Image Forensics through Illumination Cues and Deep Learning

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

Aniruddha Mazumdar

Department of Electronics and Electrical Engineering

Indian Institute of Technology Guwahati

Guwahati - 781 039, INDIA.

January, 2022

Dedicated to

My Beloved Family,

My Teachers,

and

My Friends

Certificate

This is to certify that the thesis entitled “Image Forensics through Illumination Cues and Deep Learning”, submitted by Aniruddha Mazumdar (136102029), a research scholar in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, for the award of the degree of Doctor of Philosophy, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:

Place: Guwahati

Prof. P. K. Bora
Dept. of Electronics & Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781039, Assam, India.

Declaration

I declare that this written submission represents my ideas in my own words, and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dated: 10^{th} January, 2022

Place: Guwahati

Aniruddha Mazumdar

Roll No.: 136102029

Acknowledgements

Firstly, I would like to express my deep gratitude towards my supervisor, Prof. P. K. Bora, for the endless support and patience during these years of my Ph.D. period. I have learnt a lot from his knowledge and experience, which have made it possible to complete the thesis. He has taught me how to investigate a topic in the scientific literature thoroughly and analytically, and also taught me to effectively communicate my research work to the researchers' community in oral and written format.

I would like to thank my doctoral committee members Dr. Kannan Karthik, Prof. Manas Kamal Bhuyan, and Dr. Prithwijit Guha, for their insightful comments and suggestions.

I wish to thank all my friends for their love and support during my entire education period. Special thanks go to Nayanjyoti Kakati, who has been there in all my highs and lows, constantly supporting me in all my endeavours. I would also like to thank Abhishek Paikray and Shashank Kulkarni for the endless talks and discussions regarding various topics. I thank my seniors, Pranjal Barman, Bhaskarjyoti Medhi and Dibyajyoti Das, who have taught me valuable lessons about diverse topics and also taught me how to enjoy the campus life during the stressful Ph. D. journey. I thank my friends in the IPCV lab, Nayan Moni Baishya, Riju Rangslang, Pallab Jyoti Dutta, Pradipta Sasmal, Debajit Sarma, and Govindaraj P. for the helpful discussions and the wonderful experiences. I thank my friends from other labs, Arghya Chakraborty, Samarjeet Das, and Tilendra Chaudhary, for their joyful company. I also thank my childhood friends Gautam Talukdar, Parag Thakuria, Dibakar Thakuria (both), and Champak Thakuria for their love and support.

I would like to thank the technical and office staff members of the department, Mr. Sanjib Das, Mr. Sidananda Sonowal, Mr. Mukut Baruah, Late Mr. Uday Shankar Uzir, Mr. Dasarath Das, Mr. Sundeep Borah, for their help throughout my Ph.D. work.

Lastly, I would like to take this opportunity to express my heartfelt thanks to my parents for their endless sacrifices and the unconditional love and support, which have helped me to be the person I am today. I wish to express my sincere thanks to my sister Tarali Mazumdar and my

brother Abhijit Mazumdar for their love and guidance that have helped me to achieve things in my life. Special thanks go to my brother-in-law Jitu Talukdar, my little nephew Sannidhya (Rig), and my cousins Jagadish Mazumdar and Simanta Mazumdar for their untiring support and love.

(Aniruddha)

"Great things are not done by impulse, but by a series of small things brought together."

Vincent van Gogh

Abstract

The advancements in image editing and manipulation techniques have enabled people to create visually plausible forged images with little effort. These forged images can then be published on social or electronic media with malicious intentions. Image forensics utilizes computer vision and machine learning techniques to check the authenticity of images.

This thesis proposes a number of forensics methods for detecting and localizing various types of forgeries present in images. The first method focuses on exposing splicing forgeries involving human faces in the front pose. It utilizes the inconsistencies in the lighting environments (LEs) in different faces present in the image under investigation. A novel LE estimation method is proposed based on a low-dimensional lighting model, created from a set of front pose face images captured under different directional light sources. The limitation of the method is that it can detect splicing forgeries only in images containing front-pose faces. This limitation is overcome by the proposed second method that can detect spliced faces of any pose. It is based on extracting an illumination-signature that captures the information regarding the illumination source colours from all the faces and checking the inconsistencies in them for exposing the spliced faces. The dichromatic plane histogram, which is created by utilizing the dichromatic reflection model and 3D Hough transform, is proposed as the illumination-signature. Although this method can detect spliced faces of any pose, it assumes that the skin colours of different faces are the same and hence fails in images containing persons of different races. To detect spliced faces of any pose and skin colour, a deep learning-based method is proposed, that checks the inconsistencies in the visual features present in the face regions of the illumination map (face-IM) computed from the input image. A siamese convolutional neural network (CNN) is employed for comparing the face-IMs in a pair-wise manner and trained on a set of authentic

and artificially-created spliced face-IM pairs. After the training, the CNN part of the siamese network is used for extracting features from the face-IMs. Then, a support vector machine (SVM) is employed for classifying the concatenated features of face-IMs of a pair.

In an attempt to detect different types of image editing operations and also to detect and localize various forgeries in a single framework, a universal forensics method is proposed. The method trains a siamese CNN for checking whether a pair of image patches is modified using the same type of editing operation or not. The trained siamese network is then used for classifying images modified using image editing operations present in the training phase and also those not present in the training stage in a one-shot learning framework. Using the trained siamese network, a method is proposed for localizing and detecting various types of forgeries. To further improve the forgery localization performance, a method is proposed, that trains a two-steam encoder-decoder network specifically for localizing different types of forgeries. One stream of the network learns low-level image manipulation-related features and the other stream learns high-level manipulation-related features. The outputs of the decoders of both the streams are fused using the late-fusion technique for generating a single output prediction map.

Contents

List of Figures

List of Tables

List of Acronyms

DL	Deep Learning
JPEG	Joint Photographic Experts Group
QF	Quality Factor
2D	2-Dimensional
3D	3-Dimensional
PRNU	Photo Response Non-Uniformity
CFA	Colour Filter Array
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
LSTM	Long Short-Term Memory
LE	Lighting Environment
IE	Illumination Environment
LC	Lighting Coefficient
SH	Spherial Harmonics
PCA	Principal Component Analysis
VAF	Variance Accounted For
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
SVM	Support Vector Machine
CFA	Colour Filter Array
CRF	Camera Response Function
FAR	False Alarm Rate
TNR	True Negative Rate

TPR	True Positive Rate
FNR	False Negative Rate
FPR	False Positive Rate
DPH	Dichromatic Plane Histogram
DRM	Dichromatic Reflection Model
GGE	Generalized Gray-Edge
IIC	Inverse-Intensity Chromaticity
IM	Illumination Map
ReLU	Rectified Linear Unit
GMM	Gaussian Mixture Model
SPAM	Subtractive Pixel Adjacency Matrix
SRM	Spatial-domain Rich Model
LBP	Local Binary Pattern
FS	Forensics Similarity
SP	Similarly Processed
DP	Differenty Processed
GB	Gaussian Blurring
MF	Median Filtering
UA	Unaltered
RS	Resampling
AWGN	Additive White Gaussian Noise
GC	Gamma Correction
MCC	Matthews Correlation Coefficient
ISED	Image-Stream Encoder Decoder
NSED	Noise-Stream Encoder Decoder
FCN	Fully-Convolutional Neural Network
HPF	High-pass Filter
cIoU	Per-Class Intersection Over Union

Mathematical Notations

$\mathbf{I}$	Vector
$\mathbf{u}$	Vector
a	Scalar
P	Set
$F(a, b)$	Function with arguments a, b
$(.)^T$	Transpose Operation
$diag(a_1, a_2, ..., a_N)$	$N \times N$ Diagonal Matrix with a_1, a_2, ..., a_N in its leading diagonal positions
$\mathbf{1}$	All-ones Matrix
$d(.,.)$	Distance between two Vectors
corr(.,.)	Correlation Measure between two Vectors
$\max_i(\mathbf{a}_i)$	Maximum Value of the Array $\mathbf{a}_i$
$\min_i(\mathbf{a}_i)$	Minimum Value of the Array $\mathbf{a}_i$
$\|\mathbf{a}\|_1$	L_1 Norm of a Vector $\mathbf{a}$
$\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N\}$	Set of vectors $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N$
$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$	Column Vector with Elements a_1, a_2, a_3
$H(a, b)$	2D Histogram
$\overline{H}$	Mean Value of the Histogram H
$\sigma(.)$	Sigmoidal Activation Function
$\|a\|$	Absolute Value of the Scalar a
$\text{argmax}_c \mathbf{a}_i$	Arguments of the Maxima of the Array $\mathbf{a}_i$

1(.)	Indicator Function
∩	Intersection Operation between Sets
∪	Union Operations between Sets
$\|P\|$	Cardinality of the Set P

1

Introduction

Images have become one of the most important tools to convey news and stories in social and print media nowadays. An image draws greater attention than texts and can spread information more quickly. However, digital images are more vulnerable to forgery. The availability of numerous image manipulating software, *e.g.*, Adobe Photoshop and Gimp, has made the creation of visually plausible forgery a simple task. These forged images, when used in platforms like electronic media or courts, may mislead people and sometimes create chaos in the society. These concerns necessitate the development of image forensics techniques that can check the authenticity of images before using them as critical information.

Image forensics aims to detect forged or manipulated images by utilizing computer vision and machine learning techniques. The existing literature has utilized various types of traces for exposing forgeries. These are discussed in more detail in Section 1.2. Among the different traces, the illumination traces have proven to be very effective in detecting splicing forgeries, particularly those involving human faces, due to the difficulty in matching the exact illumination condition in the forged images [7], [8], [9].

Deep learning (DL) techniques [10], [11], [12] have become the most effective strategy for learning optimal features from speech, text, and image data for different types of tasks, such as prediction and classification. For instance, convolutional neural networks (CNNs) [10] have outperformed the hand-crafted feature-based methods by large margins in various image classification tasks, such as medical image classification [13] and road scene segmentation [14]. These successes of DL in various computer vision tasks have inspired the forensics community to develop DL-based methods for exposing various types of forgeries. DL techniques, such as CNNs, generative adversarial networks (GANs) [12], long short-term memory (LSTM) network [11], and encoder-decoder networks [15], [14], have been utilized by many researchers for detecting and localizing forged regions in images [16], [17], [6].

This thesis focuses on developing forensics methods that exploit the illumination traces and DL techniques for the detection and localization of various types of forgeries.

1.1 Image Forgeries

There are various ways of creating image forgeries. Depending upon the creation procedure, the forgeries can be broadly divided into various categories. The most common forgeries are the following: i) *splicing*, ii) *copy-move*, iii) *retouching*, and iv) *content-removal*. They are described below.

1.1.1 Splicing Forgery

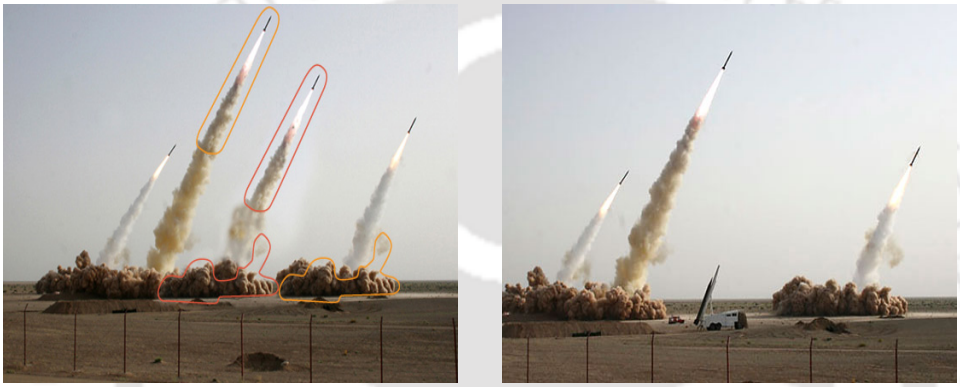
In splicing forgery, parts from different images are copied and pasted onto a single image to create a scene that never took place. Various post-processing operations, *e.g.*, brightness or contrast change, colour change, and rotation, are carried out to make the parts from different images look visually similar. Figure 1.1 shows an example of a splicing forgery depicting American politician John Kerry sharing a stage with anti-Vietnam war activist Jane Fonda [1]. The forged image was circulated via the Internet around the world and created a political storm in America. This incident shows the impact of forgeries on society.

Figure 1.1: A famous example of the splicing forgery involving American politician and diplomat John Kerry and political activist Jane Fonda. The spliced image (on the right side of the arrow) was created by copying Jane Fonda (the person on the right side in the spliced image) from a different image and pasted on the left image with John Kerry [1].

1.1.2 Copy-Move Forgery

In copy-move forgery, regions from an image are copied to a different location to either clone some objects or hide some regions in the image. The copied regions are additionally post-processed with different image processing operations like resizing, rotating, brightness or

contrast change. Figure 1.2 shows a copy-move forgery example (Figure 1.2a) and the authentic image (Figure 1.2b) from which the forgery was created. The forged image was shared by Iran's state media showing the successful launch of four missiles. However, the same media later shared another image, presumably authentic (Figure 1.2b)), captured from the same vantage point at the same time [2]. In the forged image, the missile and the smoke regions encircled by red colour are cloned from the authentic regions encircled by orange colour.



a) Forged Image b) Authentic Image

Figure 1.2: An example of copy-move forgery [2], where a rocket is clonned to create the forged image.

1.1.3 Retouching Forgery

Image retouching does not change the content of an image significantly. It enhances certain attributes of parts of an image, such as texture, shape, lighting, brightness, contrast, and colour. A famous example of the retouching forgery is shown in Figure 1.3a), where the mugshot of former American football star O. J. Simpson was darkened, and the lighting was modified by Time magazine to give him a more menacing look. Figure 1.3b) shows the original mug shot of Simpson [1].

1.1.4 Content-Removal Forgery

In content-removal forgery, regions from an image are removed and filled with inpainted pixels. This type of forgery is done to hide some critical information in an image. Figure 1.4 shows an example of a content-removal forgery, where former Soviet Union's dictator Stalin managed to remove one of his aides, Nikola Yezhov, a secret police officer from the photo-

a) Retouched Image b) Authentic Image

Figure 1.3: An example of retouching forgery. The skin tone of the famous American football star and a murder convict O. J. Simpson is changed in the forged image by Time magazine to incite negative sentiment about him [1].

graphic records [3]. Yezhovm, who is seen standing left to Stalin in the authentic image shown in Figure 1.4a), is removed in the forged image shown in Figure 1.4b) and the missing pixels are filled from surrounding pixels using image inpainting techniques.

a) Original Image b) Content-Removed Image

Figure 1.4: An example of content-removal forgery. Figure a) shows an authentic image where former Soviet Union's dictator Stalin is seen with Nikola Yezhov, a secret police official, on his left. In the content-removed image in Figure b), Stalin's censors removed Yezhov from the image and filled the region with surrounded pixels using inpainting techniques [3].

In addition to the aforementioned forgery techniques, different types of image editing or image manipulation operations are applied to images for various reasons. For instance, image editing operations, such as *Gaussian blurring*, *median filtering*, *gamma corrections*, are applied to artificially enhance some features from images to make them look visually different from their raw versions captured by the camera devices. Most of the time, these image editing oper-

ations are carried out for harmless reasons, such as to denoise an image or change the contrast of the image. However, these operations can also be applied for malicious purposes. For example, in creating forgeries, various image editing operations are applied as a post-processing operations to make the forged image look visually undetectable. The image editing operations can also used to fool certain image forensics methods, as discussed in Section 1.3.

1.2 Image Forensics

A perfectly curated forged image looks authentic to the human eyes. However, every forgery process leaves behind a unique trace of manipulation in the forged image. For example, in a spliced image, there may be regions copied from different images captured under different camera models. This will introduce inconsistencies in the camera sensor-based features, such as sensor noise [18] and camera response functions [19], between the authentic and the forged regions. In a copy-move forgery, the forged regions may be resized and rotated to make the image look visually plausible. These resizing and rotation operations will introduce resampling traces in the forged regions. The goal of image forensics is to develop various techniques to expose forgeries by utilizing these manipulation-related traces.

The image forensics methods can be broadly divided into two categories: a) *active* and b) *passive*. In active image forensics, a signature or watermark is embedded imperceptibly in an image at the time of its acquisition. If any manipulation is performed on the image later, the signature will change. The authenticity of the image can be decided by matching the earlier and the later signatures. Passive image forensics methods do not require any prior knowledge about the image. They check the authenticity of an image using its visual contents only. Since most of the image capturing devices, such as digital cameras and smartphones, do not embed any signature while capturing an image, the passive forensics approaches are more suitable for detecting the forgeries involving the images captured by these devices.

H. Farid led the pioneering work in passive image forensics by introducing the usage of different types of manipulation-related traces, that are left behind in forged images. Popescu and Farid proposed one of the earliest passive forensics methods in [20], where they introduced the idea of detecting the resampling traces for detecting images that have undergone resizing

operations. In [21], Popescu and Farid introduced the usage of the correlation of image pixels due to colour filter array interpolation for forgery detection. Johnson and Farid [22] proposed the first forensics method that utilizes the inconsistencies in illumination for exposing splicing forgeries.

Based on the type of traces utilized to expose the forgeries, the image forensics methods can be further divided into various categories, *e.g., JPEG compression-based* [23], [24], *noise-based* [25], [26], *lighting environment-based* [27], [22], [7], [28], *illumination colour-based* [29], [9], [8], *resampling-based* [20] and *camera sensor-based* [30], [31]. Furthermore, a number of *deep learning-based data-driven methods* [32], [16], [17] have been proposed recently that do not assume any prior knowledge about a particular type of trace but rather learn the traces from the training data itself. These methods are briefly explained below.

1.2.1 JPEG Compression-based Methods

Since a large proportion of digital images available are compressed and stored in JPEG format, many forensics methods have focused on utilizing the JPEG-related traces to expose forgeries. For instance, in [23], the mismatch in the JPEG quality factors (QFs) used for compressing the spliced and authentic regions of a forged image is used as a cue of the forgery. In [33], the misalignment of the 8×8 block grids is utilized for localizing the splicing and copy-move forgeries.

1.2.2 Noise-based Methods

The noise-based methods work based on the assumption that different parts of an authentic image will have similar noise characteristics. The noise is introduced to images during either acquisition or in-camera processing stage. The types of noise that get introduced during acquisition are thermal noise, shot noise, *etc*. The in-camera processing of the raw pixel values introduces various types of noise to the image, such as impulse noise due to analog-to-digital conversion error and noise due to errors during quantization of the pixel values. In an authentic image, it is reasonable to assume that the noise level will be almost similar at different parts. In a spliced image, the forged regions will have different noise characteristics than that of the authentic regions. Therefore, by checking the inconsistencies in the noise statistics at different

parts of an image, the forged regions can be identified. For instance, in [25] and [34], the mismatch in the local variances of noise extracted from the authentic and spliced regions is utilized for exposing the forgery.

1.2.3 Lighting Environment-based Methods

The formation of images in a camera is a very complex process, which is largely determined by the interaction of light sources with the surfaces present in the scene. Given a surface of known geometry, it is possible to estimate the light source directions with respect to the camera, under certain assumptions. In an authentic image, captured under distant light sources, all the different objects are captured under the light sources from the same directions. However, in a spliced image, there is a high chance that the spliced regions is captured under light sources at different locations than the authentic regions. The lighting environment-based methods are based on checking the inconsistencies in the lighting environments or the light source directions that illuminates the objects present in an image. The methods proposed in [22], [8] estimates the lighting environments in terms of the spherical harmonics coefficients [35] from different objects present in an image and then compares them to check the inconsistencies for exposing the splicing forgery. The limitation of the approaches proposed in [22] and [8] is that they require the knowledge of the 3D geometry of the surfaces present in the scene, finding which is an ill-posed problem.

1.2.4 Illumination Colour-based Methods

The colours reflected by surfaces present in an image are determined by the colours of the illumination sources and the surface albedos. There are various methods available in computational colour constancy [36], [37], [38] that can estimate the colour of the illumination source by making some assumptions about the surface albedos. Since the spliced regions in a forged image come from different images, there is a high chance that they were captured under different illumination sources. Therefore, the source illumination colours estimated from the authentic parts of a spliced image will be different from those estimated from the spliced parts. Based on this motivation, in illumination colour-based methods, the mismatch in the illumination source colour is exploited for revealing the splicing forgeries. In [29], [39], [40], the illumination

colour estimated from different parts of an image is used for exposing the splicing forgery. Since these methods assume a single illumination throughout the image, they cannot be applied to images under multiple illuminations. In [9], [41], the illumination colours estimated from the face regions of the persons present in an image are used for creating an intermediate representation of the face images. Then, various hand-crafted features are extracted and classified using different pattern recognition techniques for detecting the spliced faces. In [42], a pre-trained CNN is used extract features from the intermediate representation, which are classified using a support vector machine (SVM) to detect the spliced faces. The limitation of the approaches, proposed in [9], [41] and [42], is that the features extracted are not optimal for image forensics.

1.2.5 Resampling-based Methods

While creating a realistic-looking forgery, it is almost necessary to resize, rotate or stretch the forged regions to match the authentic region. The resampling-based methods detect the artificial correlation traces introduced due to the resizing, rotating, and stretching of the manipulated regions. For example, a resampling-based method is proposed in [20], where the periodic pixels in a resampled regions in a forged image are detected using the expectation-maximization (EM) algorithm. In [43], the resampled images are detected by computing the Radon transformation [44] of the derivative of the image pixels.

1.2.6 Camera sensor-based Methods

In a camera, an image is formed when the sensor records the pixel values from the input light that falls on it. While recording the image pixel values, the sensor introduces various unique fingerprints, such as the photo response non-uniformity (PRNU) noise and the colour filter array (CFA) interpolation algorithms, camera response function (CRF), *etc.* The PRNU is a type of fixed pattern noise present in images due to the imperfections in the sensor, which results in a deterministic pattern of bright and dark pixels in the images. Most of the digital cameras record only one colour information out of the RGB colours, in each sensor by employing a single 2D array of sensors in conjunction with a CFA, *e.g.,* the Bayer filter [45]. The missing two colour information are computed by applying a demosaicing algorithm, *i.e.*, by interpolating the adjacent pixel values. This interpolation introduces specific correlations among the neighbouring

pixels, which can be used as a unique fingerprint for the camera model. Every image capturing device employs a CRF to map the scene irradiance to pixel intensity values non-linearly. Since the sensor of each camera model have a unique CRF, it is also used as a camera model fingerprint. The camera sensor-based methods expose forgeries by checking the inconsistencies in these sensor-based fingerprints. For instance, Chen *et al.* uses the PRNU noise for detecting the source camera device and locate forgeries. In [46] and [47], the inconsistencies in the reconstruction error of the demosaicing algorithms, known as the CFA artefacts, is utilized for detecting forgeries. In [48], the consistency in the CRFs at different edges of an image is used for exposing splicing and copy-move forgeries.

1.2.7 Deep learning-based data-driven methods

Following the successful application of DL techniques, such as CNNs, encoder-decoder networks, and GANs, in various computer vision tasks, the forensics community has focused on developing deep learning-based methods for detecting various image manipulations and forgeries [49], [50], [32], [6], [51], [17]. Unlike the methods that assume the knowledge about the type of forensics traces, the DL-based methods learn directly from the training data without using any hand-crafted features to detect the manipulation traces. Hence, these methods can learn more optimal forgery-related features than the methods that extract hand-crafted features to detect particular forensics traces. In [49], a method is proposed to classify images that are edited using median filtering operation by using a CNN, where the first layer has a fixed set of weights for computing median filtering residuals. In [50], a method is proposed that can detect multiple image editing operations in a single framework using a CNN, where the first layer learns a set of filters adaptively from training data for computing high-pass residuals. However, this method has the limitation that all the image editing operations have to be known *a priori* during the training stage. Furthermore, the DL-based methods can learn and fuse multiple forensics cues in a single end-to-end framework to expose various types of forgeries. In [32], a multi-task fully-CNN is employed to localize splicing forgeries. A two-stream forensics method is proposed in [6], where the first stream employs the Radon transform [44] and the long short-term memory (LSTM) network [11] for computing the low-level feature related to resampling traces,

and the second steam employs a CNN for learning high-level features related to artificial contrast and unnatural object edges. Then a single decoder is employed to localize various types of forgeries, *i.e.,* splicing, copy-move, and content-removal, in a single framework. In [17], a GAN-based discriminative segmentation model is trained in a mixed adversarial setting to localize different types of forgeries. In [16], a two-stream region-based CNN (R-CNN) is utilized for learning both the high-level features and the low-level features automatically from the training data for localizing the above-mentioned forgeries. In [52], a siamese neural network-based method is proposed to extract the camera sensor noise (*i.e.,* PRNU) from a single image itself, and use it to localize the splicing forgeries. Although the above two-stream methods show that the fusion of high-level and low-level features are more effective in localizing forgeries, they have the following limitations: 1) the low-level feature computed in [6] is a hand-crafted one and hence may not be optimal, 2) the method proposed in [16] can only give bounding box-level localization of the forged regions.

1.3 Image Anti-Forensics

Although the image forensics methods are developed to expose the forgeries in images, many *anti-forensics* methods are available in the literature, that can fool the forensics methods. The anti-forensics methods apply various image editing operations on the forged image to remove the traces of manipulations left by the forgery process. For example, the traces of resizing operation, which is generally applied on forged regions to make them look visually undetectable, can be erased by applying median filtering [53].

1.4 Research Motivation and Problem Statement

Although image forensics is a well-researched area, there are several research issues. We have identified a number of research challenges. Our research is motivated by the following:

1) The lighting environment-based and the illumination colour-based forensics methods are more effective in detecting face splicing forgeries in real-life images, *i.e.,* highly compressed and low-resolution images. Also, it is hard to match the exact illumination condition in a spliced image [9], [54] as human eyes are not very good at judging the inconsistencies in the illumination condition in images [55], [56]. Furthermore, various anti-forensics methods have been

proposed to fool different types of forensics methods, such as the compression-based and the camera-based forensics methods [57], [58]. In our opinion, the traces related to lighting environment and illumination colour are more likely to be present in a spliced image. Hence, the lighting environment-based and the illumination colour methods are more reliable in detecting perfectly curated realistic-looking splicing forgeries. As already mentioned, the existing lighting environment-based methods require the knowledge of 3D surface geometry for accurately estimating the lighting environment. On the other hand, the existing illumination colour-based methods have the following limitations: i) some of these methods are applicable only to images captured under a single illumination source, ii) the illumination colour-related features extracted by these methods are not optimal for forgery detection. Therefore, there is scopes for research in lighting environment and illumination colour-based image forensics.

2) It also an essential task in image forensics to detect other types of forgeries, such as copy-move, retouching, and content-removal, that involve arbitrary regions in a forged image. This is important because, in real-life forensics scenarios, the type of forgery and the region involving the forgeries are generally not known *a prioi*. Thus, there is a need to develop forensics methods that can detect all types of forgeries, involving arbitrary regions, in a single framework.

3) Another important task in image forensics is to detect the image editing operations on images as it helps in exposing various forgeries and also in checking whether an image has gone through image anti-forensics operations. As discussed in Section 1.2.7, the DL-based methods are very effective in detecting various types of image editing operations carried out on images and also in detecting and localizing various forgeries involving arbitrary image regions. However, there are certain limitations of the existing methods: i) the existing image editing operation detection methods cannot detect manipulations not present in the training stage, and ii) there is room for developing end-to-end forensics methods that learn the high-level and the low-level manipulation-related features for a more precise pixel-wise localization of the forgeries.

Based on these motivations, this thesis focuses on the following points:

(i) Developing lighting environment-based forensics methods that can estimate the lighting environments from the 2D images more precisely without requiring any 3D surface information, and hence can detect forgeries more accurately.

(ii) Developing illumination colour-based methods to detect splicing forgeries in images under single or multiple illumination sources and also can learn optimal illumination colour-related features automatically from training data, therefore removing the need to compute hand-crafted features.

(iii) Developing DL-based forensics methods for detecting various types of image editing operations carried out on images, that may not be known during the training stage.

(iv) Developing DL-based methods localizing various types of forgeries involving arbitrary image regions by learning more optimal high-level and low-level manipulation-related features.

1.5 Research Contributions and Thesis Organization

The thesis has four major research contributions. They are presented in the following chapters.

- **Chapter 2: Estimation of Lighting Environment for Exposing Image Splicing Forgeries**

  In this chapter, a forensics method is proposed for detecting splicing forgeries involving human faces in the front pose, *e.g.,* those in formal group portraits. The method is based on checking the inconsistencies in the lighting environments (LEs) estimated from the faces present in the image under investigation. Firstly, a low-dimensional lighting model is created from a set of front pose face images of a single individual under different directional lighting environments. For this, the set of images is decomposed using the principal component analysis. This low-dimensional model, which captures the lighting

variation in faces, is then used to estimate the LE from a given near-front pose face image. In a spliced image, the LE estimated from the spliced face will be different from that estimated from the original faces. Therefore, finding the inconsistencies among the LEs estimated from different faces could reveal the splicing forgery. The experimental results on Yale Face Database B and a set of authentic and forged real-life images show the efficacy of the proposed method.

- **Chapter 3: Exposing Splicing Forgeries in Digital Images through the Discrepancies in Dichromatic Plane Histograms**

  This chapter proposes a forensics method to detect spliced human faces of any pose utilizing the source illumination colour as a cue. The method is based on extracting an illumination-signature from the faces of the persons present in an image using the dichromatic reflection model. The dichromatic plane histogram (DPH), which is computed by applying the 3D Hough Transform on the face images, is used as the illumination-signature. It is assumed that the skin colours of different persons' faces are the same. The correlation measure is employed to compute the similarity between the DPHs obtained from different faces present in an image. Finally, a simple threshold on this similarity measure exposes splicing forgeries in an image. Experimental results on two standard splicing datasets, DSO-1 and DSI-1, show the efficacy of the proposed method.

- **Chapter 4: Deep Learning-based Classification of Illumination Maps for Detecting Spliced Faces**

  This chapter proposes a novel image forensics method that can detect splicing forgeries group portraits involving faces of any pose and skin colour. The method converts an input image to an illumination map (IM), and the facial regions of the IM are compared in a pair-wise manner using machine learning techniques to check the presence of splicing forgery. A siamese convolutional neural network (CNN) is first trained on an external training set to differentiate between face-IM pairs coming from similar and different illumination environments (IEs). Once trained, the CNN part of the siamese network is

used as a feature extractor for each face present in a test image. The pair-wise features are concatenated and classified using a support vector machine classifier for forgery detection. The advantage of the proposed method is its ability to learn features capable of differentiating faces coming from different IEs. The experimental results on two public datasets, DSO-1 and DSI-1, show the efficacy of the proposed method with respect to the state-of-the-art.

- **Chapter 5: A Siamese Convolutional Neural Network-based Approach towards Universal Image Forensics**

  This chapter proposes a novel deep learning-based method that can detect different types of image editing operations carried out on images. Unlike most of the existing methods, which can only detect the editing operations considered in the training stage, the proposed method can generalize to manipulations not seen in the training stage. The method is based on the classification of image pairs as either similarly or differently processed using a deep siamese neural network. Once the network learns features that can discriminate different editing operations, it can check whether an image is processed with an editing operation, not present in the training stage, using the one-shot classification strategy. An image forgery detection and localization technique is also proposed using the trained siamese network. The experimental results on multiple datasets show the efficacy of the proposed method in detecting different editing operations and also show the ability in detecting and localizing image forgeries.

- **Chapter 6: Two-stream Encoder-Decoder Network for Localizing Image Forgeries**

  In this chapter, a novel two-stream encoder-decoder network is proposed, which utilizes both the high-level and the low-level image features for precisely localizing forged regions in a manipulated image. This is motivated by the fact that the forgery creation process generally introduces both the low-level and the high-level artefacts to the forged images. In the proposed two-stream network, one stream learns the low-level manipulation-related features in the encoder side by extracting noise residuals through a set of high-pass

filters in the first layer of the encoder network. In the second stream, the encoder learns the high-level image manipulation features from the input image RGB values. The coarse feature maps of both the encoders are upsampled by their corresponding decoder network to produce dense feature maps. The dense feature maps of the two streams are concatenated and fed to a final convolutional layer with sigmoidal activation to produce the pixel-wise prediction. We have carried out experimental analyses on multiple standard forensics datasets to evaluate the performance of the proposed method. The experimental results show the efficacy of the proposed method with respect to the state-of-the-art.

The thesis is concluded in Chapter 7 with a summary of the research and an outline of the possible future research.

2

Estimation of Lighting Environment for Exposing Image Splicing Forgeries

A common type of image forgery is *splicing*. In this forgery, a composite image is created by copying objects from multiple images. Splicing forgeries containing human faces are of greater concern, as their impact on society may be more serious. Therefore, image forensics to detect spliced human faces is an important research issue.

Among the different approaches available in the literature to detect splicing forgeries discussed in the earlier chapter, the *lighting environment (LE)-based* forensics methods are more applicable to real-life images like highly compressed and low-resolution images. The human visual system is not very good at judging the inconsistencies in the LEs in images [55], [56], and it is very hard to match the illumination conditions of the spliced and the authentic parts of a composite image [9], [54]. In addition to that, there are several anti-forensics methods proposed to counter different types of forensics methods, such as the compression-based and the camera-based forensics methods [57], [58]. To the best of our knowledge, no anti-forensics method has been proposed to counter the LE-based forensics techniques. Based on these motivations, this chapter proposes a novel LE-based forensics technique for detecting spliced images involving human faces.

The rest of the chapter is organized as follows. Section 2.1 describes the related work and the motivation. Section 2.2 explains the low-dimensional lighting model. Section 2.3 presents the proposed LE estimation and splicing detection methods. Section 2.4 presents the experimental results for the lighting environment estimation and the forgery detection methods. Section 2.5 discusses the effectiveness of the proposed method with respect to the state-of-the-art. Finally, Section 2.6 presents a summary of the chapter.

2.1 Related Work and Research Gap

In lighting environment-based image forensics, the mismatch in the LE [22] of one part of the image with the rest is exploited to check its authenticity. The assumption here is that the LEs estimated from different parts of an authentic image are similar. In a composite image, there may be some spliced parts that were captured under different LEs.

Johnson and Farid [27] proposed the first LE-based forensics method, which compares the

light directions estimated from different parts of an image. Assuming the surfaces to be Lambertian and illuminated by a point light source, the authors could estimate the 2D lighting directions from the pixel intensity and occluding contour normals. Riess *et al.* [59] extended the method by estimating the lighting directions from multiple coloured surfaces, resulting in improved accuracy and broader applicability of the method. These two methods, however, work only in images with a single dominant light source, and can estimate 2D lighting directions only and hence have the 3D ambiguity.

To estimate an arbitrarily complex LE, Johnson and Farid [22] proposed to use spherical harmonics (SH) analysis [35] and represented the LE and the surface reflectance function in terms of the SH coefficients. The SHs form an orthonormal basis for functions defined on the surface of a sphere. They are analogous to the Fourier series for functions defined on lines or circles. Let $F(\alpha,\beta)$ denote a function on the unit sphere, where α and β are the spherical angular coordinate. In the SH domain, the function can be expressed as

$$F(\alpha,\beta) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} F_{l,m} Y_{l,m}(\alpha,\beta) \tag{2.1}$$

where $Y_{l,m}$ is the mth SH of order l, $F_{l,m}$ is the corresponding SH coefficient.

Johnson and Farid [22] applied the SH analysis and showed that, under certain assumptions, the LE could be estimated using a low-dimensional model, *i.e.,* using only the SH coefficients up to order 2. The authors made the following assumptions: 1) linear camera response function, and 2) convex and Lambertian object with constant surface reflectance. This method could also estimate the 2D LE only because the 3D surface normals of objects are not readily available in 2D images. Kee and Farid [7] proposed the first 3D LE-based forensics method, which was aimed at exposing face splicing forgeries. They created a 3D morphable face model from a set of frontal and profile-view face images and fitted this model to each face to obtain the 3D surface normals. These 3D normals were used to estimate the 3D LE in terms of the SH coefficients. Fan *et al.* [60] extended the work of Kee and Farid [7] by estimating the 3D LEs from arbitrary objects, utilizing a shape-from-shading method [61] to obtain the 3D surface normals. Peng *et al.* [8] proposed a method to estimate the 3D LE more accurately by relaxing some less realistic

assumptions about human faces.

Although the above-mentioned 3D SH methods are good at estimating the LEs from faces, their estimation accuracy depends heavily on the accuracy of the 3D face model. In addition, these methods are difficult to implement as there are multiple modules in the algorithms [62], *i.e.,* 3D face model fitting, face texture (albedo) estimation, and the SH coefficients estimation, *etc*. An error in any of these modules may lead to the incorrect estimation of the LE. For example, the construction of a 3D face model requires a face that is lit from the front and with a normal facial expression. These conditions are not always satisfied in forensics applications. Therefore, there is a need to develop forensics methods that can check the inconsistencies in the LEs estimated from test faces without requiring any prior knowledge about their 3D shapes.

2.1.1 Low-dimensional lighting subspace

Epstein *et al.* [63] and Hallinan [64] empirically showed that the set of images of an object in a fixed-pose viewed under different point sources lies on a low-dimensional subspace. The low-dimensional subspace is spanned by the first few eigenvectors (*principal components*), computed from the set of images of the object using principal component analysis (PCA). In [63] and [64], the authors experimented with human faces and other objects and reported that the first $5-6$ eigenvectors are in general sufficient to capture 90-98% of the variation in the sets of images. Therefore, they concluded that the set of images of a Lambertian object captured under different LEs lies on a low-dimensional subspace. These results are confirmed by other researchers also [65], [66].

The initial theoretical works explaining this low-dimensional subspace were proposed by Shashua [67] and Murase and Nayar [68]. They showed that in the absence of shadows, a 3D subspace is sufficient to describe the set of images of a Lambertian object under distant illumination. However, the absence of shadow is not a very practical assumption as attached shadows are always present in a real-life scene under complex illumination. Therefore, these methods are too simple to explain the empirical low-dimensional subspace. Basri and Jacob [69] and Ramamoorthi and Hanrahan [70] independently derived an analytical formula for the irradiance of a Lambertian convex object in the SH domain, considering the attached shadows explicitly.

They showed that the irradiance is the convolution of incident illumination with the Lambertian reflectance function. If the illumination and the reflectance functions are represented in SH domain, the convolution becomes multiplication of the SH coefficients of both the functions. More importantly, they proved that the Lambertian reflection acts as a low-pass filter, and the first 9 SH coefficients are sufficient to capture 99% of the irradiance. However, the connection between the low-dimensional SH subspace and the empirical eigen subspace is not obvious. Later, Ramamoorthi [71] provided a theoretical connection between the SH subspace and the eigen subspace through the analytic PCA construction and hence proved that the first $5 - 6$ eigenvectors are sufficient to capture 98% of the lighting variations in the face. The proposed LE estimation method utilizes this concept to create the lighting model comprising the first few eigenvectors, and it is later used to estimate the LE from any test face.

This chapter proposes a novel LE-based image forensics method that can expose splicing forgeries present in images of front pose human faces. The method detects the spliced faces through the inconsistencies in the LEs estimated from the facial regions of the individuals present in the image under investigation. For this, a novel LE estimation method is proposed, which can estimate the LE from any test face without requiring to create a 3D model for that face. This is an important advantage of the proposed method over the state-of-the-art, as the existing methods required to create a specific 3D face model for each individual for the accurate estimation of the LE.

The main contributions of this chapter are as follows: 1) It proposes a novel method to estimate the LEs from human faces without requiring to create the 3D face models. The proposed method can estimate the LEs more accurately than the state-of-the-art with the advantage of being simple. 2) Based on the LE estimation method, a forensics technique is proposed, which can expose splicing forgeries present in images involving human faces in the front pose. The proposed method is appropriate for detecting splicing forgeries in real-life forged images involving any individual, as it does not need to create any 3D face model for LE estimation.

2.2 Lighting Model

In this section, we describe the low-dimensional lighting model for human face images. This is used by the proposed method for estimating the LE from human faces present in an image under investigation. The inconsistencies among the LEs estimated from different face images can indicate the presence of splicing forgeries in the image.

In a real-life scene, lighting can be very complex. For example, there may be multiple light sources located at different locations in the scene. Epstein *et al.* [63] and Hallinan [64] empirically showed that the set of images of a (nearly) Lambertian object (*e.g.*, human faces) in a fixed pose viewed under different LEs lies on a low-dimensional subspace. This subspace is spanned by the first few eigenvectors of the set of face images.

