81** |
| MF | 88.77 | 97.32 | 97.60 | 99.40 | **99.76** |
| CE | 28.84 | 50.21 | 65.56 | 86.53 | **90.15** |
| JPEGAF [138] | 56.77 | 87.45 | 96.85 | 97.99 | **98.42** |
| JPEGAF [139] | 63.93 | 93.57 | 97.68 | **98.87** | 97.86 |
| MFAF [140] | 95.16 | 99.29 | 99.42 | 99.64 | **99.76** |
| CEAF [141] | 76.44 | 93.34 | 95.35 | **97.37** | 97.17 |
| **Overall Avg.** | 70.13 | 87.18 | 93.45 | 96.03 | **97.07** |

Table 4.10: Performance comparison of different multi-purpose forensic schemes on Dresden dataset by considering multiple image processing operations.

| | **Dresden Dataset** | | | | |
|---|---|---|---|---|---|
| | Chen [142] | Bayar [85] | Yang [23] | Singh [23] | **Ours** |
| OR | 39.07 | 23.48 | 35.54 | 81.40 | **92.22** |
| JPEG | 99.36 | 99.72 | **100.00** | **100.00** | 99.98 |
| GB | 99.91 | 94.90 | 99.79 | **99.91** | 99.74 |
| AWGN | 98.95 | 96.02 | 98.91 | **99.94** | **99.94** |
| RS | 82.2 | 84.26 | 98.33 | 99.27 | **99.81** |
| MF | 93.55 | 97.39 | 99.81 | 99.96 | **100.00** |
| CE | 53.06 | 74.79 | 78.94 | **88.32** | 85.03 |
| JPEGAF [138] | 51.03 | 79.95 | 97.85 | 99.01 | **99.49** |
| JPEGAF [139] | 59.75 | 70.93 | 95.99 | 95.72 | **97.43** |
| MFAF [140] | 95.37 | 99.08 | 99.87 | 99.94 | **100.00** |
| CEAF [141] | 58.55 | 66.84 | 89.45 | 92.55 | **98.63** |
| **Overall Avg.** | 75.53 | 80.67 | 90.41 | 96.00 | **97.48** |

### 4.4.4 Comparative Analysis with Existing Approaches

We compared our MSRD-CNN with existing multi-purpose forensic schemes [142, 85, 23, 106] by considering multiple images processing operations including anti-forensic techniques using the same training and testing datasets as defined in Section III-A. We provide the diagonal entries of confusion matrices in Table 4.9 and 4.10 for different methods for ease of comparison. The proposed model provides better detection as

Table 4.11: Performance comparison of different multi-purpose forensic schemes by considering cross dataset testing when trained

| | **Models trained on BOSSBase and tested on Dresden dataset (BOSSTrain-DRESTest)** | | | | |
|---|---|---|---|---|---|
| | Chen [142] | Bayar [85] | Yang [23] | Singh [106] | **Ours** |
| OR | 9.96 | 1.74 | 2.03 | 0.04 | 0.17 |
| JPEG | 97.09 | 99.59 | 99.42 | 99.59 | **99.96** |
| GB | 99.94 | 99.06 | 99.27 | 99.83 | 99.76 |
| AWGN | 94.28 | 78.17 | 93.45 | 89.89 | **95.09** |
| RS | 74.72 | 41.64 | 69.47 | 82.37 | **96.08** |
| MF | 74.47 | 95.09 | 92.74 | 99.34 | **99.98** |
| CE | 33.36 | 29.24 | 31.83 | 76.03 | **92.12** |
| JPEGAF [138] | 35.90 | 71.57 | 88.24 | 92.24 | 92.78 |
| JPEGAF [139] | 69.47 | 80.95 | 88.62 | 91.13 | 89.20 |
| MFAF [140] | 96.98 | 99.62 | 98.67 | 99.74 | **99.74** |
| CEAF [141] | 58.78 | 66.32 | 67.07 | 79.22 | **86.52** |
| Overall Avg. | 67.72 | 69.36 | 75.53 | 82.67 | **86.49** |
| Overall Avg. excluding OR | 73.5 | 76.13 | 82.88 | 90.94 | **95.12** |

Table 4.12: Performance comparison of different multi-purpose forensic schemes by considering cross dataset testing

| | **Models trained on Dresden and tested on BOSSBase dataset (DRESTrain-BOSSTest)** | | | | |
|---|---|---|---|---|---|
| | Chen [142] | Bayar [85] | Yang [23] | Singh [106] | **Ours** |
| OR | 25.84 | 3.04 | 28.88 | 1.88 | 18.55 |
| JPEG | 94.71 | 98.57 | 97.21 | 98.91 | **99.83** |
| GB | 93.98 | 88.47 | 98.24 | 97.81 | 91.65 |
| AWGN | 71.92 | 79.41 | 94.34 | 92.16 | 85.60 |
| RS | 59.92 | 73.37 | 87.3 | 77.44 | **85.07** |
| MF | 81.73 | 94.35 | 96.19 | 98.54 | 98.35 |
| CE | 18.06 | 32.24 | 36.37 | 27.89 | **46.40** |
| JPEGAF [138] | 48.82 | 71.27 | 86.42 | 92.76 | **95.57** |
| JPEGAF [139] | 51.11 | 78.96 | 91.37 | 91.92 | **93.23** |
| MFAF [140] | 84.47 | 98.69 | 99.51 | 99.01 | 98.26 |
| CEAF [141] | 63.92 | 37.55 | 72.69 | 66.04 | **82.93** |
| Overall Avg. | 63.14 | 68.72 | 80.77 | 76.76 | **81.40** |
| Overall Avg. excluding OR | 66.86 | 75.29 | 85.96 | 84.25 | **87.69** |

compared to the existing approaches for all the considered image manipulations except GB, JPEGAF [139], and CEAF [141] operations, when tested on the BOSSBase dataset as shown in Table 4.9. Similarly, our network achieves better detection accuracy for all image manipulations except JPEG, GB, and CE operations for the Dresden dataset. However, it may be noted that for GB and CEAF [141] operations in the BOSSBase dataset, our model is second best and is around 0.2% lower than the best performing method. Also, for the JPEG and GB operations in Dresden dataset, our method is 0.02% and 0.17%

lower than the best performing method, respectively. Moreover, Table 4 shows that our model outperforms the recent deep learning based scheme [106] with average accuracy improvements of 1.04% and 1.48% for the BOSSBase and Dresden datasets, respectively.

### 4.4.5 Performance evaluation based on cross dataset images

In this subsection, we evaluate the performance of our network by considering cross dataset testing images. In the first experiment, the considered models, trained on the BOSSBase training dataset images, are applied on the Dresden test set images. Similarly, we also perform the experiments considering Dresden training dataset images and BOSSBase test dataset images. The average accuracy results of these cross dataset testing experiments are presented in Table 4.11 and 4.12. It is observed that our MSRD-CNN architecture outperforms the recent multi-purpose forensic schemes by providing higher detection accuracies of 86.49% and 81.40% for BOSSTrain-DRESTest and DRESTrain-BOSSTest, respectively. It is also noted from Table 5 that all the considered forensic methods do not perform well for the original images because the proposed model focuses on the artifacts introduced by the image manipulation operations in the image. But, the original images do not have any manipulation artifacts except the camera fingerprint-related features. Moreover, the original images of these two datasets are acquired from different camera models/devices. Therefore, we also provided the overall average accuracies excluding the original images as shown in Table 4.11 and 4.12. These results are also in favour of proposed MSRD-CNN, with 95.1% and 87.7% accuracies in two settings considered. This highlights the overall best generalization ability of the proposed approach.

## 4.5 Summary

In this chapter, we propose a new method (SNRCN2) for source social media network (SSMN) identification of an input image. We rely on patch-based driven approach in contrast to image resizing approach and suppressing the image content information by utilizing the steganalysis based high-pass SRM filters for noise residuals extraction. These extracted noise residuals are then given to our CNN model to better learn the artifacts left in the image during the post-processing operations of social media networks and perform classification. We have examined the effectiveness of the proposed model against existing methods by considering two different datasets and also a combined dataset. The proposed method consistently provides the superior performance, with image-level accuracy of 99.53% and 100%, and F1-score of 99.42% and 100% on VISION and Forchheim dataset, respectively. On the combined dataset, the proposed method provides the image-level accuracy of 99.10% which is 2% improvement compared to second-best performing method. On all these datasets, the patch-level accuracy results of the proposed method were also the best. We have examined the performance CMI methods on the social media network post-processed images. The proposed CMI method achieves ILA of 83.20% on post-processed images when trained in all social media images. It is observed there is

improvement of 1.40% in terms of ILA when trained on individual social media network images. It is observed that the performance of CMI methods including our CMI method decreases on post-processed images.

We have extended the SSMN identification to IPOs detection and proposed a general-purpose forensic approach for image manipulation detection. Our MSRD-CNN employs a multi-scale residual module to learn the prediction error features adaptively by suppressing the image content information. A feature extraction network further processes these low-level forensic features to provide high-level image manipulation features for better classification. The results consistently show that our model can effectively classify different image processing operations, including anti-forensic attacks. Our model provides overall accuracy improvements of 1.04% and 1.48% as compared to the recent forensic method [106] on BOSSBase and Dresden datasets, respectively. Even in cross dataset testing settings, our model outperforms other approaches and exhibits good generalization ability.

# Chapter 5

# A Dual-Branch CNN for Multispectral Camera Device Identification

## 5.1 Introduction

Determining the image acquisition device is one of the most important aspects of image provenance. With the numerous applications of multispectral images, it is important to identify the multispectral image acquisition camera. For this reason, we propose an approach based on the dual branch convolution neural network (CNN) for the identification of the multispectral image acquisition camera. The proposed method applies a pre-processing layer that extracts Photo-Response Non-Uniformity (PRNU) based noise residual using a wavelet-based denoising filter. These noise residuals are utilized by dual branch CNN model for feature extraction and further classification. We perform the extensive experiments on the newly created dataset that consist of images from multiple multispectral image datasets.

With the advancement of the low-cost image acquisition devices, images have become a major medium of information. There is already lots of applications which rely on RGB images captured using digital cameras or smartphone cameras. However, with technological advancement, multispectral cameras are being adapted for numerous applications related to forensic science that deal with analysing the evidence gathered at the crime scene. The utilization of multispectral imaging in the analysis of gunshot residue on the clothing of a target is of considerable significance in the field of forensic investigations [145, 146]. This analytical approach plays a crucial role in discerning key aspects such as the shooting distance, the presence of primer residue, and the identification of metal particles originating from discharged bullets. Multispectral images also serve as a valuable tool, enabling nondestructive and noncontact analysis of forensic evidence, particularly in the examination of biological fluids [147, 148, 149]. Multispectral images also help in the document analysis that holds pivotal significance within the field of forensic science, serving as a fundamental task for the identification of document forgery [150, 151, 152]. Document forgery encompasses a range of potentially fraudulent modifications applied to documents, including the obliteration or addition of text. The multispectral images also actively used in the biometric related applications [153] like

iris liveness detection [154], contactless palm-vein authentication [155]. In crime cases pertaining to domestic violence and child abuse, the chronological assessment of bruise progression serves as a critical source of evidentiary information. Multispectral image analysis and diffusion theory were employed to visualize skin vasculature and monitor the progression of fresh skin bruises [156]. It may be noted that from last few years, there is rapid growth in the use of multispectral images in different domains in terms of the number of research papers published as shown in Figure 5.1. We have used "multispectral" as a keyword to find the related research papers. Multispectral cameras provide more than three channel-based information compared to the commonly used RGB image cameras. Therefore, the multispectral cameras has the ability to provide non-destructive and real-time analysis of the crime scene, and these analyses can be presented as evidence to the court to prove the crime against criminals. However, with the increase utilization of these multispectral cameras and efficacy of the applications related to multispectral cameras, it raises a question regarding the trustworthiness of multispectral image. The camera device identification is one of the essential task to define the integrity and trustworthiness of the multispectral image. The camera device identification of the multispectral image helps in the linking to the owner of multispectral image. It can help in the investigation when the multispectral image is presented in the court as a piece of evidence. It can also be useful in image manipulation detection for verification.

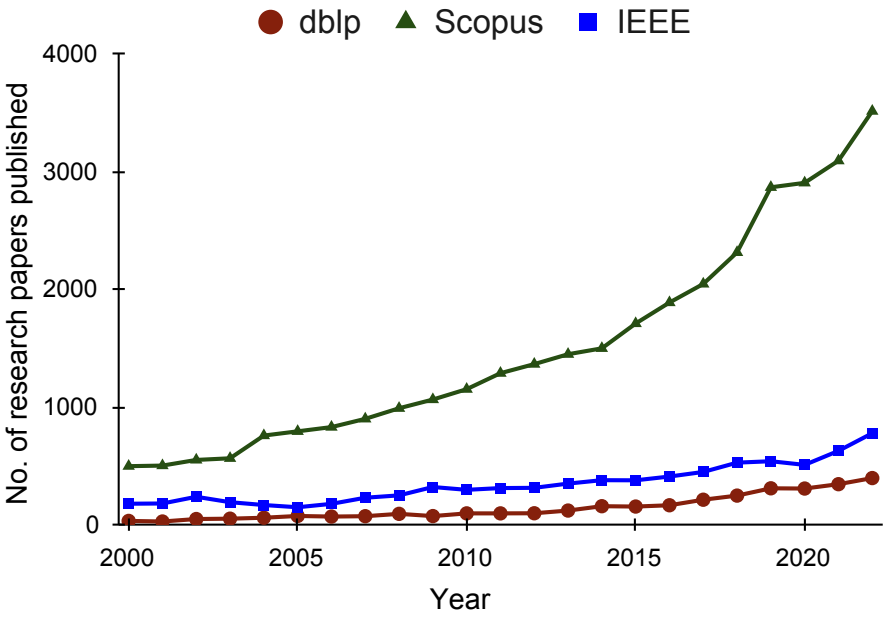



Figure 5.1: Growth in the use of multispectral images in different domains in terms of the number of research papers published.

