

University of Sheffield

Deep Learning-based Error Level Modelling for Image Manipulation Detection and Localization



Ziyong Wang

Supervisor: Dr Charith Abhayaratne

A report submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Electronics and Electrical Engineering

June 11, 2025

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name:

Signature:

Date:

Abstract

The widespread accessibility of multimedia content through social platforms has significantly increased the risk of image manipulation, thereby amplifying the dissemination and influence of fabricated visual information. Addressing this critical challenge necessitates the development of robust, efficient, and explainable methods to verify the authenticity of visual content. This thesis investigates novel AI-based approaches for image manipulation detection, with a particular emphasis on achieving strong generalization across challenging datasets and enhancing model interpretability.

Specifically, three deep learning architectures are proposed to address different tasks within image manipulation detection and localization: 1) **WCBnet** introduces an adaptive cross-block weighting mechanism at the convolutional block level, allowing the network to dynamically fuse low-level and high-level features based on their relevance to manipulation cues. This hierarchical feature weighting strategy enables WCBnet to achieve fast and accurate fact-checking with minimal additional computational overhead (only a 2.3% increase in trainable parameters), making it particularly suitable for large-scale scenarios involving newly generated manipulated images. 2) **DenseWCBnet** extends WCBnet by integrating multi-scale receptive fields across multiple convolutional blocks. Through adaptive fusion across different spatial dimensions, DenseWCBnet generates densely weighted feature representations that further enhance manipulation detection accuracy and significantly improve robustness against diverse and challenging manipulation types. 3) **WSWCBnet** proposes a novel weakly supervised localization framework, combining image-level manipulation heatmaps and semantically irrelevant segmented maps to localize manipulated regions without the need for pixel-level annotations. By leveraging only image-level supervision, WSWCBnet achieves pixel-level localization performance comparable to fully supervised methods, substantially reducing the annotation burden while maintaining high localization precision.

Extensive experiments demonstrate that WCBnet and DenseWCBnet consistently outperform state-of-the-art methods across six widely-used datasets in terms of classification accuracy and F1-score. Furthermore, when evaluated on the particularly challenging DeepfakeArt generative dataset, WCBnet and DenseWCBnet achieve classification accuracies of 94% and 97%, respectively. Additionally, WSWCBnet demonstrates its effectiveness in manipulation localization, achieving comparable performance to fully supervised models despite relying solely on weak supervision. Overall, this thesis advances the field of image forensics by providing robust, explainable, and annotation-efficient deep learning solutions for image manipulation detection and localization, with strong applicability to real-world forensic scenarios.

List of Abbreviations

AI	Artificial Intelligence
AUC	Area Under Curves
Au	Authentic Image Class
CAM	Class Activation Map
CBWM	Cross Convolutional-blocks Weighting Module
CM	Copy-move Image Class
CNN	Convolutional Neural Network
DFWF	Dense Feature Weighting and Fusing
EL	Error Level
F1-score	Harmonic Mean of Precision and Recall
FLOPS	Floating Point Operations Per Second
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
Grad-CAM	Gradient-weighted Class Activation Map
GT	Ground Truth
IoU	Intersection over Union
SAM	Segment Anything Model
SEL	Signed-value Error Level
SNR	Signal-to-noise Ratio
SOT	State-of-the-Art
Sp	Splicing Image Class
TN	True Negative
TP	True Positive
WEL	Wavelet-based Error Level

List of Symbols

$B(\cdot)$	Heatmap Enhancement Function based on Bayesian Inference
$C(\cdot)$	CNN-Based Feature Extractor
$E(\cdot)$	Error Level Extraction
$\hat{E}(\cdot)$	Enhanced Error Level Extraction
F_i	Feature Class Activation Map between Y'_j and Class Predictions
f_i	Feature Class Activation Map between y_i and Class Predictions
H	Final Manipulation Heatmap combining F_1 to F_5
$S(\cdot)$	Signed-Value Error Level Extraction
$\hat{S}(\cdot)$	Enhanced Signed-Value Error Level Extraction
$Sim(\cdot, \cdot)$	Similarity Function based on Distance Transform
x	Input Manipulated Image
x'	JPEG-Compressed Image
x_m	Pristine Background Region
x_p	Manipulated Region
y_i	Features Extracted from the i -th Convolutional Block
y'_i	Feature y_i reshaped to a Fixed Shape
y_{ij}	Feature y_i reshaped to the Shape of the j -th Convolutional Block
Y	Feature Obtained by Concatenating All Five y_i
Y'_j	Feature Obtained by Concatenating All Five y_{ij}
RefNet-i	Reference Network using the i -th Partial Structure of CBWM ($i = 1, 2, 3$)

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Charith Abhayaratne, for his invaluable guidance and academic inspiration throughout my PhD journey. His profound insights and dedication have greatly shaped my research and personal growth.

I am deeply indebted to my parents, who have supported me emotionally and financially over the years, allowing me to focus wholeheartedly on my studies. Their unwavering belief in me has been a constant source of strength.

To my colleagues, thank you for fostering an engaging and collaborative research environment, and for the friendships we've built along the way. Your support has made the challenging moments more bearable and the successes more meaningful.

Finally, my heartfelt thanks go to my fiancée, Siyang Jia, whose unwavering love and companionship have been my greatest source of comfort during this demanding journey.

Her patience, understanding, and encouragement have carried me through every challenge, and for that, I am forever grateful. As I once wrote for you:

Just like two streams diverging, winding their way through twists and turns, carrying silt and pebbles along the journey, they will eventually converge and flow together endlessly.

Thank you for being my endless flow of support and love.

Contents

List of Abbreviations	iii
List of Symbols	iv
1 Introduction	1
1.1 Overview	1
1.2 Background and Motivation	1
1.3 Aims and Objectives of the Research	2
1.4 Main Contributions	3
1.5 Thesis Structure	4
1.6 Publications Associated with This Thesis	5
2 Literature Review	6
2.1 Image Manipulation	6
2.2 Image Manipulation	7
2.2.1 Conventional Image Manipulation Methods	7
2.2.2 Non-intentional Image Forensics	12
2.2.3 Learning-based Image Manipulation Methods	12
2.3 Image Manipulation Detection Methods	15
2.3.1 Hand-crafted Feature-based Manipulation Detection	15
2.3.2 Learning-based Methods for Conventional Manipulations	17
2.3.3 Generative AI (Gen-AI) Detection	19
2.4 Challenges of Current Methods	20
2.5 Research Questions Addressing the Challenges	21
2.6 Summary	21
3 Image-wise Manipulation Detection Using CNN-based Method Modelling Error Levels (WCBnet)	23
3.1 Introduction	23
3.2 Related Work	25
3.3 Proposed Method	26
3.3.1 Signed-value Error Level	27
3.3.2 Learned Feature Extractor	29
3.3.3 Peak-shaped feature reshaping	30
3.3.4 Cross Convolutional-blocks Weighting Module	31
3.3.5 Binary Classification Module	32
3.4 Experimental Results and Discussions	32
3.4.1 Experimental Setup	32
3.4.2 Performance Comparison with State-of-the-art Methods	36

3.4.3	Comparing image-level performance with other error-modelling methods	36
3.4.4	Evaluation of WCBnet for Identifying Manipulation Types	39
3.4.5	Ablation Studies	40
3.4.6	Robustness Test Against Perturbations	45
3.4.7	Extended Experiment for Generative AI Detection	46
3.4.8	JPEG-AI and Its Impact on Error Level	47
3.5	Conclusions	49
4	Explainability of Image-wise Manipulation Detection methods and generating dense-view manipulation heatmaps (DenseWCBnet)	51
4.1	Introduction	51
4.2	Methodology	54
4.2.1	Overview	54
4.2.2	The SEL Processing and Multi-level Feature Extraction	54
4.2.3	Single-view CBWM	56
4.2.4	Dense Feature Integration	56
4.2.5	Binary Classification Module	57
4.2.6	Explainability: Feature Maps and Dense-view Heatmap	57
4.3	Experimental Results and Discussions	59
4.3.1	Experimental Setup	59
4.3.2	Comparison with State-of-art Methods Regarding Image-level Performance	60
4.3.3	Effectiveness of Signed-value Error Level	61
4.3.4	Effectiveness of CBWM	64
4.3.5	Effectiveness of Dense Feature Weighting and Fusion	66
4.3.6	Robustness Test Against Perturbations	68
4.3.7	Extended Experiments for Generative AI Detection	69
4.4	Conclusions	70
5	Weakly-supervised Pixel-wise Image Manipulation Localization Using Multi-resolution Learned Features (WSWCBnet)	71
5.1	Introduction	71
5.2	Related Work	73
5.2.1	Gradient-weighted Class Activation Mapping (Grad-CAM)	73
5.2.2	Non-semantic Image Segmentation	74
5.3	Methodology	74
5.3.1	Overview	74
5.3.2	Image-wise Manipulation Classification	75
5.3.3	Feature Map Generation	76
5.3.4	Generating Segmentation Maps Using Pre-trained Models	78
5.3.5	Combining Manipulation Heatmap and Segmentation Maps	79
5.4	Experimental Results and Discussions	80
5.4.1	Dataset	80
5.4.2	Experimental Setup	80
5.4.3	Evaluation of Heatmaps Generated by DenseCAM	81
5.4.4	Intuitive Comparison of Segmentation Networks on Manipulated Images	83

5.4.5	Evaluating the Combination of Multi-view Heatmaps and Segmented Masks	83
5.5	Conclusions	85
6	Conclusions	87
6.1	Research Summary and Objectives	87
6.2	Addressing the Five Research Questions: Contributions and Insights . . .	87
6.2.1	Key Contributions Addressing Five Research Questions	87
6.2.2	Summary of Findings	91
6.3	Future Work	92
6.3.1	Multi-Branch Hybrid Feature Representation	92
6.3.2	Advancing Explainability and Localization	93
6.3.3	Toward Practical, Adaptive, and Scalable Manipulation Detection	94
	Appendices	111
A	Enhanced heatmaps of WSWCBnet	112
A.1	Heatmaps with Effective Combination	112
A.2	Heatmaps With Ineffective Combination	114

List of Figures

2.1	The categories of common image manipulation methods	7
2.2	Example of three most conventional image manipulation methods	8
2.3	Several samples of different conventional image manipulation datasets. . .	10
2.4	More samples of different conventional image manipulation datasets. . . .	11
2.5	The basic Generator and Discriminator of GAN.	13
2.6	Fake zebra images using trained Cycle-GAN, and the upper row shows the original horse images while the bottom row shows the transferred zebra images.	13
2.7	General framework of conventional splicing detection	16
2.8	General framework of conventional copy-move detection.	17
3.1	Error levels are considered as manipulated traces that distinguish between manipulated regions and pristine background; The red annotations illustrate the two common conventional image manipulation techniques: copy-move and splicing.	24
3.2	Three examples of pristine, splicing and copy-move images, and their corresponding ground truth, EL maps and SEL maps; Forged areas in similar positions are marked with red boxes and zoomed in	26
3.3	The five stages of proposed WCBnet, and different colors of features in the figure represent different weights.	27
3.4	The comparison of manipulated images that have different relative compression factors, and their normal Error-level maps (ELA) and proposed Signed-value Error-level (SEL); Top row shows the normal condition that forged regions has lower image quality while the bottom row is the opposite condition; The error-level maps are enhanced for visualization.	28
3.5	The histograms of normal EL and proposed SEL, extracted from the same image.	29
3.6	The comparison between normal U-shaped (which refers to <i>Peak1</i> peaked-shaped reshaping) and proposed peak-shaped feature reshaping	31
3.7	Self-defined pixel-level Distance Score for evaluating the similarity between ground truth masks and error level maps.	34
3.8	Comparisons between different error levels or noises, extracted from manipulated images.	37
3.9	Three-class manipulation detection results of WCBnet on subsets of De-facto for identifying different manipulation types, represented as confusion matrix.	39
3.10	The performance of WCBnet on three subsets of CASIA2.0, regarding 100 compression factors	42