An arbitrary light source can be expressed as a summation of point light sources at infinity [63], [64]. Therefore, an image of a Lambertian object under an arbitrary LE can be written as a linear combination of the images of the object due to different point sources at infinity [72]. Let $\mathbf{B}_{\alpha,\beta} = \mathbf{B}_{\alpha,\beta}(\theta,\phi)$ be an image, called the *boundary* image, captured under the point light source located at infinity, where (α,β) and (θ,ϕ) are the spherical angular coordinates of the light source and surface normals respectively. If the irradiance of the scene is bounded by a fixed value, the set P of all possible images will be compact and convex [72], [64]. Now, a finite set of boundary images $\{\mathbf{B}_{\alpha_i,\beta_j} : i \leq M_s, j \leq N_s\}$ can be created by uniformly sampling the set P, where M_s and N_s are the numbers of the longitudes and the latitudes corresponding to the finite set of boundary images. An image $\mathbf{Z}$ of the object under an arbitrary LE can be synthesized by adding the boundary images corresponding to each point light source [64]

$$\mathbf{Z} = \sum_{i=1}^{M_s} \sum_{j=1}^{N_s} w(\alpha_i,\beta_j)\mathbf{B}_{\alpha_i,\beta_j} \tag{2.2}$$

where $w(\alpha_i,\beta_j)$ is the weighting factor for each light source.

An orthogonal basis $\{\mathbf{u}_k\}$, $k = 1, 2, ..., K$ for the finite set of boundary images $\{\mathbf{B}_{\alpha_i,\beta_j} : i \leq M_s, j \leq N_s\}$ can be found by applying PCA on the set. These bases are the first K dominant eigenvectors, and can be used to approximate $\mathbf{B}_{\alpha_i,\beta_j}$ as

$$\mathbf{B}_{\alpha_i,\beta_j} \approx \sum_{k=1}^{K} \gamma_{i,j,k}\mathbf{u}_k \tag{2.3}$$

where $\gamma_{i,j,k}$ is the weighting factor for each eigenvector $\mathbf{u}_k$. Therefore, using Equations (2.2) and (2.3), $\mathbf{Z}$ can be approximated as

$$\mathbf{Z} \approx \sum_{k} b_k\mathbf{u}_k \tag{2.4}$$

where

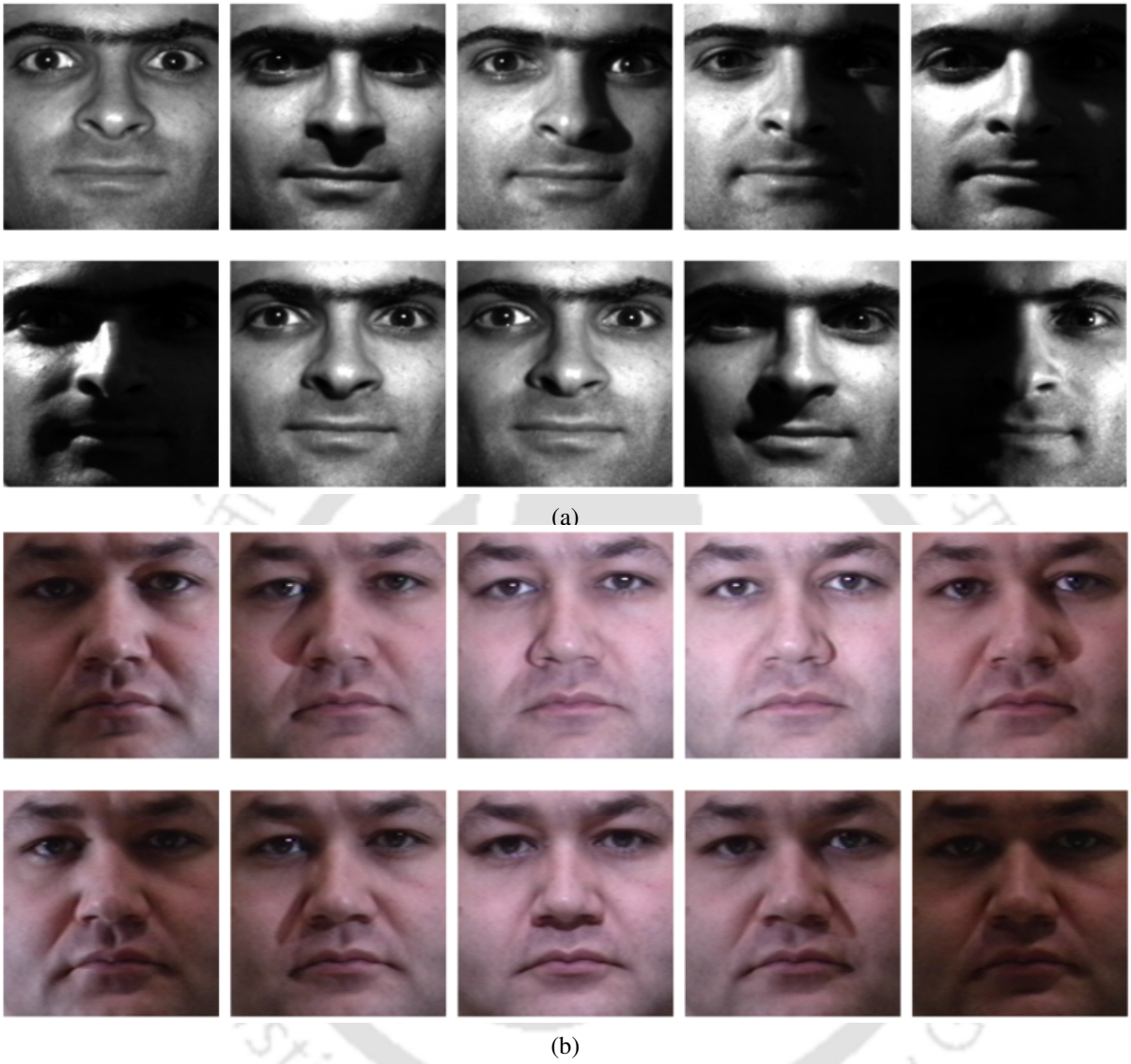
$$b_k = \sum_{i=1}^{M}\sum_{j=1}^{N} w(\alpha_i,\beta_j)\gamma_{i,j,k} \tag{2.5}$$

These basis vectors, being the eigenvectors of the covariance matrix, point to the directions of variation in the set of boundary images. As the boundary images are of the same object captured under different LEs, the dominant variations in these images are because of lighting only. Therefore, these basis vectors capture the lighting variations in the set of boundary images. The first few eigenvectors form a low-dimensional lighting model. Figure 2.2a shows the first six eigenfaces computed from a set of front pose face images of a male individual. These eigenfaces can be interpreted as faces lit from front, side, top/bottom, extreme side, corner, and extreme corner [64]. This lighting model is used in the later sections to estimate the LEs from any test face image.

2.2.1 Theoretical Analysis of the Low-dimensional Lighting Model

As already defined, $\mathbf{B}_{\alpha,\beta}(\theta,\phi)$ denotes the intensity of a single pixel in the boundary image $\mathbf{B}_{\alpha,\beta}$. Assume that the surface in the image is Lambertian and convex, the point light source is at infinity, the camera response is linear, and the surface albedo is uniform. Under these assumptions, $\mathbf{B}_{\alpha,\beta}(\theta,\phi)$ becomes proportional to the irradiance due to a point light source at (α,β).

A low-dimensional lighting model can be constructed by applying the PCA on the set of boundary images. First, an observation matrix $\mathbf{Q}$, which contains all the observations, is created by uniformly sampling the light source positions (α_p,β_p) and the surface normal coordinates (θ_i,ϕ_i) of the boundary image $\mathbf{B}_{\alpha,\beta}$. The matrix $\mathbf{Q}$ has the form



(a)

(b)

Figure 2.1: Examples of images of a single subject under different light sources from (a) Extended Yale B database [4] and (b) Multi-PIE [5] dataset.

$$Q_{ip} = \mathbf{B}_{\alpha_p,\beta_p}(\theta_i, \phi_i) \tag{2.6}$$

where i indexes the pixels in the boundary image, and p indexes the light source positions. To find the principal components or *eigenimages*, we have to compute the eigensystem of the covariance matrix $\mathbf{T} = (\mathbf{Q} - \mu\mathbf{1})(\mathbf{Q} - \mu\mathbf{1})^T$, where $\mathbf{1}$ is the all-ones matrix and μ is the mean irradiance obtained by averaging over the pixels and all the light sources. Each element of $\mathbf{T}$ is

computed as

$$T_{ij} = \sum_{p} (\mathbf{B}_{\alpha_p,\beta_p}(\theta_i, \phi_i) - \mu)(\mathbf{B}_{\alpha_p,\beta_p}(\theta_j, \phi_j) - \mu) \tag{2.7}$$

Note that μ is same for all the pixels. This is because of the assumption that the surface is convex and the light sources are uniformly sampled over the entire sphere.

The SH domain representation of $\mathbf{B}_{\alpha,\beta}(\theta, \phi)$ is given by [69], [70]

$$\mathbf{B}_{\alpha,\beta}(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \hat{A}_l L_{l,m}(\alpha, \beta) Y_{l,m}(\theta, \phi) \tag{2.8}$$

where $Y_{l,m}$ is the mth SH of order l, $L_{l,m}$ is the corresponding SH coefficient of the LE and $\hat{A}_l$ is the normalised lth order SH coefficient of the reflectance function. In [69] and [70], the authors showed that $\hat{A}_l$ decays rapidly for Lambertian surfaces, and 99% energy of the Lambertian reflection is captured by the first 9 SH coefficients of the LE up to $l = 2$.

Plugging Equation (2.8) in Equation (2.7), we get

$$T_{ij} = ((\hat{A})_0^2 - 4\pi\nu\mu^2) Y_{0,0}(\theta_i, \phi_i) Y_{0,0}(\theta_j, \phi_j) + \sum_{l=0}^{\infty} \sum_{m=-l}^{l} (\hat{A}_l)^2 Y_{lm}(\theta_i, \phi_i) Y_{lm}(\theta_j, \phi_j) \tag{2.9}$$

where ν is the number of images taken with different light source positions. Equation (2.9) is the analytic form of the elements of the covariance matrix $\mathbf{T}$. The PCA is used for finding the principal components, which are the eigenvectors of $\mathbf{T}$ computed by solving the following eigensystem:

$$\mathbf{Tu} = \lambda \mathbf{u} \tag{2.10}$$

where λ is the eigenvalue. Representing the eigenvector in terms of SHs and substituting the value of T_{ij} from Equation (2.9), we get

$$\sum_{p,q} M_{l,m:p,q} c_{p,q} = \lambda c_{l,m} \tag{2.11}$$

where $c_{l,m}$ is the coefficient corresponding to mth SH of order l of the eigen vector $\mathbf{u}$, and

$$M_{l,m:p,q} = \sum_{j} Y_{l,m}(\theta_j, \phi_j) Y_{p,q}(\theta_j, \phi_j) \tag{2.12}$$

Assuming that the light source samples are infinitely dense, the summation operation in

Equation (2.12) becomes integration over the angular coordinates (θ, ϕ) given by

$$M_{l,m:p,q} = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} Y_{l,m}(\theta,\phi) Y_{p,q}(\theta,\phi) \sin\theta \, d\theta \, d\phi \tag{2.13}$$

Here, the term $M_{l,m:p,q}$ captures the orthogonality between various SHs. When the image pixels correspond uniformly to the entire sphere of surface normals, the SHs will be orthogonal to one another. Thus, the eigenvectors will simply be the SHs themselves, and the first 9 eigenvectors will capture 99% of irradiance. However, in the case of a single image, where only the front-facing surface normals are visible, the pixels will be distributed over the upper hemisphere only. While the orthonormality of SHs guarantees that none of their linear combination can have norm 0, Ramamoorthy [71] showed that the norm of certain linear combinations come very close to 0 when the domain of integration is restricted to the upper hemisphere. Therefore, the eigenvectors will be the linear combinations of SHs with the same value of m, *i.e.,* $m = 0$, $m = 1$ and $m = -1$. SHs corresponding to $m = \pm 2$, *i.e.,* Y_4 and Y_8, are not affected by this rearrangement and become eigenvectors alone. Because of this intermingling of SHs, fewer eigenvectors will now capture most of the irradiance. The first six eigenvectors now capture about 98% of the irradiance, and thereby proving the empirical low-dimensional model proposed by Hallinan [64].

2.2.2 Computation of Low-dimensional Lighting Model

To create the lighting model, a set of front pose face images of a single individual captured under different point light sources is collected. The face parts are cropped from these images and geometrically aligned so that all the images have identical eye locations. Then, the PCA is applied to this set to get the principal components. The principal components are the eigenvectors of the covariance matrix of the set of face images represented as vectors. As reported in [64] and [63], we have also observed that the eigenvectors corresponding to the first few significant eigenvalues are sufficient to capture the lighting variations in the image set. Hence, these eigenvectors can be used as a lighting model to estimate the LE from a test face image.

Let $\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_Z$ be Z face images in the image set, each of size $M \times N$. Each face $\mathbf{F}_i$ is rearranged as a vector $\mathbf{I}_i$ of dimension MN. The PCA is applied on this set to find the orthonor-

mal vectors $\mathbf{u}_k$, $k = 1, 2, ..., MN$, along which the face vectors are varying. The eigenvalue λ_k associated with the eigenvector $\mathbf{u}_k$ represents the amount of variations in the set of $\mathbf{I}_i$s along the eigenvector $\mathbf{u}_k$. The covariance matrix $\mathbf{C}$ of the set of face vectors is given by

$$\mathbf{C} = \frac{1}{Z}\sum_{i=1}^{Z} \mathbf{J}_i \mathbf{J}i^T \tag{2.14}$$

where $\mathbf{J}_i = \mathbf{I}_i - \mathbf{\Gamma}$ is the mean subtracted face vector and $\mathbf{\Gamma} = \frac{1}{Z}\sum_{i=1}^{Z} \mathbf{I}_i$ is the mean face vector and the symbol T represents transpose operation. The eigenvectors and eigenvalues are computed by decomposing the covariance matrix $\mathbf{C}$ as

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T \tag{2.15}$$

where $\mathbf{U} = [\mathbf{u}_1 \mathbf{u}_2 \ ... \ \mathbf{u}_{MN}]$ is a matrix with each column representing an eigenvector, and $\mathbf{\Sigma} = diag(\lambda_1, \lambda_2, ..., \lambda_{MN})$ is a diagonal matrix containing the eigenvalues in its leading diagonal positions. The eigenvalues in $\mathbf{\Sigma}$ are arranged in a descending order, *i.e.,* $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_{MN}$, and the eigenvectors in $\mathbf{U}$ are arranged in the decreasing order of their corresponding eigenvalues. The first eigenvector is the most significant as it captures the largest variation in the dataset. If L number of eigenfaces are sufficient to capture most of the lighting variations, the matrix

$$\mathbf{W}_L = [\mathbf{u}_1 \mathbf{u}_2 \ ... \ \mathbf{u}_L] \tag{2.16}$$

can be used as the low-dimensional lighting model.

2.3 Proposed method

This section explains the proposed LE estimation and the splicing detection methods.

2.3.1 Estimation of Lighting Environment

We propose to estimate the LEs from near front pose face images using the low-dimensional lighting model computed from a set of front pose face images. The intuition behind this is that the first few eigenfaces point to the directions of changes in the LEs of the faces in the image set, and hence projecting a front pose face of any individual on the low-dimensional subspace results in the LE only.

The dominant L eigenfaces span an L-dimensional subspace of the original MN-dimensional image space. Using these L eigenfaces, we create the low-dimensional lighting model W_L as shown in Equation (2.16). Given a test face image $\mathbf{F}$, it is converted to grayscale and resized to a dimension of $M \times N$. The image is then rearranged as an MN-dimensional vector, $\mathbf{I}$. This face vector $\mathbf{I}$ is then projected onto the L-dimensional subspace as

$$\mathbf{\Omega} = \mathbf{W}_L^T \mathbf{I} \tag{2.17}$$

where $\mathbf{\Omega}^T = [\omega_1 \omega_2 \ ... \ \omega_L]$ is the lighting coefficient (LC) vector, ω_k representing the contribution of $\mathbf{u}_k$ to the LE in the input face image. Thus, $\mathbf{\Omega}$ gives the estimate of the LE in the test face image. For example, if we project a face image with frontal LE, the first eigenface (shown in Figure 2.2a) will have the highest correlation, and hence the LC corresponding to this eigenface will be the largest. Algorithm 2.1 shows the steps for computing the lighting model, and Algorithm 2.2 shows the steps involved in the proposed LE estimation method.

Algorithm 2.1

%Lighting Model Computation%

Input: M, N and L; A set of face images, $\{\mathbf{F}_i : i = 1, 2, ..., Z\}$ of dimension $M \times N$, of a single person under different LEs

Output: Low-dimensional lighting model $\mathbf{W}_L$

Steps:

(i) For each $i \epsilon \{1, 2, ..., Z\}$, convert $\mathbf{F}_i$ to gray-scale and then rearrange it as the vector, $\mathbf{I}_i$, of dimension MN.

(ii) Compute the covariance matrix $\mathbf{C}$ of the set of face vectors using Equation (2.14)

(iii) Compute the matrix $\mathbf{U}$ of MN eigenfaces using Equation (2.15) and construct the L-dimensional lighting model $\mathbf{W}_L = [\mathbf{u}_1 \mathbf{u}_2 \ ... \ \mathbf{u}_L]$ using the L dominant eigenfaces.

Algorithm 2.2

%LE Estimation%

Input: M, N; A face image $\mathbf{F}$ of dimension $M \times N$

Output: LC vector $\mathbf{\Omega}$

Steps:

(i) Get the low-dimensional lighting model $\mathbf{W}_L$ from Algorithm 2.1.

(ii) Convert $\mathbf{F}$ to a gray-scale image of dimension $M \times N$ and rearrange as a MN-dimensional vector $\mathbf{I}$.

(iii) Project $\mathbf{I}$ onto the L-dimensional subspace using Equation (2.17) to get the LC vector $\mathbf{\Omega}$.

The LC vector $\mathbf{\Omega}$ is used for detecting the splicing forgeries present in images containing human faces as described below.

2.3.2 Splicing Detection

We propose a forgery detection method that can expose splicing forgeries in images containing at least two near-front pose human faces. The method is based on the following assumptions: 1) in an authentic image, the LEs are similar at different parts of the image, and 2) in a spliced image, there will be objects from images captured under different LEs. Therefore, the comparison of the LEs estimated from different parts of an image gives a clue about the authenticity of the image.

In the proposed method, first, the faces are manually extracted from the image under investigation. Although various automated face detection methods are available in the literature, we prefer manual face cropping. This is because the automated face detection methods sometimes may give false positive or false negative results, in which case the proposed method will produce inaccurate results. Suppose there are D faces, $\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_D$, present in the image, and their corresponding LC vectors are $\mathbf{\Omega}_1, \mathbf{\Omega}_2, ..., \mathbf{\Omega}_D$. The proposed method detects forgeries by comparing the LEs estimated from the faces present in the image in a pair-wise manner. Since

there are D faces, there will be $\frac{D(D-1)}{2}$ comparisons. The distance between any two LC vectors $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_j$ is calculated using a correlation-based distance measure [22] given by

$$d(\mathbf{\Omega}_i, \mathbf{\Omega}_j) = \frac{1}{2}\left(1 - \text{corr}\left(\mathbf{\Omega}_i, \mathbf{\Omega}_j\right)\right) \tag{2.18}$$

where

$$\text{corr}\left(\mathbf{\Omega}_i, \mathbf{\Omega}_j\right) = \frac{\mathbf{\Omega}_i^T \mathbf{\Omega}_j}{\|\mathbf{\Omega}_i\|_1 \left\|\mathbf{\Omega}_j\right\|_1} \tag{2.19}$$

represents the normalized correlation measure between $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_j$, and $\|.\|_1$ represents the L_1 norm of a vector. This distance measure is invariant to multiplicative factors in the coefficient vectors and produces a value in the range $[0, 1]$.

If the two faces $\mathbf{F}_i$ and $\mathbf{F}_j$ come from two different LEs, the distance d between $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_j$ will be large. Otherwise, it will be small. In an authentic image, all the $\mathbf{\Omega}_i$s will point almost to the same direction as all the faces come from the same LE. So, the distance d between all the pairs $\{\mathbf{\Omega}_i, \mathbf{\Omega}_j; i, j \leq D\}$ will be small. In a spliced image, there will be at least one pair $\{\mathbf{\Omega}_i, \mathbf{\Omega}_j\}$ for which the distance will be higher as there will be at least one face coming from a different image (*i.e.*, a different LE). Therefore, the image will be considered spliced if the maximum distance among all the pairs is more than a predefined threshold. Thus, if

$$\max_{(i,j)}(d(\mathbf{\Omega}_i, \mathbf{\Omega}_j)) > r_{th} \tag{2.20}$$

the image is considered to be spliced. The discrimination threshold r_{th} is set through experimentation as explained in a later section.

The steps of the proposed method is outlined in Algorithm 2.3.

Algorithm 2.3

Input: Image under investigation, r_{th}

Output: Decision about the authenticity of the test image.

Steps:

(i) Extract the faces $\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_D$ from the test image. Note the number of faces D.

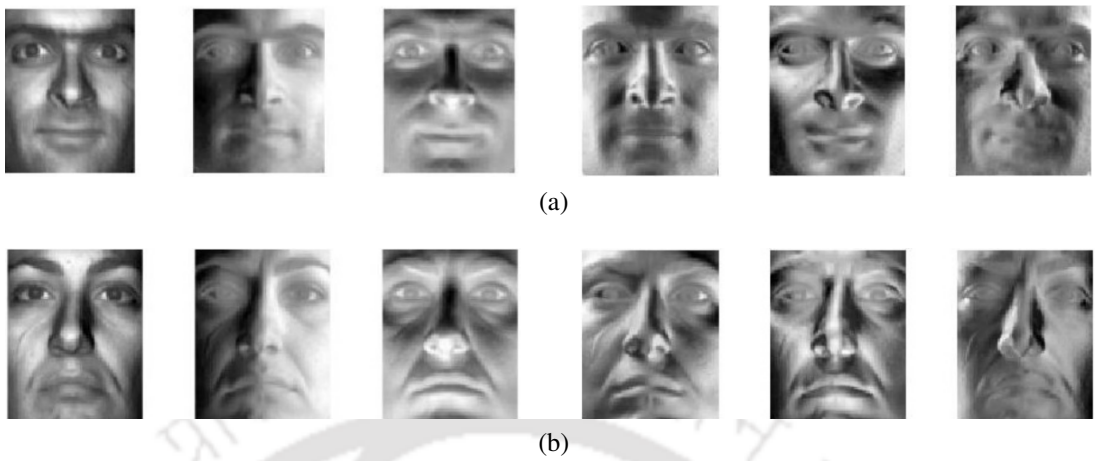


Figure 2.2: First six eigenfaces computed from the set of images of (a) a male individual and (b) a female individual.

(ii) Get $\mathbf{W}_L$ from Algorithm 2.1.

(iii) For each $\mathbf{F}_i$, estimate the LC vector $\mathbf{\Omega}_i$ using Algorithm 2.2.

(iv) Compute the pair-wise distance between the $\mathbf{\Omega}_i$s using Equation (2.18). For D number of faces present in the image there will be $\frac{D(D-1)}{2}$ distances.

(v) Decide the image as spliced if the inequality in Equation (2.20) is satisfied. Otherwise, decide the image as authentic.

2.4 Experimental Results and Analysis

A number of experiments were carried out to validate the efficacy of the proposed method. Subsection 2.4.1 describes the experimental results on the lighting model computation. Subsection 2.4.2 discusses the results on LE estimation. Subsection 2.4.3 presents the experimental results on the classification of consistent and inconsistent LEs. The performance of the proposed method in discriminating the consistent and the inconsistent LEs estimated from near-frontal face images is explained in Subsection 2.4.4. Subsection 2.4.5 presents the experimental results on real-life splicing forgery detection.

2.4.1 Lighting Model Computation

To compute the lighting model, we have used the images in the Extended Yale Face Database B [4], [73]. The database contains 16, 128 single light source images of 28 different subjects,

each seen under 9 poses and 64 lighting conditions. We have used the front pose images only to compute the lighting model. Figure 2.1(a) shows 10 images of a single subject in the dataset under different directional light sources.

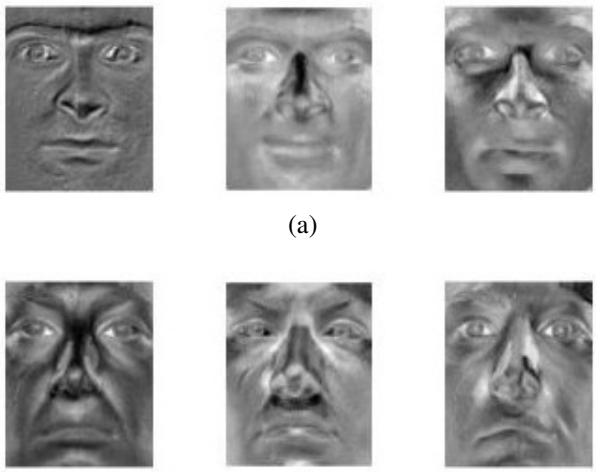
We have computed the lighting model from the face images of all the individuals separately. Figure 2.2 shows the first six eigenfaces computed from the image sets of two individuals. Except for the changes in the sign of the eigenfaces and the change in their order, these six eigenfaces are similar across individuals. These eigenfaces can be interpreted as faces lit from front, side, top/bottom, extreme side, corner, and extreme corner (not necessarily in this order), as pointed out in [64]. On the other hand, the next eigenfaces are not similar for different individuals, as can be seen in Figure 2.3. This is because the higher-order eigenfaces capture the variations due to the change in the face geometry or noise in the training images.

To see the number of eigenfaces required to represent the lighting variations in the image set of the subject shown in Figure 2.1(a), we have carried out the *variance accounted for (VAF)* analysis [64]. The VAF is the sum of the eigenvalues corresponding to the eigenvectors considered in the analysis. The VAF represents the percentage of lighting variance on the set of face images explained by the first k dominant eigenvectors corresponding to the largest k eigenvalues, and is given by the following equation:

$$VAF(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k) = \frac{\lambda_1 + \lambda_2 + ... + \lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_{MN}} \tag{2.21}$$

Figure 2.4 shows the VAF plot against the number of the eigenvectors. The plot clearly shows that the first six eigenfaces capture around 96% of the lighting variations in the image set, which is in accordance with the analytical results provided by Ramamoorthy [71]. Because of these reasons, we use the first six eigenfaces (*i.e.*, $L = 6$) to create the low-dimensional lighting model.

We have performed the following experiment for visual verification of the 6-dimensional lighting model. We have reconstructed the face vector $\mathbf{I}_R$ for different values of L using $\mathbf{I}_R = \mathbf{W}_L\mathbf{\Omega}$. Figure 2.5 shows the reconstruction of the LE from a face image using different values of L. In this case, the test image and the training images are of the same individual. As shown in



(a)

(b)

Figure 2.3: The 7th, 9th and 10th eigenfaces next to (a) eigenfaces shown in Figure 2(a), and (b) eigenfaces shown in Figure 2(b).

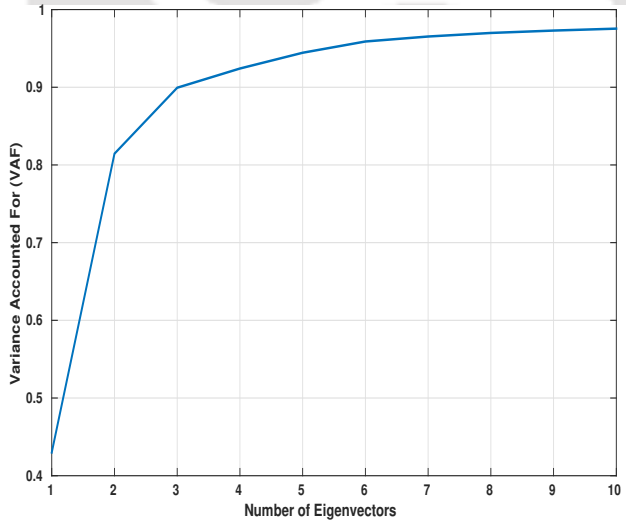
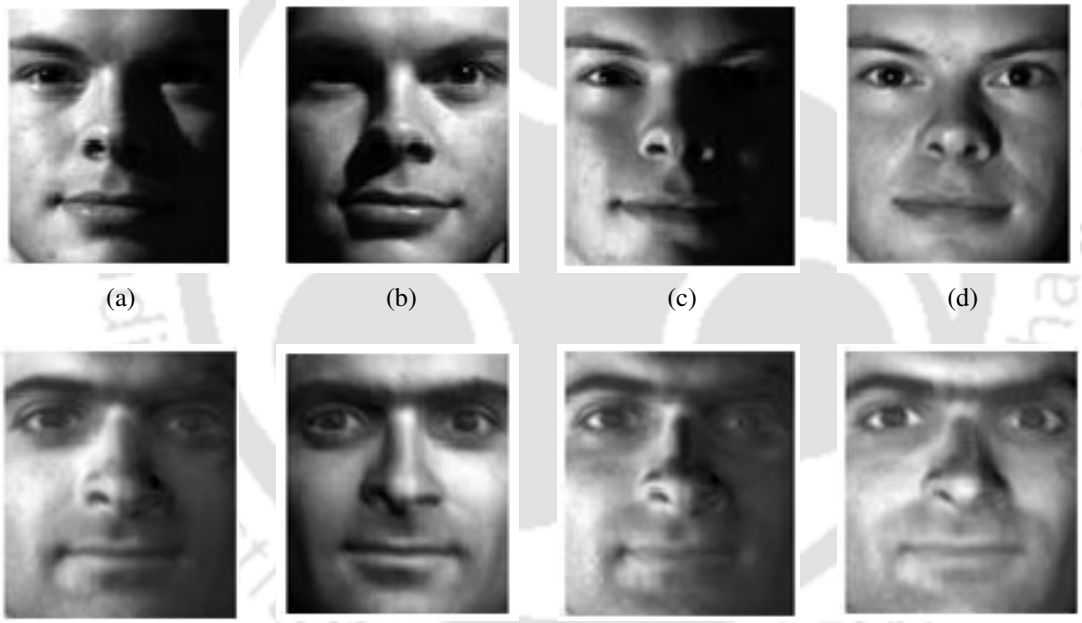


Figure 2.4: VAF analysis of the eigenvectors computed from the image set of the subject shown in Figure 2.1(a).

the figure, the first three eigenfaces cannot fully reconstruct the LE (Figure 2.5b), whereas the first six eigenfaces capture the LE very well (Figure 2.5c). Figure 2.5d shows the reconstructed face using the first ten eigenfaces. It can be observed from the figure that the faces reconstructed by the first six and the first ten eigenfaces are similar. On the other hand, the faces reconstructed by the first three and the first six eigenfaces are significantly different. This confirms visually that the first six eigenfaces are sufficient to represent the lighting variations in the set of images used to compute the lighting model.

(a) (b) (c) (d)

Figure 2.5: Reconstruction of LE of a face image using different numbers of eigenfaces. (a) is the original image, and (b), (c) and (d) are reconstructed using 3, 6 and 10 eigenfaces respectively.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.6: Reconstruction of the LEs estimated from face images of an individual different from the one used in the computation of the lighting model. (a)-(d) are the original images, and (e)-(h) are the corresponding reconstructed images.

2.4.2 Lighting Environment Estimation

To see the efficacy of the proposed method in estimating the LE from face images, we have applied the method on two datasets: (1) Yale Face Database B [4], and (2) Multi-PIE dataset [5]. Figure 2.1 shows 10 examples of front pose face cropped images from both datasets. Yale B dataset contains images of 10 individuals under 9 different poses and 64 different directional lighting conditions. The images are captured under a single strong light source without any

(a) (b) (c) (d)

Figure 2.7: Test images used for demonstrating the contribution of eigenfaces in capturing the LEs.

ambient light. We have used the front pose images of all the 10 individuals. Thus, we have a total of $10 \times 64 = 640$ images. Multi-PIE dataset [5] includes images of 330 individuals under 19 different lighting conditions and 15 different face poses. The images in this dataset are captured under single light sources and also have an ambient light source. For our experiments, we have used 30 individuals. We convert the images to gray-scale, crop the face parts (as shown in Figure 2.1) and then resize them to 100×80 pixels. The six eigenfaces of an arbitrarily selected subject, shown in Figure 2.2a, are used to create the low-dimensional lighting model, and the LE from each face is estimated using Algorithm 2.2.

To visualize the performance of the proposed LE estimation method, we have reconstructed the LEs, estimated from the front pose face images of different individuals of Yale B dataset, using the six eigenfaces shown in Figure 2.2a. Figure 2.6 shows the reconstruction of the LEs estimated from four face images of an individual other than the one used for computing the lighting model. The first row shows four images of an individual under different LEs, and the second row shows the reconstruction of the LEs of the face images of the individual, projected on the low-dimensional model created from the face images of a different individual. It is seen that the proposed method can estimate the LEs from the faces very well. From the figure, it is clear that the lighting model captures only the LE information and not other features such as shadows, face geometry, and noise.

We have considered four images of different individuals from Yale B dataset captured under different LEs and estimated the LC vectors from each of them to show the contribution of

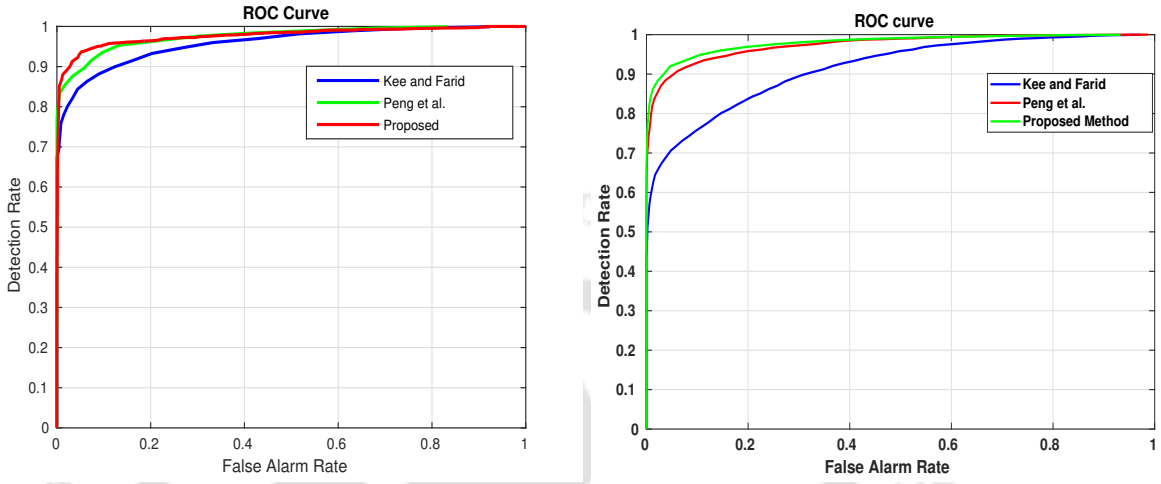
Table 2.1: Contributions of eigenfaces to capture the LEs for the images shown in Figure 2.7.

	1	**2**	**3**	**4**	**5**	**6**
(a)	0.34	**0.38**	0.04	0.21	0.02	0.03
(b)	**0.67**	0.1	0.09	0.13	0.01	0.02
(c)	0.15	0.29	0.05	**0.34**	0.16	0.18
(d)	0.16	0.22	0.09	0.22	0.05	**0.25**

different eigenfaces in capturing the LEs. Figure 2.7 shows these images. Table 2.1 shows the LCs representing the contribution of each of the eigenfaces in the estimation of the LEs for these images. Each column of the table shows the contribution of each of the 6 eigenfaces to the estimation of the LEs of the images shown in Figure 2.7. The image in Figure 2.7a is captured under a side lighting condition, and the second eigenface, which captures the side lighting condition, has the highest coefficient value. Likewise, the faces in Figure 2.7b and Figure 2.7c are captured under front lighting and extreme-side lighting conditions, respectively, and it can be seen that the eigenfaces corresponding to these two LEs have the highest LCs. The face in Figure 2.7d is captured under an extreme corner-lighting condition, which is well represented by the sixth eigenface as indicated by the large value of the sixth coefficient in the table. From this analysis, it can be argued that different directional lighting conditions are captured by the different eigenfaces, and hence any complex lighting condition will be captured by the linear combination of the six eigenfaces.

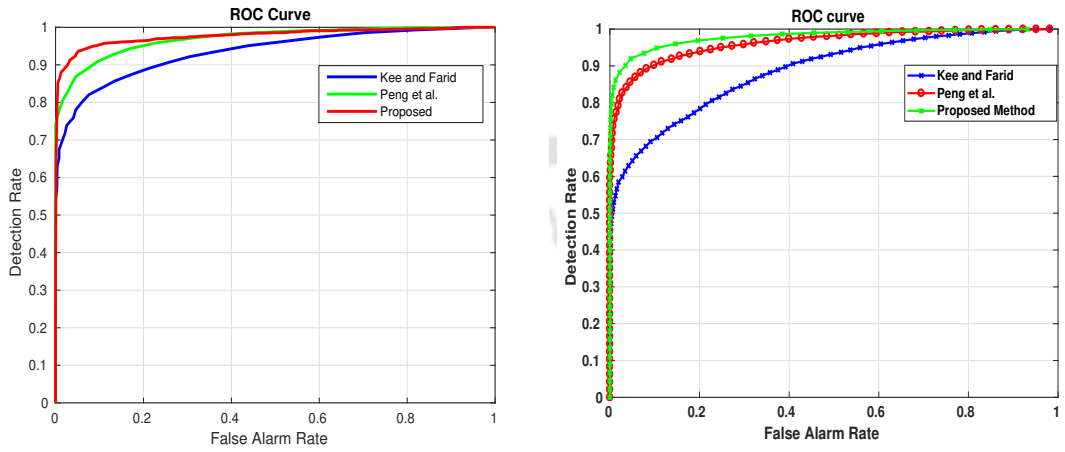
2.4.3 Classification of consistent and inconsistent LEs

Experiments were carried out to see the effectiveness of the proposed method in discriminating consistent and inconsistent LEs. We have randomly sampled 10000 pairs of images with different LEs and 10000 pairs with the same LEs for each of Yale B and Multi-PIE datasets, as in [8]. The pairs from different lighting conditions are considered *inconsistent* and the pairs from the same lighting condition are considered *consistent*. For each pair of faces, we have calculated the distance between the LC vectors, estimated from the two images, using Equation (2.18). We have computed the receiver operating characteristic (ROC) curve to show the discrimination ability of the proposed LE estimation method. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds,



(a) (b)

Figure 2.8: ROC curves for different methods showing the ability to discriminate the consistent and the inconsistent LEs on (a) Yale B dataset and (b) Multi-PIE dataset, when using the specific 3D model for each individual in Kee and Farid's and Peng *et al.*'s methods.



(a) (b)

Figure 2.9: ROC curves for different methods showing the ability to discriminate the consistent and the inconsistent LEs on (a) Yale B dataset and (b) Multi-PIE dataset, when using a generic 3D face model for all the individuals in Kee and Farid's and Peng *et al.*'s methods.

Table 2.2: Comparison of the discriminative power of the proposed method with the state-of-the-art methods on Yale B and Multi-PIE datasets, when the specific 3D face model is used for each individual in Kee and Farid's and Peng *et al.*'s methods.

	Yale B		Multi-PIE	
	AUC (%)	DR(%)@10%FAR	AUC (%)	DR(%)@10%FAR
Kee and Farid	96.1	89.4	92.3	76.2
Peng *et al.*	97.7	93.9	97.4	93.3
Proposed	**98.1**	**97.4**	**97.7**	**94.5**

Table 2.3: Comparison of the discriminative power of the proposed method with the state-of-the-art methods on Yale B and Multi-PIE datasets, when a generic 3D face model is used for all the individuals in Kee and Farid's and Peng *et al.*'s methods.

	Yale B		Multi-PIE	
	AUC (%)	DR(%)@10%FAR	AUC (%)	DR(%)@10%FAR
Kee and Farid	93.2	82.9	90.8	70.0
Peng *et al.*	96.4	90.9	95.8	90.5
Proposed	**98.1**	**97.4**	**97.7**	**94.5**

separating the consistent and the inconsistent pairs. We consider the inconsistent case as the positive class and the consistent case as the negative class.