For effective multispectral camera identification, we understand and analyze the process of multispectral image acquisition and we use the forensic traces left during the acquisition process. A multispectral image of size $M \times N \times K$ has $K$ channel and is mainly generally captured using two different classes of the multispectral imaging systems: (i) mutli-shot

imaging systems, and (ii) single-shot imaging systems. The mutli-shot imaging systems use multiple acquisitions with the help of multiple color filter arrays or multiple illumination while the single-shot imaging systems use a single acquisition to capture the multispectral imaging system. The first category of multi-shot multispectral imaging system uses $K$ shot to capture $K$-channel multispectral image as each shot with specific color filter array captures the $k^{th}$ channel [157]. These imaging systems require a mechanical system attached to camera which helps in changing the filter array or different illumination each time. Another category of the multi-shot multispectral imaging systems capture one row of the $M$ pixel values for all the $K$ channels at each acquisition and thus require $N$ number of acquisition to capture an multispectral image of size $M \times N \times K$. These multispectral imaging systems have push broom line scan technology that needs a focusing mirror lens so that the imaging system can only capture a small portion of the scene. The single-shot imaging systems either use $K$ imaging sensor to capture the $K$ channel simultaneously or use a multispectral filter array similar to color filter array in color camera to capture a multispectral image with single imaging sensor in a single acquisition. The systems with multispectral filter array use multispectral image demosaicking methods [158, 159] to generate the complete multispectral image. Each acquisition process of the multispectral camera may leave discernible noise or artifacts on the multispectral image. Therefore, the multispectral camera device identification method may utilize these discernible noises or artifacts generated during the acquisition process.

The metadata of an image can provide information related to camera. It is embedded in EXIF file of the image and can be forged easily. Also, most of the multispectral images are in .mat extension. So, it is easy to remove the camera information or any other metadata of the multispectral image as .mat files are easily editable. Therefore, the task of camera identification of multispectral images is difficult.

A number of methods have been proposed for the identification of camera of RGB image. A review of most of the methodologies have presented in paper [1]. Initial approaches use the specific artifacts generated during acquisition process of the image with a hypothetical model. These specific artifacts includes color filter array (CFA), lateral chromatic aberration, lens distortion, demosaicking algorithms and image quality matrices (IQM), dust traces, and sensor pattern noise (SPN) [14]. Some methods utilized the statistical features (local binary pattern and co-occurrences) along with machine learning classifiers. However, all of these outlined methods explored the handcrafted features.

Recently, the researchers start using the data-driven methods for camera identification. These data-driven methods use CNNs to extract the camera forensics features from the image and these features further passed to a classifier. These methods use the CNNs to extract features from the RGB image only and mainly varies in terms of number of convolutional layers and the activation functions. The methods [14], [19], [3], [15], [28], and [160] employed 4 layer CNN, 13 layer CNN, 2 layer CNN, a ResNet model, DenseNet models and 3 layer CNN model, respectively. Some CNN based methods apply pre-processing on the image to suppress the content information. This provide

an advantage to CNN to focus more on forensics artifacts. The methods proposed by Rafi et al. [24] , Liu et al. [25], Wang et al. [22], and Tauma et al. [20] have employed remnant blocks for prepossessing with 6 layer CNN, Res2Net based preprocessing with VGG16, local binary patterns with 3 layer CNN, and a $5 \times 5$ high-pass filter with 3 layer CNN, respectively. Few methods also explored the fusion-based CNNs that utilized multiple branches to extract features from the image. Each branch extracts distinct features by applying different filters from other branches. You et al. [29] utilized three-branch CNN with prepossessing, Ding et al. [51] employ a four-branch preprocessing module and ResNet based CNN. Each branch employs a distinct Gaussian-based denoising. Yang at al. [50] and Bayar et al. [21] employed multi-branch fusion-CNN, where each branch extracts different artifacts from the image. Despite the fact that many methods including above mentioned, have been proposed for camera identification of RGB images, there is no work pertaining to camera identification of multispectral images. Also, none of these methods provide the scalability scheme for the multispectral image as the acquisition process of the RGB image is quite different from the acquisition process of the multispectral image. Therefore, we require a dedicated camera identification method for multispectral images.

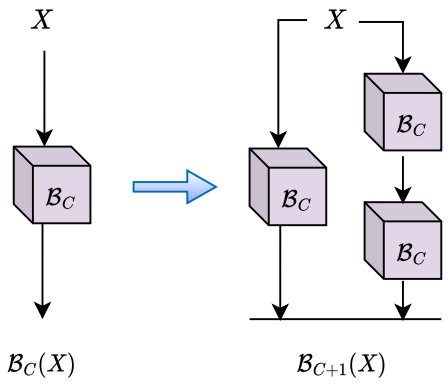



Figure 5.2: Fractal expansion rule

We propose an approach based on the dual branch CNN for the identification of the multispectral image acquisition camera. The proposed method applies a pre-processing layer that extracts PRNU based noise residual using a wavelet-based denoising filter. These noise residuals are utilized by dual branch CNN model for feature extraction and further classification. We construct a dual-branch CNN based on FractalNet [161] for effective feature extraction from multispectral images and better multispectral camera device identification (MCDI). We prepare a new dataset for multispectral camera identification studies, based on multiple publically available multispectral images datasets. Extensive experimental results on multispectral image datasets, considering 4, 5, and 6 channels images, demonstrate the efficacy of the proposed Dual-Branch Convolutional-Batch normalization-ReLU network (DBCBRN) method for MCDI. We extend the existing RGB image-based camera identification approaches for MCDI and also perform the comparative

analysis. It is the first work of its kind, concerning the camera device identification of multispectral images. The remainder of the chapter is formatted as follows: The section 5.2 provides a comprehensive explanation of our proposed dual-branch CNN-based framework. Experiments and results are presented in Section 5.3, along with a description of the dataset employed and a comparison with alternative methods.

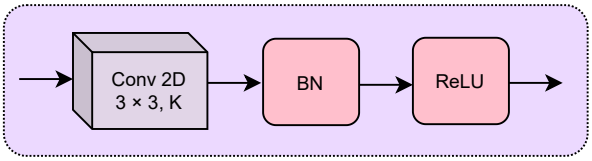



Figure 5.3: CBR block ($\mathcal{B}$) of proposed dual-branch CNN. K is total number of kernels

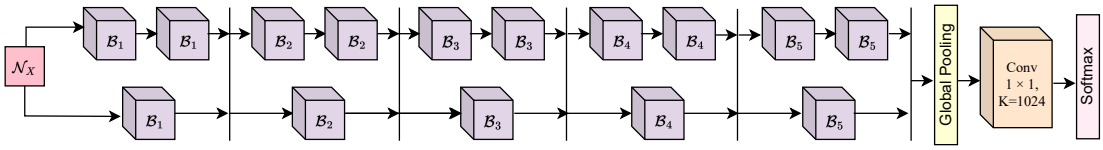



Figure 5.4: The architecture of the proposed dual-branch method. The straight line (|) represent the channel-wise concatenation operation and $\mathcal{N}_X$ represents the noise image of input multispectral image.

## 5.2 Proposed Method for Multispectral Camera Device Identification

This section describes the proposed method, DBCBRN, for the camera identification of a given multispectral image. The architecture of the proposed method is provided in Figure 5.4. The proposed method consists of two key steps: first, the extraction of PRNU based noise residuals from the given input multispectral image and second, the extraction of high-level features related to multispectral cameras from these noise residuals using a FractalNet-based CNN. These steps are further described in more detail in the following two subsections.

### 5.2.1 Noise Extraction

Most camera identification methods rely on suppressing the content information to make the identification method more robust and content-independent. Also, due to more channels in the multispectral image compared to the RGB image, it is beneficial to apply content-suppressing pre-processing before high-level feature extraction from the input image. Also, the CNN-based model tends to learn content-dependent information that is not appropriate for camera identification methods. Therefore, we apply PRNU-based noise extraction pre-processing for extracting noise residuals from the multispectral input image. It is shown in [11] PRNU-based noise can be used for extracting fingerprints related to the camera and also contains the patterns used for camera identification [11, 55].

The noise residuals are extracted using the denoising filter applied independently on each channel of the multispectral image. Considering the input image $X$ and denoising filter $F$, the noise residual can be computed using the following equation,

$$\mathcal{N}_{X_i} = X - F(X_i) \quad i \in \{1...C\}, \tag{5.1}$$

where $i$ is the $i^{th}$ channel of the multispectral input image. We use an adaptive wavelet filter mentioned in [11] as the denoising filter. The denoising filter is the outcome of two-stage process. Firstly, the local image variance is calculated. Secondly, the local Wiener filter is applied in the wavelet domain to obtain a denoised image estimation. The size of noise residuals is the same as the size of input multispectral image.

### 5.2.2 FractalNet-based DBCBRN for High-Level Features Extraction and Classification

The PRNU-based noise residuals are further passed to FractalNet [161] based DBCBRN model. The DBCBRN aims to extract the high-level camera fingerprint features from the noise residuals. The FractalNet has a more straightforward design and shows competitive performance compared to ResNet-based models. The FractalNet-based networks can be built using expanding the basic building block as shown in Figure 5.2. $\mathcal{B}_C$ is the fractal block that represents a specific operation, e.g., convolutional layer or combination of computational layers. The initial FractalNet consist of only one block. The subsequent FractalNet can be expended by recursively increasing the width $C$ as shown in eq.5.2.

$$\mathcal{B}_{C+1}(X) = \mathcal{B}_C(X) \oplus \mathcal{B}_C(\mathcal{B}_C(X)) \tag{5.2}$$

The FractalNet branches are joined together using a concatenation or addition layer. Further, the pooling layer can be applied in depth-wise expansion to reduce the spatial dimension.

The proposed model is a two-branch FratcalNet with a width of 2 ($C = 2$) and depth of 5 as shown in Figure 5.4. Here, the depth defines the number of concatenation layers. We define the CBR block ($\mathcal{B}$) as a fractal block of the network that is a sequence of $3 \times 3$ convolution layer, batch normalization (BN), and ReLU activation function as shown in Figure 5.3. The number of kernels in convolutional layer of CBR block is denoted as $K$. Each branch of DBCBRN consists of a sequence of CBR blocks. The design of DBCBRN is such that the number of CBR blocks in first branch is double of the second branch. The concatenation layer (||) between the branches combines the features from two branches using concatenation by channels and thus, doubles the number of channels. The number of channels after the five concatenations would be sixteen times the number of output channels of the first CBR block. The channels sharing at the concatenation layer provides disparate features from both branches and further providing significant high-level features for classification. It may be noted that we use five type of CBR blocks on the basis of number of channels($K$) as the number of channels doubles after every concatenation layer.

These CBR blocks can be denoted as $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$,, and $\mathcal{B}_5$ with corresponding number of kernels ($K$) as 32, 64, 128, 256, and 512 respectively. Each concatenation layer is followed by a max-pooling layer to down-sample the spatial dimension and pass only the prepotent features to the subsequent layer. The last concatenation layer outputs the feature maps of size $8 \times 8 \times 1024$ for the input noise residual of size $128 \times 128 \times 3$. The output of the last concatenation layer is passed to the global average pooling layer, which reduces the incoming features into $1 \times 1 \times 1024$. Finally, we pass the average-pooled high-level features to $1 \times 1$ convolutional layer with a total number of kernels equal to the number of classes ($N$). Further, the output vector is passed to the softmax layer to generate probabilities for the $N$ multispectral cameras and classification is done using highest probability value. The training loss $\mathcal{L}$ of DBCBRN is carried out by computing cross-entropy loss between the target and the estimated output of the DBCBRN model. The loss function is formulated as:

$$\mathcal{L}\left(f_s\left(f_c\left(H_c; W_c\right)\right); Y\right) = -\log\left(\hat{Y}\right) \times Y, \tag{5.3}$$

where $Y$ is the one-hot encoded target and $\hat{Y}$ is the corresponding output of softmax layer. The function $f_s$ depicted the softmax layer and function $f_c$ depicted the convolutional layer operation on high-level input features $H_C$. The function $f_c$ is parameterized by kernels $W_c$.

## 5.3 Experimental results

In this section, to assess the performance of the proposed method for the camera identification, we conducted a set of experiments and perform comparative analysis of the proposed method on newly constructed dataset. We also evaluate the effectiveness of the noise residuals for the camera identification.

### 5.3.1 Dataset

To the best of our knowledge, there is no standard dataset for multispectral camera identification. Moreover, there is no dataset available that contains images captured by different multispectral cameras. Therefore, we have created a new dataset that contains 609 images captured from 11 different multispectral cameras. All of these images are available in different multispectral datasets, along with various multispectral image-related applications. In [162], a summary of multispectral image datasets and accompanying multispectral cameras is provided. We merge the datasets for which images are acquired using the same camera device. We carefully select the cameras that capture the multispectral image in 420nm-700nm as the common spectral range. We discard the datasets with fewer images and have images of poor resolution in terms of width and height. These images from different datasets have different bit-size and datatype. Therefore, We perform max-normalization [163] to bring each channel of the multispectral image in the [0, 1] range. The list of selected camera along with the spectral range is mentioned

in Table 5.1. For our experiments, we have experimented with k-channel multispectral images where k$\in 4, 5, 6$. To generate the K-channel multispectral image, we select K channels with equal spectral separation, beginning with the first channel. We select (420, 510, 600, 690), (420, 490, 560, 630, 700), and (420, 470, 520, 570, 620, 670) channels from this spectral range 420-700 nm to create 4-channel, 5-channel, and 6-channel image, respectively for our experiment. It is worth noting that 4, 5, and 6-channel images contain distinct channels except the 420nm. Therefore, This provides the comprehensive analysis of all approaches. The newly created dataset will be shared publicly for the multispectral camera identification.