3.11	Different perturbation methods are applied on the CASIA2.0 image manipulation dataset, which are (a) Gaussian noise, (b) Salt&Pepper noise, (c) blurring, (d) contrast, (e) scaling and (f) JPEG compression. The detailed parameters are illustrated in the TABLE 3.11.	45
3.12	Inpainting subset of DeepfakeArt challenge, which contains leaning-based splicing fake contents.	48
4.1	Overview of proposed DenseWCBnet, which models manipulation-related error level by densely weighting and fusing multi-level features of hierarchical convolutional blocks.	52
4.2	Workflow of proposed image-level manipulation detection method DenseWCBnet which consist of five core components; The blue lines and orange lines in Dense Feature Fusion represent down-scaling and up-scaling; In the figure, ConvBlock_i is the convolutional block i of backbone networks, CBWM is the Cross Convolutional-block Weighting Module (as shown in the Figure 4.3), GAP is Global Average Pooling layer.	55
4.3	Single-view CBWM at the dimension of one convolutional block, which is similar to the one illustrated in Chapter 3 while such CBWM of different scales is included for multiple times.	56
4.4	The generation of feature maps for explainability; The dotted lines represent the grad-CAMs, while different colors mean the feature maps in two stages.	58
4.5	Comparison between the foreground and background of pristine and manipulated image, regarding RGB, EL and SEL maps.	62
4.6	Comparison of Foreground (FG) and Background (BG) Histograms in RGB, ELA, and SEL Formats, with Pristine Foreground in Green , Shared Background (BG) in Blue , and Splicing Foreground in Red	63
4.7	Various-stage feature maps generated from image-level WCBnet using grad-CAM; The feature maps in middle columns show the attention of five convolutional blocks (shown as f_i in Figure 4.2), while the last column shows the single-view heatmaps of weighted block (F_4 in Figure 4.2) which is generated using single-stage CBWM with fixed feature shape; The first three rows show the results of splicing images, and the last three are copy-move maps.	66
4.8	Dense-view heatmaps generated from image-level DenseWCBnet using grad-CAM; Heatmaps in the middle columns show the attention of densely weighted features (F_1 to F_5 in Figure. 4.2), which are processed by different CBWMs at the level of 5 weighted blocks; The last column shows manipulation heatmap H which fuses the dense-view heatmaps with weights, obtained from the activation values of the dense layer; The first three rows show the results of splicing images, and the last three are copy-move. . . .	68
5.1	Several examples of heatmaps from image-level WCBnet and segmented maps from non-semantic segmentation models, which show the challenges combining them.	72
5.2	The work-flow of proposed weakly-supervised manipulation localization model; Different colors represent four steps of proposed WSWCBnet. . . .	75
5.3	The workflow of applying Bayesian inference to enhance heatmap H generated from DenseWCBnet.	79

5.4	The features maps of the backbone ResNet50, WCBnets and DenseWCBnet; The WCBnet _i means the CBWM with block i's receptive field, while WCBnet _m is their geometric mean; D-CAM is the proposed DenseCAM; Ref label is the pixel-wise labels just for reference.	82
5.5	Comparison between Grad-CAM, Layer-CAM, Grad-CAM++ on manipulated images, regarding different views of WCBnet.	82
5.6	Another example of comparison between Grad-CAM, Layer-CAM, Grad-CAM++ on manipulated images, regarding different views of WCBnet.	83
5.7	Comparison of Segmentation Networks on Manipulated Images, generated by three state-of-art pre-trained image segmentation methods.	84
5.8	Refined manipulation localization results using combinations of different classification models and segmentation models. ResNet , WCBnet and DenseCAM denote different classification models for generating Grad-CAM-based heatmaps. D , S , and P represent segmentation masks from DeepLab, SAM (Segment Anything Model), and PSPNet, respectively.	85
6.1	Conceptual design of a future multi-branch hybrid feature network for challenging image manipulation detection.	93
6.2	Conceptual design of advanced semi-supervised image manipulation localization framework, which also includes multi-branch manipulation detection model and gradient-guided fused heatmap.	93
A.1	Enhanced heatmaps generated by effective combinations of WSWCBnet (part 1); The first column shows the reference labels, second and third columns are the selected models, and the last column is the enhanced heatmaps.	112
A.2	Enhanced heatmaps generated by effective combinations of WSWCBnet (part 2).	113
A.3	Enhanced heatmaps generated by effective combinations of WSWCBnet (part 3).	114
A.4	Invalid manipulation heatmap that the image-level models focus on wrong regions in the manipulation heatmap.	114
A.5	Invalid segmented maps that the target regions are not clearly extracted by the non-semantic segmentation models.	115
A.6	Invalid combination metric that both heatmaps and segmap are valid while the method to combine them is ineffective.	115

List of Tables

2.1	The parameters of common conventional image manipulation datasets, and the best performance of current methods over them. The number in this chart means the number of pristine and forged images respectively.	9
2.2	The common available image/video manipulation dataset that are generated by deep-learning methods for different purposes.	14
2.3	The classification accuracy of conventional splicing detection methods on three common splicing datasets.	16
2.4	The performance of different conventional copy-move detection methods on three common copy-move dataset.	17
3.1	This thesis reports results regarding six commonly used image manipulation datasets.	33
3.2	Performance of the proposed method and the state-of-art models (The best and the second-best methods for each dataset are shown in red and blue , respectively).	36
3.3	The image-level performance of proposed method and backbone on different error levels or noise levels of manipulated images.	38
3.4	The similarity score between ground truth masks and error levels of manipulated images, of which the compression quality factor ranging from 10 to 90; where Cos is the Cosine Distance between two masks while Dis is the self-defined Distance Score; The optimal performance is marked as red and the sub-optimal performance is blue	41
3.5	The Distance Score between GT and error levels of pristine images; The optimal performance is marked as red and the sub-optimal performance is blue	41
3.6	Performance of WCBnet on three subsets of CASIA2.0, regarding three different input types	42
3.7	Ablation study of peak-shaped feature reshaping, where the data in the table represents F1-score; The top performance is marked as red	43
3.8	Ablation study of peak-shaped feature reshaping, where the data represents AUC; The top performance is marked as red	43
3.9	Comparison of RefNet with Partial and Full CBWM on CASIA2.0 subsets; RefNet- <i>i</i> models are equipped with partial CBWM structures.	44
3.10	Ablation study of CBWM on SEL results across multiple testing datasets; RefNet- <i>i</i> models are equipped with partial CBWM structures; The top performance is marked in red	44
3.11	The image-wise performance of the backbone and WCBnet on the CASIA2.0 dataset with different perturbations; Signal-to-noise ratio (SNR) of the examples in Figure 3.11 is also included.	46

3.12	The image-level performance of WCBnet on Deepfake datasets regarding different reshaping peaks; The best performance is marked as red	47
4.1	Image-level performance of the proposed method and the state-of-art models; The best and the second-best methods for each dataset are shown in red and blue , respectively); This table is extended from Table 3.2 of Chapter 3	61
4.2	The comparison of histogram similarity, Distance_S represents the distance between background and manipulated foreground while Distance_A is between background and authentic foreground.	62
4.3	The image-level F1-score of backbone network, WCBnet and DenseWCBnet, on three CASIA2.0 subsets regarding RGB, ELA and SEL respectively; The best performance is marked as red	64
4.4	The image-level performance of proposed method and backbone on different error levels or noise levels of manipulated images; The best performance is marked as red ; This table is extended from Table 3.3 of Chapter 3.	64
4.5	The image-wise performance of the backbone, WCBnet and DenseWCBnet on the CASIA2.0 with different perturbations; The best performance for each perturbation method is marked as red ; This table is extended from Table 3.11 of Chapter 3.	69
4.6	The image-level performance of DenseWCBnet on Deepfake datasets, compared with the best model in DeepfakeArt challenge and optimal WCBnet (shown in the Table 3.12 of Chapter 3); The best performance is marked as red	69
5.1	The pixel-wise performance comparison between the backbone network, proposed method and a fully-supervised ManTraNet; In which the weakly supervised method combines image-wise models and segmentation models.	84

Chapter 1

Introduction

1.1 Overview

The primary aim of my PhD research is to investigate the potential of Artificial Intelligence (AI), specifically deep-learning models in advancing the field of image manipulation detection. This chapter begins by outlining the fundamental theories and methodologies underlying current image manipulation detection approaches, followed by an in-depth discussion of the key challenges and limitations of existing models. These challenges serve as the foundation for the research motivation, which leads to specific research aims and objectives designed to address these limitations.

1.2 Background and Motivation

In modern society, social media has become an integral platform for individuals to share thoughts, acquire information, and communicate. While this mode of interaction is faster and more efficient than traditional face-to-face communication, the private images and videos shared online are increasingly vulnerable to multi-media visual content manipulation by malicious attackers [1, 2]. Such manipulation can lead to the dissemination of misleading or even harmful information through the addition, alteration, or removal of objects within the image. With the proliferation of available tools, users without background in image processing can easily alter multimedia content. This growing threat underscores the urgent need for robust manipulation detection techniques to ensure the authenticity and security of digital media [3, 4, 1].

Image manipulations could generally be divided into two categories based on the techniques that attackers use, which are conventional and learning-based. The former means that attackers manually change the visual contents in the image using Photoshop, Adobe After Effects and so on [5]. The commonly used types of conventional image manipulation are **Splicing** that inserts objects or regions from a source image into another background, **Copy-move** that duplicates partial region within the same image, **Removal** that covers the objects by copied background [6]. Fake multimedia contents generated from such manipulation type remain the biggest trouble, because of its convenience and dissemination. The later learning-based type, based on Artificial Intelligence techniques such as OpenAI, MidJourney and Stable Diffusion, has brought a revolution to image manipulations and greatly increased the difficulty of manipulation detection [7]. Such AI tools could learn from massive real-world images and generate totally new contents that have the similar characteristics as real ones. The learning-based manipulation, also called **Deepfake**, with the increase in popularity and the development of learning models, has

gradually become the most difficult challenge for image manipulation detection methods [5].