We have compared our method with two existing methods, namely Kee and Farid's [7] and Peng *et al.*'s [8] methods. These methods are also specifically designed to detect composite images containing human faces. The ROC curves of the three methods on Yale B and Multi-PIE datasets are shown in Figure 2.8a and 2.8b, respectively. In the case of Kee and Farid's and Peng *et al.*'s methods, specific 3D models are used for each of the 10 subjects. The area under the curve (AUC) values computed from the ROC curves and the detection rate at 10% false alarm rate (DR(%)@10%FAR) are shown in Table 2. On Yale B dataset, Kee and Farid's method achieves an AUC of 96.1% and a DR of 89.4%, and Peng *et al.*'s method achieves an AUC of 97.7% and a DR of 93.9%. The proposed method achieves an AUC of 98.1% and a DR of 97.4% on Yale B dataset. On Multi-PIE dataset, Kee and Farid's method achieves an AUC of 92.3% and a DR of 76.2%, and Peng *et al.*'s method achieves an AUC of 97.4% and a DR of 93.3%. The proposed method achieves an AUC of 97.7% and a DR of 94.5% on Multi-PIE dataset. This implies that the proposed method can discriminate the consistent and inconsistent LEs well and perform better than the state-of-the-arts.

Another set of experiments is performed to see the effect of using a single 3D face model for all subjects in the discriminative power of the methods by Kee and Farid and Peng *et al.* This is important as in real forensics scenarios, it may be difficult to create specific 3D face models for each individual present in an image. We have applied these two methods on Yale B and Multi-PIE datasets using a single generic 3D face model for all individuals. The ROC curves for three methods on Yale B and Multi-PIE datasets are shown in Figure 2.9a and Figure 2.9b respectively. The AUC values and DRs at 10% FAR are listed in Table 3. As can be seen from Table 2.2 and Table 2.3, the AUC value achieved by Kee and Farid's method drops from 96.1% to 93.2%, and the DR drops from 89.4% to 82.9% on Yale B dataset, when a generic 3D face model instead of a specific model for each individual. On Multi-PIE dataset, the AUC value of Kee and Farid's method drops from 92.3% to 90.8%, and the DR drops from 76.2% to 70.0% when a single 3D model is used instead of specific 3D models. The AUC value achieved by Peng *et al.*'s method drops from 97.7% to 96.4%, and DR drops from 93.9% to 90.9% on Yale B dataset. On Multi-PIE dataset, AUC value achieved by Peng *et al.*'s method drops from 97.4% to 95.8%, and DR drops from 93.3% to 90.5% when a generic 3D face model is used. The drop in the accuracy of Kee and Farid's and Peng *et al.*'s methods in the case of a single generic 3D face model is expected as the computed 3D normals are not accurate in this case. On the othe hand, the proposed method's AUC and DR values are same as the earlier case, *i.e.,* AUC of 98.1%, DR of 97.4% and AUC of 97.7%, DR of 94.5% on Yale B and Multi-PIE datasets, respectively. This is because the proposed method does not depend on any specific face models. Therefore, in real-life forensics scenarios, the proposed method is more reliable than the state-of-the-art methods.

2.4.4 Performance on non-frontal face images

A set of experiments is carried out to see the performance of the proposed method in discriminating the consistent and the inconsistent LEs when faces are not in the front pose. In the first experiment, we have estimated the LEs from the images of all the 10 individuals of Yale Face Database B with a near-frontal pose as shown in Figure 2.10a. We use the lighting model, created using the front pose face images (*i.e.,* the six eigenfaces shown in Figure 2.2a),

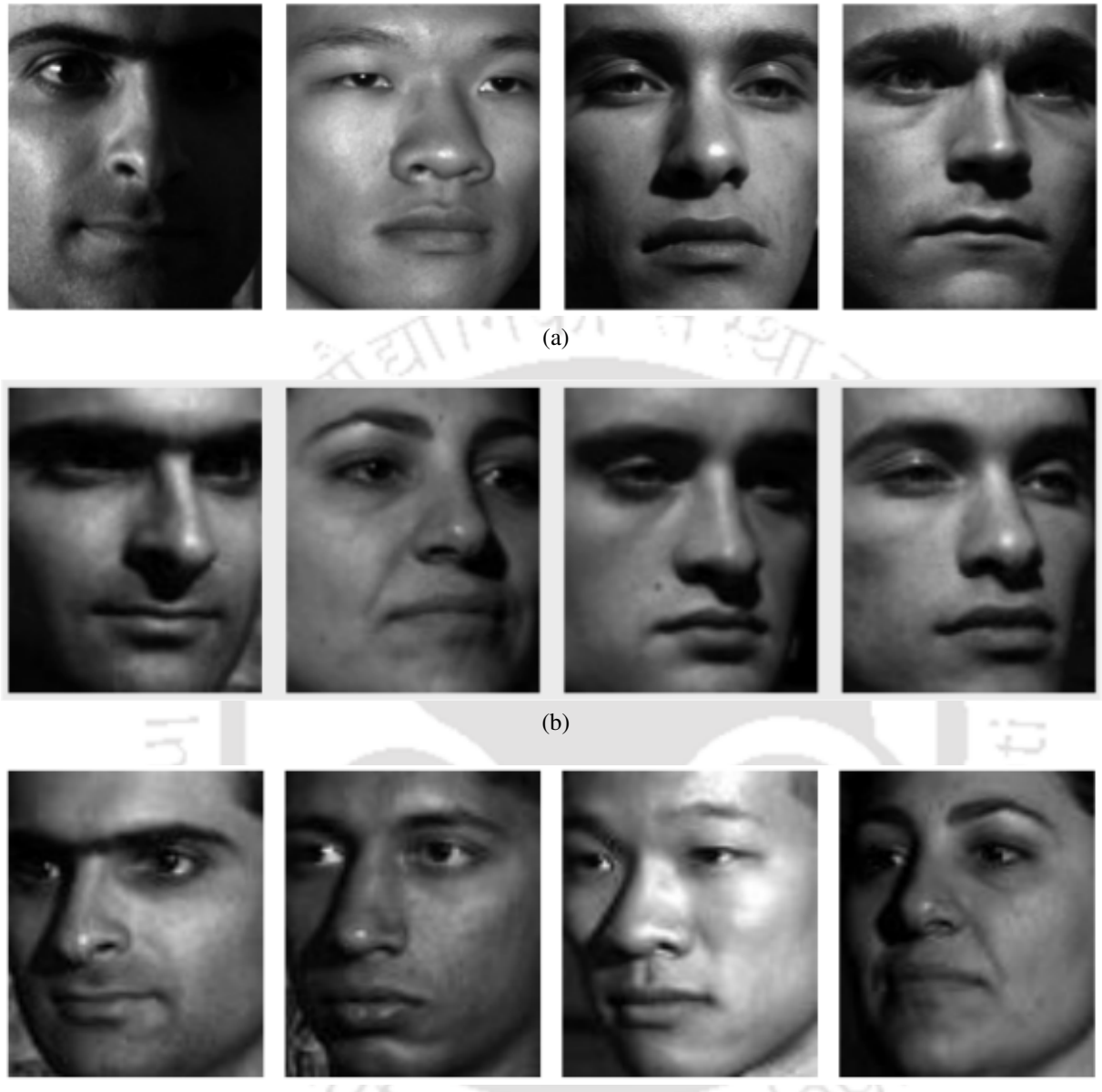
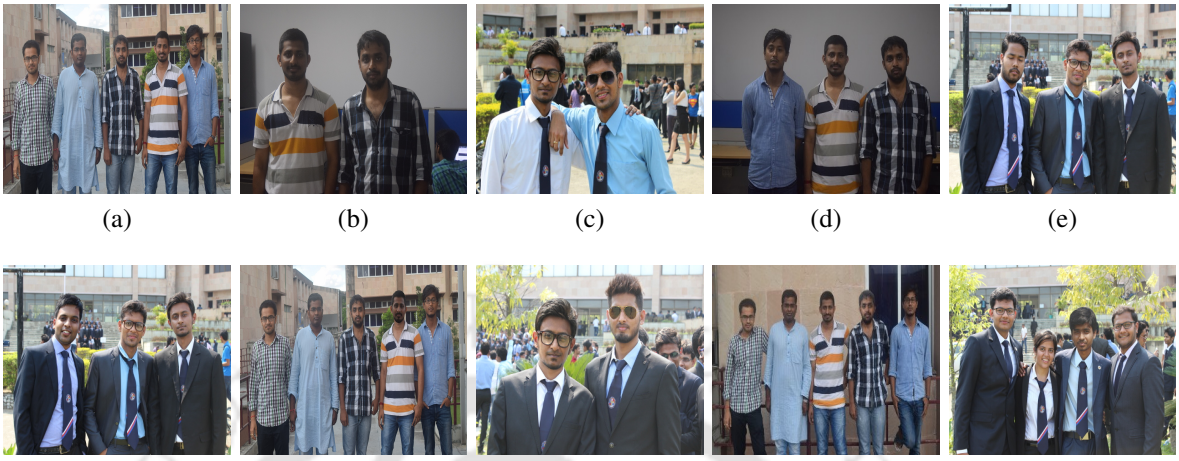


Figure 2.10: Examples of (a) near front pose faces and (b) non-frontal faces used to test the performance of the proposed method in detecting LEs from faces with poses different from the frontal pose.

to estimate the LEs from the nearly fronta face images. The ROC is computed using the pair-wise distance of LEs estimated from 10000 spliced face pairs and 10000 authentic face pairs, as already explained. From the ROC, the AUC value is found to be 96.6%, which is not very different from the AUC value calculated from frontal pose faces. This experiment suggests that the proposed method can accurately estimate the LE even in the case of nearly pose images. However, it is observed that the performance of the proposed method drops as the face poses



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 2.11: In the figure: (a)-(e) are five authentic images, and (f)-(j) are five spliced images created using the authentic images.

deviates from the front pose. For example, the method achieves an AUC of 91.16% when applied to face images with moderate side-view pose as shown in Figure 2.10b. When applied to a more extreme side-view pose, as shown in Figure 2.10c, the AUC drops to 68.81%. This is expected as the proposed method is based on the assumption that the major variations in the face images are because of lighting differences only. In the case of a test face with a pose significantly different from the front pose, the variation due to facial geometry is more than that due to lighting difference. Therefore, projecting the non-frontal test face onto the low-dimensional subspace, constructed from front pose faces, will not give the accurate LE.

2.4.5 Performance in Splicing Detection

We have performed a set of experiments to see the performance of the proposed method in detecting realistic forged images. We have created a dataset that contains 15 authentic images and 15 spliced images. Figure 2.11 shows 10 images from the dataset. The images contain more than two persons with near frontal pose faces. The spliced images are created by copying the head of at least one person from an image and pasting it onto another image.

Table 2.4: Forgery detection accuracy on the images shown in Figure 2.11.

Method	Kee and Farid	Peng et al.	Proposed
TNR(%)	0.8	0.93	**1**
FNR(%)	0.4	0.2	**0.07**

Figure 2.12: An example of a famous forged image, where former US president Bill Clinton (left) is seen shaking hands with Dimitry de Angelis, a conman from Australia.

Table 2.5: Pair-wise distances between the faces present in the spliced image shown in Figure 2.13a.

	ID_1	ID_2	ID_3
ID_1	0	**0.19**	0.18
ID_2	**0.19**	0	0.07
ID_3	0.18	0.07	0

To check the authenticity of an image, the distance $d(\mathbf{\Omega}_i, \mathbf{\Omega}_j)$ between the lighting coefficient vectors $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_j$ for each pair of faces is calculated and compared with a threshold, r_{th}, as explained in Section 2.3.2. In this experiment, we have used a single generic face model for Kee and Farid's, and Peng *et al.*'s methods. We use the threshold which corresponds to 10% FAR in the ROC curves, shown in Figure 2.8b, as r_{th} used in Equation (2.20). In the ROC curves, the thresholds corresponding to 10% FAR are 0.16, 0.11 and 0.15 for Kee and Farid's, Peng *et al.*'s and proposed methods, respectively. Table 3 shows the true negative rate (TNR), which is the probability of classifying the authentic image as authentic, and false negative rate (FNR), which is the probability of classifying the spliced as authentic for the three methods. The TNR of the proposed method is 100% *i.e.,* all the authentic images are correctly classified as authentic and the FNR is 7%, *i.e.,* the proposed method misclassified one spliced image as authentic. On the other hand, Kee and Farid's method classifies 12 out of 15 authentic images correctly (*i.e.,* TNR of 80%) and 9 out of 15 (*i.e.,* FNR of 40%) spliced images correctly. Peng *et al.*'s method classifies 14 out of 15 authentic images correctly (*i.e.,* TNR of 93%), and 12 out of 15 spliced images correctly (*i.e.,* FNR of 20%). Hence, in case of real forged images also, our method outperforms the state-of-the-art methods.

(a) (b)

Figure 2.13: (a) A forged image depicting former Brazil president Luiz Inacio Lula da Silva (center) with a gang leader Rosemary de Noronha (left). (b) The authentic image from which the forged image (a) was created.

Table 2.6: Pair-wise distances between the faces present in the authentic image shown in Figure 2.13b.

	ID_1	ID_2	ID_3	ID_4
ID_1	0	0.05	0.07	0.04
ID_2	0.05	0	**0.10**	0.06
ID_3	0.07	**0.10**	0	0.09
ID_4	0.04	0.06	0.09	0

We have analysed two famous splicing forgeries downloaded from the Internet [1]. The first forged image involves former US president Bill Clinton and Dimitry de Angelis, a conman from Australia, and is shown in Figure 2.12. To check the authenticity of the image, we have computed the lighting coefficient vectors from both faces and then calculated the distance d between them using Equation (2.18). The distance d is found to be 0.17, which is bigger than the threshold $r_{th} = 0.15$ computed from the ROC curve. Therefore, the proposed method correctly classifies the image to be forged. The second forged image we analyze is shown in Figure 2.13(a) depicting former Brazil president Luiz Inacio Lula da Silva (center) with Rosemary de Noronha (left), an undercover gang leader. However, Noronha was not present in the authentic image, shown in Figure 2.13(b). To check the authenticity of both the images, we have analysed them using Algorithm 2.3. The pair-wise distances between the LEs estimated from the faces present in the spliced and authentic images are listed in Table 2.5 and Table 2.6, respectively. It is clear from Table 2.5 that the pair-wise distance between the lighting coefficient vectors estimated from the fake face (*i.e.*, ID_1 in Figure 2.13(a)) and the other two faces are greater

[1] These images are also analyzed in [41]

than r_{th}, and the distance between the authentic faces (*i.e.*, ID_2 and ID_3) are very small. The maximum distance d among the lighting coefficient vectors estimated from the faces present in the forged image (Figure 2.13(a)) is 0.19, which is more than the threshold value r_{th}. Therefore, the forged image is correctly classified as forged. On the other hand, all the pair-wise distances in the authentic image (Figure 2.13(b)) are below the threshold r_{th}, as can be seen in Table 2.6. The maximum distance d for the authentic image is found to be 0.10, which is less than r_{th}. Hence, the authentic image is also correctly classified as authentic by the proposed method.

2.5 Discussions

From the experimental results, it is clear that the proposed method can estimate the LE more accurately than the state-of-the-art methods [7], [8]. To estimate the LE accurately from a test face image, the state-of-the-art methods need a specific 3D face model for that face. In forensics scenarios, however, it may be difficult to create a specific 3D face model for each individual. This is because the existing 3D face modeling software (*e.g.*, FaceGen used by Peng *et al.*) needs a well-lit and expressionless front pose (sometimes both the front and the side-view) face image to create an accurate 3D face model of each subject. Most of the time, these conditions are hardly met, and a single generic 3D face model has to be used to fit each subject present in an image under investigation. When a single generic 3D face model is used for all the individuals, the estimation accuracies of these methods drop significantly. The proposed method, on the other hand, does not need any specific face model and can estimate the LE from any face using a single lighting model. Another advantage of the proposed method is that it is simpler and hence easier to implement than the method [8].

The obvious limitation of the proposed method is that it is applicable only to images containing near front pose human faces. Nevertheless, it can find application in many scenarios where the frontal views of the faces are available, *e.g.*, in formal group photos.

2.6 Summary

This chapter proposed a new LE-based forensics method that can detect splicing forgeries in images involving human faces in the front pose. The proposed method checks the inconsistencies in the LEs, estimated from different faces present in the image under investigation.

The LEs are estimated by projecting the front pose face images onto a low-dimensional lighting model, computed from a set of front pose face images of a single individual through the PCA. While the state-of-the-art methods need to create a specific 3D face model to estimate the LE from a test face image accurately, the proposed method can estimate the LE from any test face using a single lighting model. The experimental results on Yale Face Database B, Multi-PIE, and our own database show the efficacy of the proposed method with respect to the state-of-the-art. The limitation of the proposed method is that it can detect splicing forgeries only in images containing near-front pose human faces. When the faces deviate from the front pose, the LE estimation of the proposed method gives inaccurate results.

3

Exposing Splicing Forgeries in Digital Images through the Discrepancies in Dichromatic Plane Histograms

The last chapter proposed an LE-based forensics method for detecting spliced faces present in images. The method extracted the LEs from the face regions of the persons present in the image using a subspace-based LE estimation method. The method is applicable only to human portraits involving near front-pose faces. This chapter proposes a forensics method to detect spliced faces of any pose utilizing the source *illumination colour* as a cue.

The knowledge of source illumination colour is very useful in many computer vision tasks. For instance, in human-computer interaction [74], the ability to remove the effect of illumination colour from the input images and videos, known as *colour constancy*, is desirable for better performance. This is because the colours of object surfaces change as the illumination colour changes, which affects the computer vision systems that rely on the object colour information. Most of the computational colour constancy methods [75], [36], [37] achieve the colour constancy by first estimating the illumination colours from the input images and then normalizing them using the illumination colours to produce the canonical images under a white light source [76]. In image forensics, the illumination colour has proven to be an effective cue for detecting splicing forgeries [29], [9]. The current and the next chapters will discuss more about the use of illumination colour for exposing splicing forgeries.

Similar to LE-based forensics methods, the illumination colour-based methods are considered to be effective since it is not easy to match the exact illumination colour in a composite image [9], [54]. As in the case of LE-based methods, there is no anti-forensics method available to counter the illumination colour-based forensics techniques. These observations motivate us to propose an illumination colour-based forensics method for detecting splicing forgeries in human group portraits.

The proposed method extracts a novel *illumination-signature* from the face region of each person present in an image. To be effective in forensics, this illumination-signature should be similar for the faces coming from the same illumination environment and different for the faces coming from different illumination environments. This chapter proposes to use the *dichromatic plane histogram (DPH)* [77] as the illumination-signature for detecting the face splicing forgeries. It is computed from the facial region of each person present in the image by applying the

3D Hough transform. The *dichromatic reflection model (DRM)* [78] is exploited for computing this histogram. Assuming the skin material of the facial region to be the same for all persons, the DPHs are expected to be similar for faces coming from the same illumination colour. On the other hand, for faces coming from different illumination environments, the DPHs will show inconsistency.

The rest of the chapter is organized as follows. Section 3.1 provides an overview of illumination colour-based forensics methods. Section 3.2 gives a detailed background on the DRM and the DPH. Section 3.3 presents the proposed method, and Section 3.4 discusses the experimental results on splicing detection. Finally, Section 3.5 presents a summary of the chapter.

3.1 An Overview of Illumination Colour-based Image Forensics

In illumination colour-based image forensics, the source illumination colours extracted from different parts of an image are utilized for detecting splicing forgeries. The motivations for using the illumination colour as a cue for detecting splicing are as follows. In an authentic image, all the different parts are lit by the same illumination sources. A spliced image may include parts copied from images captured under different illumination sources. Therefore, comparing the illumination colours estimated from different parts of an image could reveal the splicing forgery. Here, the key assumption is that the spliced and the authentic parts of a forged image may look visually similar, but the illumination colour extracted from them will differ.

Gholap and Bora [29] introduced the use of illumination colour as a cue for detecting splicing forgeries. This method estimates the illumination colour from different parts of an image using the DRM [78]. This model is elaborated in Subsection 3.2.1. For estimating the illumination colours from the image, the method requires specular regions to be manually extracted from the image. If there is more than one illumination colours present in the image, the method decides it as a spliced image. The limitation of this method are the following: 1) it fails in images that are captured under multiple illumination sources, as it assumes the presence of a single illumination source, and 2) it requires the presence of specular highlights in the images, which are manually selected.

Francis *et al.* [40] proposed a method, where the illuminat colour is estimated from the nose-tip of each person present in the image using the DRM. The illuminant colours, extracted from different persons present in an image, are compared with each other to judge the authenticity of the image. This method has limitations similar to those of Gholap and Bora's method. More specifically, it also assumes the presence of a single illumination and requires manual selection of the nose-tip from the faces. Wu and Fang [39] proposed another method where an image is divided into different non-overlapping blocks. Then, the illuminant colour is estimated from each block using the *generalized grey-edge (GGE)* method [37]. Assuming one block as the reference block, the angular error between the illuminant colours estimated from each block and the reference block is computed. If the angular error is more than a pre-defined threshold, the image is decided as spliced. Since the method requires the manual selection of a reference block, the result changes when a different reference block is selected.

Riess and Angelopoulou [79] proposed to create a new image, called the *illuminant map (IM)*, using the illumination colours estimated from the input image. First, the input image is segmented into homogenous regions, called *superpixels*, using the graph-based segmentation method [80]. Then, the illumination colour from each superpixel is estimated using a modification of the *inverse-intensity chromaticity (IIC)* method [38], and the superpixel is recoloured using the estimated illumination colour. The intuition behind this method is that the parts in the IM of an authentic image will have similar colour features, as all the parts of an authentic image are captured under the same illumination sources. The spliced regions in the IM of a spliced image will have colour features different from those in the authentic regions. In this method, as the forgery is detected manually by observing the IM, it might introduce human errors.

Carvalho *et al.* [9] proposed a method for automatic detection of face splicing forgeries by classifying the face regions of the IM (face-IM) using a machine learning-based classifier. The authors created two IMs from each image using two different illumination estimation methods, namely the IIC [38] and the GGE [37] methods. The authors observed that the face-IMs computed from an authentic image have similar visual features. On the other hand, in a spliced image, the spliced faces have visual features different from those of the authentic faces. Based

on this observation, the authors proposed to extract texture [81] and gradient-based [82] descriptors from the face regions of the IM. The IM is converted to YCbCr colour space, and the Y channel is utilized for computing both the descriptors. Then, the features are classified in a pair-wise manner using a support vector machine (SVM) classifier. More specifically, for each face-IM pair, computed using the same illumination estimation method, the same type of features extracted from the two face-IMs are concatenated and classified using the SVM. In their follow-up work, Carvalho *et al.* [41] proposed to extract three types of features from the face-IMs, namely texture, shape and colour features. More specifically, the authors proposed to compute three texture descriptors in [81], [83], [84], two shape descriptors in [85], [86], and four colour descriptors in [87], [88], [89], [90]. Also, in this work, Carvalho *et al.* proposed to convert the IM to three different colour spaces, namely Lab, HSV, and normalized RGB colour spaces. Similar to [9], here also, the features are classified in a pair-wise manner by concatenating similar features of the two face-IMs computed from the same type IM converted to the same colour space. The *k*-nearest neighbour classifier is utilized for classifying the features. Although these methods are very effective, their performances drop in low-resolution and highly compressed images. This is because in the case of low-resolution and highly compressed images, the IM computation becomes less accurate and hence the features computed from them become less discriminative.

3.2 Background

This section presents a background on the DRM, which is utilized to compute the proposed illumination-signature, DPH.

3.2.1 Dichromatic Reflection Model (DRM)

In computer vision and computer graphics, substances are divided into two general categories on the basis of optical properties: a) *homogeneous* and b) *non-homogeneous*. Non-homogeneous substances are composed of vehicle particles at the surface layer and colourant particles in the layer below this [91]. Typical examples of non-homogeneous substances are human skin, most paints, paper, plastic, *etc*. For these substances, light interacts with both

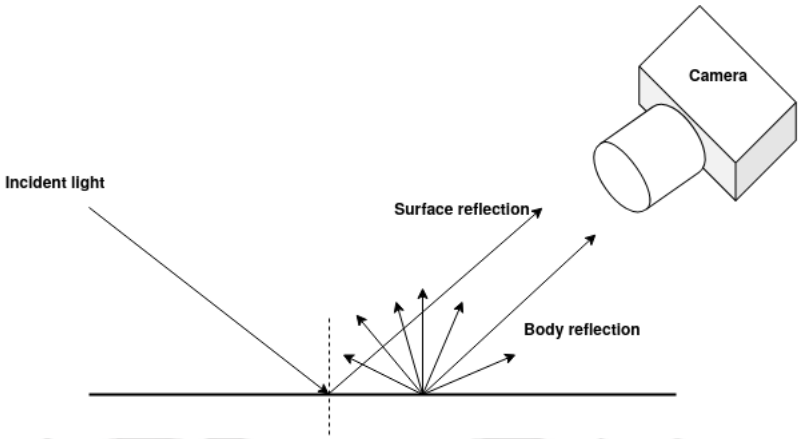


Figure 3.1: Surface and body reflections from a non-homogeneous surface according to the DRM.

the surface and the body. Shafer [78] proposed the DRM to explain the reflections from non-homogeneous substances. Homogeneous substances, *e.g.*, metals and many crystals, do not exhibit the two types of reflections as exhibited by the non-homogeneous surfaces.

According to the DRM [78], the total radiance of the light reflected from a non-homogeneous material is the sum of radiances of the light reflected from the *surface* and the *body*. Figure 3.1 shows these two reflections. The *surface* or *specular* reflection is the mirror-like reflection at the surface of the object. The *body* or *diffused* reflection occurs when the incident light penetrates through the surface and suffers scattering by the colourant particles present beneath the surface. The scattered light finally re-emitted through the surface. Thus, the total radiance $L(\theta, \lambda)$ is given by

$$L(\theta, \lambda) = m_i(\theta)C_i(\lambda) + m_b(\theta)C_b(\lambda) \tag{3.1}$$

where θ is the angle between the incident light and the viewing directions; λ is the wavelength of light; C_i and C_b are the spectral power distributions, and m_i and m_b are the geometric factors of surface and body reflections, respectively. The two vectors C_i and C_b span a two-dimensional space, known as the *dichromatic plane*. In RGB colour space, Equation (3.1) can be expressed as

$$\begin{bmatrix} f_R(\mathbf{x}) \\ f_G(\mathbf{x}) \\ f_B(\mathbf{x}) \end{bmatrix} = m_i \begin{bmatrix} f_R(\mathbf{x}) \\ f_G(\mathbf{x}) \\ f_B(\mathbf{x}) \end{bmatrix}_i + m_b \begin{bmatrix} f_R(\mathbf{x}) \\ f_G(\mathbf{x}) \\ f_B(\mathbf{x}) \end{bmatrix}_b \tag{3.2}$$

where $f_R(\mathbf{x})$, $f_G(\mathbf{x})$, and $f_B(\mathbf{x})$ are the sensor responses for the red, green and blue colour channels respectively at the pixel location $\mathbf{x}$, and the subscript i and b represent the interface and body reflection components respectively. According to *neutral interface reflection* [92], the spectral power density of the interface reflection is the same as that of the illuminant source. Thus, the RGB values of an object lie on the dichromatic plane defined by the illuminant colour and the object colour, as expressed by Equation (3.2). Under uniform illumination, the dichromatic planes of two differently coloured surfaces in the same scene intersect at the illumination colour. This is because the illumination colour is common for both the dichromatic planes estimated from the two surfaces. Therefore, the intersection of different dichromatic planes gives the estimate of the illuminant colour. However, in real-life noisy images, this method fails to give the true illuminant colour. This is because the noise may cause the pixels, belonging to a single dichromatic plane, to lie on multiple dichromatic planes. Hence, the intersection of these planes may not give the true illuminant colour.

3.2.2 Dichromatic Plane Histogram (DPH)

In [77], the authors proposed to use the 2D Hough transform [93] to find the dichromatic planes in the RGB colour space. According to the DRM, all the dichromatic planes pass through the origin. Therefore, in the RGB space, the equation of the dichromatic plane is given as

$$f_R(\mathbf{x})\sin(\theta)\cos(\phi) + f_G(\mathbf{x})\sin(\theta)\sin(\phi) + f_B(\mathbf{x})\cos(\theta) = 0 \tag{3.3}$$

where θ and ϕ are respectively the polar and azimuth angles of the plane in a spherical coordinate system. All the pixels satisfying Equation (3.3) for a specific pair of (θ, ϕ) lie on the same dichromatic plane defined by the pair (θ, ϕ). A DPH, $H_d(\theta, \phi)$, represents the distribution of pixel values lying on different dichromatic planes. $H_d(\theta, \phi)$ is created by applying a 2D Hough transform, where each bin represents the number of pixels belonging to a dichromatic plane specified by the pair (θ, ϕ). Therefore, $H_d(\theta, \phi)$ gives the likelihood of the presence of different dichromatic planes in an image corresponding to different pairs of angles (θ, ϕ). In this chapter, the DPH is used as the illumination-signature, as shown in the following section.

3.3 Proposed Method

Illumination colour-based image forensics methods work under the assumption that, in a forged image, the spliced and the original parts are captured under different illuminant colours. Therefore, it is a common practice in most of the illumination based image forensics methods [29], [39], [40], to use the illumination colour as the illumination signature, and directly compare these signatures extracted from different parts (image patches) of an image to check its authenticity. However, in real forensic scenarios, it does not produce a very convincing result. The reasons are as follows:

- The forgery detection method relies solely on the accuracy of the illuminant estimation method. But, due to the presence of noise in real images and the small size of the image patch, the illuminant estimation methods hardly give the true illuminant colour.
- The majority of the illuminant estimation methods assume the presence of a single illuminant in the scene. But in real-life images, there may be more than one illuminant. In this case, the illuminant estimation methods will fail to give the true scene illuminant colour. Therefore, using the illuminant colour as the only feature/signature for forgery detection may not produce reliable results.

In this work, we propose to use H_d as the illumination-signature for detecting splicing forgery. More specifically, we extract H_d from each face present in the test image as our aim is to detect face splicing forgeries. Since a dichromatic plane is spanned by the illuminant colour vector and body colour vector, it contains the illuminant colour information. If we consider two faces of the same skin colours illuminated by a single light source, the dichromatic planes estimated from the two faces should ideally be same as both the dichromatic planes are spanned by the same vectors. On the other hand, if we consider two faces of the same skin colours illuminated by two different light sources, the dichromatic planes estimated from these two faces will be different as the planes are spanned by different vectors.

Extending this idea to multiple illumination conditions, we will have more than one dichromatic plane for a single face image as there will be more than one illumination colour. These

dichromatic planes will be captured in the DPH. Therefore, H_ds estimated from the faces present in an authentic image will be similar. However, in the case of a spliced image, where one or more faces may come from different images, H_ds computed from different faces will not match with each other. This is because the faces coming from different images may not have the same illumination colours. Moreover, since the noise is almost similar in all parts of an authentic image, it will affect H_d, computed for each of the faces present in the images, in the same way.

To calculate the DPH, we use Equation (3.3) with a little modification. The right-hand side of this equation is zero only in the ideal case, *i.e.*, without any noise and with a single illumination. But, in real-life images, these conditions are hardly satisfied. Therefore, to relax these two conditions, Equation (3.3) is modified to an inequality as following:

$$|f_R(\mathbf{x})\sin(\theta)\cos(\phi) + f_G(\mathbf{x})\sin(\theta)\sin(\phi) + f_B(\mathbf{x})\cos(\theta)| < \delta \tag{3.4}$$

where δ is a small positive constant. Assigning a very small value to δ will exclude the true pixels that belong to a particular dichromatic plane, and assigning a very high value will include pixels that originally do not belong to the plane. Therefore, δ has to be chosen carefully through experimentation.

To calculate the DPHs from all the faces present in the image, as in Chapter 2, the faces are manually cropped. We have not employed the automated face detection methods as these methods sometimes detect non-face regions as faces or include non-face regions in the face bounding box. In these cases, the method will produce inaccurate results. After calculating H_ds for all the faces present in a given image, they are compared with one another. If there are M faces present in an image, then there will be total $\frac{M(M-1)}{2}$ comparisons. To compare any two H_ds, the similarity between them is computed by employing the correlation measure, given by

$$r(H_d^m, H_d^n) = \frac{\sum_\alpha \sum_\beta (H_d^m(\alpha,\beta) - \overline{H_d^m})(H_d^n(\alpha,\beta) - \overline{H_d^n})}{\sigma_m \sigma_n} \tag{3.5}$$

where H_d^m and H_d^m are H_ds of the m^{th} and n^{th} face respectively, and $\overline{H_d^m}$ and $\overline{H_d^n}$ are their respective means, and σ_m and σ_m are given by

$$\sigma_m = \sqrt{\sum_{\alpha}\sum_{\beta}(H_d^m(\alpha,\beta) - \overline{H_d^m})^2}$$
$$\sigma_n = \sqrt{\sum_{\alpha}\sum_{\beta}(H_d^n(\alpha,\beta) - \overline{H_d^n})^2} \tag{3.6}$$

In the case of an authentic image, the H_ds of all the faces will be similar as they come from the same illumination condition. Therefore, the correlation, $r(H_d^m, H_d^n)$, will be high. However, in the case of a spliced image, at least one face will come from a different image (*i.e.*, different illumination colour). So all the H_ds will not be similar, and at least one pair of H_ds will have a very small correlation. Therefore, an image is decided as forged if the minimum among all the correlations is less than a pre-defined threshold. Formally, if

$$\min_{(m,n)}(r(H_d^m, H_d^n)) < r_{th} \tag{3.7}$$

the image under investigation is decided as forged. Otherwise, the image is considered authentic. The threshold r_{th} is computed from the ROC curve obtained by applying the proposed method on a set of images. In this work, we have selected the threshold as the one which gives the optimal operating point in the ROC curve. This is explained in detail in Experiment 1 in Section 3.4.

Although our method can differentiate faces from different illumination conditions effectively, it also has limitations. For instance, it will fail in images that contain people from different ethnicities, *e.g.*, European and African, where the difference in skin colour is very large. This is because our method assumes that all faces are made of similar material. Nevertheless, our method can be applied to images containing people of almost the same skin colours.

The proposed method is summarized in Algorithm 3.1.

Algorithm 3.1

Input: Image under investigation **I**, δ, r_{th}

Output: Decision about the authenticity of the image.

Steps:

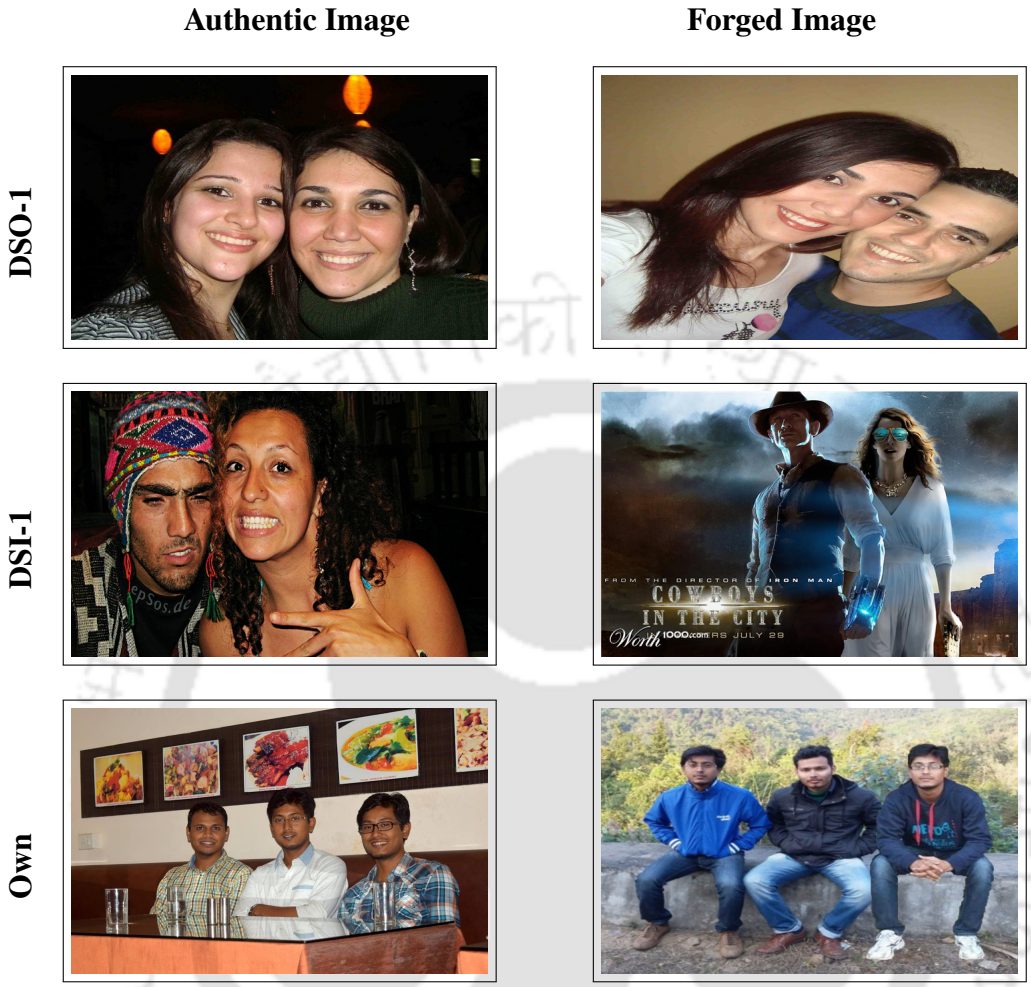


Figure 3.2: Example images from the three datasets used in this chapter.

(i) Extract all the faces present in the image under investigation. Suppose there are M faces.

(ii) Compute the DPH, H_d^m, for each face using Equation (3.4).

(iii) Compute the correlation $r(H_d^m, H_d^n)$ between the DPHs of each face pair, (H_d^m, H_d^n), using Equation (3.5).

(iv) Compute $\min_{(m,n)}(r(H_d^m, H_d^n)), (m,n) \in \{1,2,...,M\} \times \{1,2,...,M\}, m \neq n.$

(v) Decide the image to be forged if the inequality in Equation (3.7) satisfied.

3.4 Experimental Results

We have tested our proposed method on four different datasets which contain images involving human faces. These are (i) DSO-1 dataset [9], (ii) DSI-1 dataset [9], (iii) our own image

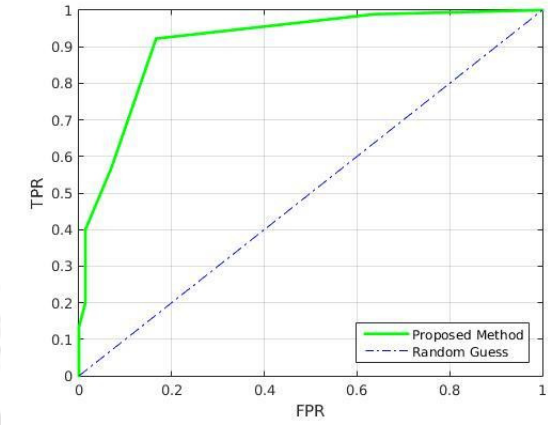


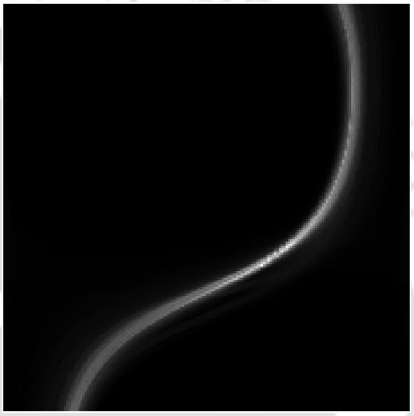
Figure 3.3: Performance of the proposed method on the "Combined" dataset, created by combining images of people of similar skin colours from DSO-1 and DSI-1 datasets.

dataset, and (iv) a dataset containing some famous forged images downloaded from the Internet. The DSO-1 dataset contains 100 authentic images and 100 forged images with the resolution of 2048×1536. The DSI-1 dataset contains 25 authentic and 25 spliced images of different resolutions downloaded from the Internet. The third dataset is our own dataset, which contains 40 authentic images and 40 spliced images of different resolutions. Here, the spliced images are created by copying one or more persons from different source images and pasting them onto a single image using the GIMP software. Figure 3.2 shows some example images from each of these datasets.