Table 5.1: Details of camera devices in dataset for the experiments.

| Dataset | Camera | Number of images | Spectral range | Number of channels |
|---|---|---|---|---|
| ICVL [164] | Specim PS Kappa DX4 | 201 | 400−700 | 31 |
| CAVE [165] | Apogee Alta U260 | 31 | 400−700 | 31 |
| Natural Scenes [166] | Pulnix TM1010 | 8 | 410−710 | 31 |
| Natural Scenes [167, 168, 169] | Hamamatsu C47429512ER | 71 | 400−720 | 33 |
| TokyoTech [170] | Monochrome camera | 51 | 420−720 | 31 |
| Harvard [171] | Nuance FX, CRI Inc. | 77 | 420−720 | 31 |
| UGR [172] | Photon V-EOS | 14 | 400−1000 | 61 |
| UWA Scenes [173] | Monochrome CCD camera, Basler Inc. | 15 | 400−720 | 33 |
| HS-SOD [174] | NH-AIK hyperspectral camera, Eba-Japan Co | 60 | 350−1100 | 81 |
| SpecTex [175] | Imspector V8 | 60 | 400−780 | 39 |
| Finalyson [176] | The Spectracube Camera | 21 | 400−780 | 31 |

### 5.3.2 Experimental Settings

All the experiments regarding training and testing of the model are performed with a system consisting of RTX 2080Ti GPU of 8GB memory and 3.20 GHz Intel Core i7-8700 CPU with 64GB RAM. The newly created dataset has been divided into two sets i.e., training and testing. We follow a 80-20 % split with 482 images for training and 127 for the testing. Each image is divided into non-overlapping patches of size $128 \times 128$ to meet the model input requirement. Importantly, it also provide more data for training of the model. We extract patches from the centre of spatial dimension of the images and ignore the boundary patches of small size. The label of each patch is same as the label of the corresponding image. It is noticeable that we have not used any resizing on the model input image as there might be information loss due to resizing. Further, The final label of the input image is estimated using the majority voting. However, this provides higher degree of accuracy for proposed model when most patches are classified correctly. We implement the proposed model in PyTorch 1.8.0 framework. We utilize mini-batch stochastic gradient descent and use the batch size of 64 for training of proposed model. We employ the Adam optimizer with a 0.0001 initial learning rate. We train each model under consideration for 100 epochs and all models were converging by 100 epochs. Also, The proposed model converges before the all comparative methods. We choose the model with the highest accuracy on test dataset.

### 5.3.3 Results and Analysis

In this section, we evaluate the performance of the proposed model using a series of experiments and a comparative study. Since, there is no CNN-based method for camera identification of multispectral images, we investigate the state-of-the-art methods used for camera identification of RGB images. All of these methods assume a three-channel RGB input image; however, none of these methods provide or explain scalability to images with more than three channels. Therefore, we have redesigned the initial layer of these methods to comply with multispectral input image and further perform the comparative analysis with proposed method. We consider four methods for comparative analysis i.e. Mayer et al. [177], Liao et al. [80], Rafi el al. [24], and Bennabhaktula et al. [27]. The methods in [177], [24], and [27] have performed CNN-based preprocessing to suppress the image content. Suppressing the content related information leverages better extraction of camera related information. The method in [80] have employed a combination of three dynamic filter and a static filter for suppressing the content information. All of these methods have demonstrated outstanding accuracy on the Dresden dataset [30], and the method in [27] have achieved the highest accuracy of 99.01%. The Dresden dataset is standard RGB image dataset which contains images captured using multiple digital cameras.

To perform evaluation of all the methods including proposed method, we define two evaluation matrices i.e patch-level accuracy and image-level accuracy. The patch-level accuracy is calculated by dividing the total number of correctly classified patches by the total number of patches. It is formulated as:

$$\textit{Patch-level accuracy} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{P_i}\mathbb{I}\left(\hat{y}_{ij} = y_{ij}\right)}{\sum_{i=1}^{T} P_i}, \tag{5.4}$$

where $y_{ij}$ and $\hat{y}_{ij}$ denote the actual and predicted labels of $j^{th}$ patch of $i^{th}$ image, respectively. $P_i$ represents the total number of patches in $i^{th}$ image and $T$ is total number of images. $\mathbb{I}(\mathfrak{C})$ is the indicator function such that $\mathbb{I}(\mathfrak{C}) = 1$ if and only if the condition $(\mathfrak{C})$ is true, and $\mathbb{I}(\mathfrak{C}) = 0$ otherwise. Image-level accuracy is calculated by dividing the total number of correctly classified images by the total number of images. It is formulated as:

$$\textit{Image-level accuracy} = \frac{\sum_{i=1}^{N}\mathbb{I}\left(\hat{Y}_i = Y_i\right)}{T}, \tag{5.5}$$

where $Y_i$ and $\hat{Y}_i$ denote the actual and predicted labels of $i^{th}$ image, respectively. First, we evaluate the proposed method in terms of the patch-level accuracy. Further, we pick the model with highest patch-level accuracy for the evaluation in terms of image–level accuracy.

#### Results on 4 channel image dataset

Table 5.2 shows the comparative analysis of the proposed method. The proposed method achieves the highest patch-level accuracy of 97.52% and image-level accuracy of 97.63% on

Table 5.2: Comparative analysis of different methods on 4,5, and 6 channel dataset

| Method | 4 channel | | 5 channel | | 6 channel | |
|---|---|---|---|---|---|---|
| | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** |
| Mayer et al. [2018] [177] | 95.20 | 88.70 | 94.00 | 93.54 | 92.18 | 89.51 |
| Liao et al. [2021] [80] | 93.28 | 88.00 | 93.68 | 90.40 | 89.16 | 83.20 |
| Rafi et al. [2021] [24] | 92.60 | 84.67 | 95.41 | 95.16 | 96.09 | 93.70 |
| Bennabhaktula et al. [2022] [27] | 94.17 | 92.12 | 94.21 | 95.27 | 94.69 | 95.28 |
| **DBCBRN with noise residuals** | **97.52** | **97.63** | **97.95** | **98.42** | **97.68** | **97.63** |

the 4 channel images of newly created dataset as shown in table 5.2. The method in [27] achieves the second highest image-level accuracy of 92.12%. Our proposed method shows an improvement of 5.51% in image-level accuracy as compare to second best method. All other methods do not perform well considering image-level accuracy. It may be noted that all methods have less image-level accuracy than the patch-level accuracy, the reason is that the incorrect classified images have less number of patches and more than half number of patches are incorrectly classified.

### Results on 5 channel image dataset

On the 5 channel image dataset, each of the comparing methods perform admirably. The patch-level and image-level accuracy for each comparative method surpasses 90% as shown in Table 5.2. However, the proposed method outperforms all other methods by achieving the patch-level and image-level accuracy of 97.95% and 98.42%, respectively.

### Results on 6 channel image dataset

The results on 6 channel image dataset is shown in Table 5.2. The proposed method achieve an accuracy of 97.29% at the patch level and 96.85% at the image level. The proposed method performs better than second-best method in [27] by a margin of 1.57% in term of image-level accuracy.

It is observed that the proposed method performs consistently best in each 4, 5, and 6 channel image dataset. All other methods do not perform consistent on 4, 5, and 6 channel image datasets. All the methods have higher patch-level and image-level accuracy in case of 5 channel image dataset. The reason might be that the channels in 5 channel images provide better distinct information compared to channels in 4 and 6 channel images.

Table 5.3: Comparative analysis of the proposed model with different pre-processing

| Method | 4 channel | | 5 channel | | 6 channel | |
|---|---|---|---|---|---|---|
| | **PLA** | **ILA** | **PLA** | **ILA** | **PLA** | **ILA** |
| DBCBRN without noise residuals | 95.99 | 94.48 | 95.65 | 96.06 | 95.77 | 95.27 |
| DBCBRN with constrained convolutional layer | 96.52 | 94.48 | 95.35 | 92.91 | 96.12 | 94.48 |
| **DBCBRN with noise residuals** | **97.52** | **97.63** | **97.95** | **98.42** | **97.68** | **97.63** |

### 5.3.4 Significance of Noise Residuals

Mostly, different RGB image camera device identification methods uses different preprocessing techniques to suppress the image content. In our work, we apply PRNU based preprocessing to extract noise residuals. To validate the significance of noise residuals in the proposed network setting, we perform experiments with different preprocessing scenarios and present the comparative analysis in Table 5.3. We train the proposed DBCBRN classifier with 4, 5, and 6 channel images for the considered preprocessing settings. Firstly, we train and evaluate the DBCBRN without any pre-processing. It has been observed that the DBCBRN with pre-processing is performing significantly better than DBCBRN trained without pre-processing. In terms of image-level accuracy, the DBCBRN with noise residuals outperforms by 3.15%, 2.36%, and 4.36% when compared to DBCBRN without any pre-processing on 4, 5, and 6 channel datasets, respectively. The noise residuals provide much better artifacts related to the camera device. Interestingly, from Table II and Table III, it can be noted that even without using noise residuals the proposed DBCBRN performs better than all of the other comparative methods. This signifies that the proposed DBCBRN model is a better feature extractor and classifier. In addition, the inclusion of noise residuals further enhance the model prediction capabilities and make it more robust.

Secondly, we test the DBCBRN model with constrained convolutional-based pre-processing, which is a standard technique also used by [177]. The constrained convolutional layer applies dynamic high-pass filtering before DBCBRN. From the results presented in Table III, it has been observed that the performance of DBCBRN with a constrained convolutional layer is significantly lower than the DBCBRN with noise residuals. The DBCBRN with noise residuals shows an improvement of 3.15%, 5.51%, and 3.15% as compared to DBCBRN with constrained convolutional layer on 4, 5, and 6 channel image datasets, respectively,in terms of image-level accuracy. Also, it can be noted that the patch-level and image-level accuracy of DBCBRN with constrained convolutional-based pre-processing is significantly higher than the Mayer et al. [177] method that originally uses constraint convolutional-based pre-processing with CNN, results reported in Table II. This highlights the efficacy of the proposed DBCBRN and shows that it is a better feature extractor than the CNN model in [177].

### 5.3.5 Results on Dresden Dataset

We conduct the experiments to assess the efficacy of the proposed method on RGB images using the Dresden [30] dataset, which encompasses RGB images from 25 camera models. We specifically choose the image samples from 25 camera devices, each representing a unique camera model. We partition the dataset into a 80% training set and a 20% test set. Table IV provides a comparative analysis of various camera device identification methods on the Dresden dataset. The results indicate that the proposed DBCBRN method attains a 100% accuracy at the image level, mirroring the performance achieved by the method

introduced by Bennabhaktula et al. [27]. Furthermore, it is noted that the proposed method attains a patch-level accuracy of 99.10%, surpassing the second-best method by 1.37%. These findings show the effectiveness of the proposed method on RGB images.

Table 5.4: Comparative analysis of the proposed model on the Dresden dataset.

| Method | PLA | ILA |
|---|---|---|
| Mayar et al. [177] | 94.26 | 99.1 |
| Liao et al. [80] | 81.63 | 94.28 |
| Rafi et al. [24] | 95.17 | 99.55 |
| Bennabhaktula et al. [27] | 97.73 | 100 |
| **DBCBRN with noise residuals** | **99.10** | **100** |

## 5.4 Summary

For the high level of inspection, the advance security applications introduce the use of multispectral cameras. In this chapter, we address the problem of camera identification of a multispectral image, which plays a crucial role in multispectral image source forensics. The domain of source forensics has widely been explored but limited to RGB images only. This is the first work that majorly focuses on the source forensics of multispectral images. The proposed method utilized the PRNU-based noise residuals of the given multispectral image for camera identification. These noise residuals are further fed into a FractalNet-based dual-branch CNN to learn the high-level forensics features and further camera identification.

This problem is first of its kind, so there is no standard dataset that contains images from multiple multispectral cameras. Along with an efficient camera identification method, this work contributes in presenting a new dataset that consists of images acquired from multiple multispectral camera devices. We also explored and extended the state-of-the-art RGB image based camera identification methods for the multispectral images. Multiple experiments have been conducted on the developed dataset with 4, 5, and 6 channel images, and the comparison analysis demonstrates that the proposed method outperforms all state-of-the-art camera identification methods. The proposed model achieves highest image-level accuracy of 97.63%, 98.42%, and 96.85% on 4, 5, and 6 channels image dataset, respectively. As there is no publicly available standard dataset for camera identification of multispectral images and there is a scope of further enlarging the developed dataset, we intend to analyse the suggested work on a greater number of cameras in the future. We can also investigate methods for matching multispectral cameras across different spectral bands. Matching multispectral cameras that capture images in different parts of the spectrum poses additional challenges but could be crucial for comprehensive camera device identification. We can also investigate potential vulnerabilities of camera device identification systems to adversarial attacks. Understanding and mitigating vulnerabilities will be important for the reliability of such systems in practical applications.

# Chapter 6

# IITRPR-CMI: A dataset for camera model identification

## 6.1 Introduction

The use of digital imaging is becoming increasingly popular, with smartphones making it easier than ever to take, share, and view images and videos. However, this convenience has also opened the door to potential misuse, raising concerns about the accuracy and authenticity of multimedia content. Examples of illegal material, copyright infringement, and intentional deception through manipulated media demonstrate the need for reliable image forensics. The CMI investigates the feasibility of connecting digital pictures to the device model that was used to take them. Being able to recognize the brand and type of camera that took a certain photograph is critical in a variety of investigative circumstances, particularly in legal matters. Over the past two decades, researchers in the field of image forensics have been working hard to develop methods to address these issues. This research has evolved from heuristic techniques that focused on modeling imaging artifacts to more recent deep learning-based approaches, such as CNN. The CNN model extract the important artifacts related to camera models from the images. Multiple CNN-based method [14, 28, 24, 27] have been proposed for the CMI task. These methods need the training data to learn or extract the inherent fingerprint present in the images. Further, these method need to be evaluated on unseen test images.

The introduction of datasets, such as the Dresden dataset [30] has played a pivotal role in evaluating and advancing forensic algorithms. However, these datasets come with limitations, with Dresden predominantly featuring DSLR and compact cameras and a notable absence of post-processed social media images. The need for comprehensive datasets reflecting the diverse and realistic impact of post-processing on forensic traces is evident. The absence of a latest and diverse dataset for CMI tasks has been a recognized challenge.

The proliferation of smartphones as the primary device for taking pictures has drastically altered the way visual data is created and shared. To evaluate and enhance CMI techniques for smartphone cameras, datasets such as SOCRatES [96], Forchheim [4], VISION [95], and Daxing [178] have been developed. The ever-evolving nature of imaging technology and the widespread use of smartphones necessitates the need for ongoing progress in CMI tasks and related datasets. Therefore, it is essential to upgrade the methods used to identify

camera models, and diverse datasets are essential for assessing these methods. With reliable datasets and a comprehensive understanding of CMI methods, we can improve image forensics and open up new possibilities in the field.