To address both image manipulation types, various methods have been developed to reveal the imperfect manual operation traces or defeat the learning-based manipulations using more powerful learning-based models. The conventional manual manipulations unavoidably leave specific clues that could be tracked to recognize manipulated images. Consequently, several hand-crafted features, such as frequency-domain distributions, textures, JPEG compression errors, ..., are extracted from images to detect the anomaly or inconsistency existing in conventional manipulated regions [8, 9, 10]. Moreover, majority of image manipulation detection methods are built upon learning-based models because of its ability to automatically learn and extract specific features that distinguish between manipulated and pristine images. What makes learning-based manipulation detection unique from other image processing problems is that the detailed information that contains the fragile manipulation traces plays vital roles as abstract global information extracted by deep-learning models [11]. The local information from manipulated images is re-involved by 1) combining the hand-crafted features as supplements; 2) applying attention module to enhance the spatial or channel representation of features; 3) fusing the global or local features extracted by shallow or deep layers of learning-based models [12, 13, 14].

However, there still exist several challenges or shortcomings regarding the existing either hand-crafted or learning-based image manipulation detection methods. The hand-crafted features are mostly extracted to expose traces of specific image characteristics, which are hardly adaptive to various manipulation datasets. The variety of image manipulation datasets also have negative influence on learning-based methods. The low- or high- level features extracted from different layers of learning models play different roles when dealing with these datasets of different image resolutions, sizes or operations, while they are equally fused or not even considered in existing methods [15, 16]. Moreover, most image-level learning-based models that achieves high classification accuracy lacks explainability to show the reason behind decisions [17]. Pixel-level fully-supervised models that could localize the manipulation regions are limited to the conditions that, pixel-level manually annotations are necessary for their training but are rare for most of manipulation datasets.

1.3 Aims and Objectives of the Research

To tackle the mentioned challenges of existing image manipulation methods, this study divides the PhD research into three objectives from building a fast fact-checking image manipulation detection model to the aim of achieving explainability behind it. The main objectives of the research are defined as:

- Image-level manipulation detection method for fast fact-checking:

The first aim is to design an image-level manipulation detection model as a fast fact-checking tool. It is achieved by deep-learning model which could learn the proper features that differentiate between manipulated and pristine images, along with proper hand-crafted feature extraction methods containing delicate conventional manipulation traces;

- Explainability of image-level method:

The next aim is to reveal the reasons behind image-level models when detecting manipulations, via different Class Activation Map (CAMs) methods [17]. The ex-

plainability of specific model component, such as feature fusion, is also visualized in order to demonstrate the validity of the proposed method and increase the persuasiveness of the image-level predictions;

- Weakly-supervised image manipulation localization without manual annotations:

The final aim is to achieve pixel-level manipulation localization, only based on the image-level models' results rather than applying pixel-level annotations. Since CAMs show models' focus when making decisions, it is applicable as coarse manipulation heatmaps as long as image-level models capture the relevant manipulation traces. The coarse maps are refined by other non-semantic technique for the final purpose of image manipulation localization.

1.4 Main Contributions

This thesis introduces a series of deep learning-based models for tackling both image-level manipulation detection and pixel-level manipulation localization, with a focus on improving detection accuracy, computational efficiency, and interpretability. The key contributions are summarized as follows:

- **Image-level manipulation detection model: WCBnet**

A novel convolutional neural network, *WCBnet*, is introduced, representing the first architecture to explicitly model the differing receptive fields and semantic roles of convolutional blocks. This is achieved through a proposed **Cross-Block Weighting Module (CBWM)**, which adaptively fuses multi-level features by assigning distinct weights to each convolutional block output. Despite introducing only 2.3% additional trainable parameters relative to the backbone, the model demonstrates substantial improvements in manipulation classification accuracy across a variety of datasets. These improvements are also robust under several common perturbations and promising against a challenging learning-based manipulations by generative AI.

- **Enhanced model DenseWCBnet and explainability via multi-view feature fusion**

To further enhance performance, the *DenseWCBnet* model is proposed by densely connecting multiple CBWMs, capturing a more comprehensive and diversified feature representation through multi-view fusion. This enhancement yields improved accuracy and robustness across multiple benchmark datasets. Moreover, the model introduces a novel explainability pathway by densely fusing class attention maps across views, effectively bridging image-level classification with pixel-level localization. This enables the generation of coarse manipulation heatmaps without the need for precise pixel-level annotations.

- **Weakly-supervised localization framework: WSWCBnet**

Building on the reliable heatmaps generated by DenseWCBnet, the *WSWCBnet* model introduces a weakly-supervised localization mechanism that leverages these heatmaps as priors. Under the assumption that image-level classifiers have learned to capture manipulation traces accurately, the method treats the class activation maps as probabilistic evidence. Using principles from Bayesian Inference, these

coarse maps are refined by incorporating object region masks from pre-trained, non-semantic segmentation models. This fusion results in boundary-enhanced manipulation localization, achieving performance comparable to fully-supervised methods, despite operating in a weakly-supervised setting.

1.5 Thesis Structure

The thesis is structured into seven chapters, each addressing a core aspect of image manipulation detection and localization through proposed deep-learning models.

Chapter 2 introduces the background of image manipulation, including both conventional and learning-based manipulation techniques. Definitions, categories, and widely-used benchmark datasets are discussed, with a particular focus on *intentional content-changing manipulations*, which form the primary scope of this research. The chapter further categorizes detection approaches based on the manipulation types, including methods leveraging hand-crafted features and recent deep neural network architectures. The chapter concludes by highlighting open challenges and formulating key research questions that motivate the proposed work.

Chapter 3 presents the first proposed model, WCBnet, a CNN-based image-level manipulation detection method. By introducing a novel Cross-Block Weighting Module (CBWM) at the convolution block level, WCBnet effectively captures the differences in receptive fields and semantic roles across hierarchical feature layers. Despite introducing only a 2.3% increase in trainable parameters compared to the backbone, WCBnet demonstrates significant improvements in classification accuracy and robustness under various perturbations and generative AI-generated content.

Chapter 4 introduces DenseWCBnet, which further enhances WCBnet by fusing multiple CBWM-generated feature views for more comprehensive feature representation. This structure not only improves classification performance across multiple datasets but also bridges image-level classification and pixel-level localization by generating coarse manipulation heatmaps through the dense fusion of class attention maps, even without pixel-level ground truth annotations.

Chapter 5 focuses on weakly-supervised image manipulation localization via WSWCBnet. This model assumes that reliable manipulation traces can be learned at the image level. It uses DenseWCBnet-generated heatmaps as prior knowledge and refines them through Bayesian inference by integrating object-level segmentation maps. This fusion of image-level priors with non-semantic segmentation boundaries enables WSWCBnet to achieve localization accuracy comparable to fully supervised methods, without requiring pixel-wise labels.

Chapter 6 summarizes the findings of the research. Each research question is revisited and addressed based on experimental evidence, validating the proposed models' effectiveness and their ability to overcome previously identified challenges. The chapter also discusses limitations and future research directions.

The final chapter contains the bibliography and supplemental materials. These include additional segmentation maps, enhanced manipulation heatmaps, and negative fusion examples from the WSWCBnet pipeline, supporting a more complete interpretation of the models and results.

1.6 Publications Associated with This Thesis

This thesis is partially based on the following three research articles, which have been published or submitted during the course of this PhD work:

- **WCBnet: Weighted Convolutional Block Modelling of Signed-Value Error Levels for Image-Wise Copy-Move and Splicing Detection**

In Proceedings of the IEEE 25th International Workshop on Multimedia Signal Processing (MMSP 2023).

Related Content: Chapter 3.

Reference: [18]

- **DenseWCBnet: Densely Weighting Convolutional Blocks for Modelling Error Levels of Manipulated Images**

Status: Submitted to IEEE Transactions on Information Forensics and Security.

Related Content: Chapter 4.

- **WSWCBnet: Weakly-supervised Localization of Manipulated Image Regions Using Multi-resolution Learned Features**

In Proceedings of the Workshop on Media Authenticity in the Age of Artificial Intelligence, The 35th British Machine Vision Conference (BMVC 2024).

Related Content: Chapter 5.

Chapter 2

Literature Review

Image manipulation has evolved significantly in various applications such as photography, media, and digital forensics in modern society. From a technical and functional perspective, image manipulations can be broadly categorized into non-intentional image-polishing edits (e.g., contrast enhancement, resizing, blurring) and intentional content-changing manipulations (e.g., splicing, copy-move, object removal, and generative AI-based alterations) [19]. This thesis specifically focuses on the latter, which refers to **image manipulation** that intentionally change the visual content of an image, often with the intent to deceive or misrepresent factual content. These manipulations pose a growing threat to the authenticity and trustworthiness of digital multi-media content, especially given the increasing ease with which such alterations can be made using advanced tools.

Within the scope of this research, **image manipulation** refers specifically to operations that are either conventional or learning-based which intentionally alter the semantics, structure, or spatial consistency of images. With the development of modern software and mobile applications, it has become highly convenient for users to edit image information manually. In parallel, the rise of deep learning-based tools such as Generative Adversarial Networks (GANs) has enabled the creation of synthetic or modified content that is often indistinguishable from genuine images [20]. These advanced techniques have transformed the manipulation landscape, leading to sophisticated alterations that challenge human perception and digital forensic methods alike. As illustrated in Figure 2.1, manipulations can be classified first by generation method such as conventional or deep-learning-based, which are further categorized by specific manipulation goals or tools. For instance, splicing introduces foreign objects into a scene to generate misleading information, whereas StyleGAN performs artistic or identity-based transformations through generative modeling [21]. This chapter provides an in-depth review of both categories of manipulation techniques and their corresponding detection strategies, highlighting existing challenges and gaps that motivate the research.

2.1 Image Manipulation

In the context of this research, **image manipulation** refers to intentional alterations of visual content that affect the semantics, spatial structure, or contextual integrity of an image. Such manipulations are facilitated by both conventional editing tools (e.g., splicing, copy-move, object removal) and modern deep learning techniques (e.g., GAN-based generation), making it increasingly difficult to distinguish authentic images from tampered ones. With the widespread availability of editing software and AI-powered applications, manipulated images can be created or modified with minimal expertise, posing serious

threats to media authenticity and digital trust [5]. As illustrated in Figure 2.1, image manipulation techniques can be broadly classified into conventional and learning-based categories, each encompassing various manipulation types and use cases [19]. These manipulations not only challenge human perception but also raise the demand for robust and explainable detection frameworks, which are explored in the following sections.

2.2 Image Manipulation

In the context of this research, **image manipulation** specifically refers to manual operations that intentionally alter the original characteristics of visual content, such as its semantics, structure, or spatial consistency. It is very convenient in modern society for consumers or attackers to modify the image information by well-developed software or applications in mobile phone. Furthermore, the application of deep-learning manipulation tools (such as Generative Adversarial Network) makes it available to modify or even create false image content that cannot be perceived by the human eyes [20]. This has become a threat to personal data on social platform, authenticity of multi-media visual contents, and security of cyber society. As shown in the Figure 2.1, these image manipulations could be generally divided into two categories according to the manipulation operators which are deep-learning and conventional methods, and further divided according to specific purposes or tools. For example, image Splicing inserts an object into a background generating additional fake information while StyleGAN transfers the art style of the whole image through generative learning-based model [21].