Experiment 1: As already mentioned, the proposed method cannot handle images containing people of very different skin colours. Therefore, in the first experiment, we see the performance of the proposed method on images containing persons of similar skin colours. We have removed 60 images that contain people from different ethnicities from the DSO-1 dataset. As our method requires a parameter to be tuned, we have created two sets of images from the DSO-1 and DSI-1 datasets. From the remaining 140 images in the DSO-1 datasets, 55 authentic and 55 spliced images are selected randomly, downsampled the size by half and JPEG compressed with quality factor 50, and merged with the DSI-1 dataset to create a single dataset. The reason for resizing and compressing the images from DSO-1 dataset is to make them similar to DSI-1 dataset images as the images in DSI-1 datasets are mostly of low resolution and highly compressed. This "combined dataset", which contains 80 authentic and 80 forged images, is used for testing

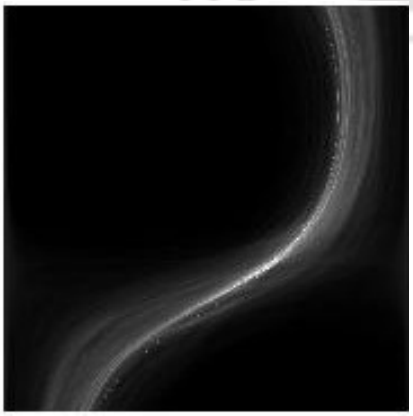
(a) Authentic

(b)



(c)

(d) Forged

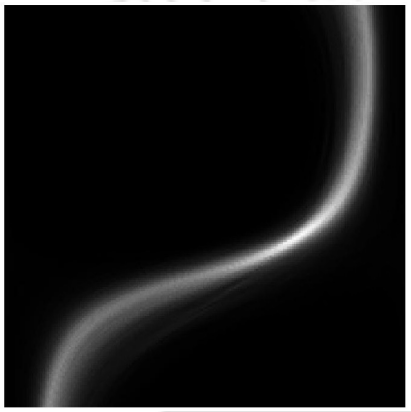


(e)

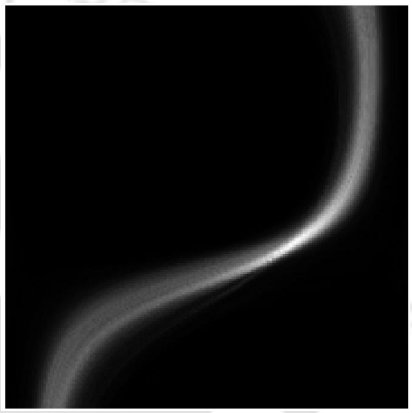
(f)

Figure 3.4: Figure (a) shows an authentic image from the DSO-1 dataset, while (b) and (c) are the DPHs of the two persons present in the image. The correlation value between the two DPHs is 0.92. Figure (d) shows a forged image, while (e) and (f) are the DPHs of the two people, and the correlation between the two DPHs is 0.73.

(a) Authentic

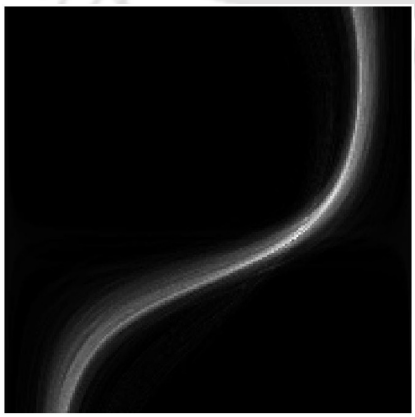


(b)

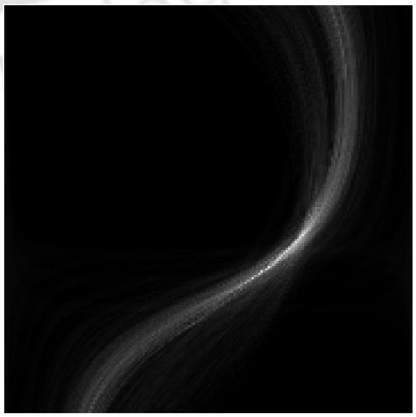


(c)

(d) Forged



(e)



(f)

Figure 3.5: Figure (a) shows an authentic image from the DSI-1 dataset, and (b) and (c) are the DPHs of the two persons present in the image. The correlation value between the two DPH is 0.97. Figure (d) shows a forged image from the same dataset, while (e) and (f) are the DPHs of the two people, and the correlation between the two DPHs is 0.39.

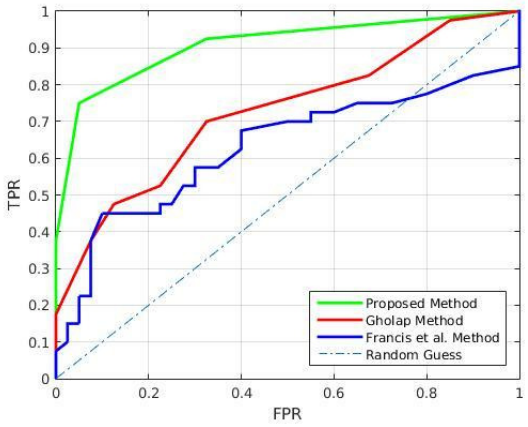


Figure 3.6: Comparison of the proposed method with existing methods on our own dataset

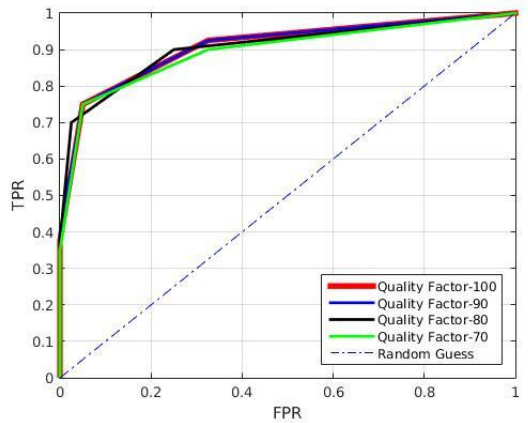


Figure 3.7: Performance of the proposed method at different JPEG compression levels on own dataset

the proposed method. The rest 30 images in the DSO-1 dataset are used to tune the parameter δ. After a number of experiments, we have set the parameter $\delta = 0.5$.

In the test phase, we have calculated the ROC curve for evaluating the proposed method on the combined dataset created by taking the images from both the DSO-1 and the DSI-1 datasets, as already explained. The ROC curve is calculated by varying the threshold on the correlation values in Equation (3.7). Figure 3.3 shows the performance of the proposed method on this dataset. The AUC value is computed to evaluate the performance of the method quantitatively. On this dataset, the proposed method resulted in an AUC of 91.2%. The optimal threshold, r_{th}, is selected as the one which yields the optimal operating point in the ROC curve. From the ROC curve, shown in Figure 3.3, the optimal operating point is found to be the one with 16% FPR and 92% TPR, and the threshold which generates this optimal operating point is $r_{th} = 0.8$. This threshold is used in the later sections to decide whether an image is authentic or forged.

In Figure 3.4, one authentic image and one forged image from the DSO-1 dataset are shown along with their DPHs. The correlation value between the DPHs of the two persons in the authentic image is 0.92, whereas it is 0.73 between the DPHs of the two persons in the forged image. In Figure 3.5, one authentic and one forged images from the DSI-1 dataset are shown along with their DPHs. The correlation value between the DPHs of the two persons in the authentic image is 0.97, whereas it is 0.39 between the DPHs of the two persons in the forged image. Therefore, in these cases, the correlation values between the DPHs of authentic images are above the optimal threshold r_{th}, and that of the DPHs of spliced images are below r_{th}.

Experiment 2: In this experiment, we have tested the performance of our proposed method on the dataset created by us. The dataset contains 40 authentic and 40 spliced images, as already mentioned. This dataset comprises images of people from India. The skin tone of Indians is brown with some amount of variations. Hence, the results on this dataset show the performance of the proposed method on images involving skin tone different from that of the DSO-1 dataset, where the skin tones of the people were mostly white with only a few dark faces.

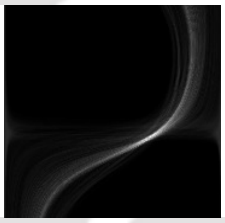
We have compared the proposed method with the existing two methods, *i.e.,* Francis *et al.* [40] and Gholap and Bora [29], as these two methods are also based on the DRM. Since

Table 3.1: AUC values achieved by the proposed method on our own dataset.

Method	AUC (%)
Gholap and Bora [29]	66.6
Francis *et al.* [40]	72.6
Proposed	**90.8**

(a)

(b)



(c)

Figure 3.8: In the image (a) Dimitri (right) is shown to be side by side with former US president Bill Clinton (left); (b) DPH of Bill Clinton, and (c) DPH of Dimitri

the original work in [29] is not intended for face images, for evaluation purposes, we have calculated the dichromatic line for each face present in an image, and then the angular error between the pair of faces is used for forgery detection. The ROC curve for the three methods are shown in Figure 3.6 and the AUC values are listed in Table 3.1. As can be seen, the ROC curve for the proposed method is well above the other two methods. The AUC value calculated from the ROC curve is 90.8% for the proposed method, while for Francis *et al.*, and Gholap and Bora methods AUC is 66.6% and 72.6% respectively. Therefore, the proposed method clearly outperforms the two existing methods. Moreover, applying the optimal threshold, *i.e.,* $r_{th} = 0.8$, computed in Section 4.1, on this dataset produces a TPR of 75% and an FPR of 5%.

To evaluate the robustness of the proposed method against the JPEG compression, we have created three more versions of our own dataset by JPEG compressing it with QFs 70, 80 and 90. The AUC values are found to be 90.8%, 90.6%, and 89.6% for JPEG QFs 90, 80, and 70 re-

Table 3.2: Performance of the proposed method at different JPEG compression levels on own dataset.

Quality Factor	AUC (%)
70	89.6
80	90.6
90	**90.8**

spectively, also shown in Table 3.2. Figure 3.7 shows the ROC curves for different compression levels are shown. This again shows the robustness of the proposed method against the JPEG compression.

Experiment 3: Another experiment is carried out to compare the performance of the proposed method with the machine learning-based state-of-the-art methods, *i.e.,* Carvalho *et al.* [9], [41]. In this experiment, we have used all the images of DSO-1. Similar to the last experiment, we have used 30 images from DSO-1 dataset for determining the optimal value δ, and these are not used for computing the performance of the proposed method. The AUC values and classification accuracies achieved by the proposed and two recent state-of-the-art methods are presented in Table 3.3. On DSO-1 dataset, the Carvalho *et al.* methods [9] and [41] achieve AUC values of 97.2% and 86.3%, respectively, and classification accuracies of 94.0% and 79.0% respectively. The proposed method achieves an AUC of 78.5% and classification accuracy of 71.6%. As can be seen, the performances of the state-of-the-art methods [9], [41] are better than that of the proposed method. This is due to the fact that the proposed method assumes the faces of all the persons present in an image to be of the same skin colour. However, there are many images in DSO-1 where persons from different skin colours are present. Therefore, from the previous and the current experiment, it is evident that the proposed method's performance drop by a large margin when applied to images with persons of different ethnicities, *i.e.,* skin colours.

We have also studied the comparative performance of the proposed method on images with different compression levels, *i.e.,* JPEG compression with different QFs. We have compressed the images in DSO-1 dataset, with QFs 70, 80, and 90. Table 3.4 shows the classification accuracies achieved by the proposed and Carvalho *et al.*'s [9] methods. The classification accuracies of Carvalho *et al.*'s are taken from [9]. It can be seen that when the images undergo JPEG

Table 3.3: AUC values achieved by the proposed method on DSO-1 dataset.

Method	AUC (%)	Accuracy (%)
Carvalho *et al.* [41]	**97.2**	**94.0**
Carvalho *et al.* [9]	86.3	79.0
Proposed	78.5	71.6

Table 3.4: Performance of the proposed method at different JPEG compression levels on DSO-1 dataset.

Quality Factor	Carvalho *et al.* [9]	Proposed
70	63.5	**69.3**
80	64.0	**70.2**
90	69.0	**71.4**

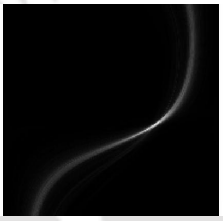
compression, the performance of the method by Carvalho *et al.* drops by a large margin. On the other hand, the proposed method is not affected by JPEG compression that much and outperforms Carvalho *et al.* at QFs 70, 80, and 90. This indicates the robustness of the proposed method against JPEG compression. This is expected because the JPEG compression affects the different faces present in an authentic image in the same way. Hence, the effect of compression on the DPH of each face will be almost similar. On the other hand, the difference in the illumination environment in a spliced image will be present even after it is compressed. Therefore, the DPHs computed from the original and the spliced faces present in a compressed spliced image will also show inconsistencies.

These experiments show that the proposed method is more applicable to real-life forensics scenarios, as most of the real-life forgeries undergo multiple compressions.

3.4.1 Analysis of Some Famous Forged Images

There are numerous forged images available on the Internet, which involve some famous persons. The first forged image that we analyze is downloaded from the Internet and shown in Figure 3.8. The image shows Dimitri de Angelis (right), a conman from Sydney, shaking hands with former United States president Bill Clinton (left). The DPHs calculated from the faces of both persons are shown in Figure 3.8(a) and 3.8(b). Although it is almost impossible to judge the authenticity of the image visually, the DPHs calculated from the two persons are clearly different from each other. The correlation value between these two histograms is found to be 0.77, which is below the threshold r_{th}, computed in Experiment 1. Therefore, the proposed

(a)



(b)

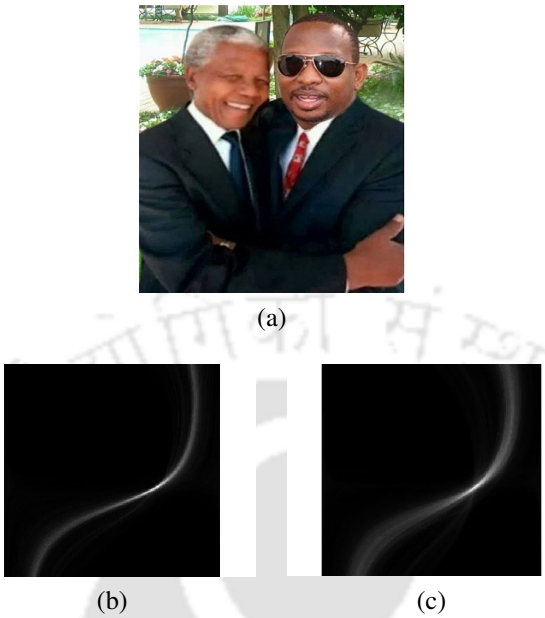
(c)

Figure 3.9: (a) An authentic image of Nelson Mandela (left) with Muhammad Ali (right), (b) DPH of Mandela's face, and (c) DPH of Ali's face

method classifies the image to be forged, which is true.

The second forged image analyzed is shown in Figure 3.10(a). In this image, a Kanyan senator named Mike Sonko is seen along with famous South African politician Nelson Mandela. The original image, however, contains Nelson Mandela and boxer Muhammad Ali, as shown in Figure 3.9(a). The forged image was created by Mike Sonko by replacing the head of Muhammod Ali in the original image with his head. The DPHs of the two persons in the authentic image are shown in Figure 3.10(b), and 3.10(c), and those of the forged image are shown in Figure 3.9(b), and 3.9(c). The DPHs computed from the two persons in the authentic image are almost similar, as shown in Figure 3.9. On the other hand, the DPHs calculated from the two persons in the forged image are different from each other, as shown in Figure 3.10. The correlation value computed between the two histograms in the authentic image is 0.97, which is above the threshold r_{th}. Therefore, the proposed algorithm classifies it to be an authentic image. The correlation value computed between the two histograms in the spliced image is 0.76, which is lower than the threshold r_{th}. Hence, the image is truly classified as spliced by our algorithm.

As already seen, the correlation values between the DPHs of the authentic faces are higher than 0.9 and those between the DPHs of the spliced and authentic faces of real-life forged



(a) (b) (c)

Figure 3.10: (a) A forged image which was created by replacing the head of Muhammad Ali by the head of Mike Sonko, (b) DPH of Mandela, and (c) DPH of Sonko

images are less than 0.8. Although the correlation values between the DPHs of the authentic and the spliced faces for both the forged images are below the threshold $r_{th} = 0.8$, they are very close. Hence, a better threshold seems to be the one at the midway point between 0.8 and 0.9. Since r_{th} is obtained from the ROC curves computed from the images of DSO-1 and DSI-1, it comes down to 0.8. This is because the correlation values between the DPHs of authentic and forged faces of DSI-1 images are very low, as can be seen in Figure 3.5. However, if we can compute the threshold from the ROC curves of more visually plausible forgeries, we expect the threshold to come somewhere in the middle of the range $0.8 - 1.0$.

3.5 Summary

This chapter proposed a new illumination colour-based forensics method to expose the splicing forgery in digital images containing human faces. The DPH, computed from the face region by applying 2D Hough transform based on the DRM, is used as the illumination-signature. To calculate the similarity between these histograms, the correlation between the DPHs of each pair of faces is computed. A threshold on this correlation measure is employed to expose the forgery. The proposed method outperformed other illumination colour-based forensics methods, which also utilize the DRM for the illumination-signature extraction. Although the proposed method

works perfectly fine for images containing people of different skin tones, it may fail in cases where the skin tones of two persons are very different from each other, *e.g.*, people from different ethnicities. This is because the method assumes the skin colour of the facial regions to be the same for all persons.

4

Deep Learning-based Classification of Illumination Maps for Detecting Spliced Faces

Chapter 3 proposed an illumination colour-based forensics method for exposing splicing forgeries involving human faces of any pose. Assuming the skin colours of all the persons present in an authentic image to be the same, the method extracts an illumination-signature from each face. The illumination-signatures from different faces are compared for detecting the spliced face. The limitation of this method is that it is not applicable to images containing persons of different ethnicities with different skin colours. Also, the method proposed in Chapter 3 assumes that the whole face part is lit by either a single illumination or a combination of multiple illumination sources. However, in real-life scenarios, different parts of a face can be lit by different illumination sources due to spatially varying illumination conditions. In this case, the method in Chapter 3 will not be able to estimate the illumination-signatures from the faces accurately.

The method proposed in Chapter 3 extracts an illumination colour-related histogram-based feature from the face parts for detecting the splicing forgery. It is nowadays well-known that hand-crafted features are not optimal for many computer vision tasks. Therefore, the features extracted by the method may not be optimal for detecting the splicing forgery. The deep learning-based methods can learn better features automatically from the training images, which are optimal for the considered tasks.

Considering the above points, this chapter proposes a novel illumination colour-based forensics method that can detect face splicing forgeries in human portrait images containing faces of different skin colours and poses. Motivated by the effectiveness of the *illumination map (IM)* as an intermediate representation of the input image for splicing detection [41], [42], the proposed method converts the input image to the IM and extracts forgery-related features from it using a convolutional neural network (CNN), trained in a siamese framework [94]. The siamese CNN is trained to classify the consistent and the inconsistent face-IM pairs present in a training dataset. Once trained, the CNN part of the siamese network is used to extract features from the face-IMs. A support vector machine (SVM) classifier is used to classify these features for splicing detection. The main contributions of this chapter are as follows: A siamese network is utilized for learning more effective features to discriminate face pairs coming similar and

different illumination conditions. The CNN part of the siamese network is then used to extract features from test face-IMs for exposing splicing forgery.

The rest of the chapter is organized as follows. Section 4.1 discusses the related work and highlights the research gap. Section 4.2 presents the proposed method, and Section 4.3 discusses the experimental results on splicing detection. Finally, Section 4.4 presents a summary of the chapter.

4.1 Related Work and Research Gap

Illumination colour is one of the important cues for detecting splicing forgeries. This is because the illumination colours estimated from the spliced regions will be different from that of the authentic regions, as already discussed in Chapter 3. The illumination colour-based forensics methods can be divided into two groups: 1) global illumination estimation-based and 2) local illumination estimation-based methods. The first group includes the methods proposed in [29], [40], [39], where the illumination colours are estimated from each face image globally. The illumination colours are then directly used for forgery detection. The method proposed in Chapter 3 belongs to this group. The second group includes the methods proposed in [79], [9], [41], [42]. In these methods, the illumination colours are estimated locally from small patches of the face images. An intermediate representation of the input image, *i.e.,* the IM, is created by using the estimated illumination colours. The IM is used for discriminating the authentic images from the forged ones. The advantage of the methods in the latter group is that they can take the spatial distribution of illumination into consideration and hence are more applicable to images captured under spatially varying illumination sources.

4.1.1 Illumination Colour Estimation

There are mainly two types of illumination colour estimation methods available in the literature [75]: (a) the statistics-based [37], [95] and (b) the physics-based methods [78], [92], [38]. Although various techniques are available from each type of illumination colour estimation methods, the following two techniques are used by the existing forensics methods for creating the IM: 1) statistics-based *generalized gray-edge (GGE)* method [37] and 2) physics-based *inverse-intensity chromaticity (IIC)* method [38]. These two methods are elaborated below.

1) Generalized gray-edge (GGE) method: The statistics-based illuminant colour estimation methods exploit the relationship between the image pixel distributions and the statistical knowledge about common surfaces and illumination sources. The GGE method [37] is based on the *gray-edge* hypothesis, according to which the average reflectance colour difference in a scene is achromatic. This is inspired by the classical *gray-world* hypothesis [36], according to which the average reflectance colour in a scene is achromatic.

Consider a Lambertian surface under an illumination source $e(\lambda, \mathbf{x})$. The image pixel value $\mathbf{f}(\mathbf{x}) = [f_R(\mathbf{x})\ f_G(\mathbf{x})\ f_B(\mathbf{x})]^T$ formed at pixel location $\mathbf{x}$ is given by

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} (e(\lambda, \mathbf{x})\, s(\lambda)\, \mathbf{c}(\lambda))\, d\lambda. \tag{4.1}$$

Here ω is the visible spectrum of light, λ is the wavelength of the incident light, $s(\lambda)$ is the surface reflectance and $\mathbf{c}(\lambda) = [c_R(\lambda)\ \ c_G(\lambda)\ \ c_B(\lambda)]^T$ is the camera response functions. In [37], the authors incorporated Minkowski norm and local smoothing in the gray-edge hypothesis yielding the following generalized gray-edge hypothesis: the pth Minkowski norm of the derivative of the reflectance in a scene is achromatic. Mathematically,

$$\left(\frac{\int \left| s^{\sigma}_{\mathbf{x}}(\lambda, \mathbf{x}) \right|^p d\mathbf{x}}{\int d\mathbf{x}} \right)^{\frac{1}{p}} = k \tag{4.2}$$

where the integration is over the image pixel domain, k is a constant term, p is a non-negative real number, the subscript $\mathbf{x}$ denotes the spatial derivative, and the superscript σ denotes the local smoothing with a Gaussian filter, $\mathbf{G}^{\sigma}$, with standard deviation σ. With the GGE hypothesis, the illuminant colour $\mathbf{e}$ is estimated as the pth Minkowski norm of the derivative of the image pixels. Mathematically,

$$\mathbf{e}^{n,p,\sigma} = \frac{1}{k} \left(\int \left| \frac{\partial^n \mathbf{f}^{\sigma}(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{\frac{1}{p}} \tag{4.3}$$

where $|.|$ is the absolute value, $\frac{\partial}{\partial \mathrm{x}}$ is the partial derivative operator, and n is the order of derivative. By varying the values of n, p, and σ, different illuminant estimation algorithms can be obtained. For instance, setting $n = 0, p = 1, \sigma = 0$ leads to the gray-world algorithm, setting $n = 1, p = 7, \sigma = 5$ leads to first-order gray-edge algorithm, and so on.

2) Inverse-intensity chromaticity (IIC) method: The physics-based methods depend on the physical process of reflections and image formation. Tan *et al.* [38] proposed a method, called the IIC, for estimating the illumination colour. It is based on the DRM [78] for non-homogenous materials, *e.g.*, human faces. The DRM is explained in detail in Chapter 3. According to DRM, the light reflected from a non-homogenous surface (*i.e.*, specular surface) comprises two different types of reflection: interface reflection and body reflection. The body reflection component contains the information about the colour of the object, while the interface reflection contains the illumination colour information. Under the DRM, the pixel value $\mathbf{f}(\mathbf{x})$ recorded by the camera sensor is given by

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} \left(e\left(\lambda, \mathbf{x}\right) + e\left(\lambda, \mathbf{x}\right) s\left(\lambda\right)\right) \mathbf{c}\left(\lambda\right) d\lambda \tag{4.4}$$

Using the above equation, Tan *et al.* showed that there exists a linear relationship between the inverse of the pixel intensity, the pixel chromaticity (*i.e.*, normalized RGB-value), and the illumination chromaticity as shown below.

$$\chi_c(\mathbf{x}) = p_c(\mathbf{x}) \frac{1}{\sum_i f_i(\mathbf{x})} + \Gamma_c(\mathbf{x}) \tag{4.5}$$

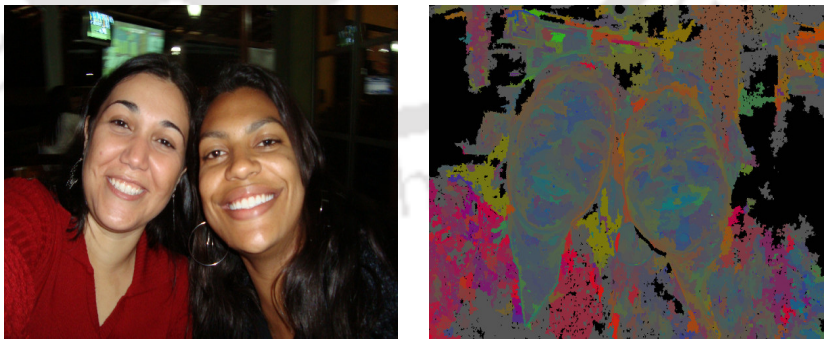
where $f_c(\mathbf{x})$ is the sensor response at pixel location $\mathbf{x}$ for each colour filter $c \in \{R, G, B\}$, $\chi_c(\mathbf{x})$ is the image chromaticity, and $\Gamma_c(\mathbf{x})$ is the specular or illuminant chromaticity, and $p_c(\mathbf{x})$ is a parameter that depends mainly on the surface geometry. The illumination chromaticity, $\Gamma_c(\mathbf{x})$, is estimated using Equation 4.5. The equation shows a linear relationship between the inverse-intensity, $\frac{1}{\sum_i f_i(\mathbf{x})}$, and chromaticity, $\chi_c(\mathbf{x})$. Therefore, a per colour channel 2D space, called IIC-space, can be created by taking $\frac{1}{\sum_i f_i(\mathbf{x})}$ as the horizontal axis and $\chi_c(\mathbf{x})$ as the vertical axis. If the image pixels are projected on the IIC plane, the vertical intercept gives the illumination chromaticity.

However, the IIC method may not give the proper estimate of the illumination colour in real-life images due to the presence of noise. To solve this issue, Riess and Angellopolou [79] proposed a modification of the IIC method. The authors proposed to compute the illuminant

chromaticities over a large number of small image patches and do majority voting to get the final estimate. It helps the illumination estimation method to become more robust against the noise present in images.

4.1.2 Illumination Map (IM)-based Image Forensics

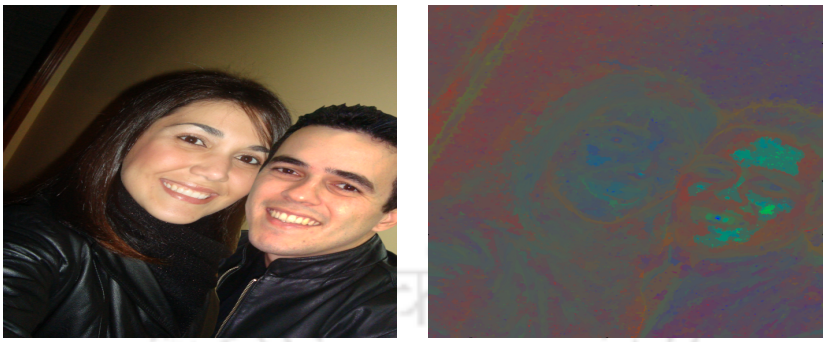
Riess and Angelopoulou [79] first proposed to use the IM as a visual cue for detecting splicing forgeries manually. To create the IM, the input image is segmented into homogenous regions, and a new image is created by recolouring each homogenous region by the estimated illumination colour from that region. The IM suppresses the surface information and highlights only the illumination colour information. Therefore, if two faces are captured under the same illumination source, they will have similar visual features in the IMs. Figure 4.1 shows an authentic image containing two faces and its corresponding IM. Although the faces in the figure have different skin colours, the face regions in the IM have similar visual features. On the other hand, if two faces are captured under different illumination sources, they will have different visual features in the IMs. For example, Figure 4.2 shows a spliced image containing two faces and the corresponding IM. Although both the faces present in the spliced image have the same skin colour, the face regions in the IM look different, *i.e.,* they have different visual features.



(a) (b)

Figure 4.1: An authentic image (a) and its corresponding IM (b).

Carvalho *et al.* [9], [41] have shown that the IM serves as an effective intermediate representation for exposing the splicing forgeries, particularly those involving human faces. As can be seen in the authentic image and its corresponding IM in Figure 4.1, the two faces have similar visual features, *i.e.*, colour, shape, and texture, in the corresponding IM. However, in case



(a) (b)

Figure 4.2: A spliced image (a) and its corresponding IM (b).

of the spliced image shown in Figure 4.2, the two faces do not have similar visual features in their corresponding IMs. Based on this observation, Carvalho *et al.* [9], [41] proposed to extract shape, texture, and colour features from the face parts of the IMs (face-IMs) using various feature descriptors. More specifically, [41] proposed to extract 12 shape features, 18 texture features, and 24 different colour features. Then, for each face-IM pair, the features extracted from both the face-IMs using the same feature descriptor are concatenated and classified using machine learning-based classifiers.

Inspired by the success of deep CNNs in computer vision, a number of deep learning-based forensics methods have been proposed in the literature [96], [50], [32], [51], [97], [17]. For instance, in [96] and [50], CNN-based methods are proposed for detecting image editing operations carried out on images. In [32], a multi-task fully-convolutional network is proposed for localizing different types of forgeries. Recently, Pomari *et al.* [42] proposed a deep learning-based method that removes the need to extract hand-crafted features from the IMs. They extracted features using a residual network (ResNet-50 [98]), pre-trained for the object recognition task, and classified them using an SVM classifier. However, the features extracted are not optimal for forgery detection as the CNN was trained for the object recognition task, which is completely different from splicing detection.

Based on the above literature survey, the following open problem is identified: The existing IM-based methods extract either hand-crafted features or deep features using a pre-trained network from the IMs. However, neither the hand-crafted features nor the deep features extracted

using a pre-trained network are optimal for extracting forgery-related features from the IMs. Therefore, there is room for learning more accurate features from the face-IM by training a network specifically for face splicing detection.

4.2 Proposed Method

We propose a method for detecting face splicing forgeries by extracting more accurate and optimal features from the face-IMs through a deep neural network, which is trained to discriminate similar and different face-IM pairs. As other illumination colour-based forensics methods [29] [9], the proposed method is based on the following assumptions:

- All the objects in an authentic image are captured under the same illumination sources. Hence, the illumination colours-related features estimated from different objects in the authentic image should be similar.
- Since the spliced objects in a forged image come from different images, there is a high probability that they were captured under a different illumination environment than that of the authentic objects. Therefore, there will be a mismatch in the illumination colour information estimated from the spliced object and that estimated from the authentic objects.

The proposed method extracts features from the face-IMs that capture the illumination colour-related information useful for exposing the spliced faces.

4.2.1 Overview of the Method

The main steps of the proposed methods are:

(1) IM Computation and Face Extraction: First, the input image is converted to an IM, as objects coming from different images (*i.e.*, different illumination sources) become more distinct in the IM [9], [41]. Since our focus is on face splicing forgery, the face parts from the IMs are manually extracted.

(2) Face-IM Pair Classification for Splicing Detection: Once the face-IMs are extracted, they are pair-wise classified for the detection of possible splicing forgery. If two faces come from the same image (*i.e.*, the same illumination source), their IMs will have the same set of features.

On the other hand, if they come from two different images, their IMs will have different sets of features. We consider a pair of face-IMs as an *authentic pair* if they come from the same illumination source (*i.e.*, the same image) and as a *spliced pair* if they come from two different illumination sources.

To differentiate between the authentic and the spliced face-IM pairs, we employ a siamese neural network [94], which takes pairs of face-IMs as inputs and predicts the class label of the pairs, *i.e.*, either authentic or spliced. The siamese network has twin CNNs, sharing the same set of parameters. The CNNs learn features that can discriminate the spliced pairs from the authentic ones. Once the siamese network is trained, we use the CNN part to extract features from the face-IMs. The features are later classified using an SVM classifier for the detection of spliced faces present in a test image. A test image is considered as spliced if at least one face pair is classified as spliced; otherwise, the image is considered as authentic.

The above steps are elaborated below.

4.2.2 IM Computation and Face Extraction

The methods proposed in [79], [9], [41] establish the effectiveness of the IM as an intermediate representation for checking the inconsistencies in the illumination environments. Inspired by this, we propose to transform the input image to the IM. To compute the IM, the input image is first segmented into homogenous regions, called *superpixels*, using the graph-based segmentation method proposed by Felzenszwalb and Huttenlocher [80]. The illumination colours are then estimated from each of the superpixels. In this work, we have experimented with two methods: 1) statistics-based GGE method [37] with $n = 1, p = 1, \sigma = 3$ and 2) physics-based IIC method [38]. After that, each superpixel is recoloured using the estimated illumination colour. As already discussed, the IM suppresses the surface information and enhances the illumination colour information. Therefore, the spliced parts become more prominent in the IM representation.

Once the IM is created, the facial regions are manually extracted from the IMs. As in [9], [41], we have also extracted the faces manually. This is because we have observed empirically that automated face detection methods sometimes give incorrect results. For example, in many

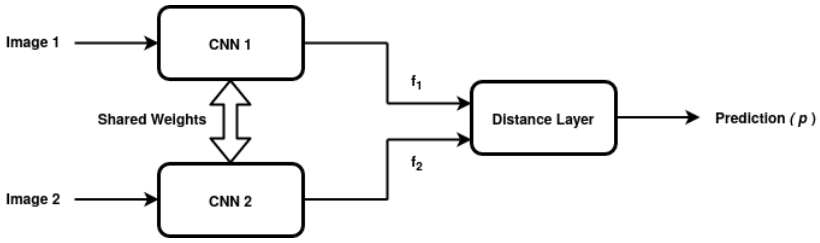


Figure 4.3: Siamese network for face-IM pair classification.

cases, the automated face detection methods detect non-faces as faces or miss to detect faces. Also, the automated methods often include non-face regions in the detected face bounding box. Since the proposed method is based on comparing the visual features in the face-IMs, any mistake in detecting the facial region will lead to misclassification of authentic and forged images. Due to these reasons, in the proposed method, we manually select the bounding boxes around the faces present in the images.

4.2.3 Face-IM Pair Classification for Splicing Detection

It has been observed that the face images captured under the same illumination environment will have the same visual features (*i.e.*, texture, shape, and colour) in their corresponding IMs [9], [41]. In case the faces come from different illumination environments, these features will be different. Therefore, by checking the consistencies of these visual features in a pair-wise manner, the authenticity of an image can be decided. If there are N faces present in an image, there will be $N(N-1)/2$ pairs. An image is considered as spliced if at least one face pair is classified as spliced; else, the image is considered as authentic.

4.2.3.1 Siamese Network for Feature Learning

A siamese neural network-based method was first proposed by Broomley *et al.* [99] for solving the signature verification problem. A siamese network consists of two identical sub-neural networks accepting one input each. The outputs of the two sub-networks are passed to a distance layer, which computes a distance metric between them. The distance metric produces a high value if the two images in a pair come from two different classes and a low value if they come from the same class.

To learn features from the face-IMs that can differentiate the authentic face pairs from the spliced ones, we propose to employ a siamese network consisting of twin CNNs [94]. In this

way, the CNN part of the network learns the features helpful for detecting the spliced faces automatically from the training data itself. The proposed siamese network takes two inputs, $\mathbf{F}_1$ and $\mathbf{F}_2$, one for each of the CNNs and the outputs are fed to the distance layer [94]. The distance layer computes a distance metric between the feature vectors learned by the twin networks. The weight sharing ensures that the same set of features are learned from both inputs. Figure 4.3 shows the block diagram of the siamese network. It contains twin CNNs, *i.e.*, CNN1 and CNN2, which compute the features $\mathbf{f}_1$ and $\mathbf{f}_2$ from the input face-IM pairs. The distance layer first finds a difference vector by computing the absolute difference between the corresponding components of the features $\mathbf{f}_1$ and $\mathbf{f}_2$. The difference vector is then fed to a fully-connected layer consisting of a single neuron with sigmoidal activation function. The output of the distance layer is in the range [0, 1]. Hence, it can be considered as the class prediction of the pair of face-IMs. A high value of this prediction indicates that the face-IMs in the pair have different visual features, and in turn indicates that the two faces come from different illumination environments, *i.e.,* spliced pair. Similarly, a low value of this prediction indicates that the two face-IMs are similar and hence are from the same illumination environment, *i.e.,* authentic pair.

4.2.3.2 Network Architecture

1) CNN

The CNN part of the siamese network is responsible for learning visual features from the training face-IM pairs. These features help in discriminating the spliced pairs from the authentic ones. Each of the twin CNNs used in this method takes an input image of size 155×155. We have experimented with smaller input sizes, such as 32×32 and 64×64. However, the results on these smaller input sizes were worse than on 155×155. The reason for this might be the following: Since our focus is on extracting features from face-IMs, we have to keep as much illumination-related information as possible. When the image size is reduced, the illumination estimation from the superpixels becomes less accurate due to the lower number of pixels in the superpixels. The number of superpixels in the face regions also reduces with the reduced input size. This results in homogenous face-IMs with only a few illuminant colours over the face regions. The dataset we use for training the siamese network contains images where the face

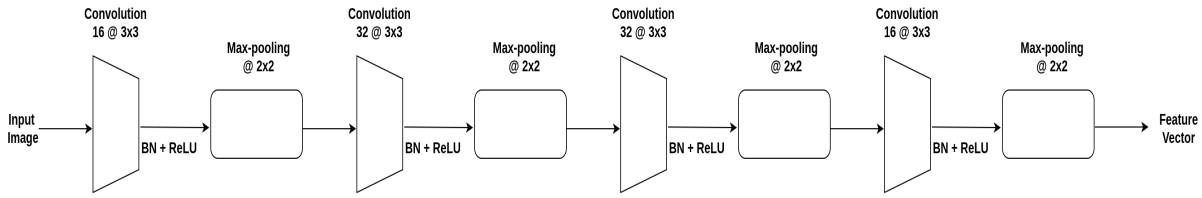


Figure 4.4: The convolutional network architecture used in the siamese network.

regions are at least of the resolution 155×155. In case the face regions in the test images are of different sizes, we resize the face-IMs to 155×155.

In the initial investigations, we experimented with deep CNNs, similar to the one proposed in [100] (VGGNet) with 16 layers. However, due to a large number of parameters, the deep CNNs overfitted the training data. This is because our training set contains a small number of images (as discussed in Section 4.3, below) compared to the number of images in datasets used to train deep networks like VGGNet [100]. Therefore, we have proposed to employ a shallow CNN containing only a few convolutional layers. The shallow CNNs with a few layers have shown to be effective in other computer vision tasks, *e.g.,* as in [101]. Finally, the CNN part of the proposed siamese network is designed to have 4 convolutional layers and 4 max-pooling layers, as shown in Figure 4.4. The number of filters in the first, second, third, and fourth convolutional layers are 16, 32, 32, and 16, respectively, with a kernel of size 3×3 and stride 1 in all the layers. The batch normalization technique [102] is used after each convolutional layer, and it is followed by rectified linear unit (ReLU) nonlinearity being applied on the output. All the four max-pooling layers employ kernels of size 2×2 with stride 2.

2) Distance Layer

The distance layer computes the weighted-L_1 distance between the features $\mathbf{f}_1$ and $\mathbf{f}_2$, computed by CNN1 and CNN2, respectively. The sigmoid nonlinearity is applied to this layer to map it to the range [0, 1]. Hence, the output of this layer can be considered to be the class prediction of the input face-IM pair. For this, a new vector first is formed by computing the element-wise absolute difference between $\mathbf{f}_1$ and $\mathbf{f}_2$. The difference vector is fed to a fully-connected single neuron with the sigmoid activation function. So, this layer outputs the weighted-L_1 distance between the features as $p = \sigma(\sum_j \alpha_j \left|\mathbf{f}_1^j - \mathbf{f}_2^j\right|)$, where $\sigma(.)$ and $|.|$ denote the sigmoidal

activation function and the absolute value, respectively. The α_js are the learnable weights of the final fully-connected layer, representing the importance of each component of the difference vector. There are 256 parameters in the distance layer.