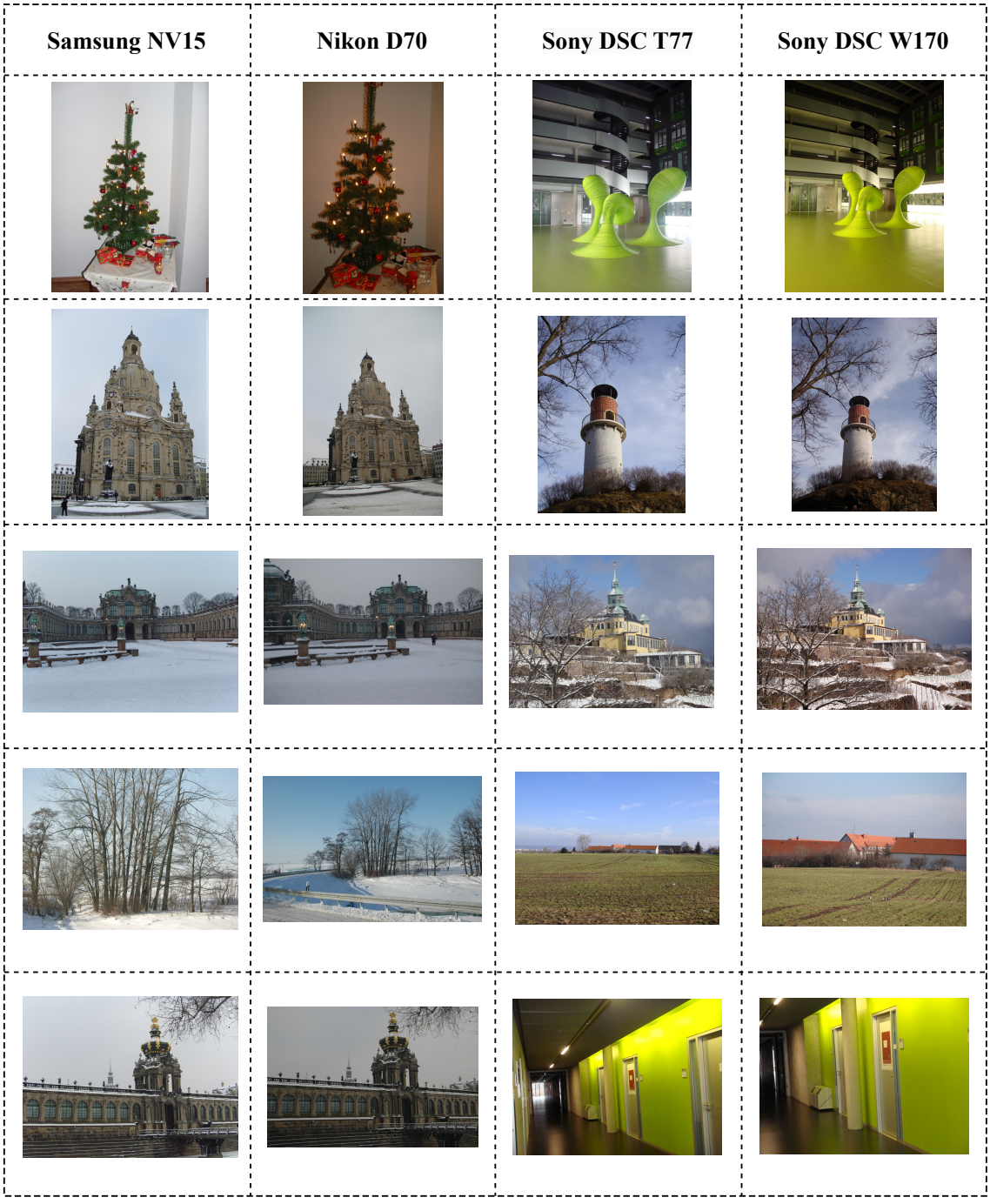



Figure 6.1: Sample images of the Dresden dataset.

In this chapter we have presented a summary of CMI datasets in section 6.2. Further, we have proposed the metholodology for creating a new dataset for evaluating CMI methods in section 6.3. In section 6.4, we have presented results of recent CMI methods on proposed dataset.

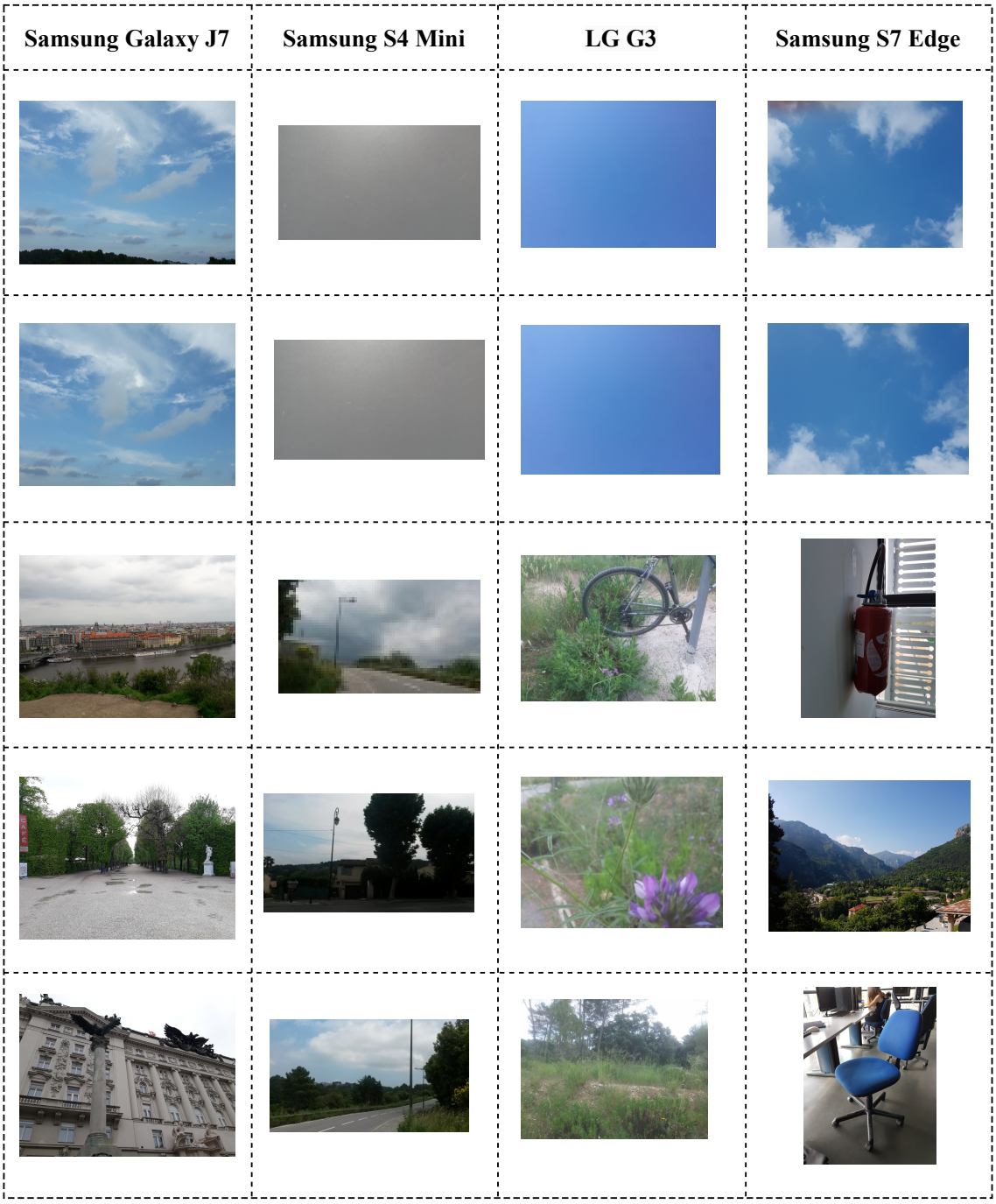



Figure 6.2: Sample images of the Socrates dataset.

## 6.2 Related Work

Multiple datasets have been published to address the evolving challenges of evaluating CMI tasks. The main condition of the CMI dataset is that there should be multiple images from different cameras. However, there is no universal dataset capable of comprehensively evaluating CMI methods, as cameras are evolving and managing all images from devices is very challenging. Consequently, researchers are consistently constructing new datasets over time to facilitate the ongoing evaluation of CMI methods.

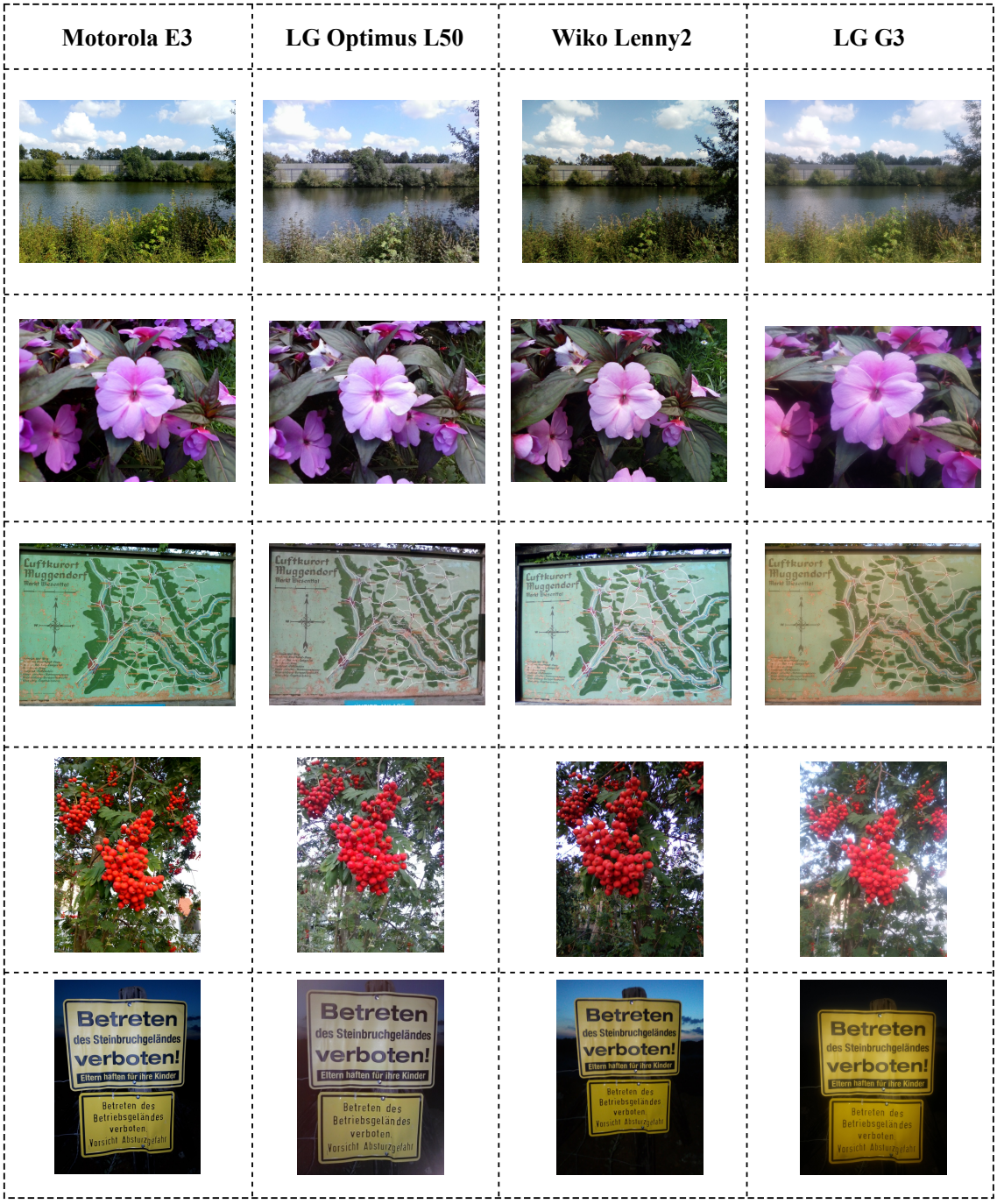



Figure 6.3: Sample images of the Forchheim dataset.

The Dresden dataset [30] is the first of its kind to evaluate forensic problems, providing a large number of images and devices. It includes around 14999 JPEG images and 1491 RAW images from 73 camera devices of 26 camera models. The images are divided into indoor and outdoor scenes, captured using traditional non-smartphone digital cameras. This dataset is the main dataset for the CMI method, offering images from more than two devices belonging to 18 camera models. However, it does not contain images of smartphone cameras, which are widely used by a large population. Additionally, there are no splits (train or test) based on scenes or other parameters for evaluting CMI methods. The