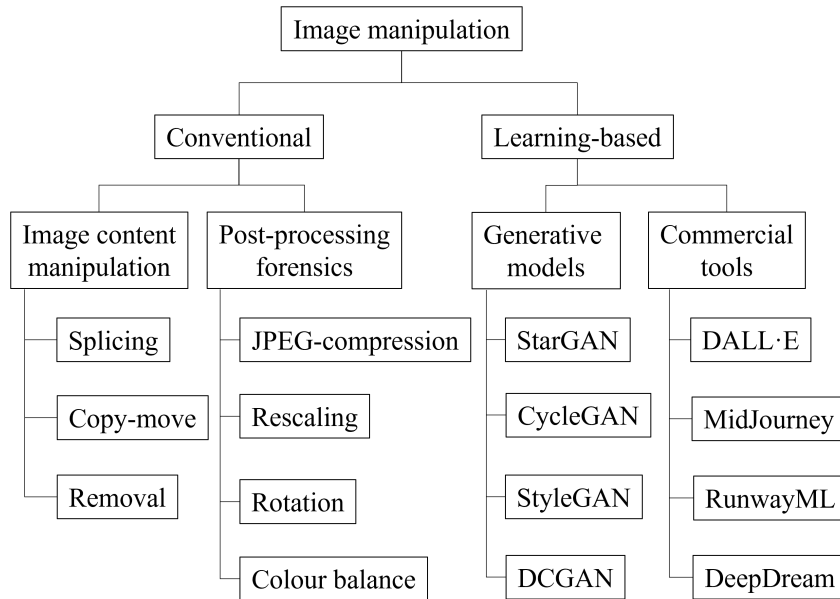


Figure 2.1: *The categories of common image manipulation methods*

2.2.1 Conventional Image Manipulation Methods

Conventional image manipulation methods encompass a range of techniques from simple geometric transformations to complex compositing processes [11]. Any users without any background knowledge can easily tamper with multimedia information through ready-made conventional tools (such as Photoshop[®], Adobe After Effects[®]) or mobile phone

software. Since it is easy to operate and could cause massive fake information, the conventional content-changing image manipulations and following post-processing that retouches operation traces are the core problems in image manipulation detection. Among them, the most common and harmful image manipulations that produce fake, distorted, misleading, and fabricated visual content are splicing, copy-move and object removal [22].

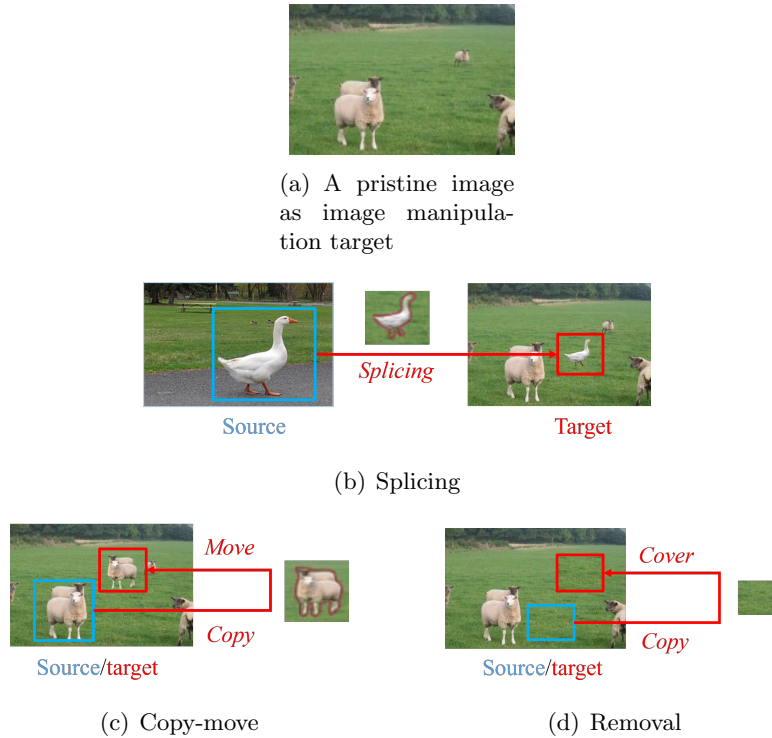


Figure 2.2: Example of three most conventional image manipulation methods

Image splicing involves combines elements from different image sources like inserting an area or object from one image to another background. This technique can introduce inconsistencies in lighting, shadows, and textures, leaving traceable manipulation traces between regions from different sources. Image copy-move , another common manipulation method, involves copying a region from an image and pasting it elsewhere within the same image. This technique is typically used to conceal or duplicate objects, leveraging the existing content to avoid discrepancies in color and noise patterns, thus complicating the detection process. Object removal, or in-painting, is the process of erasing an object from an image and filling the void with plausible background information. An example shown in the Figure 2.2 intuitively compares these three manipulation types. For splicing manipulation, an area or object is cropped and inserted in another background while the tampered object in the image is copied and pasted to different areas on the same picture by copy-move manipulation. Contrary to the copy-move operation, object removal is copying the background area and moving it on target object for covering. Some other conventional manipulations do not directly modify the image contents but have negative effect on the manipulation traces afterwards, which are called post-processing operations. For example, compression during image uploading or saving vanishes the detailed edge information while rotating and resizing changes the original coordinate system that affects the ability of manipulation detection methods.

A list of commonly-used conventional image manipulation datasets and their basic

information, including image number, size, current best performance and so on are shown in the Table 2.1 while several samples from different conventional image manipulation datasets are illustrated in Figure 2.3 and 2.4. These reveal clear differences in image size, resolution, the complexity of manipulation operations, as well as the size and type of manipulated objects. These variations introduce significant challenges in designing and building robust image manipulation detection networks. In summary, traditional image manipulation methods remain the most widespread and accessible techniques for creating falsified visual content, as they directly alter meaningful elements within images to generate deceptive information. At the same time, the post-processing operations, such as sharpening, whitening, shadow removal, and de-noising, which can affect the quality and characteristics of manipulations, should also be considered when evaluating the robustness of image manipulation detection methods. To accommodate the diverse attributes of these datasets, convolutional neural networks must possess the ability to adapt their receptive fields, allowing them to effectively capture and learn image information across different levels of detail.

Table 2.1: *The parameters of common conventional image manipulation datasets, and the best performance of current methods over them. The number in this chart means the number of pristine and forged images respectively.*

Dataset	Category	Number	Size	Best accuracy	Post-processing	Detection method
Columbia[23]	Splicing	182&180	757*568 to 1152*768	90.04%	None	Camera CRF features and RBF kernel SVM classification [24]
CASIA1.0[25]	Splicing& Copy-move	800&921	374*256	85.60%	Pre-processing	Image chroma component, statistical features and SVM classification [24]
CASIA2.0[25]	Splicing& Copy-move	7200&5123	320*240 to 800*800	96.00%	Pre and post processing	Image chroma component, statistical features and SVM classification [24]
DSO-1[26]	Splicing	100&100	2048*1536	86.30%	Color and brightness	Fused input of illuminant, SASI features HOG edges, and RBF kernel SVM classification [27]
Korus[28]	Splicing	220&220	1920*1080	F1-scores =0.69	Shadows or reflections	Use random grid CRF to maximize F1 score [29]
NC2016[30]	Splicing& Copy-move	560&564	500*500 to 5616*3744	79.80%	Post-processing	CNN+SVM [31]
CMFDdb[32]	Copy-move	80&80	1024*768	98.16%	Arbitrary small size forgeries	Zernike features and kd-tree matching [33]
CoMoFoD[34]	Copy-move	260&260	512*512 to 3000*2000	F1-score =0.9726	Resizing, rotation, compression translation, distortion	DCT features extraction, lexicographical sorting and Euclidean distance comparison [35]
Coverage[36]	Copy-move	100&100	400*486	93.8%	Translation, scaling, rotation, distortion.	Dense field features, fast approximate nearest-neighbour and PatchMatch [35]



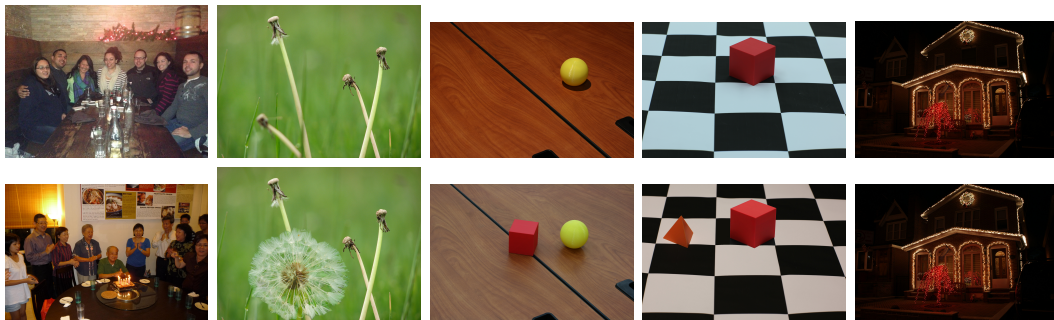
(a) Columbia (top: pristine, bottom: manipulated)



(b) CASIA (top: pristine, bottom: manipulated)

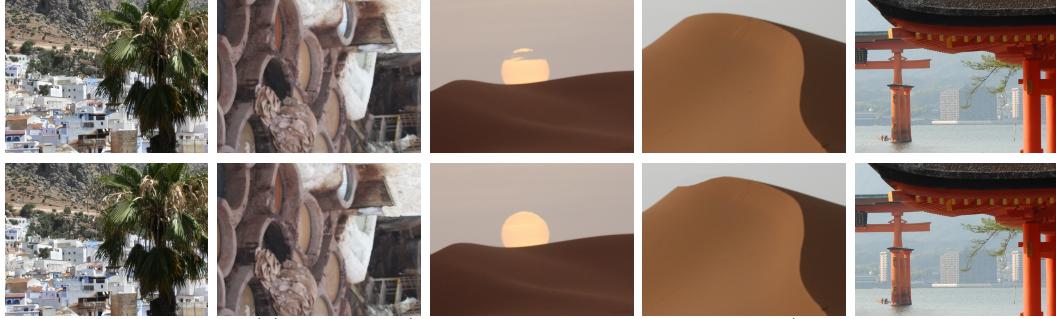


(c) Korus (top: pristine, bottom: manipulated)



(d) NC2016 (top: pristine, bottom: manipulated)

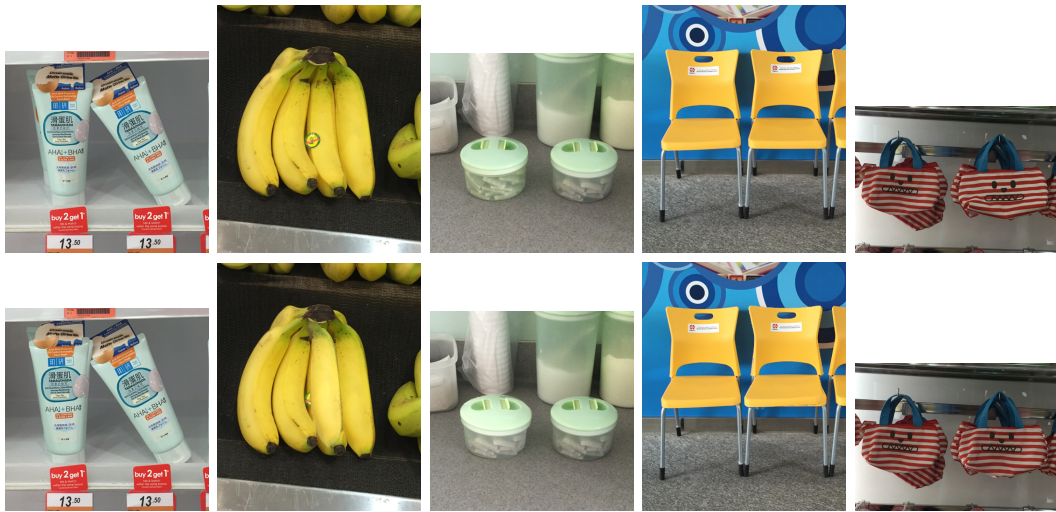
Figure 2.3: Several samples of different conventional image manipulation datasets.