3) Learning

The proposed siamese network is trained by minimizing the cross-entropy loss over a mini-batch of face-IM pairs. As the sigmoid activation function is used in the final layer of our siamese network, the cross-entropy loss is the preferred cost function to train the network. We assign the ground-truth label $g(\mathbf{F}_1, \mathbf{F}_2) = 0$ when both the images, $\mathbf{F}_1$ and $\mathbf{F}_2$, in the pair, come from the same image. Otherwise, we assign $g(\mathbf{F}_1, \mathbf{F}_2) = 1$. The cross-entropy loss is given by

$$\mathcal{L}_{CE} = \frac{1}{M} \sum_{i=1}^{M} g(\mathbf{F}_1^i, \mathbf{F}_2^i) \log p(\mathbf{F}_1^i, \mathbf{F}_2^i) + (1 - g(\mathbf{F}_1^i, \mathbf{F}_2^i)) \log(1 - p(\mathbf{F}_1^i, \mathbf{F}_2^i)) \tag{4.6}$$

where $p(\mathbf{F}_1, \mathbf{F}_2)$ is the prediction of the network and M is the number of face pairs in the mini-batch.

4.2.3.3 Feature Extraction

There is no face splicing dataset that contains a sufficient number of authentic and spliced face pairs to train the siamese network. Therefore, we propose to generate the authentic and the spliced pairs artificially from a set of authentic images. We convert the images to IMs, extract the face parts, and create pairs of face-IMs. If both faces of a pair come from the same image, we label the pair as authentic. If the two faces of a pair come from two different images, the pair is labeled as spliced. The siamese network is trained on these artificially created spliced and authentic face-IM pairs.

After the training process, the CNN part of the network is used to extract features from the face-IMs present in real-life spliced and authentic images. Though the siamese network learns to differentiate between the authentic and the artificially-created spliced face pairs present in the training set, the difference in the artificial spliced face-IM pairs is more as compared to the spliced face pairs present in real-life spliced images. The siamese network trained on the artificial training dataset will not be able to correctly classify the real spliced faces. It is experimentally found that the siamese network trained on the artificiall-created spliced and authentic

face pairs performed poorly in real-life splicing detection. Therefore, we do not employ the whole siamese network to classify real spliced faces. Also, it is not possible to fine-tune the network on real splicing forgeries as there is no big dataset available that contains a sufficient number of spliced faces. Due to these reasons, the CNN part of the trained siamese network is used as a feature extractor. More specifically, a face-IM is fed to the CNN, and its output is the feature of the face-IM, as shown in Figure 4.4.

4.2.3.4 Splicing Detection using an SVM

Once the feature vectors are extracted from each of the faces present in an image, they are classified in a pair-wise manner for detecting the splicing forgery. If there are N faces in an input image, we create $N(N-1)/2$ joint features by concatenating the feature vector of each face of the pairs. More specifically, if $\mathbf{f}_1 = [f_{11} f_{12} ... f_{1L}]^T$ and $\mathbf{f}_2 = [f_{21} f_{22} ... f_{2L}]^T$ are the L-dimensional feature vectors of the two faces of a pair, the joint feature is created as $\mathbf{f}_{joint} = [f_{11} f_{12} ... f_{1L} f_{21} f_{22} ... f_{2L}]^T$. An SVM with a radial basis function kernel is trained on the joint features, $\mathbf{f}_{joint}$, of the training face pairs. The trained SVM is later used for classifying the face-IM pairs present in the test images. We decide an input image as spliced if at least one face pair is classified as spliced. Otherwise, we consider the input image as authentic.

4.3 Experiments and Results

The proposed method comprises two main steps: i) feature extraction from face-IMs using a CNN, which is trained in a siamese network framework, and ii) classification of the features of face-IMs in a pair-wise manner using an SVM. The siamese network part is implemented using Python 3.6, Keras 2.2.4 [103]. The SVM part is implemented using Matlab 2017b. The hyperparameters of the SVM are tuned using the grid search technique. For the classification using the SVM, a 10-fold cross-validation protocol is followed.

4.3.1 Dataset

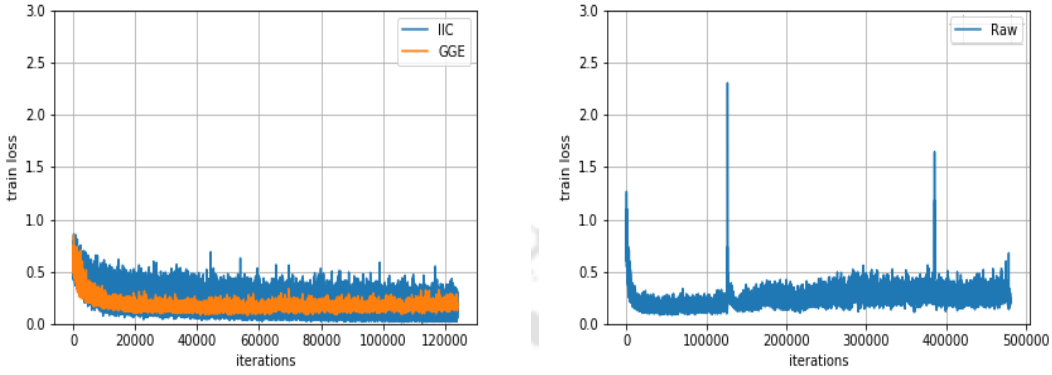
To show the efficacy of our method in exposing splicing forgeries, we have used two publicly available standard splicing datasets: DSO-1 [9] and DSI-1 [9]. The details of the datasets are as follows:

- The DSO-1 dataset contains 100 authentic and 100 spliced images of resolution $2,048 \times 1,536$. Each of the images contains group portraits of at least two persons.
- The DSI-1 dataset contains 25 authentic and 25 spliced images of various resolutions. The images are collected from various sources on the Internet.

4.3.2 Training the siamese network

To train the siamese network, we have collected a set of $6,660$ human group photos from different sources. There are $2-10$ persons in each of these images. The IMs are computed from these images using the GGE and the IIC methods, and the face parts are extracted manually. The authentic and the spliced face-IM pairs, required to train the siamese network, are created as follows. The authentic pairs are created by taking pairs of faces coming from the same image, and the spliced pairs are created by taking pairs of faces coming from two different images selected randomly. We created $57,024$ authentic pairs and $1,103,160$ spliced pairs in this manner. Out of these pairs, we have randomly selected $50,000$ authentic pairs and $50,000$ spliced pairs for training, and rest of the pairs are used for testing. The siamese network is trained separately using two different representations of the face images: (i) face-IMs computed using the GGE method, (ii) face-IMs computed using the IIC method. For the case of face-IMs computed using the GGE and the IIC methods, the network is trained for around $125,000$ iterations, and the training is stopped when the network converged. Also, we have tried to train the network using raw face images. For the case of raw face images, however, the network did not converge till $500,000$ iterations. Hence, we could not obtain a trained model for the raw face images. Figure 4.5(a) shows the plots of the training loss against the number of iteration for the network trained on IIC and GGE face-IMs, and Figure 4.5(b) shows the plot for the raw face images. As can be seen, the losses for the network trained on IIC and GGE face-IMs start to saturate at around $10,000$ iteration. On the other hand, the network trained on raw faces did not show convergence behavior. Hence a trained model could not be achieved for the raw face images.

Once the network is trained on GGE and IIC face-IMs, we test the performance of both trained models on the test set of the artificially created authentic and spliced face-IM pairs.



(a) (b)

Figure 4.5: Plot of training loss versus number of iteration for the network trained on (a) IIC and GGE face-IMs, and (b) Raw face images.

This set contains 7,024 authentic pairs and 1,053,160 spliced pairs. The network trained on GGE face-IMs achieved a classification accuracy of 95.31% on the GGE face-IM pairs of the test set. On the IIC face-IM pairs of the test set, the network trained on IIC face-IMs achieved an accuracies of 97.44%. These results suggest that the siamese network has learnt features from the IMs, that can discriminate the authentic and the spliced face-IM pairs.

4.3.3 Splicing Detection

We have carried out a set of experiments to see the performance of the two face-IM types, *i.e.,* GGE and IIC face-IMs, and the raw face images in splicing forgery detection. The performance of the proposed method in detecting the splicing forgeries are reported for DSO-1 and DSI-1 datasets.

4.3.3.1 Performance on DSO-1 dataset

Firstly, we see the performance of the features extracted using the CNN (siamese network) trained on GGE-IMs. We extract the features from GGE-IMs, IIC-IMs, and raw face images and train an SVM for each case. After the SVMs are trained, each type of feature is classified using the SVM trained for the corresponding feature type. For instance, the GGE face-IM pairs are classified using the SVM trained on GGE face-IMs only. For this trained model, we achieved detection accuracies of 79.0%, 91.0%, and 77.0% for the case of GGE-IM, IIC-IM, and raw faces, respectively. Then, we extract the features using the CNN trained on IIC-IMs

Table 4.1: Classification accuracies on DSO-1 dataset for different IM computation methods.

		Testing		
		GGE	IIC	Raw
Training	GGE	79.0	91.0	77.0
	IIC	81.0	**97.0**	77.0

Table 4.2: Classification accuracies on DSI-1 dataset for different IM computation methods.

		Testing		
		GGE	IIC	Raw
Training	GGE	83.0	90.0	84.0
	IIC	82.0	**94.0**	84.0

from GGE-IMs, IIC-IMs and raw face images. Again, an SVM is trained for each of the feature types, and the trained SVMs are used for classifying the corresponding type of features. In this case, the detection accuracies achieved are 81.0%, 97.0%, and 77.0% for GGE-IM, IIC-IM and raw faces, respectively. Table 4.1 shows the performance for all these cases on DSO-1 dataset. As can be seen, the method gives the best performance when the CNN trained on the IIC-IMs and the features are extracted from the IIC-IMs.

4.3.3.2 Performance on DSI-1 dataset

A set of experiments is carried out on DSI-1 dataset to see the performance achieved by the proposed method when the two face-IM types, *i.e.,* GGE and IIC-IMs, and the raw face images are used for testing. Here also, we first extract the features using the CNN trained on GGE-IMs from all three image representations. An SVM is trained for each type of feature and then used for forgery detection. In this case, we have achieved detection accuracies of 83.0%, 90.0%, and 84.0% for GGE, IIC, and RGB representations. Then we test the performance of the features extracted from GGE-IMs, IIC-IMs, and raw face images using the CNN trained on IIC-IMs. In this case, we achieved detection accuracies of 82.0%, 94.0%, and 84.0% for respective cases. Table 4.2 summarizes the detection accuracies for each of the cases on this dataset. One important point to be noted here is that the performance of the features extracted from raw faces is better than that of the features extracted from GGE face-IMs. One possible explanation for this behavior is as follows. As the images in DSI-1 are of low resolution and in compressed version, the IM computation from them may not be that accurate. In addition to that

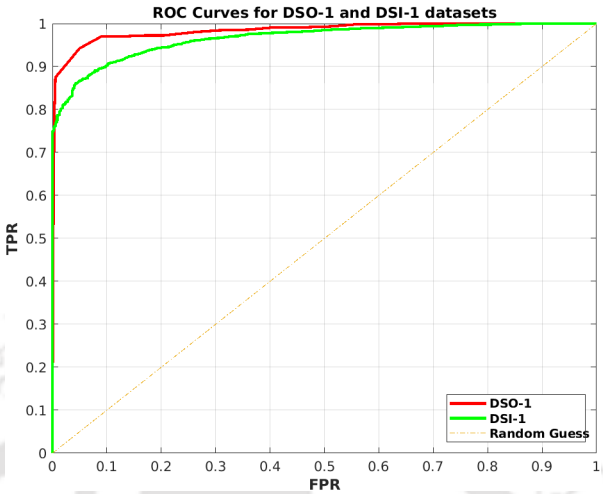


Figure 4.6: ROC curves of the proposed method for DSO-1 and DSI-1 datasets.

GGE method assumes the faces to be Lambertian. However, human faces are in real-life exhibit specular reflections and hence are not Lambertian. The combined effect of these two factors contributes to the worse performance of the GGE face-IMs in comparison to IIC and even to raw faces. Therefore, on this dataset as well, the proposed method performs better when the features are extracted from the IIC-IMs using the CNN trained on IIC-IMs.

From the results on both datasets, it is clear that the CNN trained on IIC-IMs and tested on IIC-IMs achieves the best detection accuracy among all the combinations. This is expected as the IIC method can estimate the illumination colour from face images better than the GGE method. This is because the human face is non-homogeneous, and the physics-based illuminant estimation methods assume the underlying surface to be non-homogeneous, whereas statistics-based methods assume the surface to be Lambertian.

We have also computed the ROC curves for the proposed method on DSO-1 and DSI-1 datasets and computed the AUC values. Figure 4.6 shows the ROC curves for the proposed method.

4.3.3.3 Comparison to the state-of-the-arts

To show the relative merits, we have compared the performance of the proposed method with that of the state-of-the-art methods, *i.e.*, Carvalho *et al.* [9], [41], and Pomari *et al.* [42]. We have used the best performing face-IM representation, *i.e.,* IIC-IMs, in the proposed method to compare with the state-of-the-art. We refer to this method as *IIC-IM-DL* method. Table 4.3

Table 4.3: Classification accuracies achieved by the proposed and the existing methods on DSO-1 and DSI-1 datasets.

Method	DSO-1	DSI-1
Carvalho *et al.* [9]	80.0	76.0
Carvalho *et al.* [41]	93.0	84.0
Pomari *et al.* [42]	96.0	92.0
Proposed	**97.0**	**94.0**

Table 4.4: AUC values achived by the proposed and existing methods.

Method	DSO-1	DSI-1
Carvalho *et al.* [9]	86.3	82.6
Carvalho *et al.* [41]	97.2	91.9
Proposed	**98.1**	**96.7**

shows the splicing detection accuracies of the methods on DSO-1 and DSI-1. It is clear from the table that IIC-IM-DL method outperforms the existing methods in terms of detection accuracy. Table 4.4 shows the AUC values achieved by the proposed and the competing methods. The table clearly shows that the proposed method could outperform the the methods by Carvalho *et al.* [9], [41] in terms of AUC values by a large margin. Since Pomari *et al.* [42] did not report the AUC values, we could not compare with their method. The accuracies and the AUC values of the existing methods presented here are taken from the respective papers.

We believe that the superior performance of the proposed method is due to the effectiveness of the proposed CNN in extracting more accurate visual features than those extracted by Pomari *et al.* [42]. The authors used a pre-trained network trained for another task to extract features from the IMs. The proposed method extracts features using the CNN (siamese network) specifically trained to differentiate between authentic and spliced face-IM pairs. Since Carvalho *et al.* [9], [41] extracted hand-crafted features from the face-IMs, these features are not that effective compared to the features learned by the proposed method.

4.3.3.4 Robustness to JPEG compression

We have carried out a set of experiments to see the robustness of the proposed method to JPEG compression. In real-forensics scenarios, the images generally go through various levels of compressions. Therefore, this robustness is very important from a practical point of view. We have created 3 versions of the DSO-1 dataset by compressing the images in the dataset at

Table 4.5: Performance of the proposed method at different JPEG compression levels on DSO-1 dataset.

Quality Factor	Carvalho *et al.* [9]	DPH-based method Chapter 3	Proposed IIC-IM-DL Method
70	63.5	69.3	**91.0**
80	64.0	70.2	**93.0**
90	69.0	71.4	**95.0**

three JPEG QFs, *i.e.,* 70, 80, and 90. Carvalho *et al.* also report the results at these QFs. We extract the features from the IIC face-IMs using the CNN trained on IIC-IMs. Then, we use the SVM trained using the IIC face-IMs of the uncompressed DSO-1 images to classify the face-IM pairs in the compressed versions of the datasets. Table 4.5 shows the detection accuracies achieved by the proposed method on the three compressed versions of the dataset. On the same table, we also show the detection accuracies achieved by Carvalho *et al.* [9] and the method proposed in Chapter 4. We could not compare the performance of the proposed method with that of the methods by Carvalho *et al.* [41] and Pomari *et al.* [42] as these methods did not report results for compressed images. The table shows the higher robustness of the proposed method compared to [9]. Although our method proposed in Chapter 4, *i.e.,* DPH-based method, also shows robustness to JPEG compression, its performance is inferior to that of the proposed method of this chapter by a large margin. The detection accuracy of the Carvalho *et al.*'s method drops from 80.0% to 69%, 64.0%, and 63.5% in the case of the compressions with QFs 90, 80, and 70, respectively. This drop in the accuracy of Carvalho *et al.*'s method [9] is due to the fact that the authors extracted hand-crafted features from the face-IMs, which are generally not robust against compressions. On the other hand, the proposed method learns optimal features using a siamese network, which is trained on real-life images downloaded from the Internet. In this way, the proposed method learns features present in compressed images also, as the images downloaded from the Internet are mostly JPEG compressed.

4.4 Summary

This chapter proposed a novel forensics method for the detection of face splicing forgeries through deep learning-based extraction of features from face-IMs. The method first trained a siamese-CNN to discriminate face-IM pairs coming from the same image and those coming from two different images. Once the siamese network is trained, the CNN part of the network

is used to extract features from the face-IMs present in a test image. An SVM is employed on these features to classify the real-life spliced faces present in the test images. We experimented with different types of IM computation methods and found that the IIC method performs the best among them. The experimental results on two public datasets show that the proposed is able to outperform the state-of-the-art.

5

Siamese Convolutional Neural Network-based Approach to Universal Image Forensics

The methods proposed in Chapters 2, 3, and 4 focused on detecting spliced faces present in images. The limitations of these methods are: 1) they can only detect the splicing forgery, and 2) they fail in case the spliced regions are not human faces. In real forensics scenarios, the splicing forgery may involve any arbitrary regions in an image, other than the human face. Also, there may be other forgery types like copy-move and retouching. In addition to these forgery techniques, there are different image processing operations that are applied on images for various reasons, such as to enhance the contrast or brightness of an image and to denoise an image. Therefore, it has become an important task in image forensics to detect both the forgeries and the image processing operations carried out in images. Based on these motivations, this chapter proposes a universal forensics method that can detect the presence of different image editing operations and also detect and localize forgeries in a single framework.

The detection of different types of image editing operations carried out on an image is an important part of image forensics. This is because of the following reasons: 1) images can be processed using different editing operations to artificially enhance or remove some features to make them look visually different from their raw versions, captured by the camera devices, and 2) the manipulated images are generally post-processed by different types of image editing operations to remove the traces of manipulations left by the forgery process. For example, median filtering can be applied to remove the traces of JPEG blocking artefacts present in an image [58]. However, each editing operation leaves behind a unique signature, which is utilized by researchers to detect the type of operations carried out on the image. There are many methods available in the literature, which focus on the detection of image editing operations applied to images. For example, the methods, proposed in [104], [105], [106], extracted features for detecting the traces left by the resizing and the resampling operations. The methods proposed in [107], [108], [109] extracted features related to median filtering traces. The features related to the contrast-enhancement operation are extracted in the methods proposed in [110], [111], and the JPEG artefact-related features are extracted in [112].

The processing history of an image can also expose different types of forgeries present in images. This is because the forged parts are often processed through various image editing

operations, such as resizing, blurring, and contrast enhancement, to make them look visually plausible. On the other hand, the authentic parts may not go through the same type of editing operations as the forged parts. Therefore, the presence of different types of image editing operations at different parts of an image can give a clue about its authenticity.

Although the methods proposed in [104]- [112] are capable of detecting specific types of manipulations, methods from each of the categories work only under their own assumptions about the traces of manipulations left by the forgery process. For example, the median filtering detection methods cannot detect traces left by the resampling operation. To address this limitation, researchers have focused on developing *universal forensics* methods, which can detect multiple manipulations in a single framework. The first universal forensics method was proposed by Qiu *et al.* [113], where different steganalysis features were used to detect different types of image processing operations. Fan *et al.* [114] proposed a general-purpose forensics method for detecting different types of image editing operations by creating a Gaussian mixture model (GMM) from image patches corresponding to each editing operation.

Inspired by the success in other computer vision areas, the forensics community has focused on applying DL-based methods for image manipulation detection. Chen *et al.* [49] proposed the first DL-based method for the detection of image editing operations. The authors employed a CNN [10] for detecting the median filtering operation carried out on images. In this method, the first layer of the CNN computes the median filtering residual, and the subsequent layers extract and classify the features useful for median filtering detection. Bayar and Stamm [50] proposed a DL-based universal forensics method for detecting different types of image manipulating operations. The image editing features are automatically learned from the training data by employing a CNN. The authors proposed a new convolutional layer, which suppresses the image content and enhances the features important for detecting different editing operations.

Although these universal manipulation detection methods perform well, all the manipulation operations have to be known before training the network. A large number of image editing operations are included in the image editing software, like Adobe Photoshop and GIMP. Newer image editing operations are also being developed and incorporated in these editing software.

Further, there may be multiple editing operations performed subsequently to make the forged parts look similar to the authentic parts. Therefore, it is not practical to incorporate all the editing operations in the training process as required by the existing universal manipulation detection methods, *i.e.*, [113], [114], [50]. This necessitates the development of a universal forensics method that can detect not only the image editing operations considered in the training stage but also can generalize to editing operations unknown in the training stage.

This chapter proposes a novel deep learning-based forensics method that can not only detect known manipulations but also can check the presence of manipulations not considered in the training stage. The proposed method takes two images as inputs and checks whether they have undergone the same or different manipulation operations. Also, a forgery detection and localization technique is proposed using the trained siamese network. The method first divides the image under investigation into a number of patches and then compare them in a pair-wise manner using the siamese network for localizing the forgeries.

The main contributions of this chapter are as follows:

1) A novel siamese network-based manipulation detection technique is proposed, which can i) detect image editing operations considered in the training stage, ii) detect known editing operations not considered in the training stage using the *one-shot classification technique*, and iii) check whether a test image has undergone any unknown image editing operation.

2) Based on the above method, an image forensics technique is proposed to localize and detect different image forgeries effectively.

The rest of the chapter is organized as follows: Section 5.1 describes the related forensics methods and outlines the research gaps. Section 5.2 presents the proposed manipulation detection method. Section 5.3 presents a novel forgery localization and detection technique. Section 5.4 discusses the results of the experiments carried out to show the efficacy of the proposed method. Section 5.5 presents a summary of the chapter.

5.1 Related Work and Research Gap

The forensics methods proposed for detecting image editing operations are based on the assumption that each editing operation leaves behind a unique trace of its manipulation in the image. These traces can be detected by examining the relationship between the neighboring pixels, as any image manipulation destroys the natural statistics of pixels and modifies them in a unique way [113]. Qiu *et al.* [113] proposed a universal forensics method for detecting image editing operations, namely Gaussian blurring, median filtering, JPEG compression, resampling, and gamma correction. The author explored various steganalytic features, such as *subtractive pixel adjacency matrix (SPAM)* [115], *spatial-domain rich model (SRM)* [116], and *local binary pattern (LBP)* [117], to detect different types of image processing operations. The method is based on the observation that different image editing operations destroy the natural statistics of the pixels of an authentic image in the same way steganography methods do while manipulating the pixels for embedding a message. Fan *et al.* [114] proposed to create a GMM from small image patches corresponding to each editing operation. Then, the average log-likelihood of the patches of an image under the different GMMs corresponding to different classes are compared to decide the editing operation applied to the image. The authors also showed that the forgery localization can be done by checking the presence of more than one editing operation in different parts of the input image.

CNNs have proven to be very good in different computer vision tasks, *e.g.*, object detection and recognition [118]. However, they did not perform well when applied directly to image manipulation detection [49]. This is because the conventional CNNs capture the image content rather than important forensics features. Therefore, to suppress the image content and enhance the relationship between neighboring pixels, most of the forensics methods employ prediction error filters in the first layer of the CNNs. These prediction error filters perform high-pass filtering on the input image and produce noise residuals which are then fed to the subsequent layers of the CNN. For instance, Chen *et al.* [49] proposed to train a CNN for differentiating images modified by applying median filtering from the original unaltered version. The authors first extracted the median filtering residuals for detecting median filtering operation. They experi-

mentally showed that the CNN is able to learn features that can differentiate the median filtered images from the unaltered ones. Using the trained CNN, Chen *et al.* proposed a splicing forgery localization method. Bayar and Stamm [50] proposed a CNN-based technique for classifying multiple image editing operations in a single framework. For this, the authors proposed a new convolutional layer, named *constrained convolutional layer*, as the first layer in their proposed CNN to adaptively suppress the image content for detecting different image editing operations. The constrained convolutional layer learns a set of K prediction error filters by constraining the weights in each filter as

$$\begin{aligned} w_k^1(0,0) &= -1 \\ \text{and} \sum_{l,m\neq 0} w_k^1(l,m) &= 1 \end{aligned} \tag{5.1}$$

where $w_k^1(l,m)$ denotes the weight at position (l,m) of the kth filter and $w_k^1(0,0)$ denotes the weight at the center of the corresponding filter kernel. This procedure is repeated for all the pixels by moving the kernels throughout the image. This prediction error layer extracts the local dependency of pixels with their neighbours, which is the important information from the forensics point of view [113]. The authors showed that the CNN can learn features automatically through the training process for detecting various common image editing operations, such as Gaussian filtering, median filtering, gamma correction, and JPEG compression. The authors also showed that the CNN can learn to detect multiple editing operations applied subsequently on images, *e.g.,* applying Gaussian blurring followed by gamma correction.

The idea of using the prediction error filters in the first layer is motivated by different image forensics and steganalysis methods. Steganalysis methods like the SRM [116] and the SPAM [115] utilize different prediction filters for computing the prediction errors, which are later used as features to detect hidden messages present in the stago images. In forensics, Rao *et al.* [97] proposed to initialize the first layer of a CNN with the SRM filters. Recently, Zhou *et al.* [119] proposed a forensics method, where they used 3 SRM filters as the first layer of a CNN. They also showed that increasing the number of filters does not necessarily boost performance.

The universal forensics method proposed by Bayar and Stamm [50] shows that CNNs can automatically learn features important for detecting different image editing operations. How-

ever, there are some limitations of the method. For training the CNN, all the image editing operations should be known *a priori*. If an image is subjected to an operation other than the ones used for training, the method will not be able to detect it. This is a critical limitation of the method proposed by Bayar and Stamm [50], as the knowledge about the type of editing operations carried out on images is generally not available in practical forensics scenarios. Further, since Bayar and Stamm's method cannot detect unknown manipulations, it cannot be used for localizing forgeries. This is because in a forged image, the forged regions can be manipulated using any image editing operations, which may not be considered while training the CNN. Based on these points, the following research gaps are identified:

1) There is room for developing forensics methods that can not only detect the image manipulations considered in the training stage but also check whether an image is being processed by manipulations unknown during the training stage.

2) A forgery localization and detection method can be developed utilizing the traces left by various imaged editing operations, which are generally applied as post-processing operations.

Recently, Mayer and Stamm [120] have, independently of our work, proposed their siamese network-based *Forensics Similarity (FS)* method for checking whether a pair of images has the same forensic trace or not. More specifically, they have trained a siamese network to check whether a pair of images are 1) captured by the same or different camera models and 2) modified by the same or different image editing operations. They have also shown the ability of the trained siamese network in detecting and localizing forgeries, such as splicing. Although both the proposed and the FS methods use siamese networks for manipulation detection, there are a number of differences in the network architectures. More notably, the proposed method uses a constrained convolution layer operating on the green channel of the input images. On the other hand, the FS method processes all the three RGB channels of the input images and relaxes the constraint imposed on the first convolutional layer. Also, the proposed method computes a distance metric between the twin CNN features, whereas the FS method concatenates the twin CNN features along with their element-wise multiplications. We propose to employ the

one-shot classification technique to check the presence of unknown manipulations in an image under investigation.

There are various image forgery localization and detection methods available in the literature. These methods utilize different traces for detecting the forgeries. For instance, the methods proposed in [46] (CFA1) and [47] (CFA2) utilizes the mismatch in the colour filter array (CFA) de-mosaicing artefacts present in the manipulated and the authentic regions of a forged image. The method proposed in [121] (ADQ1) exposes the forged regions by detecting the presence of aligned double JPEG quantization in a forged image. In [122] (DCT), the authors proposed to check the inconsistencies in distribution of the discrete cosine transformation coefficients computed from different parts of an image for exposing the forgery. In [33] (BLK), the non-alignment of the 8×8 block grids in a region of an image is used as a cue of forgeries. The method proposed in [123] (ELA) detects the forged regions present in an image by checking the inconsistencies in the amount of JPEG compression that different regions have undergone. the method proposed in [124] (NOI1) extracts the noise pattern using a wavelet-based filtering method, and the variance of the noise of different parts are compared to detect the forged region. The method proposed in [52] (Noiseprint) first trains a siamese network to differentiate images coming from different camera models and then use the trained network to detect spliced regions present in a forged image. This is based on the assumption that the spliced regions will come from images captured by camera models that are different from the one used to capture the authentic regions. In [32] (MFCN), a deep learning-based multi-task fully-convolutional neural network is trained specifically for localizing the splicing forgeries present in images.

5.2 Proposed Manipulation Detection Method

The proposed method is inspired by the works of Chen *et al.* [49] and Bayar and Stamm [50], where they showed that CNNs are capable of learning image manipulation features automatically from the training data. However, instead of learning features to classify images to different manipulation classes, the proposed method learns the features which can discriminate various image editing operations. To achieve this, we employ a siamese network-based metric learning technique [94]. This technique is capable of learning generic image features that can generalize

to unknown classes. The siamese network has twin CNNs accepting two images as the input and classifies the image pair as either *similarly processed* (SP) or *differently processed* (DP). Once the network learns to differentiate between SP and DP image pairs, we use it to detect image editing operations present in a gallery of manipulations in a *one-shot classification* fashion. The gallery contains images edited using the image processing operations commonly used in the anti-forensics methods. The gallery may contain images edited using manipulations that are not present in the training stage. Therefore, it can be augmented by adding more and more image processing operations, which may be required by a forensics analyst at the time of analysis. We assume that we have at least one image from each manipulation class considered in the gallery. Given a test image, we first check whether the image is unaltered or manipulated by comparing it with a reference unaltered image using the trained siamese network. If the image is classified as manipulated, we check whether it is manipulated using any of the editing operations present in the test gallery.

As already stated, the proposed method is based on the classification of image patch pairs as either SP or DP through a metric learning-based technique using a siamese network [99], [125]. More specifically, the siamese network learns a distance metric to check whether two image patches have gone through the same type of editing operations or not. The reasons for the siamese network-based classification of patch pairs are as follows:

(i) Unlike the CNN-based methods, *e.g.*, [50], [49], the siamese network can learn more generic image manipulation features through a distance metric learning-based approach. This is an important advantage of the proposed method, as it allows to discriminate/detect manipulations not present in the training stage.

(ii) Since the methods in [113], [114], and [50] learn class-specific features to classify the images into one of the different but fixed types of manipulations, they are more vulnerable to anti-forensics. Anti-forensics methods can be developed to hide the traces left by each of the operations considered in these methods [126]. On the other hand, the proposed method learns a distance metric to check whether two image patches have undergone the

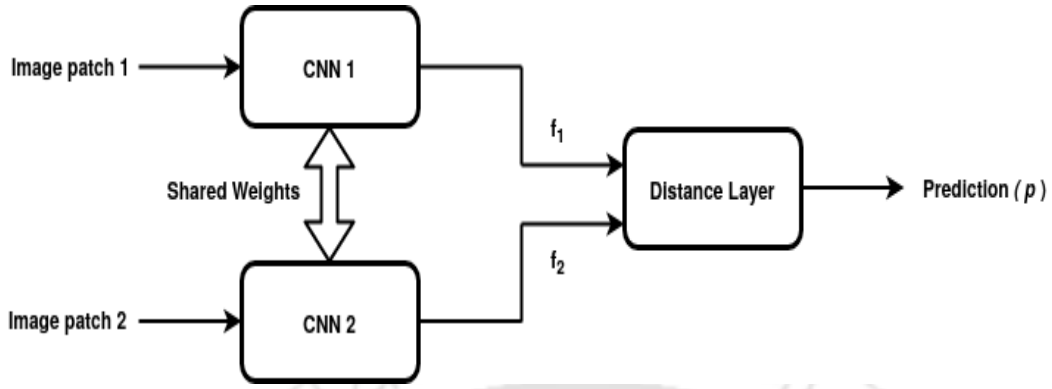


Figure 5.1: Framework of the siamese network that takes a pair of input image patches and produce prediction p indicating whether the pair is SP or DP.

same type of manipulation or not. Hence, developing anti-forensics techniques to counter the proposed method will be more difficult.

(iii) In a forged image, the tempered regions are generally post-processed with different image editing operations to make them look visually plausible. As a result, there are differences between the editing operations present in the manipulated and the authentic regions in the forged image. Therefore, checking the differences in the editing operations might expose the forgery.

The proposed manipulation detection method using siamese network-based patch pair classification is described below.

5.2.1 Siamese Convolutional Neural Network for Image Editing Operation Detection

Figure 5.1 shows the block diagram of the proposed framework. It has twin neural networks CNN1 and CNN2 sharing the same set of weights. It accepts two input image patches $\mathbf{F}_1$ and $\mathbf{F}_2$, which are parallelly processed by CNN1 and CNN2, producing two feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$. The distance layer [94] computes a distance metric between $\mathbf{f}_1$ and $\mathbf{f}_2$, and the sigmoid nonlinearity then maps the output to the range $[0, 1]$ to produce the prediction p. Because of the sharing of weights, CNN1 and CNN2 map two similar input images to very close points in the feature space. The proposed siamese CNN automatically learns the features that can check whether a pair of image patches is SP or DP.

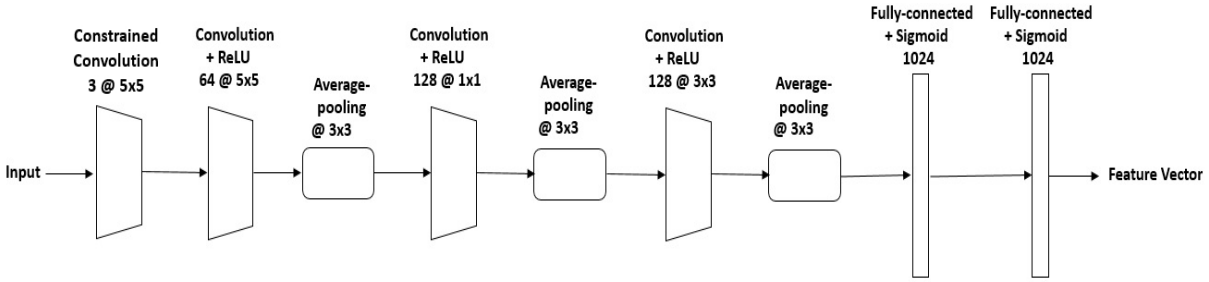


Figure 5.2: The CNN architecture used in the proposed siamese network.

5.2.2 Network Architecture

5.2.2.1 CNN

The CNN part of the siamese network has the architecture shown in Figure 5.2. The input to the CNN is a gray-scale image of size 150×150. We have experimented with other input sizes as well, namely 64×64 and 32×32. We did not experiment with input size bigger than 150×150 due to computational constraints. We empirically found that the network performs slightly better when the input size is 150×150 than the other two sizes. One possible reason for this is that the network has more number of pixels for learning the features related to the image editing operations when the input size is 150×150 than the other two. In the initial investigations, we have experimented with deep CNNs with more than 10 layers, such as ResNet-50 [98] and VGGNet-16 [127]. Although the performances of the deep CNNs were found to be good in detecting the manipulations considered in the training stage, they did not perform well in detecting manipulations not considered in training. This suggests the overfitting of the deep CNNs. After a set of experimentation, we found that the CNN with 1 constrained convolutional layer, 3 convolutional layers, 3 average-pooling layers, and 2 fully-connected layers performed the best in detecting editing operations considered in the training stage as well as unknown operations. The other hyper-parameters, namely the type HPF layer and activation function, used in the proposed CNN are based on empirical results, which are presented in the experimental section.

The block diagram of the CNN is shown in Figure 5.2. The constrained convolutional layer contains 3 constrained convolutional filters [50] whose weights follow the constraints given by Equation (5.1). This layer is followed by a convolutional layer with 64 filters of size 5×5

with stride 1. The ReLU nonlinearity is applied element-wise to the output of this layer. It is followed by an average-pooling layer with a kernel size 3×3 and stride 3. The output of this layer is fed to another convolutional layer with 128 filters of size 1×1 and stride 1. The ReLU nonlinearity is applied element-wise to the output of this layer. It is followed by an average-pooling layer with a kernel size 3×3 and stride 3. The reason for using 1×1 convolution filters are: 1) it has fewer parameters than the bigger filters, therefore reduces the chance of overfitting and 2) it combines the features present at the same location across different feature maps. The output of this layer is fed to another convolutional layer with 128 filters of size 3×3 and stride 1. The ReLU nonlinearity is applied element-wise to the output of this layer. It is followed by an average-pooling layer with a kernel size 3×3 and stride 3. This layer is followed by two fully-connected layers, each with 1024 neurons. The sigmoid nonlinearity is used in each of these layers. The neurons in the fully-connected layers are dropped out [128] with a probability of 0.5 at each iteration of the training process. The output of the final fully-connected layer represents the features learned by the CNN.

5.2.2.2 Distance Layer

After the feature vectors, $\mathbf{f}_1$ and $\mathbf{f}_2$, are computed from a pair of image patches $\mathbf{F}_1$ and $\mathbf{F}_2$ by CNN1 and CNN2, respectively, the distance layer computes a distance metric between them. More specifically, the distance layer first computes the weighted-L_1 distance between $\mathbf{f}_1$ and $\mathbf{f}_2$, and then maps it to the range [0, 1] by applying the sigmoid non-linearity. Therefore, the output of this layer can be considered as the class prediction of the input image patch pair. Firstly, a difference vector is formed by computing the element-wise absolute difference between $\mathbf{f}_1$ and $\mathbf{f}_2$. Then, the difference vector is fed to a fully-connected single neuron with the sigmoid activation function. This neuron computes the prediction of the input image pair as

$$p = \sigma(\sum_j \alpha_j |f_1(j) - f_2(j)|) \tag{5.2}$$

where $\sigma(.)$ is the sigmoid nonlinearity function, and α_j is a learnable parameter representing the importance of each component of the feature vectors in the classification of the patch pair.

5.2.3 Learning

The siamese network is trained by minimizing the cross-entropy loss over a mini-batch of image patch pairs. We assign the ground-truth label $g(\mathbf{F}_1, \mathbf{F}_2) = 0$ when both the image patches $\mathbf{F}_1$ and $\mathbf{F}_2$ in the pair come from the same image. Otherwise, we assign $g(\mathbf{F}_1, \mathbf{F}_2) = 1$. The cross-entropy loss $\mathcal{L}_{CE}$ is given by

$$\mathcal{L}_{CE} = \frac{1}{M} \sum_{i=1}^{M} g(\mathbf{F}_1^i, \mathbf{F}_2^i) \log p(\mathbf{F}_1^i, \mathbf{F}_2^i) + (1 - g(\mathbf{F}_1^i, \mathbf{F}_2^i)) \log(1 - p(\mathbf{F}_1^i, \mathbf{F}_2^i)) \tag{5.3}$$

where $p(\mathbf{F}_1, \mathbf{F}_2)$ is the prediction of the network, M is the number of image patch pairs in the mini-batch, and i is the index of the pair.

The network is trained until the loss value starts to saturate. Once the network is trained, the model weights are stored for inference.

5.2.4 Manipulation detection using One-shot Classification

Once the siamese network learns to discriminate between the SP and the DP image patch pairs, we use it to detect different image editing operations in a pair-wise manner. More specifically, given a test image patch $\mathbf{I}$ and a set of image patches $\{\mathbf{I}_c\}_{c=1}^{C}$ coming from each of the editing operations present in a gallery of manipulations, we first compute the pair-wise prediction p_c of $\mathbf{I}$ and $\mathbf{I}_c$. The prediction p_c represents the similarity between the types of editing operations applied on $\mathbf{I}$ and $\mathbf{I}_c$. Thus, the type of operation c^* applied on $\mathbf{I}$ is given by the class for which prediction is maximum. Mathematically,

$$c^* = \text{argmax}_c p_c. \tag{5.4}$$

The main advantage of using a siamese network is that once it learns to discriminate/detect different classes present in the training stage, it can generalize to unseen classes with very few images per class. This is because the siamese network learns more generic discriminative features than simple CNNs by learning a distance metric from the training data [94]. Once the network learns the metric, it can be used to compare images coming from classes not present in the training stage [125]. In the extreme case, given only one example per unseen class, the

network can check whether a test image patch has undergone any of these manipulations. This is the one-shot classification method. Therefore, once we train the proposed siamese CNN to discriminate different editing operations present in the training stage, it learns features that are capable of detecting/discriminating unseen editing operations.