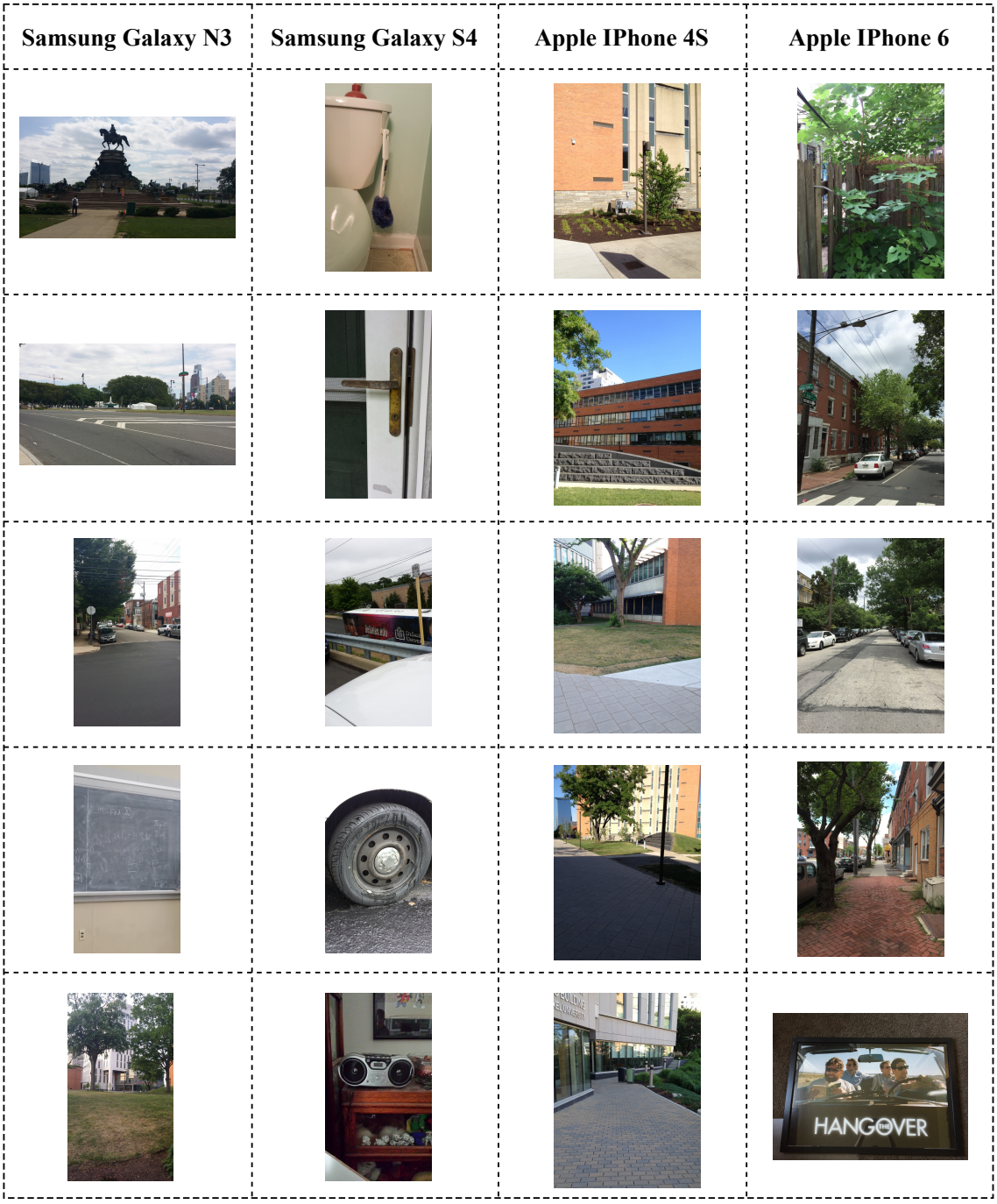



Figure 6.4: Sample images of the SP Cup dataset.

sample images of Dresden dataset are shown in Figure 6.1.

Shullani et al. [95] presented the VISION dataset, which includes 34,000 images from 35 smartphones of 11 major brands. Additionally, they provided a baseline dataset with 16,100 images evenly distributed among all the devices. The VISION dataset also contains post-processed images downloaded from Facebook and WhatsApp.

In 2018, Stamm et al. [82] organized an IEEE SP Cup contest for CMI. The contest featured a training dataset of 2750 images, equally distributed among 10 smartphone camera models. The test dataset was released without labels and the teams had to upload

the results to obtain the ILA of the test dataset. The test dataset also included IPO post-processed images, providing a valid and effective way to evaluate CMI methods. However, the lack of labels in the test dataset limited the scope of inference from the test dataset results. The sample images of SP Cup dataset are shown in Figure 6.4.

The Daxing [178] is one of the largest datasets that provides 43400 smartphone camera images and belongs to 90 smartphone devices of 22 camera models. All these cameras belong to 5 major brands. To the best of our knowledge, this is the largest smartphone image dataset in terms of total images. Images are captured using different orientations of the smartphone: 0, 90, and 180. The scene in images includes the saturated region such as wall, sky and unsaturated region such as trees, stone, objects.

Galldi et al. [96] introduced the SOCRatES dataset for camera recognition on smartphones. This is the largest CMI dataset in terms of the number of camera devices and models, containing around 9700 images taken with 103 different camera devices of 65 smartphone camera models. The main feature of this dataset is that all images were taken by camera device owners, which introduces heterogeneity in the image capturing techniques. The images in the dataset are divided into two categories: background and foreground. The background category includes images of the sky, clouds, and saturated images that contain any object, which are useful for better extraction with respect to the device. The foreground category includes images consisting of single objects such as books, water bottles, lamps, and pots, etc. Despite the large dataset in terms of camera models, the number of images per camera device is much smaller, and the variability in the scene is much lower. The sample images of Socrates dataset are shown in Figure 6.2.

The Forchheim [4] dataset holds significant importance for evaluating camera identification tasks The key importance of this dataset is that it contains images of same scene. The dataset comprises images from approximately 143 scenes, with each scene captured using all available devices. In total, there are 23,106 images, including 3,851 original images from 27 camera devices of 25 camera models. The remaining images are post-processed images uploaded and downloaded from five social media platforms. These post-processed images are important for evaluating the robustness of CMI methods. Given that the dataset offers images from the same scene, it facilitates the extraction of forensic features related to camera models. However, it is equally important to assess their performance on non-similar scene images to comprehensively evaluate CMI methods. It is noteworthy that the Forchheim dataset does not provide the non-similar images. The sample images of Forchheim dataset are shown in Figure 6.3.

There are also few datasets that are initially proposed for different forensics problem. However, these dataset can be used for CMI. The RAISE dataset [179] contains 8156 images from three different camera models of Nikon brand. The scene in images includes: saturated images, objects, and people. The SIDD dataset is presented by Abdelhamed et a. [180]. It is originally proposed for noise removal from smartphone images. However, it can be used for CMI as it provides 30000 images belonging to 5 different smartphone.

## 6.3 IITRPR-CMI dataset

### 6.3.1 Image Acquisition Protocol

The goal of creating a diverse dataset is to incorporate the various important aspects of the datasets that have been presented previously. We have proposed a methodology to create a diverse dataset, as there is no existing dataset that provides a split based on scene for CMI methods evlauation. The Forchheim dataset [4] provides the same scenes across all camera devices, but does not provide a split. Previous methods have randomly split the dataset for evaluating their respective methods. Our approach aims to provide a dataset split for effective evaluation of CMI methods. The second aspect of the CMI dataset is the homogeneity or heterogeneity among the users who have taken the images. All datasets, except for SOCRatES [96], consist of images captured by a single user or a common set of users. The heterogeneity aspect introduces a sense of realism to the captured images. We have collected images that have been taken by their respective owners. The third major aspect of the CMI dataset is that the images contain the same content or scene. This characteristic was introduced in the Forchheim dataset [4]. The images in the dataset should contain images of the same content so that the differences between the same content images will highlight the noise artifacts related to the camera model. Inspired by this, we have also collected images of the same content that have natural scenes and objects.

The proposed methodology for image acquisition involves the participation of two users per smartphone camera device. One user is owner of the camera device, while the other remains fixed across all devices. The owner of the camera device is tasked with capturing images encompassing diverse landscapes and objects, under varying lighting conditions such as sunlight, foggy, or night conditions. There are no restrictions imposed on the owner regarding the scenes captured. Concurrently, the second user is responsible for capturing images falling into four distinct categories: natural, objects, texture, and colors. Natural category images must feature a predetermined set of scenes and be captured under sunlight. The object category images should consist of a fixed set of objects against a uniform background across all images. The third set encompasses images depicting different textures on various objects, with a requirement for continuity across the surface. The color category involves scenes with multiple colors. Additionally, the camera settings, including aspect ratio and auto-focus, remain consistent across all images. A total of 20 different camera devices are included in the image acquisition process. All users have been duly informed about the purpose behind capturing these images.

### 6.3.2 Dataset Organization

This section presents the organization of the newly constructed dataset IITRPR-CMI. This dataset contains images taken with a total of 16 smartphone cameras from a variety of brands, including Vivo, Oppo, Realme, Samsung, OnePlus, and Apple. These cameras are primarily used by a large population of users in India. The dataset includes smartphones

from a wide range of prices, providing a broad spectrum of the population. Also, we have include the images from different OS platforms. All images are stored in the highest JPEG quality and were captured using the default auto-settings of the cameras, including High Dynamic Range (HDR), focus, and white-balance. The list of smartphone cameras, respective brands, operating system and their maximum resolution is shown in Table 6.1.

Table 6.1: Smartphone camera models included in the IITRPR-CMI dataset

| SNo. | Camera Id | Brand | Model | OS Version | Max. Image Resolution |
|---|---|---|---|---|---|
| 1 | D01_Samsung_Galaxy_S20Plus | Samsung | Galaxy S20 Plus | Android 13 | 4032 x 2268 |
| 2 | D02_Nothing_One | Nothing | One | Nothing OS 1.5.6 | 4096 x 3072 |
| 3 | D03_Samsung_Galaxy_A03 | Samsung | Galaxy A03 | Android 12 | 4000 x 3000 |
| 4 | D04_Samsung_Galaxy_M04 | Samsung | Galaxy M04 | Android 12 | 4160 x 3120 |
| 5 | D05_Vivo_V9_Pro | Vivo | V9 Pro | Android 9 | 4160 x 3120 |
| 6 | D06_Apple_Iphone_12Mini | Apple | Iphone 12 Mini | iOS 16.2 | 4032 x 3024 |
| 7 | D07_Apple_Iphone_11 | Apple | Iphone 11 | iOS 16.6 | 4032 x 3024 |
| 8 | D08_Redmi_Note_8Pro | Redmi | Note 8 Pro | Android 9 | 4624 x 3472 |
| 9 | D09_Samsung_Galaxy_J8_10G | Samsung | Galaxy J810G | Android 10 | 4608 x 2592 |
| 10 | D10_Samsung_Galaxy_F41 | Samsung | Galaxy F41 | Android 12 | 4624 x 2136 |
| 11 | D11_OnePlus_8T | OnePlus | 8T | Android 12 | 4000 x 1800 |
| 12 | D12_Vivo_Y02t | Vivo | Y02t | Android 13 | 3264 x 1836 |
| 13 | D13_Oppo_A17k | Oppo | A17k | Android 12 | 3264 x 1840 |
| 14 | D14_Samsung_Galaxy_S20_FE5G | Samsung | Galaxy S20 FE 5G | Android 13 | 4000 x 3000 |
| 15 | D15_Motorola_Moto_G60 | Motorola | Moto G60 | Android 13 | 4000 x 3000 |

We have two categories in the images captured by each camera. The first category consists of images that are captured without any constraint by the respective owner. The second category consist fixed number of images captured by one specific user. The first category defines images which are randomly clicked. There is no limit on the number of images. We have collected all the images available by the user. Although, we have set a constraint of minimum 150 images per camera in this category. Considering the scene details of images in this category, there might be overlap in the scene as most scenes are clicked in same place. However, there is good amount of randomness in scene content. The number of images clicked by each camera is mentioned in the Table 6.2. There are total 3033 number of images respective to first category. Few images of each category in the proposed dataset are shown in Figure 6.5.

Table 6.2: The organization of the IITRPR-CMI dataset.

| SNo. | Camera Id | Random images | Natural | Texture | Objects | Colours | No Content |
|---|---|---|---|---|---|---|---|
| 1 | D01_Samsung_Galaxy_S20Plus | 169 | 54 | 42 | 44 | 15 | 2 |
| 2 | D02_Nothing_One | 225 | 54 | 42 | 44 | 15 | 2 |
| 3 | D03_Samsung_Galaxy_A03 | 231 | 54 | 42 | 44 | 15 | 2 |
| 4 | D04_Samsung_Galaxy_M04 | 251 | 54 | 42 | 44 | 15 | 2 |
| 5 | D05_Vivo_V9_Pro | 168 | 54 | 42 | 44 | 15 | 2 |
| 6 | D06_Apple_Iphone_12Mini | 152 | 54 | 42 | 44 | 15 | 2 |
| 7 | D07_Apple_Iphone_11 | 178 | 54 | 42 | 44 | 15 | 2 |
| 8 | D08_Redmi_Note_8Pro | 196 | 54 | 42 | 44 | 15 | 2 |
| 9 | D09_Samsung_Galaxy_J8_10G | 168 | 54 | 42 | 44 | 15 | 2 |
| 10 | D10_Samsung_Galaxy_F41 | 159 | 54 | 42 | 44 | 15 | 2 |
| 11 | D11_OnePlus_8T | 171 | 54 | 42 | 44 | 15 | 2 |
| 12 | D12_Vivo_Y02t | 188 | 54 | 42 | 44 | 15 | 2 |
| 13 | D13_Oppo_A17k | 169 | 54 | 42 | 44 | 15 | 2 |
| 14 | D14_Samsung_Galaxy_S20_FE5G | 175 | 54 | 42 | 44 | 15 | 2 |
| 15 | D15_Motorola_Moto_G60 | 280 | 54 | 42 | 44 | 15 | 2 |

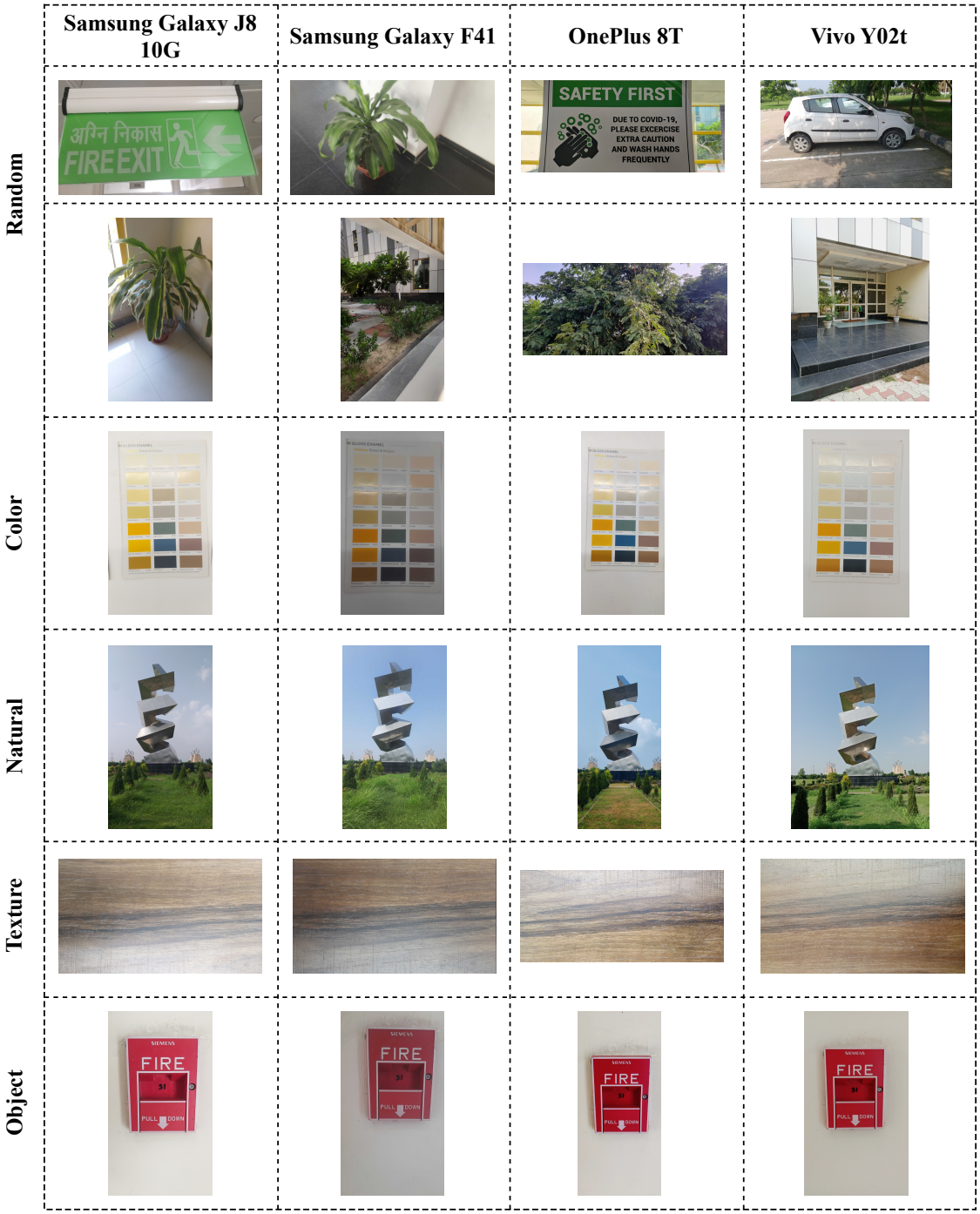



Figure 6.5: Sample images of the IITRPR-CMI dataset.