(a) CMFDdb (top: pristine, bottom: manipulated)



(b) CoMoFoD (top: pristine, bottom: manipulated)



(c) Coverage (top: pristine, bottom: manipulated)

Figure 2.4: *More samples of different conventional image manipulation datasets.*

2.2.2 Non-intentional Image Forensics

Although image manipulation detection in this study aims to identify intentionally altered content, numerous studies have shown that unintentional operations which commonly introduced during image acquisition, transmission, or storage, can destroy manipulation traces and affect detection performance. These operations, often referred as post-processing perturbations, include JPEG compression, Gaussian blurring, image rescaling, and so on [37].

Several prior works have adopted such perturbations to evaluate the robustness of manipulation detection models. For example, JPEG re-compression and Gaussian noise was used to test a CNN-based tampering detector, revealing significant performance drops [38]. Another two post-processing techniques, namely blurring and resizing was applied as perturbations and showed that manipulation cues were less detectable, especially for copy-move and splicing attacks under such perturbations [39]. The need for robustness evaluation regarding various post-processing operations was emphasized when deploying detectors in uncontrolled environments [40].

In this thesis, such post-processing operations are explicitly treated as realistic perturbations to test the robustness of proposed models against non-manipulative degradations. This evaluation helps validate the practical viability of manipulation detectors, especially when deployed in real-world scenarios where non-intentional forensics are commonplace.

2.2.3 Learning-based Image Manipulation Methods

Learning-based image manipulation refers to the use of deep neural networks to autonomously learn from large datasets and generate new visual content or modify existing images. This process involves training models to understand and replicate the underlying patterns and structures of input data, enabling the creation of realistic and contextually relevant imagery [41]. Generative AI refers to a class of artificial intelligence techniques, such as Generative Adversarial Networks (GANs) [42] and Diffusion Models [7] have introduced unprecedented capabilities in automation, precision, and the generation of highly realistic manipulated images.

Generative Adversarial Networks (GANs) Generative Adversarial Networks (GANs), for example, can transform Gaussian-distributed noise into data that closely resembles the input data, producing a continuous stream of realistic 'synthetic data'. As shown in the Figure 2.5, the core concept of GANs involves two components: a Generator, which creates forged images, and a Discriminator, which classifies these generated images [43]. In essence, the Generator strives to produce images as realistic as possible, while the Discriminator aims to accurately distinguish between fake and genuine images. Through this adversarial process, highly convincing fake visual content that is nearly indistinguishable to the human eye is generated.

Significant research has been devoted to generating fake human faces, which inherently contain sensitive personal identity information. By learning the distribution of features from real human faces, GANs can produce highly realistic synthetic faces indistinguishable from genuine ones [6], alter specific facial attributes (such as skin tone, hair color, or gender) [44], and even transfer the facial identity of one individual onto another (known as DeepFake[45]). Various GAN architectures have been developed to enhance training stability and improve the quality of the generated images. For instance, DCGAN [20] replaces the pooling layers in traditional GANs with convolutional layers of longer stride, while WGAN [46] introduces a new loss function to measure the distance between real and

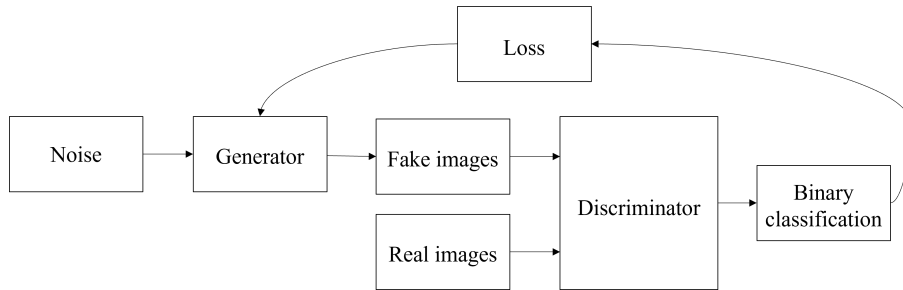


Figure 2.5: *The basic Generator and Discriminator of GAN.*



Figure 2.6: *Fake zebra images using trained Cycle-GAN, and the upper row shows the original horse images while the bottom row shows the transferred zebra images.*

fake images, as opposed to relying on cross-entropy. BigGAN utilizes larger batch sizes and class-conditional training to produce highly detailed images at large scales, making it a leading method in high-quality image synthesis [47]. StyleGAN3 improves on the temporal consistency issues observed in previous models by incorporating alias-free designs, ensuring smoother transitions and better geometric alignment across frames [21]. Different GAN architectures have also been designed for specific manipulation tasks. For example, style transfer methods typically require pairs of pristine and forged images, but CycleGAN [48] eliminates this requirement by using two generators and two discriminators. In the CycleGAN framework, images generated by Generator A are processed by Generator B before being evaluated by the corresponding discriminators. Figure 2.6 illustrates a typical CycleGAN application where horses are transformed into zebras. Another example is StarGAN [49], which accepts both target domain labels and input images to determine not only whether an image is real or fake, but also to identify the domain or style of the input image. This allows StarGAN to achieve multi-domain image translation with a single generator, enabling bidirectional conversion between multiple domains. SinGAN takes a novel approach by being trainable on a single image, enabling diverse image manipulation tasks such as image editing, harmonization, and super-resolution without the need for extensive datasets [50]. Furthermore, TransGAN, which replaces convolutional layers with transformers, captures long-range dependencies more effectively, generating images with globally coherent structures [51]. Collectively, these innovations have significantly broadened the scope of GAN-based image manipulation and deepfake generation, offering enhanced versatility and performance across a wide range of tasks.

In summary, GAN-based image manipulation generates new, false visual content

through deep-learning networks. These methods progressively create synthetic content that closely mimics natural images, making them increasingly difficult to distinguish from authentic visuals. This continual refinement is achieved through iterative training and learning processes. Table 2.2 provides an overview of the most prominent datasets used for training and evaluating learning-based image manipulation techniques.

Table 2.2: *The common available image/video manipulation dataset that are generated by deep-learning methods for different purposes.*

	Dataset	Manipulations	Number	Best accuracy	Detection Methods
Image	DFFD[52]	Various	300000	100%	Xception and attention mechanism [52]
	iFakeFaceDB[53]	StyleGAN&PGGAN	320000	92.25% on average	Xception
	PGGAN[54]	PGGAN	80000	99.97%	Xception
Video	FaceForensics++[55]	Deepfake, CGI, faceswap	4000	100%	Mesosopic Features, Steganalysis Features and Deep Learning Features [55]
	DF-TIMIT[56]	Faceswap	620	99.90%	warped facial features and VGG16 [57]
	Celeb-DF[58]	Deepfake	5639	83.60%	Xception focusing entire facial regions [59]
	DFDC-preview[60]	DeepFake, CGI	4113	93.00%	Xception face detection [60]
	UADFV[61]	Fakeapp	49	100%	Xception and capsule network focusing entire facial features [59]

Commercial AI Tools These AI models are packaged as commercial AI tools which simplify the application complexity, providing an intuitive experience that can be harnessed for various commercial applications. Packaged commercial AI tools, such as DALL·E, MidJourney, RunwayML, and DeepDream, are typically offered as products or services with accessible interfaces that make advanced AI capabilities available to a broad audience, including artists, designers, and content creators.

DALL·E, developed by OpenAI, is a transformer-based model capable of generating detailed images from textual descriptions. It combines deep learning techniques with natural language processing to create a wide variety of images, often incorporating imaginative and abstract concepts. Its greatest innovation lies in the ability for users to generate detailed images simply by inputting a vague textual description. Its ability to blend text and image generation has broadened the possibilities for creative tasks like art and design. MidJourney is another text-to-image tool that generates visually rich and highly artistic images from textual descriptions, developed on variations of GANs and diffusion models. Similar to DALL·E, it allows users to enter prompts and receive corresponding artwork. MidJourney distinguishes itself by its remarkable ability to create images with a painterly or surreal quality, making it particularly valuable for creative endeavors such as concept art, design, and storytelling. This tool empowers both artists and non-artists to explore visual storytelling through AI, fostering creativity and innovation in their projects. DeepDream is an AI tool that utilizes convolutional neural networks (CNNs) to create mesmerizing, dream-like visuals by enhancing patterns within images. It amplifies specific features or patterns that already exist, resulting in surreal and psychedelic effects. Although DeepDream is not designed for realistic image generation like GANs, it has played a crucial role in understanding how AI perceives images and patterns. This tool has gained popularity for its creative and artistic applications. RunwayML is a platform that makes advanced machine learning models accessible to artists, designers, and creators. One of its features includes text-to-image generation, allowing users to produce synthetic visuals using AI. RunwayML integrates various pre-trained models, including

those based on StyleGAN and BigGAN, providing users with the flexibility to create realistic and stylized images for design, art, and marketing.

The proliferation of these user-friendly AI tools has emerged as a significant threat to the authenticity of multimedia visual content, since these tools enable non-experts to easily create highly realistic fake images and videos that can evade human detection. The democratization of such powerful technologies raises critical concerns regarding misinformation and the potential for malicious use in various contexts, including social media and journalism. As a result, the challenges posed by these advancements necessitate the development of robust detection methods and ethical guidelines to mitigate the risks associated with AI-generated content.

2.3 Image Manipulation Detection Methods

Both conventional and AI-based image manipulation detection techniques are inherently imperfect, with manipulated regions often exhibiting subtle inconsistencies or artifacts that deviate from the visual properties of the original image. Image manipulation detection, therefore, seeks to identify and extract these anomalies to reliably differentiate between altered and authentic images, ensuring the preservation of visual content integrity. Generally, these detection methods can be classified into two categories: hand-crafted feature-based approaches and learning-based techniques, depending on the methodology employed to uncover the imperceptible traces of tampering embedded within the image [62].