Now, given a test image patch, we first compare it with a reference unaltered image using the trained siamese network. If the pair-wise prediction p is more than 0.5, the test patch is classified as unaltered. Otherwise, the test patch is considered to be manipulated, and the type of manipulation is detected using one-shot classification. We assume to have at least one reference image patch from each manipulation class included in the gallery of manipulations. We want to check whether the editing operation in the gallery is applied on the test image patch or not. To achieve this, we compare the test image patch in a pair-wise manner with one reference image patch from each class. The class of the test image patch is given by the class of the reference image corresponding to the maximum prediction score p, as given by Equation (5.4). However, we need to ensure that the maximum prediction score is sufficiently high. In this work, we assume that it is bigger than 0.5. This is because in case the test image patch is manipulated using an editing operation not present in the gallery of manipulations, the maximum prediction score will not correspond to the true manipulation class. Also, in this case, the maximum prediction score will not be very high. Hence, if the maximum prediction score is less than 0.5, we do not assign any class to the image patch and consider it as a manipulated image only. The proposed manipulation detection method is presented in Algorithm 5.1.

Algorithm 5.1

% Algorithm for detecting the class of manipulation applied on a test image patch %

Input: Test image patch $\mathbf{I}$, gallery G containing a reference unaltered image patch $\mathbf{I}_u$ and a set of reference image patches $\{\mathbf{I}_c\}_{c=1}^{C}$ manipulated using the image editing operations considered in the testing phase.

Output: Decision about the authenticity of $\mathbf{I}$/manipulation class of $\mathbf{I}$

Steps:

1) Compute the prediction score p for the pair $\mathbf{I}$ and $\mathbf{I}_u$ using the trained siamese network.

2) If $p > 0.5$, decide **I** to be unaltered and go to step 6.

3) For $c = 1, 2, ..., C$, compute the prediction score p_c of the pair consisting of **I** and $\mathbf{I}_c$.

4) Compute c^* using Equation 5.4 and find the corresponding p_{c^*}.

5) If $p_{c^*} > 0.5$, assign the manipulation class c^* to **I**. Else, decide **I** as a manipulated image patch with an unknown editing operation.

6) Stop.

5.3 Forgery Localization and Detection

We now describe the method to localize and detect different forgeries present in images using the proposed siamese network. The method is based on the following assumptions:

(i) All the parts of an authentic image go through the same types of editing operations.

(ii) While creating a forged image, there are often different image editing operations carried out on the forged parts to make them look visually plausible.

(iii) The area of the forged parts is less than the area of the authentic parts.

For example, in splicing forgery, the spliced objects may come from a different noisy environment or may be manipulated using various image editing operations. In the case of copy-move forgery, one region may be copied and pasted at a different location after performing some processing operations like up/downsampling. Therefore, detecting the presence of more than one type of image editing operation in an image can give a clue about the authenticity of the image.

To check the presence forgery in a test image, it is first divided into a number of overlapping patches, $B_i, i = 1, 2, ..., N$ where N is the number of patches. We consider 50% overlap between two adjacent patches. One of these patches is randomly selected as a reference patch, B_r, and others are compared with B_r in a pair-wise fashion using the trained siamese network. The siamese network is trained to classify a pair of patches as either SP or DP, as explained in the previous section. Since it gives the probability p of classifying a pair as SP, ideally, all the pairs of patches in an authentic image will have higher values of p. In the case of a forged image,

the reference patch may come either from the authentic part or from the forged part. If the reference patch comes from the forged part, there will be high pair-wise probabilities for forged patches and low pair-wise probabilities for authentic patches. On the other hand, if the reference patch comes from the authentic part, the forged patches will have low pair-wise probabilities, and the authentic patches will have high pair-wise probabilities. Therefore, we first decide the class of the reference patch, *i.e.*, whether authentic or forged, and then detect the forged patches accordingly.

Suppose p_i represents the p value for patch B_i with respect to the patch B_r, and per_{fb} represents the percentage of patches having p values lower than a predefined threshold th. Then,

$$per_{fb} = \sum_{i=1}^{N} 1(p_i < th)/N \tag{5.5}$$

where 1(.) is an indicator function that gives 1 whenever its argument is true and 0 otherwise. If $per_{fb} < 0.5$, the reference patch B_r is considered as authentic. Otherwise, B_r is considered as forged. Once the class of the reference patch is decided, the forged patches are detected as per the following two cases:

- *Case-1:* If B_r is authentic, the patches corresponding to the pairs having p values less than th is considered as the forged patches. This is based on the assumption that the area of the forged region is less than the area of the authentic region.
- *Case-2:* If B_r is forged, the patches corresponding to the pairs with p values greater than th is considered as forged patches.

Once the forged patches are detected, a binary mask image is created by assigning the pixels in the forged patches a value of 0 and pixels in the authentic regions a value of 1. The binary mask helps in the qualitative and quantitative analysis of the effectiveness of the forgery localization technique.

We have experimented with different image patch sizes, *i.e.*, 150×150, 64×64, and 32×32, and found that patches of size 64×64 with a stride of 32 pixels is the best option among them. Although the manipulation detection accuracy increases as the patch size increases, the

localization ability of the forgery detection method reduces with increasing patch size. If the patch size is too small, the manipulation detection will not be reliable, and hence it will produce many false alarms in the forgery detection process.

The steps of the forgery localization technique are outlined in the following algorithm.

Algorithm 5.2

% Algorithm for forgery localization using the siamese network trained to discriminate different image editing operations %

Input: Test image $\mathbf{I}$ and threshold th

Output: Binary mask image $\mathbf{I}_B$ showing the forged regions as black and authentic regions as white.

Steps:

(i) Divide $\mathbf{I}$ into N patches $B_i, i = 1, 2, ..., N$ of size 64×64 with a stride of 32.

(ii) Randomly select one patch B_r as a reference patch.

(iii) For $i = 1, 2, ..., N$ compute the prediction p_i of each patch B_i with respect B_r using the trained siamese network.

(iv) Compute per_{fb} using Equation (5.5).

(v) If $per_{fb} < 0.5$, decide the B_r as forged. Else decide it as authentic.

(vi) If B_r is forged, decide B_is for which $1(p_i > th) = 1$ as forged and the rest as authentic. Else, decide B_is for which $1(p_i < th) = 1$ as forged and the rest as authentic.

(vii) Generate a binary mask image $\mathbf{I}_B$, where the pixels in the forged patches are set to 0, and the authentic pixels are set to 1.

Once the forged regions are computed from an image under investigation, we can take the image-level decision, *i.e.*, whether forged or authentic. This is important as there may be false detection of forged regions, even in authentic images. The image-level decision is made based on the percentage of forged regions present in an image. For this, we compute the percentage of

Table 5.1: Different manipulations considered in this work

Editing Operation	Detail
Gaussian blurring (GB)	Kernel size (K_s) = 5, standard deviation (σ) = 1.1
Median filtering (MF)	Kernel size (K_s) = 5
Resampling (RS)	Scaling = 1.5, bilinear interpolation
Noise addition (AWGN)	AWGN with standard deviation (σ) = 2
Gamma correction (GC)	Parameter (γ) = 1.5
JPEG compression (JPEG)	QF = 70

forged pixels, per_{fr}, in the computed binary mask, $\mathbf{I}_B$. The test image $\mathbf{I}$ is decided to be forged if per_{fr} is greater than a predefined threshold th_{img}. Formally, $\mathbf{I}$ is decided as forged, if

$$per_{fr} > th_{img}. \tag{5.6}$$

Here, the threshold th_{img} indicates the percentage of pixels to be classified as forged in order to decide $\mathbf{I}$ to be forged.

5.4 Experimental Results

For experimentation, a dataset was created using the unprocessed raw images taken from the Dresden Image Database [129]. The database contains more than 14,000 images with resolutions of about 2000×3000 captured by 73 different digital cameras. A set of 1566 raw images were compressed in JPEG format with a QF of 100. The green channel of the images is considered in the experimentation. We cropped image patches of size 150×150 from these images, resulting in 114,000 unaltered (UA) image patches.

The proposed system was implemented using the Keras [103] deep learning library on a Tesla K20c GPU with 5 GB of RAM. The Nadam optimizer [130] was used with the parameters set as: $learning\ rate(\eta) = 0.002$, $momentum(\mu) = 0.002$ and $decay = 0.005$ and $regularization\ term(\lambda) = 0.0001$. The learning rate decay technique is used for reducing the fluctuations in convergence [131]. The training batch size was set to 16 images. We have used the batch normalization technique [102] as it helps in achieving faster convergence and higher generalization accuracy.

5.4.1 Manipulation Detection Results

A series of experiments were carried out to test the performance of the proposed siamese network in detecting/discriminating different image editing operations. For this, six different versions of the 114,000 UA patches are created by editing them with the following operations: Gaussian blurring (GB), median filtering (MF), resampling (RS), corrupting with the additive white Gaussian noise (AWGN), gamma correction (GC), and JPEG compression (JPEG). The details of the manipulations are listed in Table 5.1. This way, we obtain 114,000 patches from the unaltered as well as each of the editing operations. The experiments are described below.

5.4.1.1 Discrimination of Different Image Editing Operations

In the first experiment, we test the ability of the proposed method in detecting/discriminating different image editing operations. For this, we have trained the siamese network on SP and DP image pairs, where the images come from seven different classes: UA, GB, MF, JPEG, GC, AWGN and RS. We randomly selected 40,000 patches from each class to create the training set. 500,000 SP pairs of patches are randomly selected with both patches of a pair coming from the same class (*i.e.*, both patches of a pair come from one of the seven manipulations). Similarly 500,000 DP pairs are randomly selected, with one patch of the pair coming from one of the seven classes and the other patch coming from a different class (*e.g.*, if one patch of pair comes from GB class, the other may come from MF class). To monitor the classification performance of the network during training, we apply it on a validation set that contains 10,000 SP pairs and 10,000 DP pairs that are not in the training set.

We monitored the losses and accuracies in the training process and stopped when the network converged. It was observed that after about 8,000 iterations, the training loss and training accuracy start saturating, indicating the convergence of the network. The validation accuracy also reaches about 99% after 8,000 iterations and saturates. The training process is stopped at 10,000 iterations and the final parameters of the model are saved for future use. To evaluate the performance of the model, the trained model was tested on the test set, which consists of 50,000 SP pairs and 50,000 DP pairs. The test SP and DP pairs are also created in the same manner as the training pairs are created. Note that the test image patches also come from the same seven

classes only. On this test set, the model achieves an accuracy of 99.58%. It shows that the proposed siamese network can discriminate the different types of image editing operations.

In the second experiment, we check the ability of the proposed method to classify SP/DP pairs when only two manipulations are considered. In this case, we consider two manipulations at a time out of the five classes, UA, GB, MF, JPEG, RS, created in the previous experiment. For example, the SP/DP pairs for the manipulation pair UA vs. GB are created by considering the images coming from these two manipulations only. We consider 50, 000 SP and 50, 000 DP pairs for each manipulation pair. We have also checked the classification performance of the FS method [120] on these SP/DP pairs[1]. Table 5.2 shows the classification of the proposed and the FS methods. The proposed method achieves more than 99% classification accuracies for all the manipulation pairs. The FS method achieves 100% classification accuracies and outperforms the proposed method for 6 manipulation pairs. However, the proposed method outperforms the FS method for 3 manipulation pairs, *i.e.*, UA vs. GB, UA vs. MF, and MF vs. GB. For the manipulation pair MF vs. GB, the proposed method achieves an accuracy of 99.27%, whereas the FS method achieves only 95%.

Table 5.2: SP/DP pair classification accuracies achieved by the proposed and the FS methods when considering two manipulations at a time

Manipulation Pair	Proposed Method	Forensics Similarity [120]
UA vs. GB	**99.76%**	99.46%
UA vs. MF	**99.87%**	98.05%
UA vs. R	99.63%	**100.0%**
JPEG vs. AU	99.53%	**100.0%**
JPEG vs. GB	99.53%	**100.0%**
JPEG vs. MF	99.67%	**100.0%**
GB vs. R	99.52%	**100.0%**
MF vs. GB	**99.27%**	95.0%
MF vs. R	99.45%	**100.0%**

These results show that the proposed method can effectively differentiate image patches manipulated by different image editing operations.

5.4.1.2 Detection of different Image Editing Operations

An experiment is carried out to check the ability of the proposed method in classifying each of the seven manipulation types, including the UA class considered during training. This is

[1]We have used the source code and trained model weights provided by the authors.

Table 5.3: Confusion matrix for the classification of different manipulation classes

		Predicted Class						
		OR	**GB**	**MF**	RS	**AWGN**	**GC**	**JPEG**
True Class	**OR**	**99.63%**	0.00 %	0.09 %	0.08 %	0.04 %	0.11 %	0.03 %
	GB	0.00%	**99.53 %**	0.16 %	0.27 %	0.00 %	0.02 %	0.01 %
	MF	0.08%	0.09 %	**99.67 %**	0.08 %	0.00 %	0.07 %	0.00 %
	RS	0.10%	0.20 %	0.11 %	**99.52 %**	0.00 %	0.00 %	0.06 %
	AWGN	0.18 %	0.00 %	0.00 %	0.00 %	**99.63 %**	0.1 %	0.07 %
	GC	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	**99.91 %**	0.00 %
	JPEG	0.14 %	0.04 %	0.02 %	0.26 %	0.08 %	0.01 %	**99.45 %**

Table 5.4: Classification accuracies on different manipulation classes

Manipulation	**Proposed Method**	**Bayar and Stamm [50]**
UA	**99.63%**	99.01%
GB	**99.53%**	99.22%
MF	**99.67%**	99.34%
RS	**99.52%**	98.88%
AWGN	99.63%	**99.92%**
GC	**99.91%**	98.53%
JPEG	99.45%	**99.72%**

accomplished by using a one-shot classification approach, presented in Algorithm 5.1. The test set contains $50,000$ images from each of the seven classes. Table 5.3 shows the confusion matrix for the task of classification of the seven manipulations. It can be seen that the proposed method detects all the seven classes with high accuracies, achieving a maximum of 99.91% accuracy for gamma correction manipulation and a minimum of 99.45% for JPEG compression detection. For comparison, we have implemented Bayar and Stamm's method [50] and tested its performance on this test set. The size of the image patches used in this experiment is 150×150. The classification results are shown in Table 5.4. The proposed method classifies UA, GB, MF, RS, AWGN, GC, and JPEG manipulations with accuracies of 99.63%, 99.53%, 99.67%, 99.52%, 99.63%, 99.91%, and 99.45%, respectively, whereas Bayar and Stamm's method classified with accuracies of 99.01%, 99.22%, 99.34%, 98.88%, 99.92%, 98.53%, and 99.72%, respectively. These results show that the proposed siamese network outperforms Bayar and Stamm's method [50] on manipulations excepting the AWGN and JPEG classes. We believe that the superior performance of the proposed method is due to the ability of the siamese network in learning more discriminative features than a simple CNN.

Table 5.5: Classification accuracies on manipulations not present in the training stage

Manipulation	AWGN	GC	JPEG
Accuracy	96.61%	95.24%	97.91%

5.4.1.3 Generalization to Unseen Manipulations

a) An experiment is carried out to check the generalization ability of the proposed method in detecting manipulations not considered in the training phase. For this, we have trained the network on SP and DP pairs of patches, where the patches come from four classes, *i.e.*, UA, GB, MF, and RS. There are 500,000 training pairs of both SP and DP images, sampled from the four classes in the same way as explained in the first experiment. Once, the network is trained, we test it on a set of image patches coming from image editing operations not present in the training stage, *i.e.*, AWGN, GC, and JPEG. The test set includes 50,000 image patches from each of the three classes. To check the performance of the network in classifying these editing operations, we perform the one-shot classification by applying Algorithm 5.1. For this, we have one image patch from each of these three manipulation classes as references. We also have one reference image from each of the four classes used in the training stage. Table 5.5 shows the accuracies of the proposed method in detecting the patches coming from manipulations not seen in the training stage. It can be seen that the network classifies images coming from AWGN, GC, and JPEG classes with accuracies 96.61%, 95.24%, and 97.91% respectively. This shows the generalization capability of the proposed method to unknown types of image editing operations. This is an important advantage of the proposed method over Bayar and Stamm's method [50], as their method can only detect manipulations present in the training stage.

b) An experiment is carried out to see the generalization performance of the network in detecting editing operations with arbitrary values of the parameters. This is a practical scenario as any parameter value can be used while manipulating an image with a particular editing operation. A test set is created by manipulating the UA patches using the six manipulations with arbitrary values of the parameters, as shown in Table 5.6. This test set contains 50,000 images for each manipulation, where the values of the parameters for the manipulations are selected randomly. In this experiment, we use the pre-trained siamese network from the first experi-

Table 5.6: Editing operations with variour parameters considered in this work

Editing Operation	Detail
Gaussian blurring (GB)	σ = 1.1, 1.5, 2 for each K_s = 3 and 5
Median filtering (MF)	K_s = 3, 5, 7
Resampling (RS)	Scaling = 1.2, 1.5, 1.7, 2, bilinear interpolation
Noise addition (AWGN)	σ = 1.5, 1.7, 2
Gamma correction (GC)	Parameter (γ) = 1.5, 1.7, 2
JPEG compression (JPEG)	QF = 70, 80, 90

ment, which was trained on the UA class and six manipulations listed in Table 5.1 with a single parameter value for each manipulation. The manipulations are detected using the one-shot classification technique, where the reference image for each class is created using the manipulations listed in Table 5.1, *i.e.*, using a single parameter value for each manipulation. The classification accuracies are shown in Table 5.7. The network achieves a maximum accuracy of 95% for the MF class and a minimum accuracy of 85% for the GC class. This establishes the generalization power of the network in detecting manipulations with values of parameters other than the ones used during training.

Table 5.7: Genealization accuracies on single manipulations with arbitrary parameters

Manipulation	**GB**	**MF**	**RS**	**AWGN**	**GC**	**JPEG**
Accuracy	90.96%	95.70%	87.64%	90.42%	85.56%	90.44%

c) An experiment is carried out to test the generalization ability of the network in detecting multiple editing operations applied on a single image. The ability to detect/discriminate multiple manipulations in images is important from the forensics point of view. This is because i) it gives the information regarding the processing history of an image, and ii) it can expose forgeries, such as splicing, copy-move, and retouching, as the forged image parts are generally processed by multiple editing operations to make them look visually plausible. We have created six more versions of the dataset by manipulating each image patch present in the UA class by two subsequent editing operations. In this way, the following six versions of manipulations were created: Gaussian blurring-median filtering (GB-MF), median filtering-Gaussian blurring (MF-GB), Gaussian blurring-resampling (GB-RS), resampling-Gaussian blurring (RS-GB), median filtering-resampling (MF-RS), resampling-median filtering (RS-MF). Here, the manipulation A-B means an image patch is first manipulated using the editing operation A and then edited

Table 5.8: Genealization accuracies on double manipulations with seven training manipulation classes

Manipulation	GB-MF	MF-GB	GB-RS	RS-GB	MF-RS	RS-MF	Average
Accuracy	94.85%	93.60%	94.16%	90.42%	95.82%	95.53%	94.06%

using the operation B.

For this experiment, we use the network trained on images undergoing the single manipulation operations listed in Table 5.1. The double manipulations are then detected using the one-shot classification strategy by applying Algorithm 5.1. We assume to have one image from each of the double manipulation classes as required in the one-shot classification technique. The accuracies of the network on these double manipulation classes are listed in Table 5.8. The network classifies GB-MF, MF-GB, GB-RS, RS-GB, MF-RS, and RS-MF manipulations with accuracies 94.85%, 93.60%, 94.16%, 90.42%, 95.82%, and 95.53%, respectively. This shows that even though the network was trained only to detect single editing operations, it can generalize well to double manipulations detection with accuracies of more than 90%.

d) In this experiment, we test the generalization ability of the proposed method in detecting single and multiple manipulations after re-compression. For this, we create another version of each of the seven single manipulations and six double manipulations by re-compressing them using JPEG compression with a QF of 90. More specifically, this new version contains the manipulations as follows: OR-JPEG, GB-JPEG, MF-JPEG, RS-JPEG, AWGN-JPEG, GC-JPEG, GB-MF-JPEG, MF-GB-JPEG, GB-RS-JPEG, RS-GB-JPEG, MF-RS-JPEG, RS-MF-JPEG. Then, these manipulations are detected through one-shot classification by applying the pre-trained siamese network from the first experiment. Table 5.9 shows the detection accuracies for these manipulations. The network is able to achieve an average accuracy of 82.91% with a maximum accuracy of 87.86% for MF-JPEG manipulation and a minimum accuracy of 79.29% for MF-GB-JPEG manipulation. It is observed that the accuracies achieved by the proposed method on the double manipulations followed re-compression are lower than those

Table 5.9: Generalization accuracies on different manipulations followed by re-compression with a QF of 90

UA-JPEG	GB-JPEG	MF-JPEG	RS-JPEG	AWGN-JPEG	GC-JPEG	GB-MF-JPEG	MF-GB-JPEG	GB-RS-JPEG	RS-GB-JPEG	MF-RS-JPEG	RS-MF-JPEG
84.23%	86.46%	87.86%	84.23%	84.59%	83.94%	79.61%	79.29%	81.83%	79.46%	81.74%	81.76%

on the single manipulations followed by re-compression. This is expected as the application of the second manipulation may remove the trace of the first manipulation. When the images are re-compressed with a QF of 70, the average detection accuracy over all the 12 classes drops down to 75.25%. These results suggest that the siamese network, trained on singly manipulated images, can detect the singly and doubly manipulated images after re-compression with decent accuracies.

5.4.1.4 Dependence of Generalization Accuracy on Number of Training Manipulations

An experiment is performed to check the effect of varying the number of manipulations present in the training stage on the generalization accuracy of the network. For this, we use the siamese network trained on pairs of image patches coming from the UA class and three single manipulation classes, namely GB, MF, and RS. Then, the network is used to detect the six double manipulations, as mentioned above, using the one-shot classification technique. The detection accuracies are listed in Table 5.10. A comparison of Table 5.8 and Table 5.10 shows that increasing the number of manipulations in the training stage helps the network to generalize more.

Table 5.10: Generalization accuracies on double manipulations with four single manipulation classes during training

Manipulation	**GB-MF**	**MF-GB**	**GB-RS**	**RS-GB**	**MF-RS**	**RS-MF**	**Average**
Accuracy	83.58%	83.94%	83.87%	84.06%	83.88%	83.35%	83.78%

5.4.1.5 Detection of Unknown Manipulations

In this experiment, we test the ability of the proposed method in differentiating an unknown manipulation from the manipulations present in the test gallery. In this case, we do not assume any knowledge about the processing operation, and hence we do not have any reference image to perform the one-shot classification. We create a new set of manipulated images by adding motion blur, with a length of 9 pixels in the horizontal direction, to the 50, 000 UA test patches. Then, we applied Algorithm 5.1 on this test set to see how the method differentiates this manipulation from those present in the test gallery. We found that the method decides these images to be manipulated by an unknown editing operation with an accuracy of 91%.

5.4.2 Selection of Hyper-parameters of the Proposed CNN

A set of experiments are performed by varying the hyper-parameters of the CNN to select the best performing architecture. In the first experiment, we have varied the pooling type, *i.e.*, max-pooling and average-pooling, used in the network and applied on the test set used in Section 5.1.1. Figure 5.3 shows the test accuracy with respect to training iteration for both pooling types. It is seen that average-pooling helps the network achieve higher test accuracy than max-pooling. Quantitatively, the networks with average-pooling and max-pooling achieve test accuracies of 99.58% and 98.9%, respectively. One possible explanation for this is that average-pooling takes into consideration all the values present in the pool window, which might help the network learn more accurate features than for max-pooling.

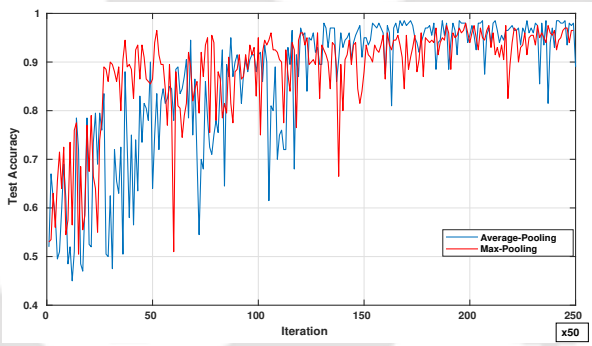


Figure 5.3: Testing accuracy versus training iteration for the network with max-pooling and average pooling.

Next, we experiment with different noise residual computation strategies in the first layer of the CNN. We tested the effect of employing fixed SRM filters and constrained filters in the first layer. In this study, we have used the 3 SRM filters used in [119] and three constrained filters. Using each of these sets of filters in the first layer, we train the siamese network on the dataset of seven single manipulations used in Section 5.4.1. We test the performance of the network on the UA images and the images modified using the six single manipulations listed in Table 5.1, and the six double manipulation operations, *i.e.*, GB-MF, MF-GB, GB-RS, RS-GB, MF-RS, RS-MF. Figure 5.4 shows the test accuracy with respect to the training iteration for the network with fixed SRM filters and constrained filters. From the figure, it is clear that the network with constrained filters converges faster than the one with fixed SRM filters. Table

5.11 shows the detection and generalization accuracies for the constrained and the fixed filter type. Column 2 and Column 3 in the table represent the average detection accuracies of the method on the single manipulation classes and the double manipulation classes, respectively. The method achieves the average accuracies of 98.12% and 99.40% by using the fixed SRM filters and the constrained convolution filters, respectively. In the case of double manipulation detection, the method achieves the average accuracies of 92.01% and 93.64% by using the fixed SRM filters and the constrained convolution filters, respectively. These results show that the network with constrained filters achieves better test and generalization accuracy than the one that uses the fixed SRM filters.

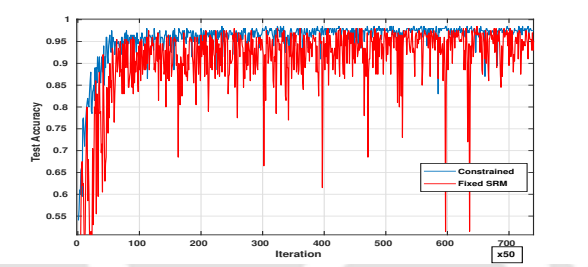


Figure 5.4: Testing accuracy versus training iteration for the network with constrained versus fixed SRM filters.

Table 5.11: Generalization accuracies of the proposed method with different types filters in the first layer of the CNN.

Filters	Test Accuracy	Generalization Accuracy
Fixed SRM	98.12%	92.01%
Constrained	99.40%	93.64%

5.4.3 Forgery Localization and Detection Results

A set of experiments are conducted to test the performance of the proposed forgery detection method. We show the localization results on three forgery types, namely splicing, copy-move and retouching.

For showing localization ability on real-life splicing forgeries, we use the DSO-1 [9] and the NIST-16 [132] datasets. The DSO-1 dataset is one of the popular datasets for splicing detection and contains 100 realistic spliced images of resolution 2048×1536 pixels, along with ground truth binary masks representing authentic and spliced parts. The NIST-16 dataset contains 564

high resolution forged images along with the corresponding binary masks. We have computed the $F1$-score and the *Matthews correlation coefficient* (MCC) [133] to quantitatively assess the forgery localization ability of the proposed method. These two measures give pixel-level forgery detection score. For the computation of these two measures, a binary mask is created for each test image by classifying the pixels in the image using Algorithm 5.2. Once the ground truth and the computed binary masks are available, the $F1$ and the MCC measures are computed as follows:

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{5.7}$$

and

$$MCC = \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \tag{5.8}$$

where TP, TN, FP, FN represent the true positive, the true negative, the false positive, and the false negative counts, respectively. We consider the forged pixels as the positive class and the authentic pixels as the negative class. Therefore, true positives mean forged pixels classified correctly as forged, true negatives mean authentic pixels correctly classified as authentic, false positives mean authentic pixels wrongly classified as forged, and false negatives mean forged pixels misclassified as authentic.

To show the relative merits of the proposed method with respect to the existing splicing detection methods, we have compared it with the following techniques: Noiseprint [52], MFCN [32], NOI1 [124], CFA2 [47], DCT [122], BLK [33], ELA [123], ADQ1 [121], CFA1 [46] and [34]. The source codes of these methods are made publicly available by Zampoglou *et al.* [134]. Table 5.12 shows the average $F1$-scores and the average MCC values achieved by the different methods on the DSO-1 images. The $F1$-scores and MCC values of the existing methods are taken from [32]. In [32] and [119], the authors varied the threshold and selected the one which corresponds to the highest value of $F1$ and MCC for each image. For a fair comparison, we have also followed the same testing protocol in this work. As can be seen in the table, the proposed method achieves an average $F1$-score of 0.4790 and an average MCC value of 0.4211. On the other hand, the state-of-the-art method, MFCN, achieves the average $F1$-score and the average MCC value of 0.4795 and 0.4074, respectively. Therefore, in terms

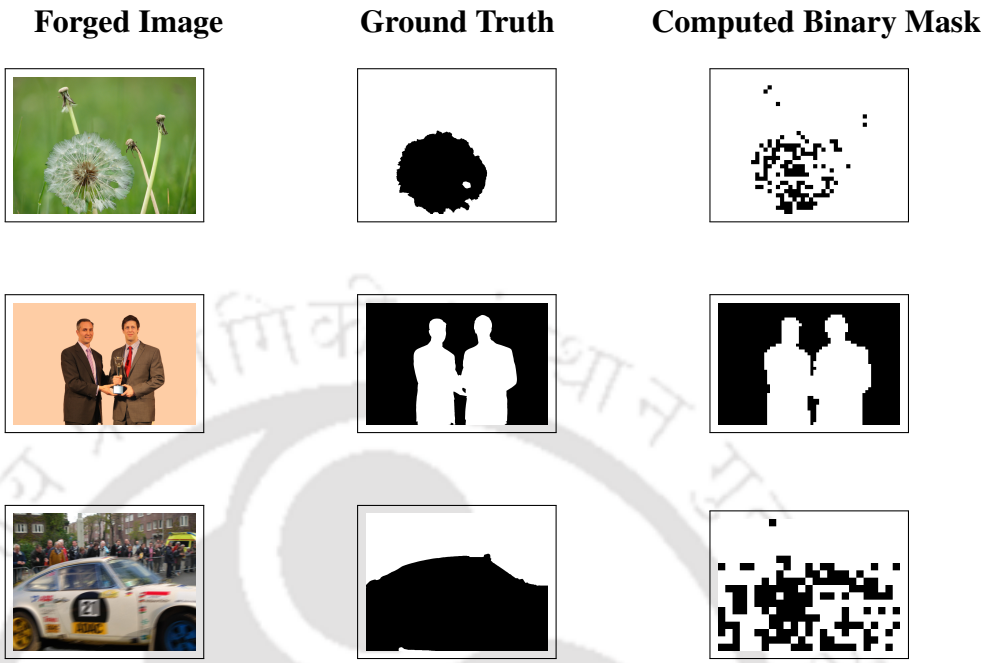


Figure 5.5: Performance of the proposed method in localizing forgeries in NIST-16 dataset. Each row shows a forged image, the ground truth binary mask and computed binary mask.

of *MCC* measure, the proposed method outperforms the existing methods shown in Table 5.12 on the DSO-1 dataset. Although the proposed method could not outperform MFCN in terms of $F1$-score, it outperforms all the other existing methods. It should be noted that MFCN is explicitly trained on real splicing forgeries, whereas the proposed method is trained only to differentiate/detect different types of image editing operations. We further applied the method to the authentic 100 images of the DSO-1 dataset. On these images, the proposed method achieves an average $F1$-score of 0.9901, which shows that the method can reliably differentiate authentic images from forged ones. Table 5.13 shows the average $F1$-scores and the average *MCC* values achieved by the different methods on the NIST-16 dataset [135]. The $F1$-scores and *MCC* values of the existing methods are taken from [52]. It can be seen in Table 5.13 that the proposed method achieves an average $F1$-score of 0.2916 and an average *MCC* value of 0.2901. However, the best performing method, Noiseprint [52], achieves an average $F1$-score of 0.395 and an average *MCC* value of 0.387. Although the proposed method does not outperform Noiseprint and NOI2, it outperforms the other methods in terms of both $F1$-score and *MCC* value.

Table 5.12: $F1$-scores and MCC values achieved by the proposed and the existing methods on DSO-1 dataset.

Method	F1	MCC
Proposed	0.4790	**0.4211**
MFCN	**0.4795**	0.4074
NOI1	0.3430	0.2454
CFA2	0.3124	0.1976
DCT	0.3066	0.1892
BLK	0.3069	0.1768
ELA	0.2756	0.1111
ADQ1	0.2943	0.1493
CFA1	0.2932	0.1614
NOI2	0.3155	0.1919

Table 5.13: $F1$-scores and MCC values achieved by the proposed and the existing methods on NIST-16 dataset.

Method	F1	MCC
Proposed	0.292	0.290
Noiseprint	**0.395**	**0.387**
NOI1	0.260	0.235
CFA2	0.227	0.184
DCT	0.234	0.195
BLK	0.233	0.204
ELA	0.184	0.145
ADQ1	0.275	0.262
CFA1	0.225	0.185
NOI2	0.314	0.296

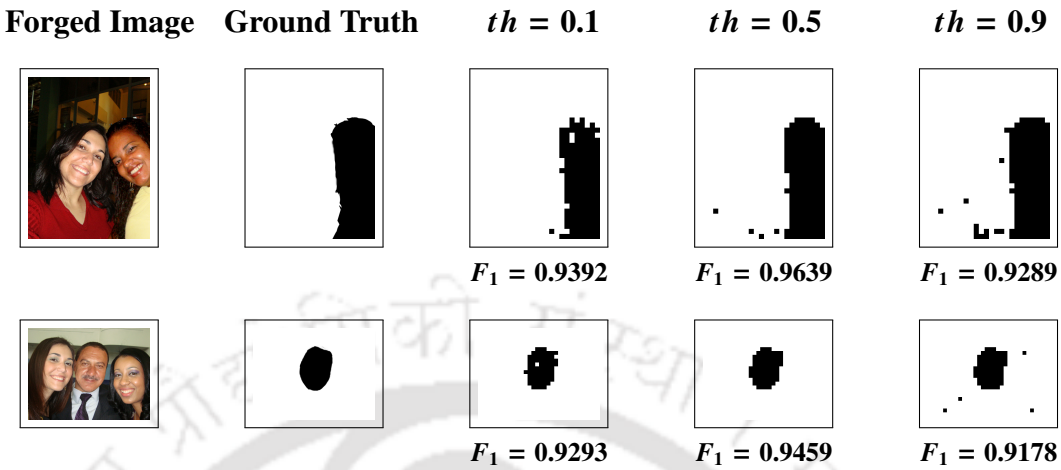


Figure 5.6: Effect of varying the threshold th in the splicing localization accuracy of the proposed method. Each row shows a forged image along with the ground truth binary mask image, and three computed binary mask images corresponding to thresholds $th = 0.1$, $th = 0.5$, and $th = 0.9$, respectively.

We have also carried out an experiment to see the sensitivity of the proposed method to the threshold th used to compute the output binary mask. Figure 5.6 shows the effect of varying the threshold in the $F1$-score on two spliced images. From the figure, it is clear that although the false positive and the false negative vary with the threshold, the overall forged area is detected in almost all the cases. Also, we have checked the performance of the method in detecting patches present in authentic images with different threshold values. Figure 5.7 shows the results on two authentic images with different thresholds. It can be observed that the method detects authentic patches as authentic with high precision even for threshold as high as $th = 0.9$ (which tends to increase false alarms more).

We also demonstrate the effectiveness of the proposed method in detecting and localizing retouched and copy-move forgeries. For this, we created a set of retouched and copy-move forgeries, and applied the proposed forgery localization method. Here we show the result for one example each from the retouched and the copy-move forgeries in Figure 5.8. The retouched image was created by applying the Gaussian blurring effect on the facial region of the girl in the image using GIMP software. The copy-move forgery was created by copying the rightmost car and then upsampling and pasting it in between the leftmost and the rightmost cars. The second and the third column show the ground truth binary mask and the computed binary mask along

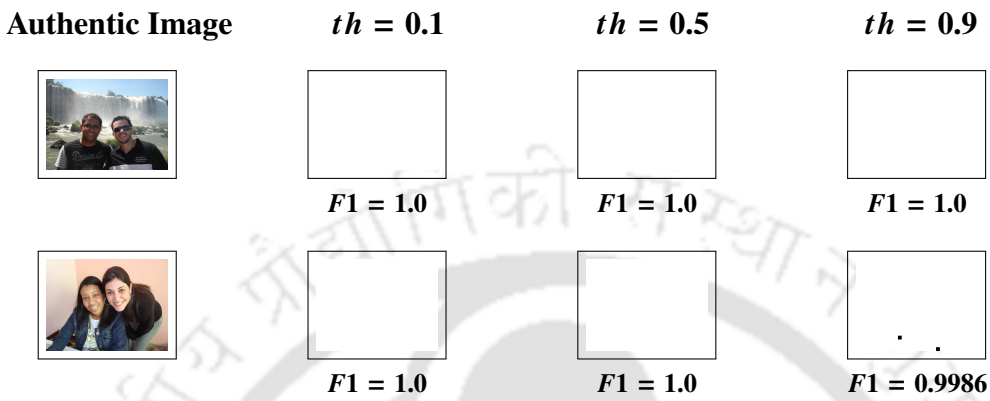


Figure 5.7: Effect of varying the threshold *th* in the false detection of forged patches in authentic images. Each row shows an authentic image along with the three binary images corresponding to thresholds $th = 0.1$, $th = 0.5$, and $th = 0.9$, respectively.

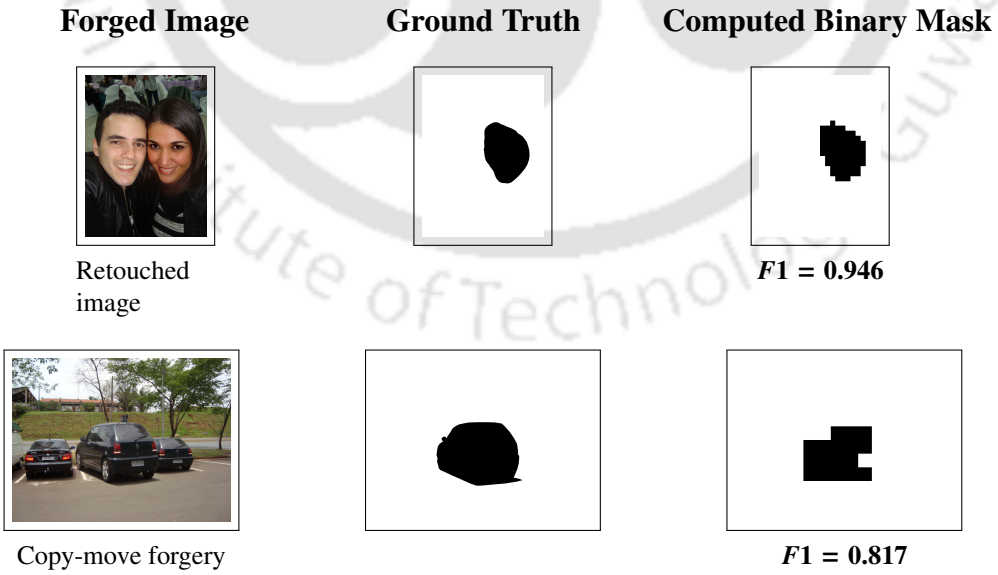


Figure 5.8: Ability of the proposed method to detect retouched and copy-move forgeries. Each row shows a forged image, the ground truth binary mask, and computed binary mask along with the $F1$-score.

with the corresponding $F1$-score. As can be seen in the figure, the proposed method could correctly localize the retouched region in the forged image with an $F1$-score of 0.946. In the case of the copy-move forgery, the proposed method could detect the forged region (middle car) with an $F1$-score of 0.817.

We have also carried out an experiment to show the efficacy of the proposed method in the image-level forgery detection task. To show the relative performance of the method, we have compared it with two state-of-the-art forgery detection methods: the FS method [120] and Bondi *et al.* [136] method. For this, we have computed the *area under the curve (AUC)* on DSO-1 dataset by varying the threshold in Equation (5.6). Table 5.14 shows the AUC values achieved by the proposed and state-of-the-art methods. The AUC values achieved by the state-of-the-art methods are taken from [120]. Although the proposed method could not outperform the FS method, it beats Bondi *et al.* by a large margin. One possible reason for the superior performance of the FS method is that it learns more subtle forensic traces, *i.e.*, camera traces, by training the siamese network for the camera model identification task. On the other hand, the proposed method learns manipulation traces by training the siamese network to differentiate different types of image processing operations. In case a forgery contains forged regions, that have not gone through any post-processing operation, the manipulation traces may not be able to expose the forgery.