## 6.4 Results

We have evaluated the state-of-the-art method on the proposed IITRPR-CMI dataset. We have performed experiments in three different settings. In the first setting, we have considered images from the second category that is images consists of same scene. The second category images are divide in 80% training and 20% test set. Most of the prior method have considered the similar scene images for the evaluations. It is assumed that similar scene images provide better camera model fingerprint [4]. Table 6.3 shows the

results of four CMI methods [27, 28, 25] including our proposed CMI method. It is observed all methods have perform better and our proposed CMI method achieves highest ILA of 96.87%. It is inferred that the similar content provide better fingerprints for CMI.

Table 6.3: Comparison of the proposed CMI method and alternative methods with adopting the native PSS of the respective method

| | **PLA** | **ILA** | **APMVC** |
|---|---|---|---|
| Bennabhaktula et al. | 91.78 | 96.09 | 94.49 |
| Rafi et al. | 74.8 | 89.64 | 80.57 |
| Liu et al. | 80.25 | 95.5 | 82.81 |
| **Proposed CMI method** | 93.73 | **96.87** | 95.94 |

Further we have evaluated the proposed CMI method with the Bennabhaktula et al. [27] on two settings. The reason of performing experiments with the method proposed in [27] is that it perform second best in the experiments with similar scene images as shown in Table 6.3. In the first setting we have trained model on similar scene images and tested on random images. In the second setting we have trained model on random images and tested on similar scene images. The comparative results are shown in Table 6.4. The proposed method have performed better than method on [27] in terms all evaluation metrics (PLA, ILA, and APMVC). However, it may be noted that in both settings the accuracies are not that good. In case of training with random images and tested on similar scene images, the ILA is 79.85% which is better than the training with similar scene images and tested on random images. It is inferred that the quality noise features learned from similar scene images are not good enough to correlate with the noise features of random images. However, the random images provide a wide range images in terms of scene to learn camera model fingerprints and provide better correlation with quality noise from the similar scene images.

In this study, we conducted a comparative evaluation of our proposed camera model identification (CMI) method against the technique described by Bennabhaktula et al. (2022). The selection of Bennabhaktula et al.'s method as a benchmark stems from its commendable performance, ranking second best in our tests involving images with similar scenes (refer to Table 6.3. This evaluation comprised two experimental setups. The first involved training the model using images from similar scenes and testing it on a set of randomly chosen images. Conversely, the second setup involved training the model on a random image collection and testing it on images with similar scenes. The outcomes of these experiments are detailed in Table 6.4.

Our findings indicate that the proposed CMI method surpasses the performance of the Bennabhaktula et al. method across all the evaluation metrics we employed, namely Patch-Level Accuracy (PLA), Image-Level Accuracy (ILA), and Average Precision for Multi-View Camera (APMVC). However, it's important to note that in both experimental setups, the overall accuracy levels were moderate.

A notable observation was that when the model was trained on random images and then applied to similar scenes, the ILA achieved was 79.85%, which is higher than the

reverse scenario (training on similar scenes and testing on random images). This suggests that the quality noise features extracted from similar scenes may not effectively correlate with those from random images. Conversely, training with a more diverse set of scenes (random images) appears to enhance the model's ability to learn distinctive camera model fingerprints, improving its correlation with the quality noise characteristics found in similar scenes.

Table 6.4: Comparison of the proposed CMI method trained on different settings

| **Mehtod** | **Trained on similar scene images, Tested on random images** | | | **Trained on random images, Tested on similar scene images** | | |
|---|---|---|---|---|---|---|
| | PLA | ILA | APMVC | PLA | ILA | APMVC |
| Bennabhaktula et al. | 36.1 | 42.16 | 68.84 | 61.28 | 68.56 | 81.94 |
| **Proposed CMI method** | **54.16** | **69.72** | **67.77** | **72.22** | **79.85** | **84.88** |

## 6.5 Summary

This chapter primarily addresses the dataset aspect of CMI methods. We introduced a new dataset called IITRPR-CMI, which comprises images taken with commonly used smartphone cameras. This dataset includes images from 15 different smartphone cameras and is categorized based on content type. It consists of two main sets: one set containing 2880 randomly taken images and another set consisting of 2355 images capturing similar scenes. The latter set comprises 157 instances of the same scene, each captured using 15 different smartphone cameras. To assess the effectiveness of our proposed CMI method and compare it with existing state-of-the-art techniques, we conducted experiments using this dataset. These experiments underscore the significance of the dataset and its potential for further advancements in the field of CMI.

# Chapter 7

# Conclusion

## 7.1 Conclusion

This thesis aims to provide novel and effective methods in the field of source camera image forensics, more specifically for camera model identification for both RGB and multispectral images. The Chapter 1 discusses the motivation for this work, considering the rapid rise of images acquisition via smartphone cameras or multispectral devices. A detailed literature review is presented in this relation in Chapter 2. The novel methods developed and the experiments discussed in Chapter 3 and 4 consider the evaluation on multiple datasets with different competitive methods and in the light of real-world scenarios. Chapter 5 focuses on camera identification for multispectral images and the Chapter 6 presents a new RGB images dataset created by us for CMI evaluation.

The new dual-branch CNN-based framework, a patch-based driven approach as described in Chapter 3, provides an effective, and robust approach for identifying the model of the camera used to capture a RGB image. Compared with prior methods on multiple datasets, the proposed approach offers significant improvements in CMI accuracy for challenging but important application scenarios. In cross-dataset settings also, where the evaluation images not only differ from the training images but are drawn from an entirely different dataset, the proposed method improves PLA between 1.8% and 1.9% and ILA between 3.5% and 5.2%, in comparison to second best performing method. The proposed method also improves CMI robustness, which we quantify in terms of a new APMVC metric that we propose for this purpose.

Addressing the challenges posed by images shared over social media networks such as Whatsapp, Telegram, Facebook, Twitter or Instagram, the thesis explores social media network identification and the performance of CMI methods on social media network post-processed images. We propose a new method (SNRCN2) for identifying the source social media network of an input (shared) image. For better performance and better noise residual extraction, we rely on suppressing the image content information by utilizing the steganalysis based high-pass SRM filters. These extracted noise residuals are then given to our CNN model to learn better artifacts left in the image during the post-processing operations of social media networks and perform classification. The proposed method consistently provides superior performance, with image-level accuracy of 99.53% and 100% on VISION and Forchheim datasets, respectively. Identifying the social media source of images streamlined the process, enabling efficient direction of images to the relevant trained CMI method, thereby reducing training time and enhancing

performance. Extensive experimentation using the Forchheim dataset is performed to validate the effectiveness of directing images to respective social media network trained CMI method. In one setting, we trained CMI methods on an augmented dataset of social media post-processed images and original images. Subsequently in another setting, we trained distinct CMI models for each social media platform. The results showed improved ILA when using CMI models trained explicitly for each social media network images, highlighting the benefits of tailored approaches in camera model identification. In both the settings the proposed dual-branch CMI method performs consistently better than existing CMI methods. The Chapter 4 further presents that related problem of general image processing operation detection and an effective method, MSRD-CNN, for this task.

Considering the use of multispectral cameras for different applications in recent times, we present in Chapter 5 the problem of camera identification of a multispectral image, a problem that plays a crucial role in multispectral image source forensics. The proposed method utilized the PRNU-based noise residuals of the given multispectral image for camera identification. These noise residuals are further fed into a FractalNet-based dual-branch CNN to learn the high-level forensics features and further camera identification. This problem is the first of its kind, so there is no standard dataset that contains images from multiple multispectral cameras. Along with an efficient camera identification method, this work contributes to presenting a new dataset that consists of images acquired from multiple multispectral camera devices. We also explored and extended the state-of-the-art RGB image-based camera identification methods for the multispectral images. Numerous experiments have been conducted on the developed dataset with 4, 5, and 6-channel images, and the comparison analysis demonstrates that the proposed method outperforms all state-of-the-art camera identification methods. The proposed model achieves the highest image-level accuracy of 97.63%, 98.42%, and 96.85% on the 4, 5, and 6-channel image datasets, respectively.

As there are few datasets related to smartphone camera images and smartphone camera technology is evolving very fast, a new dataset IITRPR-CMI is created, as described in Chapter 6, to evaluate and further improve the CMI methods. The dataset contains images acquired from 15 smartphone cameras. The dataset is divided based on the content type. One set consists of 2880 randomly clicked images and another set consists of 2355 similar scene images. The similar scene images are cumulative of 157 same scene images, each captured via 15 different smartphone cameras. The experiments have been performed on this dataset also to evaluate the performance of our proposed CMI method and prior state-of-the-art methods. The different experiments conducted using this new dataset highlights its importance and the scope for further improvement in field of CMI. The dataset may be further extended and would be made publicly available.

In summary, this thesis significantly contributes to the field of source camera image forensics, offering innovative methods and valuable datasets for CMI across diverse imaging scenarios.

## 7.2 Scope of Future Research

This thesis opens up several promising directions for future work in the field of SCIF, each paving the way for advancements in digital image forensics. The current CMI approach, similar to most existing methods, operates in batch mode where the entire set of camera devices and models are known before training. A pivotal area for future development involves creating methods capable of incremental upgrades. As new devices, such as the latest smartphone models, are introduced, there's a growing need for methods that can fine-tune to recognize these new camera models without the necessity of retraining the entire CMI method from scratch. Otherwise also, there is scope for improvement in developing effective CMI models as highlighted by newly developed CMI dataset.

In the future, we aim to expand the scope of the proposed SSMN method. This expansion seeks to encompass a larger set of real-world images, addressing the complexities introduced by the distribution of images across various social media networks and the challenges posed by potential adversarial attacks. Such an expansion is vital for enhancing the practicality and robustness of CMI methods in real-world scenarios.

As there is no publicly available standard dataset for camera identification of multispectral images and there is a scope of further enlarging the developed dataset, we intend to analyze the suggested work on a broader range of cameras and explore methods for matching multispectral cameras across different spectral bands. This area is particularly challenging due to the variety of image capture techniques across the spectrum, but it is important for a comprehensive understanding of CMI.

Another important direction for future work is the creation of large-scale datasets for CMI that represent the increasing diversity and sophistication of modern smartphone cameras. A key challenge in this domain is that modern smartphones increasingly employ computational cameras, where the image is synthesized by fusing together captures from multiple sensors. In such situations, the sensor features indicative of the camera model will likely differ in different images depending on the sensor images used in the fusion and their relative contributions.

Finally, exploring the potential vulnerabilities of camera identification systems to adversarial attacks is essential. Understanding and mitigating these vulnerabilities will contribute significantly to the reliability and applicability of these systems, ensuring their robustness and effectiveness in the dynamic landscape of camera model identification methodologies.

# References


[1] Pengpeng Yang, Daniele Baracchi, Rongrong Ni, Yao Zhao, Fabrizio Argenti, and Alessandro Piva. A survey of deep learning-based source image forensics. *Journal of Imaging*, 6(3):9, 2020.

[2] Hong Cao and Alex C Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4): 899–910, 2009.

[3] David Freire-Obregon, Fabio Narducci, Silvio Barra, and Modesto Castrillon-Santana. Deep learning for source camera identification on mobile devices. *Pattern Recognition Letters*, 126:86–91, 2019.

[4] Benjamin Hadwiger and Christian Riess. The Forchheim image database for camera identification in the wild. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI*, pages 500–515. Springer, 2021.

[5] Matthew C. Stamm, Min Wu, and K. J. Ray Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[6] Kai San Choi, Edmund Y Lam, and Kenneth KY Wong. Automatic source camera identification using the intrinsic lens radial distortion. *Optics express*, 14(24): 11551–11565, 2006.

[7] Lanh Tran Van, Sabu Emmanuel, and Mohan S Kankanhalli. Identifying source cell phone using chromatic aberration. In *2007 IEEE International Conference on Multimedia and Expo*, pages 883–886. IEEE, 2007.

[8] Ahmet Emir Dirik, Husrev Taha Sencar, and Nasir Memon. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Transactions on Information Forensics and Security*, 3(3):539–552, 2008.

[9] Sevinc Bayram, Husrev Sencar, Nasir Memon, and Ismail Avcibas. Source camera identification based on CFA interpolation. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–69. IEEE, 2005.

[10] Mehdi Kharrazi, Husrev T Sencar, and Nasir Memon. Blind source camera identification. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 1, pages 709–712. IEEE, 2004.