2.3.1 Hand-crafted Feature-based Manipulation Detection

For hand-crafted feature-based methods, the manipulated images are manually processed to extract the manipulation traces or artifacts that are hardly detectable by human eyes but are distinguishable in other domains [8, 9, 10]. These conventional methods are primarily based on the researchers' prior knowledge in the field of image processing, allowing them to infer image features that are less likely to be falsified. For example, several signal processing methods such as discrete cosine transform (DCT) [63] and Wavelet transform [64] have been applied to expose the tractable traces in frequency domain for image manipulation detection. As illustrated in Section 2.2.1, the detection principles for the two primary types of traditional image manipulation, splicing and copy-move, differ significantly. The former aims to identify discrepancies between the foreground and the background, while the latter seeks to locate the source region identical to the manipulated area. Different hand-crafted features are employed accordingly in splicing and copy-move detection.

Splicing detection Splicing is an operation that extracts objects or regions and insert such targets into another background image in order to falsify non-existent information. Its detection aims to identify discrepancies between the feature distributions of the manipulated regions and the background to uncover the most mismatched areas. As shown in the figure 2.7, splicing detection methods vary depending on the specific features they target. Pun extracts noise features by calculating the noise-level function, which reflects the relationship between noise level and brightness across multiple segmentation scales [74], while Zhang focuses on coefficient features through block-based discrete cosine transform (DCT) over different frequency ranges [75]. Similarly, Zhang uses block-wise DCT to obtain model magnitude components, with additional dimensionality reduction applied via

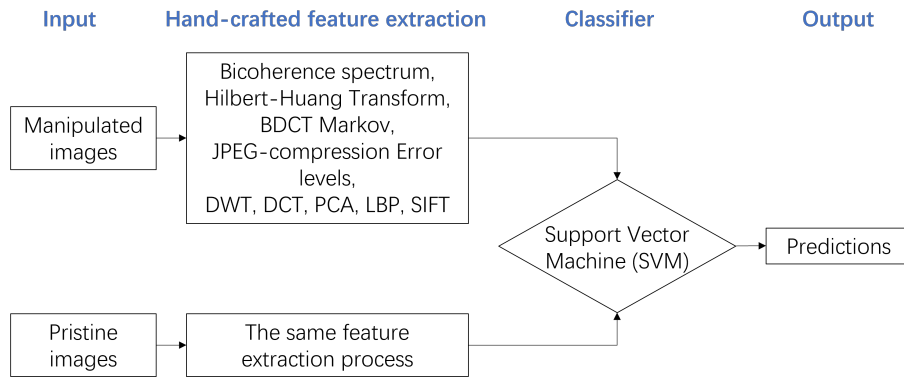


Figure 2.7: General framework of conventional splicing detection

Table 2.3: The classification accuracy of conventional splicing detection methods on three common splicing datasets.

Method	Dataset	Features	Classification model	Accuracy
[65]	Columbia	BDCT Markov & Spatial Markov	SVM+PCA	97.33%
[66]		BDCT Markov & DWT Markov	SVM+RFE	93.55%
[67]		BDCT Markov & 1D, 2D moments	SVM	91.87%
[68]		HHT & DWT	SVM	85.87%
[69]		RLRN & PCA	SVM	88.28%
[70]	CAISA1.0	BDCT Markov & Rake transform	SVM+BFS	98.00%
[71]		LBP & DCT	SVM	97.00%
[72]		SPT & LBP in Cb channel	SVM	94.89%
[73]	CASIA2.0	Markov in Cr channel		95.60%
[66]		BDCT Markov & DWT Markov	SVM+RFE	89.76%
[71]		LBP & DCT	SVM	97.50%

principal component analysis (PCA) [76]. Xu Zhang leverages SIFT features for pixel-level feature extraction, followed by coarse quantization through k-means clustering and fine quantization using optimized product quantization (OPQ) for indexing [77]. Additionally, O. Mayer transforms the similarity between image patches into forensic similarity graphs and utilizes community detection for splicing localization after feature extraction [78]. The Table 2.3 summarizes several state-of-the-art splicing detection methods, highlighting their distinct feature extraction techniques. These approaches have demonstrated strong performance across three commonly used splicing datasets [79]. The table includes the following abbreviations: BDCT (Block Discrete Cosine Transform), RLRN (Run Length and Run Number), LBP (Low Binary Pattern), SPT (Steerable Pyramid Transform), RFE (Recursive Feature Elimination), and BFS (Boosting Feature Selection).

Copy-move Detection Copy-move manipulation involves duplicating a region or object within an image and pasting it elsewhere in the same image. The purpose of this manipulation is similar to splicing: both aim to insert fake information or cover existing content. However, because the copied region comes from the same image, it shares nearly identical attributes with the source area, making it harder to detect using conventional features that rely on differences from device-specific characteristics. Despite this, even if the copied regions are modified through resizing, rotation, or blurring, copy-move detection can still be achieved by comparing features across all regions of the image. The two regions with the highest similarity are identified as the manipulated area and its corresponding source region. As shown in the Figure 2.8, the core component of the copy-move detection process is the selection and extraction of hand-crafted features. A variety of

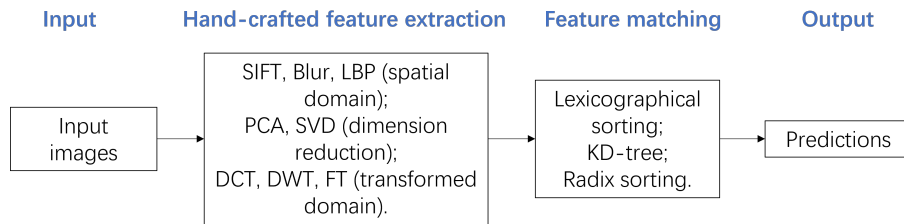


Figure 2.8: General framework of conventional copy-move detection.

Table 2.4: The performance of different conventional copy-move detection methods on three common copy-move dataset.

Method	Dataset	Features	Classification	Accuracy
[80]	MICC-F220	Variant DWT	Mean matrix sorting & Euclidean distance	98.29%
[81]		FFT & DCT	SVM	88.62%
[82]		DCT	kd-tree	79.33%
[83]		SIFT & KAZE feature	2NN and Euclidean distance	78.33%
[84]	MICC-F600	Blur moments & PCA	kd-tree	F1=59.4%
[85]		Block SIFT features	Correlation coefficient	F1=54.4%
[86]		SIFT	Agglomerative hierarchical clustering	F1=42.3%
[87]	FAU	1D colour-dependent features	lexicographic sorting	F1=89.3%
[88]		Low frequency features by Gaussian pyramid decomposition	Euclidean distance of circle regions	F1=90.92%
[89]		SIFT	Best-bin-first algorithm	F1=63.1%
[90]		SURF(Speed Up Robust Features)	Euclidian distance	F1=69.5%

traditional features are typically employed, including block-wise descriptors such as Scale Invariant Feature Transform (SIFT), blur moments, Local Binary Patterns (LBP), Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Fourier Transform (FT). Subsequently, these extracted features from different regions of the image are sequentially compared, in order to identify the region pairs of manipulated and source sequentially comparing the extracted features from different regions of the image [80, 81, 82, 85, 83, 86]. In contrast to splicing detection, copy-move manipulation employs various classification models to identify and sort image block pairs exhibiting the highest similarity. Jen implements a lexicographically sorted matrix to organize statistical features derived from the histogram of oriented gradients [91]. Similarly, Liu applies the same lexicographic sorting model following the extraction of geometric inherent features using the discrete Fourier–Mellin transform [92]. Classification is further enhanced by exploiting the correspondence between the superpixels in both the original and duplicated regions, based on the SIFT features of key points [93]. Additionally, the kd-tree structure enables efficient searching of block pairs and the computation of Euclidean distance for similarity comparisons of the extracted blur invariants [94]. The performance on three common copy-move datasets, namely MICC-F220, MICC-F600, FAU[85, 95] of several hand-crafted feature-based copy-move detection methods with different features and classification models are illustrated in the Table 2.4.

2.3.2 Learning-based Methods for Conventional Manipulations

Convolutional Neural Networks (CNNs) serve as a robust tool for the automatic learning and extraction of features, and they are extensively utilized for classifying manipulated images and identifying tampered regions. Specifically, CNNs are designed to extract op-

timal high-level features that effectively differentiate between manipulated and authentic inputs through a training process [96, 97]. To incorporate deep learning methodologies into image manipulation detection, numerous CNN-based approaches have been tailored to address specific manipulation-related contexts.

Direct Application Initially, CNN models that have demonstrated strong performance in traditional image classification and segmentation tasks are directly adapted as feature extractors for distinguishing between authentic and manipulated images. These CNNs are often trained on image manipulation datasets without any architectural modifications. For instance, the Xception model has been employed as a block-wise feature extractor, followed by two fully connected layers for classification [98]. Similarly, two deep CNN architectures, with 27 and 54 layers respectively, have been utilized to extract discriminative features [99]. In the context of copy-move detection, a 10-layer CNN has been specifically designed for feature extraction, with a subsequent Support Vector Machine (SVM) classifier used to complete the classification process [100].

Classifier for Hand-crafted Features Moreover, CNNs are leveraged as classifiers for pre-extracted hand-crafted features. For example, VGG-16 is trained on features extracted from three categories (copy-move, splicing, and video forgery) to detect and localize pixel-level image manipulations [99]. Another CNN-based classifier is trained on discrete cosine transform (DCT) coefficients to identify manipulated regions [101]. Some CNNs are specifically designed to detect double-JPEG compression traces. In one study, three CNN architectures were compared: LeNet for pixel-domain classification, a 10-layer CNN trained on noise-domain features after DWT transformation, and a third CNN trained on DCT coefficients [102]. Similarly, a CNN architecture with 4 convolutional layers, 4 pooling layers, and 2 fully-connected layers was tested to detect single, double, and triple JPEG compressions [103].

Customized Structures To enhance CNN-based manipulation detection, several structural changes have been introduced to better capture the subtle traces left by image manipulations. Unlike traditional image processing tasks, the low-level features extracted by the shallow layers of CNNs are critical for detecting fine details of manipulation, and thus need to be carefully incorporated alongside higher-level features [11]. One key advancement is the integration of multi-level information, allowing CNNs to handle the diversity and complexity of manipulations. This has led to the development of architectures capable of fusing multi-scale features from various depths of the network. CNN-based detection models achieve this by fusing both global and local features through methods like Feature Pyramid Networks (FPNs). In FPNs, feature maps from different layers are up-sampled and merged, enabling the network to simultaneously capture coarse and fine-grained manipulation traces [12]; Additionally, recent approaches have incorporated self-attention mechanisms, which enhance the spatial representation of feature maps at each level before combining them. This ensures that both positional and semantic information is retained during the fusion process, making manipulation localization more precise [13, 14]; Fully-convolutional neural networks (FCNNs) are also widely used in manipulation detection due to their ability to perform both image-level and pixel-level detection. A prominent example is an advanced multi-task FCNN architecture that includes a VGG-16 branch for learning surface labels and a fully-connected (FC) branch for capturing the edges of spliced regions, improving both detection and localization [104]. Another variation of FCNNs focuses on using residual images—generated by high-pass

filters—along with the original images as input, thereby enhancing the network’s sensitivity to manipulation traces [105]; These structural innovations in CNNs demonstrate how the field has evolved from standard architectures to more specialized models, designed to handle the specific challenges posed by image manipulation detection.