Table 5.14: Image-level detection accuracies (in terms of AUCs) achieved by the proposed and the state-of-the-art methods on DSO-1 dataset.

Method	**AUC**
Proposed	0.86
Forensics Similarity [120]	**0.91**
Bondi *et al.* [136]	0.76

The above results show the ability of the proposed siamese network-based approach in detecting different types of image editing operations and forgeries and hence establish its effectiveness as universal image forensics.

5.5 Summary

This chapter proposed a novel image forensics method to detect/discriminate different types of manipulations carried out on images. The method employed deep siamese CNNs, which take a pair of image patches as inputs and decide whether they are SP or DP. Unlike the existing methods, which detect images that are manipulated by some fixed types of manipulations, the proposed method checks whether two image patches are processed through the same operation or not. For this, a distance metric is learned, which gives a high value if the pairs of patches are manipulated by different editing operations, and a low value if they are manipulated by the same editing operation. Because of this, the network could learn more generic manipulation features than simple CNN-based methods, which helped it to generalize to manipulations not considered in the training stage. For detecting the manipulations not considered in the training stage, the one-shot classification technique is employed. Based on this manipulation detection technique, a splicing detection and localization method is proposed. For localizing the splicing forgeries present in an image, the image is divided into a number of overlapping patches. The pair-wise classifications of the patches are then utilized to detect and localize the forgery. The experimental results show the efficacy of the proposed method in detecting/differentiating different manipulations and detecting real-life splicing forgeries.

6

Two-stream Encoder-Decoder Network for Localizing Image Forgeries

The method proposed in Chapter 5 focused on developing a universal image forensics method that can detect various image editing operations carried out on images using a siamese CNN. A forgery detection and localization method is also proposed using the siamese network. The method could outperform the state-of-the-art methods in the detection of image editing operations. However, the forgery localization performance of the proposed method was below that of the state-of-the-art. This is because the method utilizes only the image editing operation-related features for detecting the forged region. Further, the method was not trained specifically for localizing the forgeries. This chapter addresses this drawback and proposes a forensics method that is specifically designed and trained for localizing various types of forgeries in a single framework.

A perfectly curated forged image looks visually plausible; nevertheless, the forgery creation process introduces various *high-level* and *low-level* artefacts in the forged image. The high-level artefacts include unnatural contrast and edges, inconsistent out-of-focus blur, *etc*. The low-level artefacts are inconsistent noise levels in different parts of a forged image, traces related to double JPEG-compression, and different types of post-processing operations carried out on the forged regions, *etc*. Different image forensics methods utilize these high-level and low-level artefacts to expose various types of forgeries.

Most of the previous forensics methods have focused on detecting a single type of forgery, *e.g.*, detecting splicing or copy-move forgeries. These works are based on some assumptions about the type of forgery. The methods developed for detecting splicing assume that the spliced parts will have certain features (*i.e.*, high-level or low-level artefacts) that will not match with those of the authentic parts. For example, in [41] and [18], the illumination-based high-level features and the camera sensor-based low-level features, respectively, are utilized to expose splicing forgery. The methods, aimed at detecting copy-move forgeries, assume the presence of cloned objects in the forged image. For example, Popescu and Farid [137] proposed to use a high-level image feature to find the duplicate regions present in a forged image.

Although the above-mentioned methods are effective in detecting a single type of forgery, they may not be able to detect other forgeries. This is because the methods developed to detect a

particular type of forgery make some assumptions about the forgery. To tackle this limitation, a number of methods [16], [6], [51], [17] have been proposed recently, which can detect multiple forgeries, such as splicing, copy-move, and content-removal, in a single framework. These methods are based on learning different high-level and low-level features using different deep learning-based techniques.

This chapter proposes a novel two-stream encoder-decoder network to detect and localize multiple forgeries in a single framework. The encoder of one stream learns to differentiate between the authentic and forged parts based on the high-level image features. The other stream learns the inconsistencies in the low-level image features between the authentic and forged regions present in forged images. The outputs of the decoders of the two streams are fused to produce the final prediction map for localizing the forgeries. The main contributions of the chapter are as follows:

1) The proposed method learns both the high-level and the low-level manipulation-related features in an encoder-decoder framework for pixel-wise forgery localization.

2) Instead of fusing the features learned by the encoders, we propose to perform the late-fusion of the outputs of the two decoder streams.

3) The whole method can be trained end-to-end without any human intervention and hence is a completely data-driven approach.

The rest of the chapter is organized as follows: Section 6.1 describes the existing forensics methods for detecting different types of forgeries and highlights the research gap. Section 6.2 presents the proposed forensics method. Section 6.3 discusses the experimental results and Section 6.4 presents a summary of the chapter.

6.1 Related Work and Research Gap

As mentioned in previous chapters, many methods have been proposed in the image forensics literature for the detection of different types of forgeries. The earlier approaches were aimed at detecting or localizing a specific type of forgery, *e.g.*, detecting splicing or copy-move

forgery. These methods were based on the detection of forgery-specific traces present in the manipulated images.

For the detection of splicing forgeries, traces such as the differences in the local noise-levels [34], [124] and the JPEG compression levels [121], [33], [122] at different locations in an image, the mismatch in the colour filter array (CFA) interpolation methods [46], [47], and the error levels between an image and its resaved JPEG version [123] are used as hand-crafted features. Recently, deep learning-based methods have been proposed, which learn the splicing-related traces by training convolutional neural networks (CNNs) [138], [139], [32], and siamese networks [140]. For example, Salloum *et al.* [32] proposed a multi-task fully-convolutional network (MFCN) to localize the forged regions present in a manipulated image. Cozzolino and Verdoliva [140] proposed to localize splicing forgeries by learning camera-related traces, through training a siamese network to differentiate between image patches coming from the same and different images.

For the detection of copy-move forgery, visual features are computed from different parts of an image and checked for the presence of duplicate regions [137]. Ardizzone *et al.* [141] proposed to detect cloned regions by matching triangles of keypoints. Wu *et al.* [142] proposed a two-branch CNN, where one branch learns to check for cloned objects based on visual similarities, and the other branch learns to check for manipulated regions based on visual artifacts.

Although the above methods can detect/localize splicing and copy-move forgeries effectively, they have the limitation of being able to detect a single type of forgery only. However, in real forensics scenarios, the type of forgery is generally unknown beforehand. There might be more than one type of forgeries present in a single manipulated image. These concerns necessitate the development of forensics techniques that can detect different types of forgeries in a single framework.

The image forensics community is currently focusing on developing deep learning-based methods to detect and localize multiple forgeries in a single framework. Zhou *et al.* [16] proposed a two-stream faster R-CNN network (RGB-N) for localizing different forgeries. One stream of the network learns the high-level manipulation-related features, while the other stream

learns the low-level manipulation-related features. Bappy *et al.* [6] proposed a two-stream method (LSTM-EnDec), where one stream applies an LSTM network [11] on the hand-crafted resampling features to produce the final low-level features, and the other stream learns the high-level features using a CNN-based encoder network. The resampling features, computed using the Radon transform [44] and the Laplacian filters [143], help detect different types of operations carried out on the forged regions while creating a forgery. The encoder network learns various high-level manipulation-related traces, such as unnatural contrast. Finally, the features of both streams are concatenated, and then a single decoder network is applied to produce a pixel-wise prediction map. Wu *et al.* [51] proposed another deep learning-based method (ManTra Net) that first extracts image manipulation trace-related features from a test image and then checks the consistency between the features extracted from different image parts to localize the forged regions. Kniaz *et al.* [17] have proposed a method, called mixed adversarial generators (MAG), where a discriminative segmentation model is trained in a mixed adversarial setting using a GAN [12] to localize different types of forgeries.

The state-of-the-art methods [16], [6] have shown that the fusion of high-level and low-level manipulation-related traces helps in detecting and localizing different forgeries more effectively. For example, Zhou *et al.* [16] showed the effectiveness of the fusion of the high-level and the low-level features, learned from the training examples, in an R-CNN framework. However, the method can give only the bounding box-level localization of the forged regions, and hence not a true pixel-wise localization. The bounding box-level localization may include many authentic pixels inside the forged bounding box depending upon the shape of the forged regions. The LSTM-EnDec fuses the low-level features, computed using Radon transform and LSTM network, and the high-level features, computed using the CNN-based encoder network, for pixel-wise localization of forged regions. However, as the low-level manipulation traces are hand-crafted features, they may not be optimal for the forgery localization task. To address the above issues, this chapter proposes to employ a two-stream encoder-decoder network that learns both the low-level and high-level manipulation-related traces automatically from the training images and can perform pixel-wise forgery localization.

6.2 Proposed Method

The proposed method aims at localizing forgeries present in a manipulated image through a two-stream encoder-decoder neural network. One stream of the encoder-decoder network is the *image-stream*, which learns the high-level manipulation traces present in forged images. The other stream is the *noise-stream*, which learns the low-level manipulation traces from the noise residuals computed from the input image. The motivation for employing a two-stream network comes from the nature of artefacts present in a forged image. When an image is manipulated to create forgeries, such as splicing, different types of artefacts are introduced in the forged image. These artefacts can be broadly divided into the following two categories:

(i) High-level artefacts: The high-level artefacts, generally introduced in a forged image, include artificial edges, unnatural contrasts, inconsistent blur, *etc*. For example, the forged regions in a splicing or a copy-move forgery may have slightly higher contrasts than the rest of the image. When an object is copied and pasted on a different location of an image, the edges around the pasted object tend to be different from the natural object edges.

(ii) Low-level artefacts: The low-level artefacts present in forged images are the inconsistencies in the noise levels in the forged and the authentic regions. For example, in a spliced image, the spliced regions may come from different images, and possibly with noise levels different from the rest of the image. Also, the application of various image editing operations, such as Gaussian blurring and contrast enhancement, changes the local dependencies between the neighbouring pixels in unique ways. This, in turn, modifies the noise residuals present in the images distinctively for different editing operations. In a forged image, the forged regions are generally edited using different post-processing operations to make it look visually plausible. Therefore, the application of different editing operations on a forged image also contributes to the low-level artefacts.

The proposed two-stream encoder-decoder network learns the features related to both the high-level and the low-level manipulation traces at the encoder side and upsamples these features to produce dense feature maps at the decoder side. The dense feature maps from both the

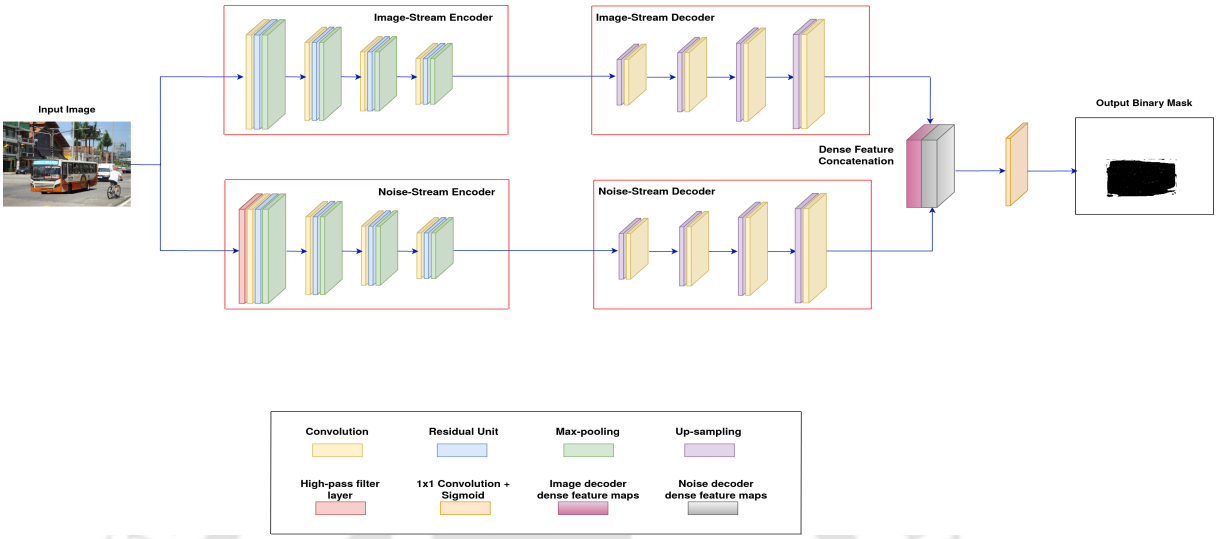


Figure 6.1: Block diagram of the proposed two-stream encoder-decoder network. The encoder in the image-stream learns high-level manipulation traces, such as artificial contrast. The encoder in the noise-stream learns the low-level traces, such as noise inconsistency, by employing a high-pass filter layer at the beginning of the network. The decoders in both the streams upsample the coarse feature maps of the encoders to produce dense feature maps, which are then concatenated and fed to a 1×1 sigmoidal convolutional layer for performing the pixel-wise classification.

streams are concatenated and fed to another convolution layer to produce the output prediction map, representing the forged and authentic pixels.

Figure 6.1 shows the block diagram of the proposed network. As shown in the figure, there are two encoder-decoder streams in the proposed network, namely the image-stream encoder-decoder (ISED) and the noise-stream encoder-decoder (NSED). Given a test image as input, it is processed by both the ISED and the NSED. The encoder in each stream performs a series of convolution, non-linear mapping, and max-pooling operations on the input image and produces the coarse feature maps. The coarse feature maps capture the relevant information about the high-level and the low-level manipulation-related traces. The corresponding decoder for each stream then performs upsampling and convolution operations on the coarse feature maps to produce the dense feature maps. The dense feature maps are of the same resolution as the input image and helpful for pixel-wise classification. The output dense feature maps of both stream decoders are concatenated and fed to a convolutional layer with a kernel of size 1×1 and sigmoid activation to produce the pixel-wise probability map. The pixel-wise probability

map gives the probability of each pixel belonging to the authentic and the forged classes.

Both the streams of the encoder-decoder network are described below.

6.2.1 Image-Stream Encoder-Decoder (ISED)

The proposed ISED architecture is inspired by SegNet architecture [14]. As in SegNet, the encoder of the ISED is a fully-convolutional neural network (FCN) [15], *i.e.,* it is a CNN without the fully-connected layers. We employ the ResNet [98] architecture instead of the VGGNet [127] architecture, used in the encoder of the SegNet. This is motivated by the fact that ResNet is easier to train than VGGNet [98] due to the presence of skip connections between convolutional layers. These help the ResNet propagate the gradients towards the initial layer without vanishing. As the aim of this stream is to learn the high-level manipulation traces that are left behind by different forgeries, the ISED operates on the RGB values of the input image. We have experimented with various numbers of the convolutional layers and residual blocks for the encoder of the ISED. From the experimental results, we fixed the architecture that gives the optimum train-test accuracies. It consists of 4 convolutional layers, 4 residual blocks, and 4 max-pooling layers. The architectures with more number of convolutional and residual layers overfitted the training data, and the ones with fewer layers could not achieve the expected training accuracies, *i.e.,* they underfitted the training data. At each convolutional layer of the proposed encoder, the ReLU activation and the batch normalization technique are used. Each residual block has 3 convolutional layers along with the ReLU activation and the batch normalization applied after each convolution operation. The output of each residual block is downsampled using the max-pooling operation. The sizes of the kernels in the convolutional and the max-pooling layers are set as 3×3 and 2×2, respectively. The number of filters in the 4 convolutional layers (and 4 residual blocks) are 32, 64, 128, and 256, respectively. The encoder takes an image of size 256×256×3 as the input and produces coarse feature maps of size 16×16× 256 as the output. An input image of any other size is first resized to $256 \times 256 \times 3$. The high-level artefacts present in forged images, such as inconsistent blur, artificial edges, unnatural contrast, will be affected by the image prefiltering operation in downsampling. For example, artificial edges present in the forged images will be smoothened by the downsampling operation,

such as bilinear averaging. However, we have trained the proposed network on images that are downsized to 256×256 from larger sizes. Because of this, the network learns the high-level artefacts present on the resized images. Also, the deep learning-based methods have proven to perform well on classifying small resolution images, such as 32×32 [144]. The high-level artefact learning task of the proposed method is similar to the learning in other computer vision tasks, such as object classification and segmentation. We have experimented with input size smaller than 256×256, namely 128×128, but the performance achieved was poor. The possible reason is that as the input image is downsized by a larger factor, the effect of prefiltering operation due to downsampling will also be larger. We did not perform any experiment with an input size bigger than 256×256 due to computational memory constraints. However, we believe that the performance of the network will be better as the resizing factor comes close to 1 due to the minimum amount of prefiltering operation being carried out on the input image.

The decoder takes the coarse feature maps as inputs and performs upsampling and convolution operations to produce dense feature maps to accommodate pixel-wise classification into forged and authentic classes. Following the symmetry in the encoder and decoder architecture as in SegNet [14], we also employed the same number of upsampling and convolutional layers in the decoder side as in the encoder side. The proposed decoder has 4 upsampling layers and 4 convolutional layers. The upsampling layers upsample the input feature maps and then convolve with trainable filters in the convolutional layers to produce the dense feature maps. In this work, we use an upsampling factor of 2 at each upsampling layer. The numbers of filters in the decoder convolutional layers are 32, 32, 16, and 16, respectively. The kernels in all the convolutional layers are of size 3×3. At each convolutional layer, the ReLU activation and the batch normalization are applied. The decoder produces the dense feature maps of size $256 \times 256 \times 32$.

6.2.2 Noise-Stream Encoder-Decoder (NSED)

This stream focuses on learning low-level manipulation-related features. The low-level image features are proven to be helpful in distinguishing different types of image manipulation operations [50]. These features are also shown to be able to localize different types of forgeries [16]. In splicing forgeries, the spliced parts are likely to have noise levels different from

those in the authentic regions. In copy-move and content-removal forgeries, the forged regions are generally passed through different types of post-processing operations to make them look visually undetectable. These post-processing operations change the local dependencies of the pixels present in the forged regions and hence introduce low-level artefacts.

The NSED takes the green colour channel of the input image to the encoder. The green channel is selected for extracting the low-level artefacts because there are 25% more green pixels than the red and the blue ones in the Bayar filters, which are used by most camera devices as the preferred CFA [45], [145]. Due to this reason, the green channel contains the least noise among the 3 colour channels. Hence, the low-level artefact extraction from this channel is more reliable [50]. The encoder in this stream is similar to the one in the ISED. However, it employs an additional convolutional layer, called the *high-pass filter (HPF) layer* [96], [50], [16], before the normal convolutional layer. The HPF layer computes the high-pass residuals from the input image to suppress the image contents and enhance the noise contents. The kernels in the HPF layer follow certain constraints to compute the high-pass residuals. Depending upon the constraints put on the kernels, the HPF layers available in the literature can be categorized as: (i) the *median filter residual* layer, (ii) the *constrained convolutional* layer, and (iii) the *SRM* filter layer. The median filter residual layer [96] employs a fixed set of weights in the kernels to compute the median filtered residuals as the output of the HPF layer. The constrained convolutional layer [50] extracts content-adaptive high-pass residuals by learning weights under a pre-defined constraint, given by

$$\begin{aligned} w_k^1(0,0) &= -1 \\ \text{and} \sum_{l,m\neq 0} w_k^1(l,m) &= 1 \end{aligned} \tag{6.1}$$

where $w_k^1(l,m)$ denotes the weight at position (l,m) of the kth filter and $w_k^1(0,0)$ denotes the weight at the center of the corresponding filter kernel. This constraint in Equation (6.1) is enforced on the weights of the constrained layer during the training stage. More specifically, at the end of each training iteration, the central filter weight of each constrained filter kernel is set to -1, and the non-central filter weights are normalized by dividing by their sum.

The SRM filter layer computes high pass residuals from the input image by applying a fixed set of SRM filters [116]. The constrained high-pass filters and the SRM filters have recently been shown to be effective in different forensics problems [16], [51]. In this work, we have experimented with the constrained convolutional and the SRM filter layers and found that the constrained high-pass filters perform better than the SRM filters. This is also intuitive as the constrained convolutional layer's content-adaptive filters learn the kernel weights from the training data itself. As the SRM filters are fixed filters, they may not be able to extract optimal features for forgery detection tasks. Hence, we have preferred to use the constrained filters for all the experiments reported in this chapter.

Therefore, the first layer of the encoder is a constrained convolutional layer with 3 filters of size 5×5. The rest of the encoder has 4 convolutional layers, 4 residual blocks, and 4 max-pooling layers. The sizes of the kernels in the normal convolutional and the max-pooling layers are 3×3 and 2×2, respectively. The numbers of filters in the normal convolutional layers are 32, 64, 128, and 256. An input image of size $256 \times 256 \times 1$ is fed to the encoder, and the feature map of size $16 \times 16 \times 256$ is produced as the output. For any other sizes, the input image is first resized to $256 \times 256 \times 1$. The low-level artefacts present in forged images, *i.e.*, the inconsistencies in the noise level and traces related to various image processing operations, are more affected by the image resizing operation compared to the high-level artefacts. In this work, we assume the forged regions to be sufficiently large. Then, there will be sufficiently many forged pixels in the resized image, which are obtained by performing the downsampling operation (*e.g.*, bilinear averaging) on the forged pixels only. Under these conditions, the low-level artefacts of the forged regions will still be available in the downsampled images. Since we train the proposed network on downsampled images, it will learn the additional low-level artefacts present in them.

The coarse feature maps produced by the encoder are then fed to the decoder, which performs upsampling and convolution to produce the dense feature maps. The decoder of this stream has the same architecture as the one in the image-stream. Therefore, the output of the NSED is the feature maps of size $256 \times 256 \times 32$

6.2.3 Feature Concatenation and Prediction Layer

The output dense feature maps of both the streams are concatenated and fed to a single convolutional layer to produce the final prediction. This way of concatenating features is known as the *late-fusion* technique. This is because the decoder of each stream processes the coarse features from the corresponding encoder to generate the dense features, which can be thought of as raw decisions about the authenticity of each pixel in the input images.

More specifically, the decoder outputs of both the streams are first concatenated to create the combined feature maps of size $256 \times 256 \times 64$. These feature maps are then fed to the final pixel-wise prediction layer, which is a 1×1 convolutional layer with sigmoid nonlinearity that produces class probability for each pixel. This layer produces a single probability map of size equal to that of the input image. The map provides the probability of each pixel being classified as either authentic or forged.

6.2.4 Learning

The encoder-decoder network parameters are learned in the training process by minimizing a loss function computed between the ground-truth and the predicted binary masks over a mini-batch of images. There are generally more authentic pixels than the forged ones in a forged image. The classical cross-entropy loss, which is computed over all the pixels, is more biased towards the authentic classes. This results in poor performance in classifying the forged pixels while maintaining good performance in classifying the authentic pixels. To handle this class-imbalance issue, we have experimented with two different loss functions, namely the weighted cross-entropy loss [146] and the Dice loss [147], given by the following equations:

$$\mathcal{L}_{wCE} = -\frac{1}{M} \sum_{c=1}^{2} \sum_{i=1}^{M} w_c g_c(i) \log p_c(i) \tag{6.2}$$

and

$$\mathcal{L}_{dice} = 1 - \sum_{c=1}^{2} \frac{2 \sum_{i=1}^{M} g_c(i) p_c(i)}{\sum_{i=1}^{M} g_c^2(i) + \sum_{i=1}^{M} p_c^2(i)} \tag{6.3}$$

where $\mathcal{L}_{wCE}$ is the weighted cross-entropy loss, $\mathcal{L}_{dice}$ is the Dice loss, w_c is the weighting factor for the class c, $g_c(i)$ and $p_c(i)$ are the ground-truth and predicted values for the pixel i being

class c, and M is the total number of pixels in the batch of images.

The weighted cross-entropy loss handles class-imbalance by assigning larger weights to the forged pixels and smaller weights to the authentic ones. We have used the median class weighting [146], where the weight of each class is computed as the ratio between the median of all the class frequencies and its own class frequency computed over the entire training dataset. The Dice loss maximizes the overlap between the predicted mask and the ground truth mask for each of the manipulated and the authentic classes. The value of the Dice loss always lies in the range $[0, 1]$ irrespective of the number of pixels in each class, and hence solves class-imbalance. We have experimentally found that the Dice loss outperforms the weighted cross-entropy in terms of generalization accuracy. This is reported in the experimental section. The superior performance of the Dice loss is also reported for different problems, *e.g.*, [148], where the class-imbalance is present. Hence, we propose to use the Dice loss as the preferred loss function for training the proposed network. We also tried the linear combination of the two loss functions, as in [148], but did not observe any improvement over the Dice loss.

The entire network is trained end-to-end on pairs of input images and the corresponding binary masks till it converges. Once the network is trained, we perform the inference on test images for predicting the forged pixels.

6.3 Experimental Results

To show the effectiveness of the proposed method in localizing different types of forgeries, experiments are performed on the following standard forgery datasets: NIST Nimble 2016 (NIST 16) [132] , NIST Media Forensics Challenge (MFC) 2018 [149], IEEE Forensics Challenge (IFC) [150], Columbia [151], DSO-1 [9], CASIA v1 and CASIA v2 [152] datasets. The details about the datasets are as follows:

- NIST16 dataset contains 564 forged images and their corresponding ground-truth binary masks covering the three types of manipulations: splicing, copy-move, and content-removal.

- MFC2018 dataset includes $4,541$ forged images and their corresponding ground-truth

binary masks.

- IFC dataset includes 450 forged images and their corresponding binary masks. The forged images are created using splicing and copy-move operations.
- Columbia dataset comprises 180 spliced images and their corresponding binary masks.
- DSO-1 dataset is a popular splicing dataset and contains 100 spliced forgeries along with their corresponding ground-truth binary masks.
- CASIA v1 and CASIA v2 are two splicing datasets containing 921 and 5, 123 spliced images, respectively.

The network is implemented in Keras with Tensorflow backend. We have used the *adam* optimizer with a learning rate of 0.00005.The training batch size is fixed to be 16 pairs of images and their corresponding binary masks.

We have used the following metrics to measure the forgery localization ability of the proposed method the $F1$-score, the *pixel-wise classification accuracy*, the *area over ROC curve (AUC)* and *per-class intersection-over-union (cIoU)* [17]. We have already discussed about the computation of the $F1$-score and the ROC in previous chapters. The pixel-wise classification accuracy is computed as follows:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.4}$$

where TP, TN, FP, FN represent the true positive, the true negative, the false positive, and the false negative counts, respectively. The cIoU is the per-class IoU computed for the forged class, and is given as

$$cIoU = \frac{|\mathrm{P} \cap \mathrm{G}|}{|\mathrm{P} \cup \mathrm{G}|} \tag{6.5}$$

where $|.|$, $\cap$ and $\cup$ represent the cardinality, intersection and union operations respectively. P and G are the predicted and the ground-truth masks, respectively.

6.3.1 Pre-training on Synthetic Dataset

There is no standard forgery dataset available, which includes a sufficiently large number of forged images to train deep neural networks. Therefore, we have first pre-trained the proposed network on a synthetic dataset. This dataset contains artificially created forgeries without any post-processing. The forged images present in this dataset imitate real-life forgeries and help the network learn different manipulation-related traces.

In this work, we have used the spliced images from the synthetic dataset created by Bappy *et al.* [6] from DRESDEN [129], COCO [153], and NIST16 datasets. To create the spliced images, the authors have copied objects from COCO dataset and pasted on different authentic images in DRESDEN and NIST16 datasets. From this dataset, we have used $13,470$ spliced images and their corresponding ground-truth binary masks. We have trained the proposed network on this synthetic dataset by using 90% for training and 10% for validation. Once the network converges on this dataset, we save the model for further fine-tuning and testing on different standard forgery datasets.

The proposed network gives a probability map as output, which is further thresholded to generate the binary map for localizing the forged regions. We have experimented with various thresholds in the range $0.2 - 0.8$ and have observed that the results are not affected by any noticeable amount. This is because the probability values corresponding to the authentic pixels are very close to 0, *i.e.* mostly below 0.01, and the probability values corresponding to forged regions are very close to 1, *i.e.*, mostly higher than 0.9. In this chapter, we have used the mid-value, *i.e.* 0.5, as the threshold for all the experiments.

To see the generalization ability of the pre-trained network, we have checked its performance on NIST16, IFC, Columbia, and DSO-1 datasets. The cIoU values on these datasets are presented in Table 6.1. We have also shown the cIoU values achieved by three recent methods, *i.e.* MFCN [32], ManTra Net [51] and MAG [17], on Columbia and DSO-1 datasets, for comparative analysis. The results reported for these three methods correspond to their respective models trained/fine-tuned on real-life forgeries. The proposed method achieves the cIoU values of 0.50, 0.47, 0.43, and 0.46 on NIST16, IFC, Columbia, and DSO-1 datasets, respectively.

On the other hand, on Columbia and DSO-1 datasets, MFCN [32] achieves the cIoU values of 0.42 and 0.37 respectively, ManTra Net [51] achieves the cIoU values of 0.58 and 0.38 respectively, and MAG [17] achieves the cIoU values of 0.77 and 0.56 respectively. Although the proposed pre-trained network could not outperform MAG, it outperformed MFCN on DSO-1 and Columbia datasets and ManTra Net on DSO-1 dataset. It is important to note that the results achieved by MFCN, Mantra Net, and MAG methods correspond to models trained on realistic forged images, *i.e.*, created manually. On the other hand, the results achieved by the proposed method correspond to the model trained on synthetically generated forgeries. These results indicate the ability of the proposed method to learn important forensics features from synthetic forged images that can localize real-life complex forgeries.

Table 6.1: cIoU values achieved by the proposed network, pre-trained on the synthetic dataset, and two other existing methods. '-' denotes the values that are not reported.

	CASIA v1	**NIST16**	**IFC**	**Columbia**	**DSO-1**
MFCN [32]	-	-	-	0.42	0.37
ManTra Net [51]	-	-	-	0.58	0.38
MAG [17]	-	-	-	0.77	0.56
Proposed (pre-trained)	0.55	0.50	0.47	0.51	0.46

6.3.2 Fine-tuning and Evaluation on Standard Forgery Datasets

The pre-trained network is fine-tuned on a training set created from NIST16, IFC, and CASIA v2 datasets. We have split NIST16 and IFC datasets into train (70%), validation (5%), and test (25%) subsets, following the same train-test split protocol as in [16], [6], and used all the spliced images of CASIA v2 for training, resulting in a total of $6,093$ images for training. Additionally, we have performed data augmentation by (1) flipping the images both horizontally and vertically, and (2) cropping the images randomly around the manipulated regions to get a zoomed-in version of the images. In this way, we have generated around $40,000$ training images, which help the network learn more diverse manipulation-related features and reduce overfitting. After the model is fine-tuned on these datasets, we have checked the test accuracies on the test images of the above-mentioned datasets using the aforementioned quantitative measures.

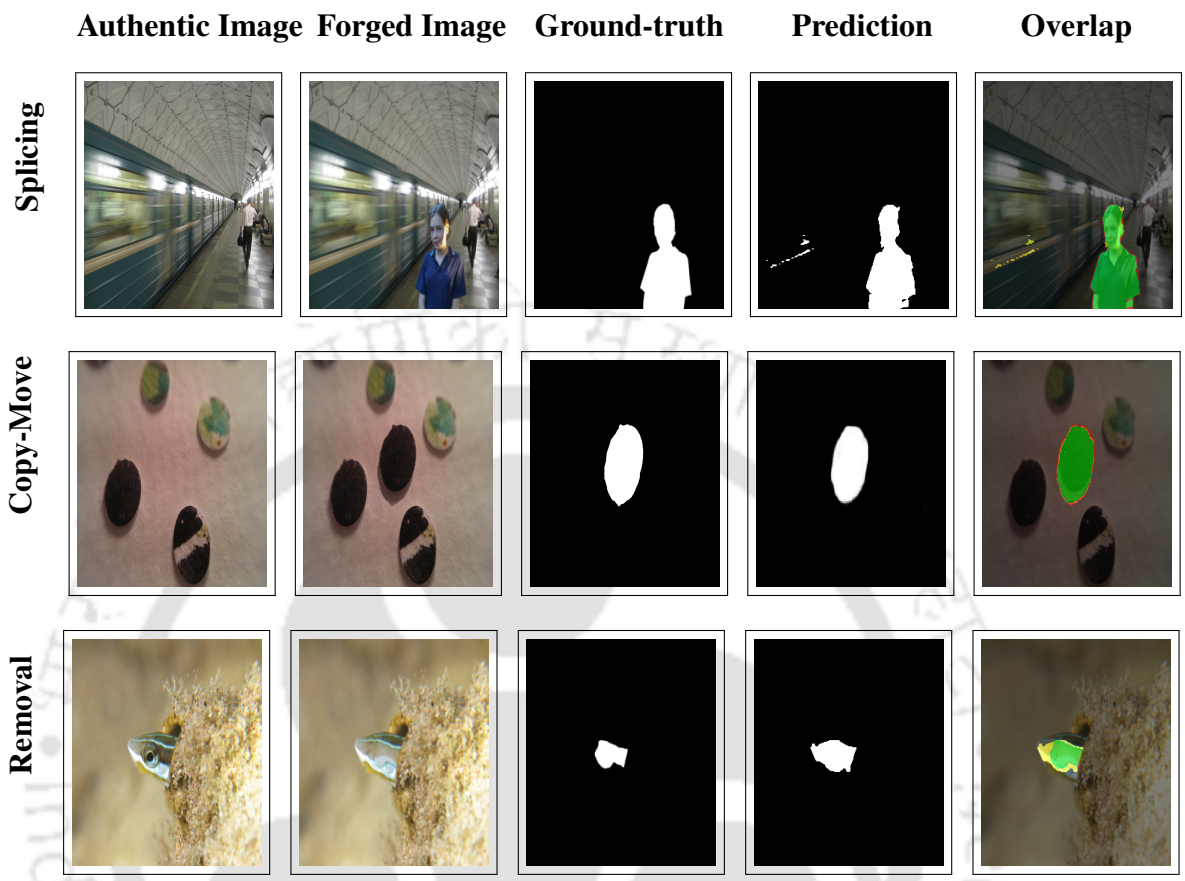


Figure 6.2: Forgery localization results of the proposed method for splicing, copy-move, and removal forgeries present in NIST16 dataset. The columns from the left show the authentic image which is used for creating the forgery, the forged image, the ground-truth binary mask, the predicted binary map, and the overlap of the ground-truth binary mask and the predicted binary map, respectively. The ground-truth, the prediction, and overlapped regions are represented by red, yellow, and green colours, respectively, on the overlap image.

A number of experiments are performed to show the forgery localization ability of the proposed method on various datasets containing different types of forgeries.

1) We show the localization ability of the proposed method on the three types of forgeries from NIST16 dataset. Figure 6.2 shows the localization results on one example image from each manipulation type, *i.e.*, splicing, copy-move, and content-removal. We have first computed the pixel-wise accuracies achieved by the proposed method on NIST16 and IFC datasets for quantitative analysis. We have also compared the performance of the proposed method and LSTM-EnDec, as this method also employs an encoder-decoder network along with an LSTM network. Table 6.2 shows the pixel-wise accuracies of the proposed method on these two datasets. It also shows the accuracies achieved by LSTM-EnDec [6] on these datasets. The

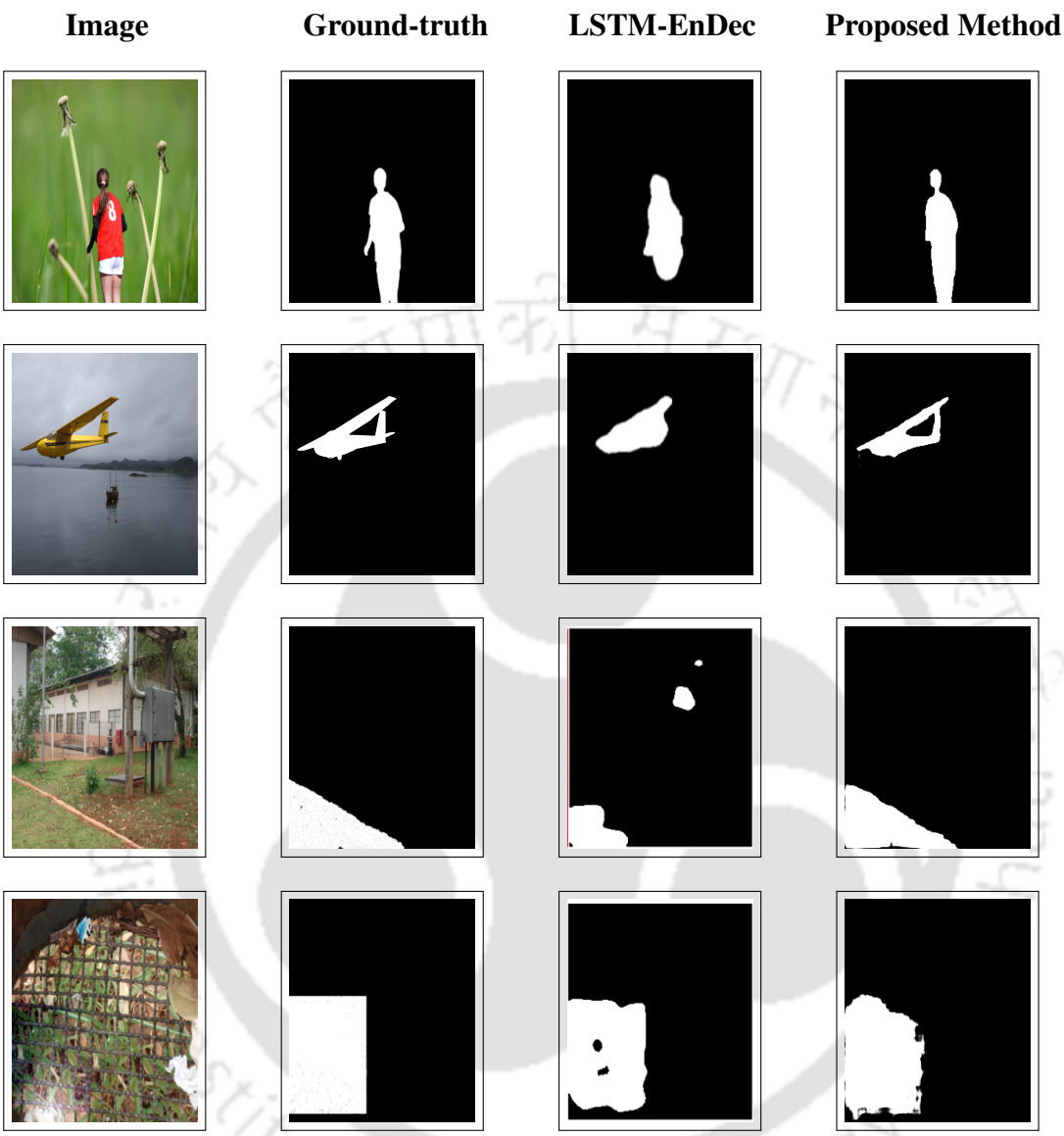


Figure 6.3: Examples of qualitative forgery localization results of LSTM-EnDec and the proposed method on NIST16 and IFC datasets. First two rows show the results on images from NIST16 and last two rows show the results on images from IFC dataset. The results of LSTM-EnDec shown in the third column are taken from [6].

proposed method achieves the pixel-wise accuracies of 95.74% and 92.32% on NIST16 and IFC datasets, respectively. On the other hand, LSTM-EnDec achieves accuracies of 94.80% and 91.19% on NIST16 and IFC datasets, respectively. These results quantitatively show the superior performance of the proposed method over LSTM-EnDec on these datasets. Figure 6.3 shows some of the qualitative results of LSTM-EnDec and the proposed method on NIST16 and IFC datasets. It can be seen that the proposed method can localize the forged regions better than LSTM-EnDec. The quantitative and qualitative results indicate the ability of the proposed

Table 6.2: Comparison of the performance of the proposed method with LSTM-EnDec [6] on two standard datasets in terms of pixel-wise accuracy.

	NIST16	**IFC**
LSTM-EnDec [6]	94.80%	91.19%
Proposed	**95.74**%	**92.32**%

method to learn more discriminative low-level features by employing an encoder network than the hand-crafted features proposed in LSTM-EnDec [6].

Table 6.3: cIoU values on DSO-1 and Columbia datasets. '-' denotes the values that are not available in the literature.