[11] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1 (2):205–214, 2006.

[12] Bingchao Xu, Xiaofeng Wang, Xiaorui Zhou, Jianghuan Xi, and Shangping Wang. Source camera identification from image texture features. *Neurocomputing*, 207: 131–140, 2016.

[13] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification based machine learning approach with high order statistics features. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1183–1187. IEEE, 2016.

[14] Luca Bondi, Luca Baroffio, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Processing Letters*, 24(3):259–263, 2016.

[15] Yunshu Chen, Yue Huang, and Xinghao Ding. Camera model identification with residual neural network. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4337–4341. IEEE, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9. IEEE, 2015.

[19] Hongwei Yao, Tong Qiao, Ming Xu, and Ning Zheng. Robust multi-classifier for camera model identification based on convolution neural network. *IEEE Access*, 6: 24973–24982, 2018.

[20] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification with the use of deep convolutional neural networks. In *2016 IEEE International workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2016.

[21] Belhassen Bayar and Matthew C Stamm. Augmented convolutional feature maps for robust CNN-based camera model identification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4098–4102. IEEE, 2017.

[22] Bo Wang, Jianfeng Yin, Shunquan Tan, Yabin Li, and Ming Li. Source camera model identification based on convolutional neural networks with local binary patterns coding. *Signal Processing: Image Communication*, 68:162–168, 2018.

[23] Lisha Yang, Pengpeng Yang, Rongrong Ni, and Yao Zhao. Xception-based general forensic method on small-size images. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pages 361–369. Springer, 2020.

[24] Abdul Muntakim Rafi, Thamidul Islam Tonmoy, Uday Kamal, QM Jonathan Wu, and Md Kamrul Hasan. RemNet: remnant convolutional neural network for camera model identification. *Neural Computing and Applications*, 33:3655–3670, 2021.

[25] Yunxia Liu, Zeyu Zou, Yang Yang, Ngai-Fong Bonnie Law, and Anil Anthony Bharath. Efficient source camera identification with diversity-enhanced patch selection and deep residual prediction. *Sensors*, 21(14):4701, 2021.

[26] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2Net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.

[27] Guru Swaroop Bennabhaktula, Enrique Alegre, Dimka Karastoyanova, and George Azzopardi. Camera model identification based on forensic traces extracted from homogeneous patches. *Expert Systems with Applications*, 206:117769, 2022.

[28] Abdul Muntakim Rafi, Uday Kamal, Rakibul Hoque, Abid Abrar, Sowmitra Das, Robert Laganiere, Md Kamrul Hasan, et al. Application of DenseNet in camera model identification and post-processing detection. In *CVPR workshops*, pages 19–28, 2019.

[29] Changhui You, Hong Zheng, Zhongyuan Guo, Tianyu Wang, and Xiongbin Wu. Multiscale content-independent feature fusion network for source camera identification. *Applied Sciences*, 11(15):6752, 2021.

[30] Thomas Gloe and Rainer Böhme. The Dresden image database for benchmarking digital image forensics. In *Proceedings of the 2010 ACM symposium on applied computing*, pages 1584–1590, 2010.

[31] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on information forensics and security*, 3(1):74–90, 2008.

[32] Xiang Jiang, Shikui Wei, Ruizhen Zhao, Yao Zhao, and Xindong Wu. Camera fingerprint: A new perspective for identifying user's identity. *arXiv preprint arXiv:1610.07728*, 2016.

[33] Sz-Han Chen and Chiou-Ting Hsu. Source camera identification based on camera gain histogram. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV–429. IEEE, 2007.

[34] Miroslav Goljan and Jessica Fridrich. Camera identification from cropped and scaled images. In *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 154–166. SPIE, 2008.

[35] Kurt Rosenfeld and Husrev Taha Sencar. A study of the robustness of PRNU-based camera identification. In *Media Forensics and Security*, volume 7254, pages 213–219. SPIE, 2009.

[36] Bei-Bei Liu, Yongjian Hu, and Heung-Kyu Lee. Source camera identification from significant noise residual regions. In *2010 IEEE International Conference on Image Processing*, pages 1749–1752. IEEE, 2010.

[37] Chang-Tsun Li. Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287, 2010.

[38] Chang-Tsun Li and Yue Li. Color-decoupled photo response non-uniformity for digital image forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(2):260–271, 2011.

[39] Yoichi Tomioka and Hitoshi Kitazawa. Digital camera identification based on the clustered pattern noise of image sensors. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–4. IEEE, 2011.

[40] Floris Gisolf, Anwar Malgoezar, Teun Baar, and Zeno Geradts. Improving source camera identification using a simplified total variation based noise removal algorithm. *Digital Investigation*, 10(3):207–214, 2013.

[41] Xiangui Kang, Jiansheng Chen, Kerui Lin, and Peng Anjie. A context-adaptive SPN predictor for trustworthy source camera identification. *EURASIP Journal on Image and video Processing*, 2014(1):1–11, 2014.

[42] Thibaut Julliand, Vincent Nozick, and Hugues Talbot. Image noise and digital image forensics. In *Digital-Forensics and Watermarking: 14th International Workshop, IWDW 2015, Tokyo, Japan, October 7-10, 2015, Revised Selected Papers 14*, pages 3–17. Springer, 2016.

[43] Bhupendra Gupta and Mayank Tiwari. Improving source camera identification performance using DCT based image frequency components dependent sensor pattern noise extraction method. *Digital Investigation*, 24:121–127, 2018.

[44] Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, and Enrico Magli. Compressed fingerprint matching and camera identification via random projections. *IEEE Transactions on Information Forensics and Security*, 10(7):1472–1485, 2015.

[45] Miroslav Goljan, Mo Chen, Pedro Comesaña, and Jessica Fridrich. Effect of compression on sensor-fingerprint based camera identification. *Electronic Imaging*, 28:1–10, 2016.

[46] Ruizhe Li, Chang-Tsun Li, and Yu Guan. Inference of a compact representation of sensor fingerprint for source camera identification. *Pattern Recognition*, 74:556–567, 2018.

[47] Arjan Mieremet. Camera-identification and common-source identification: The correlation values of mismatches. *Forensic science international*, 301:46–54, 2019.

[48] Zhonghai Deng, Arjan Gijsenij, and Jingyuan Zhang. Source camera identification using auto-white balance approximation. In *2011 International Conference on Computer Vision*, pages 57–64. IEEE, 2011.

[49] Zeno J Geradts, Jurrien Bijhold, Martijn Kieft, Kenji Kurosawa, Kenro Kuroki, and Naoki Saitoh. Methods for identification of images acquired with digital cameras. In *Enabling technologies for law enforcement and security*, volume 4232, pages 505–512. SPIE, 2001.

[50] Pengpeng Yang, Rongrong Ni, Yao Zhao, and Wei Zhao. Source camera identification based on content-adaptive fusion residual networks. *Pattern Recognition Letters*, 119: 195–204, 2019.

[51] Xinghao Ding, Yunshu Chen, Zhen Tang, and Yue Huang. Camera identification based on domain knowledge-driven deep multi-task learning. *IEEE Access*, 7: 25878–25890, 2019.

[52] Kai San Choi, Edmund Y Lam, and Kenneth KY Wong. Source camera identification using footprints from lens aberration. In *Digital photography II*, volume 6069, pages 172–179. SPIE, 2006.

[53] Kai San Choi, Edmund Y Lam, and Kenneth KY Wong. Source camera identification by JPEG compression statistics for image forensics. In *TENCON 2006-2006 IEEE Region 10 Conference*, pages 1–4. IEEE, 2006.

[54] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106, 2007.

[55] Tomás Filler, Jessica Fridrich, and Miroslav Goljan. Using sensor pattern noise for camera model identification. In *2008 15th IEEE international conference on image processing*, pages 1296–1299. IEEE, 2008.

[56] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler. Large scale test of sensor fingerprint camera identification. In *Media forensics and security*, volume 7254, pages 170–181. SPIE, 2009.

[57] Thomas Gloe, Karsten Borowka, and Antje Winkler. Efficient estimation and large-scale evaluation of lateral chromatic aberration for digital image forensics. In *Media Forensics and Security II*, volume 7541, pages 62–74. SPIE, 2010.

[58] Matthias Kirchner. Efficient estimation of CFA pattern configuration in digital camera images. In *Media Forensics and Security II*, volume 7541, pages 383–394. SPIE, 2010.

[59] Guanshuo Xu and Yun Qing Shi. Camera model identification using local binary patterns. In *2012 IEEE international conference on multimedia and expo*, pages 392–397. IEEE, 2012.

[60] Thanh Hai Thai, Remi Cogranne, and Florent Retraint. Camera model identification based on the heteroscedastic noise model. *IEEE Transactions on Image Processing*, 23(1):250–263, 2013.

[61] Simone Milani, Paolo Bestagini, Marco Tagliasacchi, and Stefano Tubaro. Demosaicing strategy identification via eigenalgorithms. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2659–2663. IEEE, 2014.

[62] Chen Chen and Matthew C Stamm. Camera model identification framework using an ensemble of demosaicing features. In *2015 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2015.

[63] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on information Forensics and Security*, 7(3):868–882, 2012.

[64] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Evaluation of residual-based local features for camera model identification. In *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 11–18. Springer, 2015.

[65] Oya Celiktutan, Bülent Sankur, and Ismail Avcibas. Blind identification of source cell-phone model. *IEEE Trans. Inf. Forensics Secur.*, 3(3):553–566, 2008.

[66] Thanh Hai Thai, Florent Retraint, and Rémi Cogranne. Camera model identification based on the generalized noise model in natural images. *Digital Signal Processing*, 48:285–297, 2016.

[67] Kapil Rana, Gurinder Singh, and Puneet Goyal. MSRD-CNN: Multi-scale residual deep CNN for general-purpose image manipulation detection. *IEEE Access*, 10: 41267–41275, 2022.

[68] Gurinder Singh and Puneet Goyal. SDCN2: A shallow densely connected CNN for multi-purpose image manipulation detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 18(3s):1–22, 2022.

[69] Luca Bondi, David Güera, Luca Baroffio, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. A preliminary study on convolutional neural networks for camera model identification. *Electronic Imaging*, 29(7):67–76, 2017.

[70] Anselmo Ferreira, Han Chen, Bin Li, and Jiwu Huang. An inception-based data-driven ensemble approach to camera model identification. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

[71] Owen Mayer and Matthew C Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15:1331–1346, 2019.

[72] Changhee Kang and Sang-ug Kang. Camera model identification using a deep network and a reduced edge dataset. *Neural Computing and Applications*, 32: 13139–13146, 2020.

[73] Belhassen Bayar and Matthew C Stamm. Design principles of convolutional neural networks for multimedia forensics. *Electronic Imaging*, 29:77–86, 2017.

[74] Owen Mayer and Matthew C Stamm. Learned forensic source similarity for unknown camera models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2012–2016. IEEE, 2018.

[75] David Güera, Fengqing Zhu, Sri Kalyan Yarlagadda, Stefano Tubaro, Paolo Bestagini, and Edward J Delp. Reliability map estimation for CNN-based camera model attribution. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 964–973. IEEE, 2018.

[76] Artur Kuzin, Artur Fattakhov, Ilya Kibardin, Vladimir I Iglovikov, and Ruslan Dautov. Camera model identification using convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3107–3110. IEEE, 2018.

[77] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A CNN-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2019.

[78] Zeyu Zou, Yunxia Liu, Wenna Zhang, and Yuehui Chen. Camera model identification based on residual extraction module and SqueezeNet. In *Proceedings of the 2nd international conference on big data technologies*, pages 211–215, 2019.

[79] Md Hasan Al Banna, Md Ali Haider, Md Jaber Al Nahian, Md Maynul Islam, Kazi Abu Taher, and M Shamim Kaiser. Camera model identification using deep CNN and transfer learning approach. In *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pages 626–630. IEEE, 2019.

[80] Xin Liao, Jing Chen, and Jiaxin Chen. Image source identification with known post-processed based on convolutional neural network. *Signal Processing: Image Communication*, 99:116438, 2021.

[81] Kapil Rana, Gurinder Singh, and Puneet Goyal. SNRCN2: Steganalysis noise residuals based CNN for source social network identification of digital images. *Pattern Recognition Letters*, 171:124–130, 2023.

[82] Matthew Stamm, Paolo Bestagini, Lucio Marcenaro, and Patrizio Campisi. Forensic camera model identification: Highlights from the IEEE signal processing cup 2018 student competition [SP competitions]. *IEEE Signal Processing Magazine*, 35(5): 168–174, 2018.

[83] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 171–180. SPIE, 2015.

[84] Xiangui Kang, Matthew C Stamm, Anjie Peng, and KJ Ray Liu. Robust median filtering forensics using an autoregressive model. *IEEE Transactions on Information Forensics and Security*, 8(9):1456–1468, 2013.

[85] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[86] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM workshop on information hiding and multimedia security*, pages 5–10, 2016.

[87] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[88] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[89] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[90] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[91] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[92] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[93] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[94] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[95] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. VISION: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017(1):1–16, 2017.

[96] Chiara Galdi, Frank Hartung, and Jean-Luc Dugelay. SOCRatES: A database of realistic data for source camera recognition on smartphones. In *ICPRAM*, pages 648–655, 2019.

[97] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[98] Matthew C Stamm, Min Wu, and KJ Ray Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[99] Pengpeng Yang, Daniele Baracchi, Rongrong Ni, Yao Zhao, Fabrizio Argenti, and Alessandro Piva. A survey of deep learning-based source image forensics. *Journal of Imaging*, 6, 2020.

[100] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[101] Tomáš Pevnỳ, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. In *Proceedings of the 11th ACM workshop on Multimedia and security*, pages 75–84, 2009.

[102] Jan Kodovsky, Jessica Fridrich, and Vojtech Holub. On dangers of overtraining steganography to incomplete cover model. In *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*, pages 69–76, 2011.

[103] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[104] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[105] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 2019.

[106] Gurinder Singh and Puneet Goyal. GIMD-Net: An effective general-purpose image manipulation detection network, even under anti-forensic attacks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[107] Aparna Bharati, Daniel Moreira, Allan Pinto, Joel Brogan, Kevin Bowyer, Patrick Flynn, Walter Scheirer, and Anderson Rocha. U-phylogeny: Undirected provenance graph construction in the wild. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1517–1521, 2017.