2.3.3 Generative AI (Gen-AI) Detection

However, the detectable traces in the aforementioned image manipulations become ineffective dealing with images generated by learning-based methods, as these methods are trained to create manipulated content that seamlessly matches the characteristics of the background. With the rapid advancement of GANs, deepfake technology, also called Generative AI, has enabled the creation of highly realistic synthetic images and videos that are visually indistinguishable from authentic content. However, despite the visual similarities, GAN-generated images exhibit subtle differences from real images at the pixel level and in high-dimensional feature spaces [6]. The solution lies in utilizing more powerful CNNs, which are trained to acquire the ability to distinguish AI-generated images from pristine ones through learning and training. As a result, deep learning-based image classification techniques have been increasingly employed for Gen-AI detection.

Fingerprint-based Methods Specifically, just as images captured by the same mobile device contain unique Photo-Response Non-Uniformity (PRNU) patterns due to specific hardware, sensors, lenses, and processing parameters, Generative Adversarial Networks (GANs) exhibit distinct ‘fingerprints’ based on their architectures and filters, which can be extracted for manipulation detection. To validate the hypothesis that GANs possess distinct noise ‘fingerprints’, high-pass filters are applied to extract noise residuals from images generated by Cycle-GAN and Progressive GAN. As more residuals are gathered, they converge into stable patterns that act as unique fingerprints for each GAN. Correlation analysis between these residuals shows that the fingerprints are distinct enough to classify images generated by different GANs. Further testing across 22 sets of noise residuals from various GANs (Cycle-GAN, Pro-GAN, and Star-GAN) and real cameras demonstrates that the residuals are most correlated with fingerprints from the same dataset. Even though some similarities exist between GANs with similar architectures, GAN-generated manipulations can be effectively distinguished from real images. Classification is achieved by minimizing the distance between residuals and GAN fingerprints, yielding near-perfect attribution. Comprint is another fingerprint-based method which extracts the noise patterns from GAN-generated datasets as unique fingerprints for each model [106].

CNN-based Classifier One prominent approach involves utilizing more powerful Convolutional Neural Networks (CNNs) as classifiers, trained to automatically extract and distinguish the features of GAN-generated images from those of authentic images [107]. Compared to traditional shallow classifiers, CNNs can capture more complex, high-dimensional features through successive convolutional operations, thereby improving the model’s sensitivity to fine-grained image details. Specifically, CNN models are trained on large datasets consisting of both real and GAN-generated images, enabling them to learn how to identify common artifacts in deepfake images, such as texture inconsistencies, visual artifacts, and lighting discrepancies [108]. Meso-4 consists of four convolutional layers with ReLU activation, batch normalization, pooling layers, and two fully-connected layers with dropout, while its variant, MesoInception-4, replaces the first two layers with Inception modules to enhance spatial feature extraction [16]. Xception is also applied to initially extract

facial regions from input images, which are then passed into a bidirectional Long Short-Term Memory (LSTM) module. This module captures temporal dependencies and learns a latent representation to effectively differentiate between facial manipulations and genuine images [15]. These models demonstrate strong performance not only in detecting images generated by popular GANs but also in exhibiting some generalizability across different datasets and generative models [109]. For deepfake video detection tasks, CNNs can be combined with temporal modeling techniques, such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks, to capture temporal inconsistencies, such as unnatural facial expressions or irregular head movements, in forged video content. This hybrid approach significantly enhances detection accuracy and has become a key area of research in deepfake detection [110, 111]. The incremental classifier is designed to adapt to new classes without retraining on the entire dataset by extracting and averaging feature vectors from images for classification. This structure has also been applied to the detection of GAN-generated images [112]. In GAN-related applications, different GANs are treated as distinct classes, allowing the incremental classifier to perform multi-classification on both current and emerging GAN-generated data. The architecture is tested using various GANs, such as Cycle GAN and ProGAN, and incorporates multiple strategies, including iCaRL and GS-classifier, while retraining models like Xception and DCGAN discriminators with each new dataset addition [113]. By employing more powerful CNN architectures as classifiers, researchers have been able to design robust and sophisticated deepfake detection systems, which are essential for combating increasingly complex forgery techniques and advancing the field of automated deepfake detection [55].

2.4 Challenges of Current Methods

Despite the significant progress made, there are still notable gaps and challenges in the current image manipulation detection fields.

1. Although a variety of manipulation traces have been extracted as hand-crafted features, their effectiveness declines on other datasets with different properties or manipulation types; For example, the methods extracting texture features could not distinguish between the forged region and source region in copy-move images since they own exactly the same texture features;
2. Most backbones of deep-learning methods are pre-trained on ImageNet or built for specific object detection purposes. However, the high-dimensional global semantic information that is important to these networks is not equally important in image manipulation detection tasks or even has side effects to the results to some extent. Both global and local information extracted by these pre-trained networks should be taken into account, however, how to individually and jointly apply these hierarchical features is an ongoing challenge;
3. Deep-learning image manipulation detection methods have demonstrated high image-level classification accuracy, but such black-box methods lack of explainability. It is unknown whether or not image-level decisions are driven by manipulation-related clues. For instance, a model might achieve excellent accuracy by focusing on irrelevant features such as the absence of animals, rather than detecting the actual manipulation artifacts. The explainability of these models is crucial and challenging to evaluate their effectiveness in identifying manipulated images;

4. Pixel-wise deep-learning image manipulation methods could also achieve high manipulation localization accuracy after training, but this often requires a large amount of manually annotated pixel-level labels as training data. In real-world scenarios where such detailed annotations are typically unavailable, achieving effective manipulation localization through weakly-supervised methods remains a significant challenge.

2.5 Research Questions Addressing the Challenges

To address the four challenges illustrated in Section 2.4, a list of research questions progressing learning-based model's structures and exploring wider application are proposed in this section:

1. To address 1st challenge regarding the variety of manipulated traces, how to extract one type of hand-crafted features which universally exist in all (or most of) manipulated images distinguishing manipulated regions from background is the first research question. The selection of optimal hand-crafted features enables the image manipulation detection models to be generally applicable to various manipulation types, which is addressed in Chapters 3 and 4;
2. For the 2nd challenge about the importance of multi-dimensional features in image manipulation detection, it is required to individually consider the roles of hierarchical features extracted by ImageNet-trained CNN backbones, and jointly combine them to adapt to various properties of image manipulation datasets. How to design and develop a CNN-based structure that makes full use of high- or low- dimensional features is the second research question, which is addressed in Chapters 3 and 4;
3. The 3rd challenge is about the explainability of learning-based methods which work like black-box, which leads to the next research question: how to explain the reasons behind decisions? For manipulation detection tasks, explainability includes visualizing models' region of interest in the manipulated images when classifying them as manipulated class. Moreover, the reasons behind any proposed components are required to be explained, such as visualizing the role of mentioned hierarchical features. This explainability-related question is addressed in Chapter 4;
4. For the last challenge of the absence in pixel-level image manipulation annotations, the research question is proposed as: how to build an image manipulation localization model in the absence of pixel-wise annotation labels, while fully-supervised manipulation localization models cannot be trained in this scenario? Methodology introduced in Chapter 5 addresses this question;
5. As discussed in Section 2.3.3, another significant challenge arises from advanced learning-based image manipulation methods, particularly generative models. Ensuring that the proposed image detection methods remain effective on challenging learning-based datasets became our 5th research question, which is addressed as extension context in both Chapters 3 and 4.

2.6 Summary

Image manipulation techniques, both conventional and learning-based, are posing significant challenges to the authenticity of multimedia content and the security of transmitted

information. Among them, techniques like image splicing and copy-move manipulations that are widely accessible and easy-to-produce, and Deepfake methods with increasing sophistication and detection difficulty, are becoming a rising concern. Although deep learning-based methods have shown effectiveness in distinguishing manipulated images from authentic ones, key challenges remain. These include the variety of manipulation datasets, the intensive need for pixel-level annotations, and the black-box mechanism of learning models that lack explainability, which collectively limit the capabilities of existing methods for accurate detection and precise localization.

Chapter 3

Image-wise Manipulation Detection Using CNN-based Method Modelling Error Levels (WCBnet)

In this chapter, an image-wise manipulation detection CNN-based model named WCBnet will be introduced to address the 1st and 2nd research question proposed in Section 2.5. It is used as a fast fact-checking tool in order to classify pristine images and manipulated (specifically, copy-move and splicing) images. To achieve this goal, WCBnet models JPEG compression-based error levels to reveal the disparities between the manipulated regions and the background areas, using the weighted fusion of the multi-level hierarchical features extracted by the backbone CNN.

3.1 Introduction

Combating the spread of visual mis/disinformation remains a challenge in current social media fact checking services and applications due to the fast growth of the presence of manipulated social media images in the contexts of current geopolitical affairs [1, 5, 2]. These images are vulnerable to the rapid advancement of both conventional and learning-based manipulation techniques. The former involves manually altering image contents or properties using image editing tools, while the latter entails generating new visual content using deep learning models [114, 115]. The most common operations in conventional image manipulation include inserting objects from one image into the background of another (splicing) or copying regions within an image and pasting them elsewhere in the same image (copy-move) [116]. These easy-to-perform and hard-to-detect manipulation methods pose a significant threat to image authenticity in the current era of information explosion. A substantial body of work has been proposed and refined for image-level manipulation detection [117] and pixel-level manipulation localization [118, 119]. In this chapter, our focus is on image-level classification of manipulated versus authentic images, specifically within the application domain of fact-checking large sets of images without pixel-level labels. This context necessitates the rapid verification of the authenticity of individual images.

This chapter presents an efficient CNN-based approach, WCBnet, starts with extracting learned features from the signed-value error levels (SEL) of JPEG-compressed

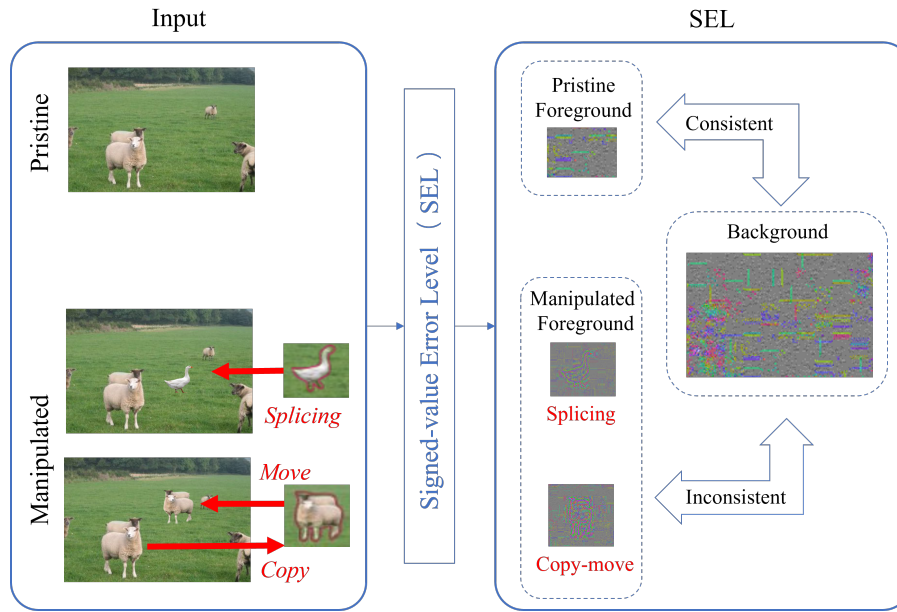


Figure 3.1: Error levels are considered as manipulated traces that distinguish between manipulated regions and pristine background; The red annotations illustrate the two common conventional image manipulation techniques: copy-move and splicing.

images on hierarchical convolution blocks. This is followed by adaptively concatenating, weighting and fusing these multi-level features by considering self-attention over all blocks according to different error levels corresponding to different manipulation types. The performance of the proposed approach is evaluated on common manipulation datasets and compared with the state-of-the-art. WCBnet trained using around 2500 images of CASIA 2.0 dataset, resulted in the best F1-score for CASIA 1.0, Defacto, Coverage and Columbia datasets after fine-tuning by a small portion of those datasets. On average WCBnet improves the F1 score with respect to the second-best performing methods by 27.5%, 34.3%, 16.2% and 6.1% for these four datasets, respectively. There are three main works that have been done to achieve image-wise manipulation detection:

- Proposal of a CNN-based model called WCBnet that can adjust the use of the multi-level features from hierarchical convolution blocks, which is achieved by a cross convolutional-blocks weighting module; This self-adaptive model performs the best on multiple image forgery datasets containing two manipulation types;
- Design of a flexible feature reshaping module, named peak-shaped reshaping, to customize the uniform shape of the multi-level features according to the needs of different image manipulation detection tasks;
- Involving a pre-processing method to extract compression-based artifacts of manipulated images, by normalizing and retaining the signed values of compression error levels to enhance the representation of small error values, which is called signed-value error levels (SEL).