	DSO-1	**Columbia**	**MFC2018**
CFA1 [46]	0.33	0.44	-
NOI1 [124]	0.21	0.40	-
DCT [122]	0.24	0.41	-
MFCN [32]	0.37	0.42	-
ManTra Net [51]	0.38	0.58	-
MAG [17]	**0.56**	0.77	-
Proposed	0.52	**0.83**	0.49

2) To show the relative merits of the proposed method with respect to other existing forensics methods, we have considered the following methods: ELA [123], DCT [122], CFA1 [46], NOI1 [124], MFCN [32], RGB-N [16], ManTra Net [51], and MAG [17]. Table 6.3 shows the cIoU values achieved by the proposed and the competing methods on DSO-1, Columbia, and MFC2018 datasets. The cIoU values of the existing methods are taken from [17]. As can be seen in the table, the proposed method achieves the cIoU values of 0.52 and 0.83 on DSO-1 and Columbia, respectively, whereas the best performing method MAG achieves 0.56 and 0.77. Although MAG slightly outperforms the proposed method on DSO-1 dataset, it outperforms MAG on Columbia dataset by a large margin. On MFC2018 dataset, the proposed method achieves the cIoU value of 0.49. We could not compare this performance of the proposed method with state-of-the-art methods, as these methods have not reported experimental results on this dataset. Since these three datasets, *i.e.*, DSO-1, Columbia, and MFC2018, are not used in fine-tuning the proposed network, these analyses show the generalization ability of the proposed method to

unseen datasets.

Table 6.4: $F1$-scores and AUCs on three datasets, '-' denotes the values that are not available in the literature

	NIST16		CASIA v1		Columbia	
	F1	AUC	F1	AUC	F1	AUC
ELA [123]	0.24	0.43	0.21	0.61	0.47	0.58
NOI1 [124]	0.29	0.49	0.26	0.61	0.57	0.55
CFA1 [46]	0.17	0.50	0.21	0.52	0.47	0.72
MFCN [32]	0.57	-	**0.54**	-	0.57	-
RGB-N [16]	**0.72**	0.94	0.41	0.80	0.69	0.86
ManTra Net [51]	-	0.80	-	**0.82**	-	0.82
Proposed	0.62	**0.95**	0.41	0.81	**0.86**	**0.88**

Table 6.4 shows the performance of the proposed method in terms of the $F1$-score and the AUC value on three datasets. It also shows the performance of other existing forgery localization methods for comparisons. The $F1$-scores and the AUC values of the existing methods are taken from [16] and [51]. As shown in the table, the proposed method outperforms all the existing methods on Columbia dataset in terms of both measures. On NIST16 dataset, the proposed method is outperformed by RGB-N method in terms of the $F1$-score. However, in terms of the AUC value, the proposed method outperforms all the existing methods on NIST16 dataset. These results quantitatively show the superior performance of the proposed method in localizing forgeries over the state-of-the-art. We believe that the superior performance of the proposed network over the state-of-the-art methods is due to the ability to learn both the low-level and the high-level artefacts for pixel-wise forgery localization in a more effective way. Figure 6.4 shows two examples of forgery localization from DSO-1, IFC, CASIA v1, Columbia, and MFC2018 datasets. These results qualitatively show the ability of the proposed network in localizing different forgeries present in multiple datasets.

3) To see the robustness of the proposed method against JPEG compression, we have compressed the images in NIST16 and Columbia datasets with QFs 50, 70, and 90. Then, we have checked the performance of the method on these compressed versions of the datasets. Table 6.5 shows the cIoU values achieved by the proposed method on these versions. Although the

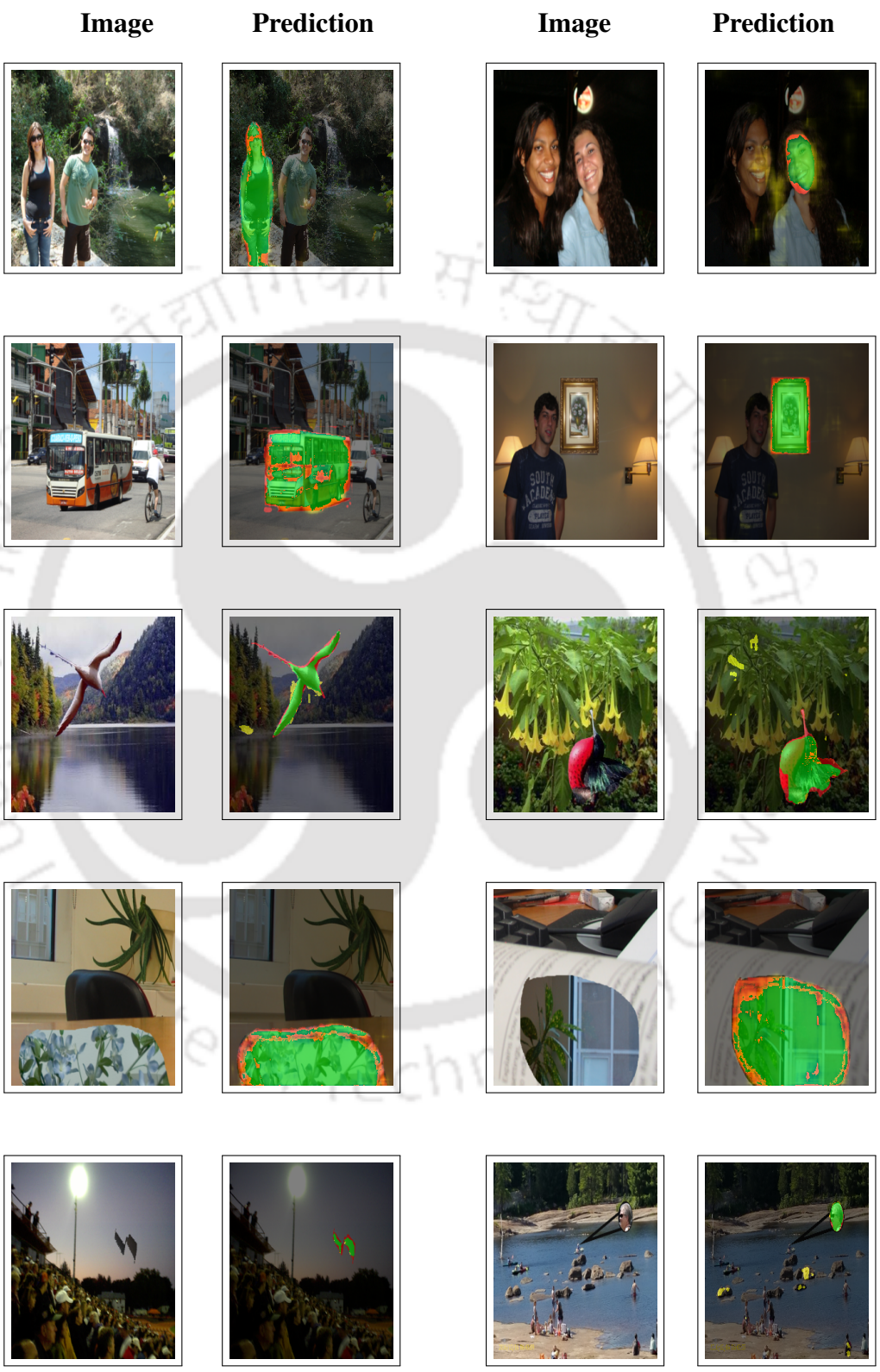


Figure 6.4: Qualitative results showing the localization ability of the proposed method on different datasets. The rows from the top are results from DSO-1, IFC, CASIA v1, Columbia, and MFC2018 respectively.

performance of the method degrades as the QF reduces, it can still achieve a decent cIoU score of at least 0.4 at a QF as low as 50. This is more than the cIoU values achieved by the non-deep learning methods reported in Table 6.4. The degradation of performance with respect to high JPEG compression (*i.e.*, low QF) is expected as most of the low-level image manipulation traces are lost when the image is compressed with a low QF.

Table 6.5: cIoU values for different compression levels.

Compression Level	NIST16	Columbia
QF_50	0.46	0.40
QF_70	0.47	0.44
QF_90	0.51	0.55
QF_100	**0.72**	**0.83**

4) We have also carried out an experiment to show the performance of the proposed method on pristine images. We have used the pristine images from DSO-1 dataset for this analysis. Figure 6.5 shows two authentic images and their corresponding predicted masks. We have also computed the cIoU values for the authentic images for a quantitative measure[1]. The proposed method achieved a cIoU of 0.965 on the authentic images of DSO-1. From the analysis, it can be argued that the proposed method does not produce many false positives in the case of pristine images.

6.3.3 Ablation Study

We have experimented with different network settings and loss functions to find out the best-performing one. Firstly, we have varied the number of encoders and decoders in the proposed method. More specifically, we have experimented with three network settings:

(i) NSED: It is a single noise-stream encoder-decoder network (shown in Figure 6.6(a)).

(ii) ISE-NSE-1-Dec: It employs two-stream encoders, *i.e.*, noise and image-streams, and then the features of both the streams are fused (early-fusion), and a single decoder is employed to compute the prediction (shown in Figure 6.6(b))

[1]The cIoU computation of the authentic images is done for the authentic class

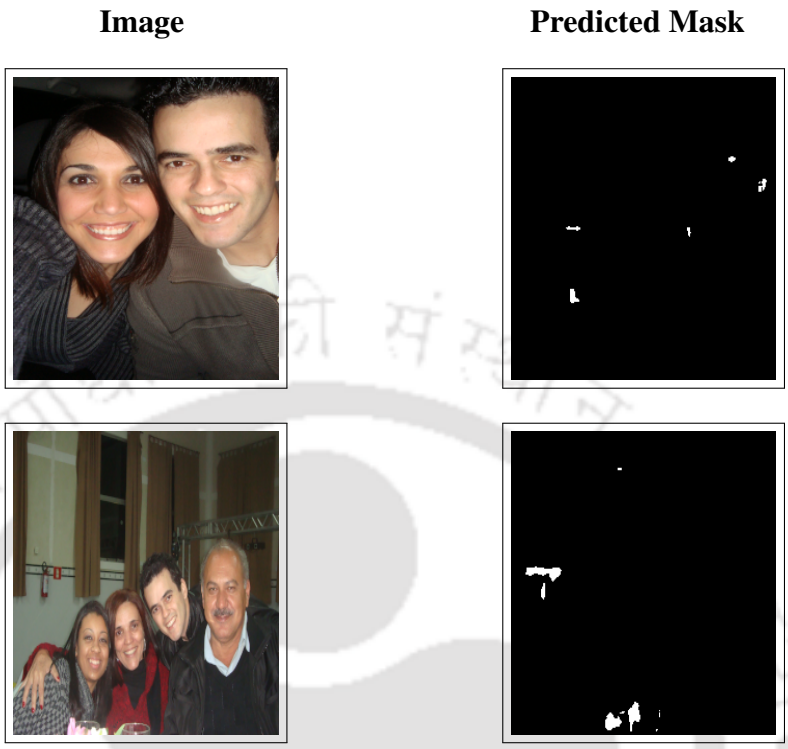


Figure 6.5: Prediction results on two pristine images from DSO-1 dataset. As can be seen, except for a few small regions, there is not much false positive in the predicted masks.

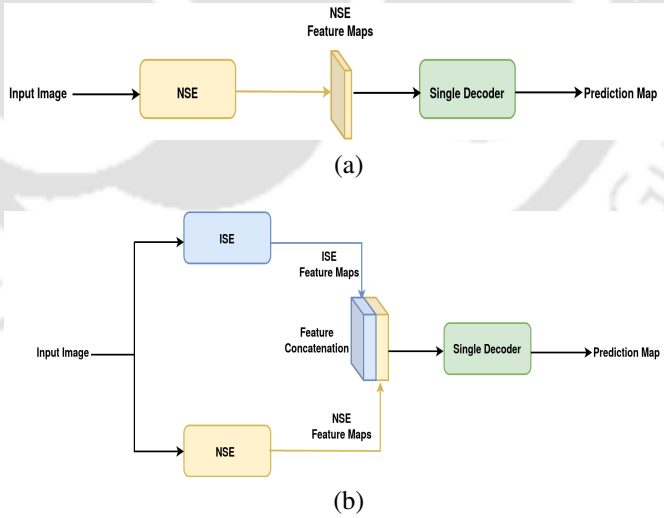


Figure 6.6: Different variants of encode-decoder architecture experimented in the work: (a) NSED and (b) ISE-NSE-1-Dec

(iii) Proposed (two-stream encoder-decoder): It has two parallel encoder-decoder networks, *i.e.*, noise and image-stream encoder-decoder, and performs fusion of the decoder feature maps of both the streams (late-fusion) for producing the prediction.

Table 6.6 shows the performance of the three network variants on NIST16 and Columbia

Table 6.6: Performance comparison of the ablated versions of the proposed network on two datasets

	NIST16		**Columbia**	
	F1	AUC	F1	AUC
NSED	0.51	0.93	0.75	0.85
NSE-ISE-1-Dec	0.50	0.92	0.77	0.88
proposed	**0.62**	**0.95**	**0.86**	**0.88**

datasets in terms of the $F1$-score and the AUC value. As can be seen, the proposed architecture performs the best on both datasets in terms of both measures. These results indicate the necessity of learning both the low-level and high-level features for accurately localizing the forgeries. The results also suggest that the late-fusion technique performs better than early-fusion.

The possible reason for this is that the features computed by the noise-stream and the image-stream encoders may have different distributions. Hence, in the case of early-fusion, a single decoder operating on the concatenated features may not be effective in computing the dense feature maps for accurate predictions. On the other hand, in the late-fusion technique, each stream first computes the dense feature maps individually, which are then concatenated and fed to a final convolutional layer. Hence, the difference in the distributions of the features of the two encoder streams does not affect the performance of the network.

Therefore, it can be argued that the superior performance of the proposed method with respect to RGB-N [16] and LSTM-EnDec [6] is due to the incorporation of the constrained convolutional layer and the late-fusion of the dense feature maps.

Finally, we have checked the performance of the network when trained with the weighted cross-entropy loss instead of the Dice loss. Table 6.7 shows the performance of the proposed (*i.e.*, two-stream encoder-decoder) method on IFC and DSO-1 datasets, when trained using the weighted cross-entropy and the Dice losses. The results show that the network trained using the Dice loss performs better than the one trained using the weighted cross-entropy loss on both datasets. Figure 6.7 shows the qualitative results of the ablation study.

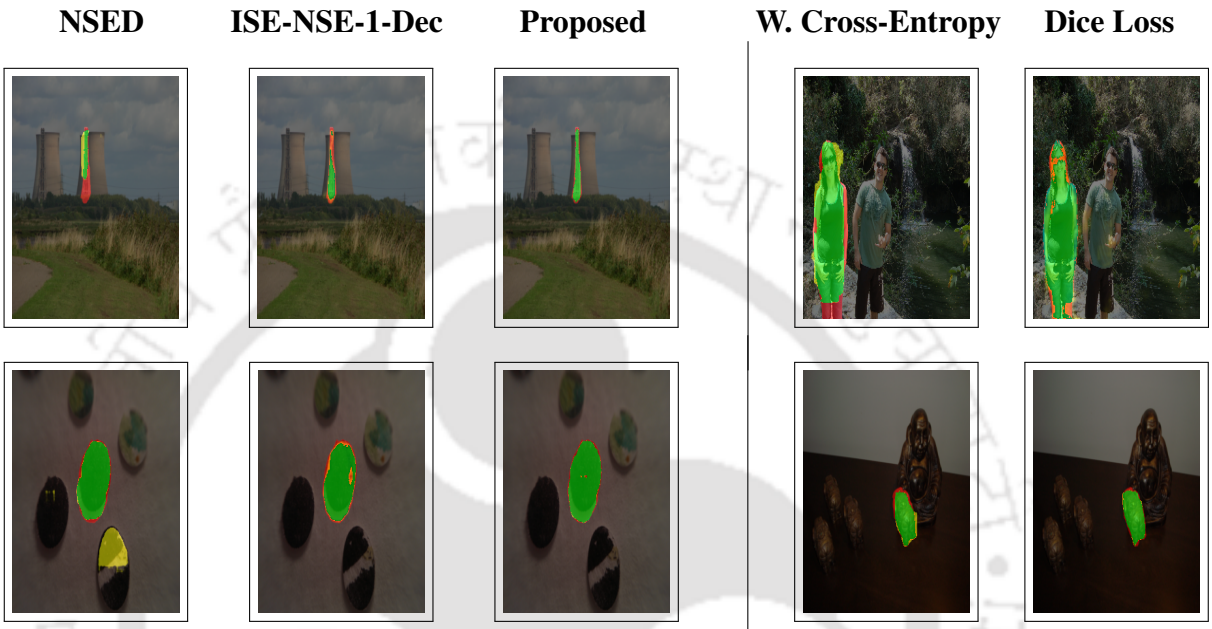


Figure 6.7: Ablation study of the proposed network, with different network settings and loss functions. The images in the first three columns are examples of predictions by NSED, ISE-NSE-1-Dec and the proposed networks with the Dice loss respectively on NIST datasets. The images in the last two columns are predictions of the proposed network using the weighted cross-entropy and the Dice losses, respectively, on IFS dataset.

Table 6.7: Comparison of cIoU values for weighted cross-entropy and Dice losses

Loss Function	IFC	DSO-1
Weighted Cross-Entropy	0.64	0.49
Dice	**0.68**	**0.52**

6.4 Summary

This chapter proposed a novel two-stream encoder-decoder network for localizing different types of forgeries, namely splicing, copy-move, and content-removal. One of the streams learns the high-level manipulation-related traces, such as unnatural contrast, from the RGB pixel values in the encoder side. The encoder of the other stream learns the low-level features, such as noise inconsistencies, by employing a high-pass filtering layer as the first layer of the encoder CNN. The decoders of both the streams perform upsampling and convolution on the coarse feature maps computed by the encoders and produce dense feature maps of the same resolution as the input. The dense feature maps of both the streams are concatenated and fed to a single convolutional layer with the sigmoid nonlinearity to produce the pixel-wise probability map. The probability map gives the probability of each pixel being classified as forged or authentic. The experimental results on multiple standard forgery datasets show the effectiveness of the proposed method with respect to the state-of-the-art methods.

7

Conclusions and Future Works

The thesis investigated the digital image forensics problem and proposed a number of methods for the detection and localization of various types of forgeries present in images. This chapter summarizes the main contributions of the thesis and provides some possible directions for future research.

7.1 Summary

Detecting spliced human faces in images is an important forensics task. The first method, presented in Chapter 2, focused on detecting splicing forgeries in images containing human faces in the front pose. The method first estimated the lighting environments (LEs) from the face regions of the persons present in an image using a novel LE estimation method. The estimation method is based on creating a low-dimensional lighting model from a set of front pose face images of a single individual under different directional LEs. The principal component analysis (PCA) is used for decomposing the set of face images, and the first six dominant eigenvectors are used for creating the lighting model, as they capture most of the lighting variation in the set of the face images. The LE of a face image is estimated by projecting the face onto the lighting model. The angular errors between LEs, estimated from all the faces, are computed in a pair-wise manner. If the angular error for any pair of LEs is more than a predefined threshold, the image is decided as spliced. Although the method is effective, it has the limitation that it can detect splicing forgeries in images containing front pose faces only.

To overcome the above limitation, Chapter 3 proposed a method that can detect spliced faces of any pose present in a forged image. The method extracted a novel illumination-signature from each face by utilizing the illumination colour as a cue of splicing forgery. We propose the dichromatic plane histogram (DPH) as the illumination-signature. It is computed by applying 3D Hough transform on the face images. The dichromatic reflection model is utilized for computing the DPH. The DPHs of all the faces are compared pair-wise using the correlation measure between them. The image is decided as spliced if the correlation value of any pair of DPHs is below a predefined threshold. The spliced face is detected as the one whose DPH has the values of correlation with the majority of other DPHs less than a predefined threshold. The limitation of this method is that it assumes all the faces to be of the same skin colour, and hence

it fails in cases where the difference of skin tones of the faces are large, *e.g.*, faces of people from different ethnicities.

Deep learning-based methods, such as CNNs, have proven to be effective in many computer vision applications. Chapter 4 presented a deep learning-based method for detecting splicing forgeries involving faces of any pose and skin colour. The method created an illumination map (IM) from the input image by first segmenting it into homogeneous regions and then recolouring the segments by the illumination colours estimated from them. The illumination colours are estimated using the physics-based inverse-intensity chromaticity (IIC) method and statistics-based generalized gray-edge (GGE) method. The face parts of the IM (face-IM) are extracted, and a siamese CNN is trained to discriminate face-IM pairs coming from the same and different images. Once the siamese CNN is trained, the CNN part of the network is used for extracting features from the face-IMs present in an image. The features extracted from the face-IMs of the image are concatenated in a pair-wise manner and classified using an SVM. The method was successful in detecting spliced faces of any pose and skin colour.

To detect various image editing operations and different forgeries involving arbitrary image regions, a universal forensics method is proposed in Chapter 5. The method employs a siamese CNN for differentiating image patches modified using different image editing operations. The siamese network is trained to classify pairs of image patches as either similarly or differently processed. The trained siamese network is used for classifying image patches processed with operations, either considered or not considered in the training stage, using the one-shot classification technique. Based on this trained siamese network, a forgery detection and localization technique is proposed. It can detect and localize various types of forgeries, such as splicing, copy-move and retouching, involving arbitary image regions.

A method is proposed in Chapter 6 that employs a two-stream encoder-decoder network for utilizing both the high-level and the low-level image manipulation-related traces for localizing various types of forgeries in a single framework. The encoder in one stream of the network learns the high-level manipulation-related traces or artefacts, such as unnatural contrasts, and that in the other stream learns the low-level artefacts, such as noise inconsistencies. The de-

coder in each stream upsamples the coarse features learned by the corresponding encoder and produces dense feature maps. The dense feature maps of both the streams are then fed to a 1×1 convolutional layer to produce the final prediction map, representing the forged and the authentic pixels.

7.2 Future Research Directions

We have identified several possible research directions that can be pursued for extending the works of this thesis. These are discussed below.

1) In chapter 3, a siamese CNN-based method is proposed to classify the face-IMs in a pair-wise manner for detecting face splicing forgeries. The method has several modules, *i.e.,* segmentation and illumination colour estimation, for creating the IMs from the input images. This increases the complexity of the method. In addition to this, in the case of low-resolution images, the accuracy of the illumination estimation methods drops. This leads to less accurate IMs, which in turn leads to less accurate splicing detection. As GANs have shown their excellence in generating artificial images, they can be utilized for generating the IMs. Employing GANs for computing the IMs will remove the need for performing segmentation and illumination colour estimation, and hence will make the method simpler. This might also help to generate the IMs more accurately even when the images are of low-resolution as GANs are effective in generating high-quality images of any resolution.

2) Chapter 6 proposed a two-stream encoder-decoder network for localizing various types of forgeries. It combined the high-level and low-level manipulation-related features through the late-fusion technique. It is also shown that late-fusion performed better than the early-fusion technique. It is open to explore the hybrid fusion techniques that take advantage of both the early and late-fusion techniques. Further, many recent works, *e.g.,* [154], [155], have shown that employing the attention module in the encoder-decoder framework improves the network's ability in many computer vision tasks. Along similar lines, the attention module can be incorporated to the two-stream encoder-decoder network, which will help the network to focus on the forged regions more than the authentic ones.

3) It has been established that adding small imperceptible non-random perturbation to input

images can fool deep neural networks to give incorrect class predictions [156], [157], [158]. This is known as the adversarial attack, and the perturbed image is known as the adversarial example [156]. An important open problem for future research is to check the vulnerability of the deep learning-based forensics methods proposed in Chapters 4, 5, and 6 against the adversarial attacks. In case the performances of the proposed methods are found to be degraded greatly due to the adversarial attacks, various techniques to improve the robustness of the methods to the adversarial attacks could be explored. For instance, the gradient penalty technique, proposed in [159], can be incorporated in the loss functions of the proposed networks. The gradient panalty technique ensures that the loss functions do not change significantly when there is a slight change in the inputs, and thereby increases the robustness of the networks to the adversarial examples. In addition to that, the networks can be trained with adversarial examples, as proposed in [160], to increase the robustness of them against the adversarial attacks.

Bibliography

[1] Photo tampering throughout history. [Online]. Available: https://www.cc.gatech.edu/~beki/cs4001/history.pdf

[2] In an Iranian image, a missile too many. [Online]. Available: https://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/

[3] Photo manipulation throughout history: A timeline. [Online]. Available: https://martynaeidukynaite.wordpress.com/2014/03/19/photo-manipulation-throughout-history-a-timeline/

[4] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[5] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[6] J. H. Bappy, C. Simons, L. Nataraj, B. Manjunath, and A. K. Roy-Chowdhury, "Hybrid lstm and encoder–decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, 2019.

[7] E. Kee and H. Farid, "Exposing digital forgeries from 3-d lighting environments," in *2010 IEEE International Workshop on Information Forensics and Security*. IEEE, 2010, pp. 1–6.

[8] B. Peng, W. Wang, J. Dong, and T. Tan, "Optimized 3d lighting environment estimation for image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 479–494, February 2017.

[9] T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, 2013.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[13] X. Gao, S. Lin, and T. Y. Wong, "Automatic feature learning to grade nuclear cataracts based on deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2693–2701, 2015.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[16] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[17] V. V. Kniaz, V. Knyaz, and F. Remondino, "The point where reality meets fantasy: Mixed adversarial generators for image splice detection," in *Advances in Neural Information Processing Systems*, 2019, pp. 215–226.

[18] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, 2008.

[19] C. Chen, S. McCloskey, and J. Yu, "Image splicing detection via camera response function analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5087–5096.

[20] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on signal processing*, vol. 53, no. 2, pp. 758–767, 2005.

[21] ——, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[22] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, 2007.

[23] H. Farid, "Exposing digital forgeries from jpeg ghosts," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009.

[24] Q. Liu, "An approach to detecting jpeg down-compression and seam carving forgery under uncompression anti-forensics," *Pattern Recognition*, vol. 65, pp. 35–46, 2017.

[25] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, no. 10, pp. 1497–1503, 2009.

[26] S. Lyu, X. Pan, and X. Zhang, "Exposing region splicing forgeries with blind local noise estimation," *International journal of computer vision*, vol. 110, no. 2, pp. 202–221, 2014.

[27] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proceedings of the 7th Workshop on Multimedia and Security*, 2005, p. 110.

[28] P. Saboia, T. Carvalho, and A. Rocha, "Eye specular highlights telltales for digital forensics: A machine learning approach," in *IEEE Int. Conf. Image Processing (ICIP)*, 2011, pp. 1937–1940.

[29] S. Gholap and P. K. Bora, "Illuminant colour based image forensics," in *IEEE Region 10 Conf.*, 2008, pp. 1–5.

[30] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[31] O. Mayer and S. M. C., "Accurate and efficient image forgery detection using lateral chromatic aberration," *IEEE Transactions on Information Forensics and Security*, 2018.

[32] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicng localization using a multi-task fully convolutional network (mfcn)," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201–209, 2018.

[33] W. Li, Y. Yuan, and N. Yu, "Passive detection of doctored jpeg image via block artifact grid extraction," *Signal Processing*, vol. 89, pp. 1821–1829, 2009.

[34] S. Lyu, X. Pan, and X. Zhang, "Exposing region splicing forgeries with blind local noise estimation," *International Journal of Computer Vision*, vol. 110, pp. 202–221, 2014.

[35] T. M. MacRobert, "Spherical harmonics: an elementary treatise on harmonic functions with applications," 1947.

[36] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26, 1980.

[37] J. Weijer, T. Gevers, and A. Gijsenij, "Edge-based color consistency," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, 2007.

[38] R. Tan, K. Nishino, and K. Ikeuchi, "Color constancy through inverse-intensity chromaticity space," *Journal of the Optical Society of America A*, vol. 21, pp. 321–334, 2004.

[39] X. Wu and Z. Fang, "Image splicing detection using illuminant color inconsistency," in *IEEE Int. Conf. Multimedia Inform. Networking and Security*, 2011, pp. 600–603.

[40] K. Francis, S. Gholap, and P. K. Bora, "Illuminant colour based image forensics using mismatch in human skin highlights," in *Proc. Twentieth National Conference on Communications (NCC)*, 2014, pp. 1–6.

[41] T. Carvalho, F. A. Faria, H. Pedrini, R. S. Torres, and A. Rocha, "Illuminant-based transformed spaces for image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 720–733, 2016.

[42] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3788–3792.

[43] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, 2008.

[44] J. Radon, "On the determination of functions from their integral values along certain manifolds," *IEEE transactions on medical imaging*, vol. 5, no. 4, pp. 170–176, 1986.

[45] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE micro*, vol. 18, no. 6, pp. 20–30, 1998.

[46] P. Ferrara, T. Bianchi, A. Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1566–1577, 2012.

[47] A. E. Dirik and N. D. Memon, "Image tamper detection based on demosaicing artifacts," in *IEEE International Conference on Image Processing*, no. 1497-1500, 2009.

[48] Z. Lin, R. Wang, X. Tang, and H. Shum, "Detecting doctored images using camera response normality and consistency," in *IEEE Computer Vision and Pattern Recognition*, 2005, pp. 1087–1092.

[49] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, no. 1849-1853, 2015.

[50] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, 2018.

[51] Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9543–9552.

[52] D. Cozzolino and L. Verdoliva, "Noiseprint: a cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, 2019.

[53] M. Kirchner and R. Bohme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, 2008.

[54] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, pp. 1–42, 2011.

[55] Y. Ostrovsky, P. Cavanagh, and P. Sinha, "Perceiving illumination inconsistencies in scenes," *Perception*, vol. 34, no. 11, pp. 1301–1314, 2005.

[56] H. Farid and M. J. Bravo, "Image forensic analyses that elude the human visual system," in *Media forensics and security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 754106.

[57] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can we trust digital image forensics?" in *15th Int. Conf. Multimedia*, 2007, pp. 78–86.

[58] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 1050–1065, 2011.

[59] C. Riess, S. Pfaller, and E. Angelopoulou, "Reflectance normalization in illumination-based image manipulation detection," in *International Conference on Image Analysis and Processing*. Springer, 2015, pp. 3–10.

[60] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "3d lighting-based image forgery detection using shape-from-shading," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1777–1781.

[61] R. Huang and W. A. Smith, "Shape-from-shading under complex natural illumination," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 13–16.

[62] B. Peng, W. Wang, J. Dong, and T. Tan, "Optimized 3d lighting environment estimation for image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 479–494, February 2017.

[63] R. Epstein, P. Hallinan, and A. Yuille, "5±2 eigenimages suffice: An empirical investigation of low-dimensional lighting models," in *IEEE Workshop on Physics-Based Vision*, 1995, pp. 108–116.

[64] P. W. Hallinan *et al.*, "A low-dimensional representation of human faces for arbitrary lighting conditions," in *CVPR*, vol. 94, 1994, pp. 995–999.

[65] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[66] A. L. Yuille, D. Snow, R. Epstein, and P. N. Belhumeur, "Determining generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 203–222, 1999.

[67] A. Shashua, "On photometric issues in 3d visual recognition from a single 2d image," *International Journal of Computer Vision*, vol. 21, no. 1-2, pp. 99–122, 1997.

[68] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International journal of computer vision*, vol. 14, no. 1, pp. 5–24, 1995.

[69] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[70] R. Ramamoorthi and P. Hanrahan, "On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object," *JOSA A*, vol. 18, no. 10, pp. 2448–2459, 2001.

[71] R. Ramamoorthi, "Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 10, pp. 1322–1333, 2002.

[72] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?" *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245–260, 1998.

[73] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[74] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel, "Visual tracking for multimodal human computer interaction," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1998, pp. 140–147.

[75] G. D. Finlayson and G. Schaefer, "Solving for colour constancy using a constrained dichromatic reflection model," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 127–144, 2001.

[76] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Computational color constancy: Survey and experiments," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2475–2489, 2011.

[77] L. Shi and B. Funt, "Dichromatic illumination estimation via hough transforms in 3d," in *IS&T Fourth European Conf. on Colour in Graphics, Imaging and Vision*, 2008.

[78] S. A. Shafer, "Using color to separate reflection components," *Color Res. Appl.*, vol. 10, no. 4, pp. 210–218, 1985.

[79] C. Riess and E. Angelopoulou, "Scene illumination as an indicator of image manipulation," *Information Hiding Workshop*, vol. 6387, pp. 66–80, 2010.

[80] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[81] A. Carkacioglu and F. T. Yarman-Vural, "Sasi: A generic texture descriptor for image retrieval," *Pattern Recognition*, vol. 36, no. 11, pp. 2615–2633, 2003.

[82] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2005, pp. 886–893.

[83] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *Journal of Visual Communication and Image Representation*, vol. 11, no. 3, pp. 327–342, 2000.

[84] M. Unser, "Sum and difference histograms for texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 118–125, 1986.

[85] F. Mahmoudi, J. Shanbehzadeh, A. Eftekhari-Moghadam, and H. Soltanian-Zadeh, "Image retrieval based on shape similarity by edge orientation autocorrelogram," *Pattern Recognition*, vol. 36, no. 8, pp. 1725–1736, 2003.

[86] D.-H. Lee and H.-J. Kim, "A fast content-based indexing and retrieval technique by the shape information in large image database," *Journal of Systems and Software*, vol. 56, no. 2, pp. 165–182, 2001.

[87] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 762–768.

[88] R. O. Stehling, M. A. Nascimento, and A. Falcao, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proc. ACM 11th Int. Conf. Inf. Knowl. Manage.,*, 2002, pp. 102–109.

[89] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proc. 4th ACM Int. Conf. Multimedia*, 1996, pp. 65–73.

[90] M. J. Swain and D. H. Ballard, "Color indexing," *Internation Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[91] S. Tominaga and B. Wandell, "Standard surface-reflection model and illumination estimation," *Journal of the Optical Society of America A*, vol. 6, no. 4, pp. 576–584, 1989.

[92] G. J. Klinker, S. A. Shafer, and T. Kanade, "A physical approach to color image understanding," *International Journal of Computer Vision*, vol. 4, pp. 7–38, 1990.

[93] K. Okada, S. Kagami, M. Inaba, and H. Inoue, "Plane segment finder: algorithm, implementation and applications," in *Int. Conf. on Robotics and Automation*, 2001, pp. 2120–2125.

[94] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning workshop*, 2015.

[95] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics and scene semantic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 687–698, 2011.

[96] J. Chen, X. Kang, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, 1849.

[97] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *WIFS*, 2016.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[99] J. Bromley, I. Guyon, Y. LeCun, E. LeCun, Y., and R. Shah, "Signature verification using a siamese time delay neural network," in *Advances in Neural Information Processing System*, 1993.

[100] K. Simonyan and A. Zisserman, "Very deep convolutional neural networks for large-scale image recognition." in *arXiv preprint arXiv:1409.1556*, 2014.

[101] K. Namdar, I. Gujrathi, M. A. Haider, and F. Khalvati, "Evolution-based fine-tuning of cnns for prostate cancer detection," in *International Conference on Neural Information Systems (NeurIPS)*, 2019.

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

[103] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.

[104] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Transactions on Information Forensics and Security*, vol. 53, no. 2, pp. 758–767, 2005.

[105] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensics detection of resampled images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 536–545, 2012.

[106] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–536, 2008.

[107] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

[108] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, 2010.

[109] C. Chen and J. Ni, "Median filtering detection using edge based prediction matrix," *Digital Forensics and Watermarking*, pp. 361–375, 2012.

[110] M. C. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *IEEE International Conference on Image Processing*, 2008, pp. 3112–3115.

[111] ——, "Forensic estimation and reconstruction of contrast enhancement mapping," in *IEEE International Conference on Acoustic Speech and Signal Processing*, 2010, pp. 1698–1701.

[112] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts." *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[113] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensics strategy based on steganalytic model." in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, no. 165-170, 2014.

[114] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, no. 1-6, 2015.

[115] T. Penvy, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[116] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[117] Y. Q. Shi, P. Sutthiwan, and L. Chen, "Textural features for steganalysis," in *International workshop on information hiding*. Springer, 2012, pp. 63–77.

[118] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of Advances in neural information processing systems*, vol. 1097-1105, 2012.

[119] P. Zhou, X. Han, V. Morariu, and L. Davis, "Learning rich features for image manipulation detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2018.

[120] O. Mayer and M. C. Stamm, "Forensic similarity for digital images," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1331–1346, 2019.

[121] Z. Lin, J. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis," *Pattern Recognition*, vol. 43, pp. 2492–2501, 2009.

[122] S. Ye, Q. Sun, and E. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *IEEE International Conference on Multimedia and Expo*, no. 12-15, 2007.

[123] N. Krawetz, "A picture's worth..: Digital image analysis and forensics," *Black Hat Briefings*, pp. 1–31, 2007.

[124] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, pp. 1497–1509, 2009.

[125] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminately, with application to face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 539–546.

[126] M. Fontani and M. Barni, "Hiding traces of median filtering in digital images," in *Proc. 20th Eur. Signal Process. Conf.*, 2012, pp. 1239–1243.

[127] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[128] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, arXiv preprint arXiv:1207.0580.

[129] T. Gloe and R. Bohme, "The 'dresden image database' for benchmarking digital image forensics." in *Proceedings of the 25th Symposium on Applied Computing*, 2010, pp. 1585–1591.

[130] T. Dozat, "Incorporating nesterov momentum into adam," in *ICLR Workshop*, 2016.

[131] M. Welling and Y. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceeding of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 681–688.

[132] NIST nimble 2016 datasets. [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/

[133] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[134] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Large-scale evaluation of splicing localization algorithms for web images," *Multimedia Tools and Applications*, vol. 76, pp. 1–34, 2016.

[135] "Nist nimble 2016 datasets. https://www.nist.gov/sites/default/files/ documents/2016/11/30/should i believe or not.pdf."

[136] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1855–1864.

[137] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2004-515*, pp. 1–11, 2004.

[138] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1855–1864.

[139] S. K. Tiwari, A. Mazumdar, and P. K. Bora, "Detection of splicing forgery using cnn-extracted camera-specific features," in *International Conference on Pattern Recognition and Machine Intelligence*. Springer, 2019, pp. 473–481.

[140] D. Cozzolino and L. Verdoliva, "Noiseprint: a cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, 2019.

[141] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy–move forgery detection by matching triangles of keypoints," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2084–2094, 2015.

[142] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Busternet: Detecting copy-move image forgery with source/target localization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 168–184.

[143] B. Horn, B. Klaus, and P. Horn, *Robot vision*. MIT press, 1986.

[144] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *International conference on machine learning*. PMLR, 2013, pp. 1319–1327.

[145] K. Bouman, N. Khanna, and E. Delp, "Digital image forensics through the use of noise reference patterns," in *International sustainable remediation forum conference*, 2016, pp. 1–7.

[146] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[147] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571.

[148] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 562–578.

[149] NIST media forensics challenge (MFC) 2018. [Online]. Available: https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2018/

[150] IEEE forensics challenge (IFC). [Online]. Available: http://ifc.recod.ic.unicamp.br/fc.website/index.py

[151] T.-T. Ng, S.-F. Chang, and Q. Sun, "A data set of authentic and spliced image blocks," *Columbia University, ADVENT Technical Report*, pp. 203–2004, 2004.

[152] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, 2013, pp. 422–426.

[153] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[154] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[155] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[156] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[157] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[158] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[159] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[160] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 3358–3369.

List of Publications

Journal

- A. Mazumdar and P. K. Bora, "Two-stream Encoder-Decoder Network for Localizing Image Forgeries," *Journal of Visual Communication and Image Representation,* Dec., 2021.
- A. Mazumdar and P. K. Bora, "Siamese Convolutional Neural Network-based Approach towards Universal Image Forensics," *IET Image Processing,* June, 2020.
- A. Mazumdar and P. K. Bora, "Estimation of Lighting Environment for Exposing Image Splicing Forgeries," *Multimedia Tools and Applications,* 2019.

Conference

- A. Mazumdar, J. Singh, Y. S. Tomar, and P. K. Bora, "Detection of Image Manipulations Using Siamese Convolutional Neural Networks," in Proc. of *International Conference on Pattern Recognition and Machine Intelligence (PReMI),* Tezpur University, India, December 2019.
- A. Mazumdar and P. K. Bora, "Deep Learning-based Classification of Illumination Maps for Exposing Face Splicing Forgeries," in Proc. of *IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, September 2019.
- A. Mazumdar and P. K. Bora, "Exposing splicing forgeries in digital images through dichromatic plane histogram discrepancies," in Proceedings of the *Tenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP),* IIT Guwahati, India, December 2016.