[108] Kapil Rana, Gurinder Singh, and Puneet Goyal. MSRD-CNN: Multi-scale residual deep CNN for general-purpose image manipulation detection. *IEEE Access*, 10: 41267–41275, 2022.

[109] Qing Wang and Rong Zhang. Double JPEG compression forensics based on a convolutional neural network. *EURASIP Journal on Information Security*, 2016 (1):1–12, 2016.

[110] Gurinder Singh and Puneet Goyal. SDCN2: A shallow densely connected CNN for multi-purpose image manipulation detection. *ACM Transactions on Multimidia Computing Communications and Applications*, 2022.

[111] Roberto Caldelli, Rudy Becarelli, and Irene Amerini. Image origin classification based on social network provenance. *IEEE Transactions on Information Forensics and Security*, 12(6):1299–1308, 2017.

[112] Roberto Caldelli, Irene Amerini, and Chang Tsun Li. PRNU-based image classification of origin social network with CNN. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1357–1361, 2018.

[113] Irene Amerini, Chang-Tsun Li, and Roberto Caldelli. Social network identification through image classification with CNN. *IEEE Access*, 7:35264–35273, 2019.

[114] Yijun Quan and Chang-Tsun Li. On addressing the impact of ISO speed upon PRNU and forgery detection. *IEEE Transactions on Information Forensics and Security*, 16:190–202, 2021.

[115] Manisha, A.K. Karunakar, and Chang-Tsun Li. Identification of source social network of digital images using deep neural network. *Pattern Recognition Letters*, 150:17–25, 2021.

[116] Gregory K Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[117] Majid Rabbani. Book review: JPEG2000: Image compression fundamentals, standards and practice, 2002.

[118] Manisha, Chang-Tsun Li, and Karunakar A. Kotegar. A multi-scale content-insensitive fusion CNN for source social network identification. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2981–2985, 2022.

[119] Usage statistics of JPEG for websites. https://w3techs.com/technologies/details/im-jpeg, 2022 (accessed Sept 25, 2022).

[120] Sebastiano Battiato, Oliver Giudice, and Antonino Paratore. Multimedia forensics: discovering the history of multimedia contents. In *Proceedings of the 17th International Conference on Computer Systems and Technologies*, pages 5–16, 2016.

[121] Cecilia Pasquini, Giulia Boato, and Rainer Bohme. Teaching digital signal processing with a challenge on image forensics [SP Education]. *IEEE Signal Processing Magazine*, 36(2):101–109, 2019.

[122] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[123] Matthias Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *Proceedings of the 10th ACM Workshop on Multimedia and Security*, pages 11–20, 2008.

[124] Nahuel Dalgaard, Carlos Mosquera, and Fernando Pérez-González. On the role of differentiation for resampling detection. In *2010 IEEE International Conference on Image Processing*, pages 1753–1756. IEEE, 2010.

[125] Xiaoying Feng, Ingemar J Cox, and Gwenaël Doërr. Normalized energy density-based forensic detection of resampled images. *IEEE Transactions on Multimedia*, 14(3):536–545, 2012.

[126] Babak Mahdian and Stanislav Saic. Blind authentication using periodic properties of interpolation. *IEEE Transactions on Information Forensics and Security*, 3(3): 529–538, 2008.

[127] Tiziano Bianchi and Alessandro Piva. Detection of non-aligned double JPEG compression with estimation of primary compression parameters. In *2011 18th IEEE International Conference on Image Processing*, pages 1929–1932. IEEE, 2011.

[128] Ramesh Neelamani, Ricardo De Queiroz, Zhigang Fan, Sanjeeb Dash, and Richard G Baraniuk. JPEG compression history estimation for color images. *IEEE Transactions on Image Processing*, 15(6):1365–1378, 2006.

[129] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012.

[130] Zhenhua Qu, Weiqi Luo, and Jiwu Huang. A convolutive mixing model for shifted double JPEG compression with application to passive image authentication. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1661–1664. IEEE, 2008.

[131] Matthias Kirchner and Jessica Fridrich. On detection of median filtering in digital images. In *Media Forensics and Security II*, volume 7541, page 754110. International Society for Optics and Photonics, 2010.

[132] Gang Cao, Yao Zhao, Rongrong Ni, Lifang Yu, and Huawei Tian. Forensic detection of median filtering in digital images. In *2010 IEEE International Conference on Multimedia and Expo*, pages 89–94. IEEE, 2010.

[133] Chenglong Chen and Jiangqun Ni. Median filtering detection using edge based prediction matrix. In *International Workshop on Digital Watermarking*, pages 361–375. Springer, 2011.

[134] Matthew C Stamm and KJ Ray Liu. Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(3):492–506, 2010.

[135] Heng Yao, Shuozhong Wang, and Xinpeng Zhang. Detect piecewise linear contrast enhancement and estimate parameters using spectral analysis of image histogram. In *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009)*, pages 94–97, 2009.

[136] Matthew Stamm and KJ Ray Liu. Blind forensics of contrast enhancement in digital images. In *2008 15th IEEE International Conference on Image Processing*, pages 3112–3115. IEEE, 2008.

[137] Matthew C Stamm and KJ Ray Liu. Forensic estimation and reconstruction of a contrast enhancement mapping. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1698–1701. IEEE, 2010.

[138] Wei Fan, Kai Wang, François Cayre, and Zhang Xiong. A variational approach to JPEG anti-forensics. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3058–3062. IEEE, 2013.

[139] Wei Fan, Kai Wang, François Cayre, and Zhang Xiong. JPEG anti-forensics with improved tradeoff between forensic undetectability and image quality. *IEEE Transactions on Information Forensics and Security*, 9(8):1211–1226, 2014.

[140] Wei Fan, Kai Wang, François Cayre, and Zhang Xiong. Median filtered image quality enhancement and anti-forensics via variational deconvolution. *IEEE Transactions on Information Forensics and Security*, 10(5):1076–1091, 2015.

[141] Hareesh Ravi, A Venkata Subramanyam, and Sabu Emmanuel. Ace–an effective anti-forensic contrast enhancement technique. *IEEE Signal Processing Letters*, 23 (2):212–216, 2015.

[142] Yifang Chen, Xiangui Kang, Z Jane Wang, and Qiong Zhang. Densely connected convolutional neural network for multi-purpose image forensics under anti-forensic attacks. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 91–96, 2018.

[143] Aniruddha Mazumdar, Jaya Singh, Yosha Singh Tomar, and Prabin Kumar Bora. Universal image manipulation detection using deep siamese convolutional neural network. *arXiv preprint arXiv:1808.06323*, 2018.

[144] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. Break our steganographic system: the ins and outs of organizing BOSS. In *International Workshop on Information Hiding*, pages 59–70. Springer, 2011.

[145] Félix Zapata, María López-López, José Manuel Amigo, and Carmen García-Ruiz. Multi-spectral imaging for the estimation of shooting distances. *Forensic Science International*, 282:80–85, 2018.

[146] Marcela Albino de Carvalho, Marcio Talhavini, Maria Fernanda Pimentel, José Manuel Amigo, Celio Pasquini, Severino Alves Junior, and Ingrid Távora Weber. Nir hyperspectral images for identification of gunshot residue from tagged ammunition. *Analytical Methods*, 10(38):4711–4717, 2018.

[147] Carolina S Silva, Maria Fernanda Pimentel, José Manuel Amigo, Ricardo S Honorato, and Celio Pasquini. Detecting semen stains on fabrics using near infrared hyperspectral images and multivariate models. *TrAC Trends in Analytical Chemistry*, 95:23–35, 2017.

[148] Samuel Cadd, Bo Li, Peter Beveridge, William T O'Hare, Andrew Campbell, and Meez Islam. The non-contact detection and identification of blood stained fingerprints using visible wavelength reflectance hyperspectral imaging: Part 1. *Science & Justice*, 56:181–190, 2016.

[149] Samuel Cadd, Bo Li, Peter Beveridge, William T O'Hare, Andrew Campbell, and Meez Islam. The non-contact detection and identification of blood stained fingerprints using visible wavelength hyperspectral imaging: Part ii effectiveness on a range of substrates. *Science & Justice*, 56:191–200, 2016.

[150] Lívia Rodrigues e Brito, Angêlica Rocha Martins, André Braz, Amanda Belém Chaves, Jez Willian Braga, and Maria Fernanda Pimentel. Critical review and trends in forensic investigations of crossing ink lines. *TrAC Trends in Analytical Chemistry*, 94:54–69, 2017.

[151] Carolina S Silva, Maria Fernanda Pimentel, Ricardo S Honorato, Celio Pasquini, José M Prats-Montalbán, and Alberto Ferrer. Near infrared hyperspectral imaging for forensic analysis of document forgery. *Analyst*, 139.

[152] Zohaib Khan, Faisal Shafait, and Ajmal Mian. Automatic ink mismatch detection for forensic document analysis. *Pattern Recognition*, 48:3615–3626, 2015.

[153] Leonidas Spinoulas, Mohamed E. Hussein, David Geissbühler, Joe Mathai, Oswin G. Almeida, Guillaume Clivaz, Sébastien Marcel, and Wael Abdalmageed.

Multispectral biometrics system framework: Application to presentation attack detection. *IEEE Sensors Journal*, 21:15022–15041, 2021.

[154] Shejin Thavalengal, Tudor Nedelcu, Petronel Bigioi, and Peter Corcoran. Iris liveness detection for next generation smartphones. *IEEE Transactions on Consumer Electronics*, 62(2):95–102, 2016.

[155] Yung-Yao Chen, Chih-Hsien Hsia, and Ping-Han Chen. Contactless multispectral palm-vein recognition with lightweight convolutional neural network. *IEEE Access*, 9:149796–149806, 2021.

[156] Lise Lyngsnes Randeberg, Eivind La Puebla Larsen, and Lars Othar Svaasand. Characterization of vascular structures and skin bruises using hyperspectral imaging, image analysis and diffusion theory. *Journal of biophotonics*, 3:53–65, 2010.

[157] Sofiane Mihoubi. *Snapshot multispectral image demosaicing and classification*. PhD thesis, Université de Lille, 2018.

[158] Vishwas Rathi and Puneet Goyal. Multispectral image demosaicking based on novel spectrally localized average images. *IEEE Signal Processing Letters*, 29:449–453, 2021.

[159] Medha Gupta, Vishwas Rathi, and Puneet Goyal. Adaptive and progressive multispectral image demosaicking. *IEEE Transactions on Computational Imaging*, 8:69–80, 2022.

[160] Na Huang, Jingsha He, Nafei Zhu, Xinggang Xuan, Gongzheng Liu, and Chengyue Chang. Identification of the source camera of images based on convolutional neural network. *Digital Investigation*, 26:72–80, 2018.

[161] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. FractalNet: Ultra-deep neural networks without residuals. In *International Conference on Learning Representations*, 2017.

[162] Haris Ahmad Khan, Sofiane Mihoubi, Benjamin Mathon, Jean-Baptiste Thomas, and Jon Yngve Hardeberg. Hytexila: High resolution visible and near infrared hyperspectral texture images. *Sensors*, 18(7):2045, 2018.

[163] Yunsong Li, Weiying Xie, and Huaqing Li. Hyperspectral image reconstruction by deep convolutional neural network for classification. *Pattern Recognition*, 63: 371–383, 2017.

[164] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural RGB images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.

[165] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K. Nayar. Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.

[166] Sérgio MC Nascimento, Flávio P Ferreira, and David H Foster. Statistics of spatial cone-excitation ratios in natural scenes. *JOSA A*, 19(8):1484–1490, 2002.

[167] David H Foster, Kinjiro Amano, Sérgio MC Nascimento, and Michael J Foster. Frequency of metamerism in natural scenes. *Josa a*, 23(10):2359–2372, 2006.

[168] David H Foster, Kinjiro Amano, and Sérgio MC Nascimento. Time-lapse ratios of cone excitations in natural scenes. *Vision research*, 120:45–60, 2016.

[169] Sérgio MC Nascimento, Kinjiro Amano, and David H Foster. Spatial distributions of local illumination color in natural scenes. *Vision research*, 120:39–44, 2016.

[170] Yusukex Monno, Sunao Kikuchi, Masayuki Tanaka, and Masatoshi Okutomi. A practical one-shot multispectral imaging system using a single image sensor. *IEEE Transactions on Image Processing*, 24(10):3048–3059, 2015.

[171] Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *CVPR 2011*, pages 193–200. IEEE, 2011.

[172] Jia Eckhard, Timo Eckhard, Eva M Valero, Juan Luis Nieves, and Estibaliz Garrote Contreras. Outdoor scene reflectance measurements using a bragg-grating-based hyperspectral imager. *Applied Optics*, 54(13):D15–D24, 2015.

[173] Zohaib Khan, Faisal Shafait, and Ajmal Mian. Adaptive spectral reflectance recovery using spatio-spectral support from hyperspectral images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 664–668. IEEE, 2014.

[174] Nevrez Imamoglu, Yu Oishi, Xiaoqiang Zhang, Guanqun Ding, Yuming Fang, Toru Kouyama, and Ryosuke Nakamura. Hyperspectral image dataset for benchmarking on salient object detection. In *2018 Tenth international conference on quality of multimedia experience (qoMEX)*, pages 1–3. IEEE, 2018.

[175] Arash Mirhashemi. Introducing spectral moment features in analyzing the spectex hyperspectral texture database. *Machine Vision and Applications*, 29(3):415–432, 2018.

[176] Steven Hordley, Graham Finalyson, and Peter Morovic. A multi-spectral image database and its application to image rendering across illumination. In *Third International Conference on Image and Graphics (ICIG'04)*, pages 394–397. IEEE, 2004.

[177] Owen Mayer, Belhassen Bayar, and Matthew C Stamm. Learning unified deep-features for multiple forensic tasks. In *Proceedings of the 6th ACM workshop on information hiding and multimedia security*, pages 79–84, 2018.

[178] Huawei Tian, Yanhui Xiao, Gang Cao, Yongsheng Zhang, Zhiyin Xu, and Yao Zhao. Daxing smartphone identification dataset. *IEEE Access*, 7:101046–101053, 2019.

[179] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. RAISE: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, pages 219–224, 2015.

[180] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700. IEEE, 2018.