3.2 Related Work

JPEG Manipulation Detection and the Role of Error Level Analysis (ELA)

JPEG compression is the most prevalent lossy image format used across digital photography, social media, and content distribution platforms. Its widespread adoption makes it a critical focus in multimedia forensics, as manipulated images are often saved or transmitted in JPEG format. JPEG compression introduces characteristic artifacts, such as block discontinuities and quantization patterns, which can serve as valuable forensic cues for detecting inconsistencies introduced by post-processing or malicious manipulation. Numerous JPEG-based manipulation detection techniques have been developed over the years, including analysis of double compression artifacts [120], quantization noise inconsistencies [121], and block boundary anomalies [122]. These methods often rely on statistical modeling, transform-domain features, or learning-based approaches to localize or classify image manipulations.

Among these, *Error Level Analysis* (ELA) stands out as a straightforward interpretable approach. ELA estimates the error residuals between an original JPEG image and its re-compressed version under the same quality factor, producing a map that highlights areas with inconsistent compression characteristics [41]. Compared to other methods that rely on complex statistical priors or compression history estimation, Error level maps have several practical advantages such as easy-to-compute, model-free, and providing direct visual cues for detecting manipulation regions. These properties make it especially suitable for integration with deep-learning models as a sufficient input.

Error-level learning-based methods

The risk of the existing data-driven learning-based approaches is that they suffer greatly from datasets with different properties or manipulation types. To address it, Error level is universally present in all inevitable image compression and re-saving operations, so it has been applied as hand-crafted features to track the JPEG compression-based artifacts during the lossy re-saving process of manipulated images [123, 124, 125]. Specifically, the discernible patterns between areas subjected to different JPEG compression processes and unaffected background regions exhibit varying error levels, thereby revealing traces of manipulation in the error-level domain [126]. During each instance of image manipulation or subsequent lossy compression, data undergoes distortion following a distinctive pattern dictated by specific quantization tables. The error-level inconsistency in compression factors or compression times of manipulated regions and background regions is learned by different deep-learning methods for image manipulation detection. AlexNet 16 is applied to learn the Error-level of CASIA2.0 image manipulation dataset and achieve 90% accuracy after 10-epoch training [127]. The Error-level is augmented using various methods and learned by an auto-encoder in [128], whose performance reaches 95.79% on a copy-move dataset named MICC F220. The application of Error-level is also proven to be effective in image spam classification (spam text is inserted in the images), which achieves best performance over several image spam datasets [129]. As depicted in Fig 3.2, error-level maps vividly illustrate the abundance of JPEG artifacts, effectively pinpointing manipulated regions resulting from both splicing and copy-move operations, while errors of lower magnitude are evenly dispersed across the background areas.

However, existing EL-based learning methods do not adapt the architecture of conventional CNNs to the specific demands of image manipulation detection, as these networks are typically pre-trained and optimized for object classification tasks such as those in Im-

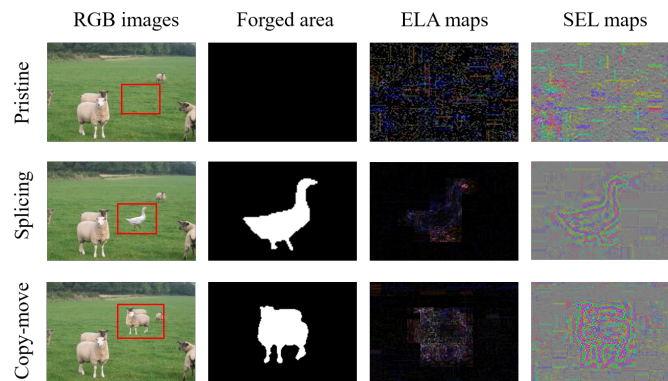


Figure 3.2: Three examples of pristine, splicing and copy-move images, and their corresponding ground truth, EL maps and SEL maps; Forged areas in similar positions are marked with red boxes and zoomed in

ageNet. Therefore, both the network architecture and the nature of the extracted features must be carefully reconsidered and tailored for the task of detecting image manipulations.

Multi-level feature fusion for image manipulation detection

Rather than directly using the high-dimensional global features extracted by the final layers of deep-learning networks, researchers have increasingly focused on the multi-level features extracted by different convolutional layers in CNNs. In the field of image manipulation detection, the local information of subtle manipulation traces present in shallow-layer features are particularly crucial. Most multi-level feature fusion structures follow a U-shaped structure. Beginning with the high-dimensional features from the final layer, the process of 'up-sampling and addition' is iteratively applied to combine the multi-scale output by each convolutional block in the CNNs. In an image manipulation localization method called EMT-Net, the multi-level features from RGB branch are first enhanced by noise-branch features and fused using the U-shaped structure [130]. Similar structure is used in TANet, but the up-sampling process is achieved by Reverse Attention Mining Block [131]. Another feature fusion structure is dense-connection, following the structure of DenseNet. The outputs of all convolutional blocks are rescaled, either through up-sampling or down-sampling, to match the size of each feature map, and are then combined across all spatial dimensions. This structure is applied in HiFi-Net, to fuse the multi-level features densely and be evaluated on a self-made dataset including 13 manipulation methods [132]. In an image forensic detection model SDCN2 [133], the local and global extracted image contents are densely integrated to enhance the residual feature representation.

Although attention modules have been introduced to enhance the output of each convolutional block, existing feature fusion architectures fail to account for the relative relationships between convolutional blocks, specifically the varying importance of multi-level features in image manipulation detection.

3.3 Proposed Method

As shown in Figure 3.3, the proposed model consists of five stages for image-wise manipulation detection. They include 1) Signed-value Error Level (SEL) image pre-processing; 2) Leaned Feature Extractor (a learning-based CNN); 3) adjustable Peak-shaped Feature

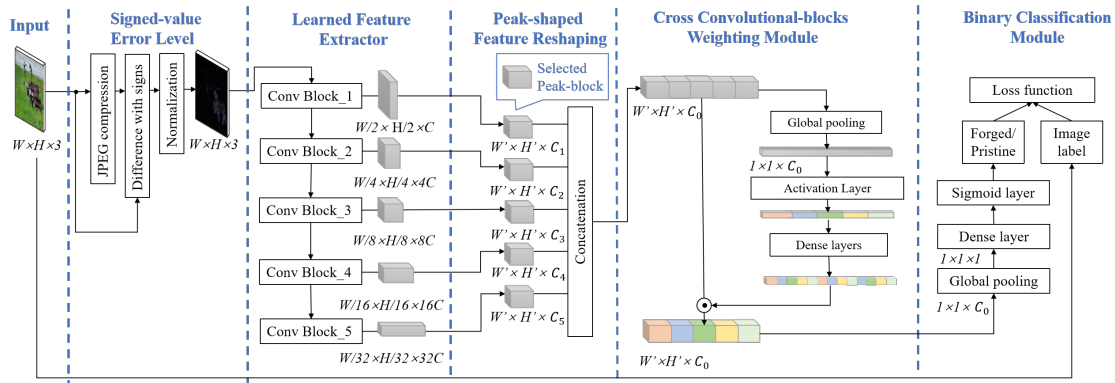


Figure 3.3: The five stages of proposed WCBnet, and different colors of features in the figure represent different weights.

Reshaping process; 4) Cross Convolutional-blocks Weighting Module (CBWM); 5) Binary Classification Module. It shows the whole process of modelling error-level image manipulation traces into binary image classes, and the details of each stage will be introduced in this section.

3.3.1 Signed-value Error Level

Rather than using raw RGB images as input for feature extraction, the present work utilizes error-level (EL) analysis to emphasize artifacts resulting from inconsistent compression processes between forged and background areas. This approach minimizes the excessive image features and distractions typically present in RGB inputs, allowing the network to focus specifically on compression inconsistencies. In JPEG-compressed images, one of the most widely used formats, data loss primarily occurs during the quantization phase, where a uniform quantization table is applied across all segmented image blocks [134, 135]. Applying EL as the pre-processing step enhances the network’s ability to detect subtle JPEG-compression artifacts. In the case of a manipulated image $\mathbf{x} \in \mathbb{R}^{384 \times 384 \times 3}$, disparities exist in data discarding patterns between the pristine background region $\mathbf{x}_p$ and the manipulated region $\mathbf{x}_m$, which typically undergoes additional image processing (e.g., resizing, rotating, enhancing) before being re-compressed with JPEG. Conventional approaches involve re-compressing the image $\mathbf{x}$ as $\mathbf{x}'$ and computing the absolute difference between $\mathbf{x}$ and $\mathbf{x}'$ to determine the error level of data discarding [136, 137]. EL maps highlight areas of an image with significant compression discrepancies, revealing potential manipulations by comparing the original and compressed versions of the image. The fundamental EL maps $E(\mathbf{x})$ is computed by the absolute difference between image $\mathbf{x}$ and compressed result $\mathbf{x}'$:

$$E(\mathbf{x}) = |\mathbf{x} - \mathbf{x}'|. \quad (3.1)$$

Subsequently, image enhancement is performed by computing a scaling factor based on the maximum pixel difference and applying brightness adjustment to $E(\mathbf{x})$ to amplify potential-manipulated regions in the image. After enhancement, the resulting image $E(\mathbf{x})$ is normalized by dividing by its maximum value, ensuring the values are scaled to the range $[0, 255]$:

$$E(\hat{x}) = E(x) \times \frac{255}{\max(\max(|E(x)|), 1)} \quad (3.2)$$

where $E(\hat{\mathbf{x}})$ is the enhanced version of $E(\mathbf{x})$.

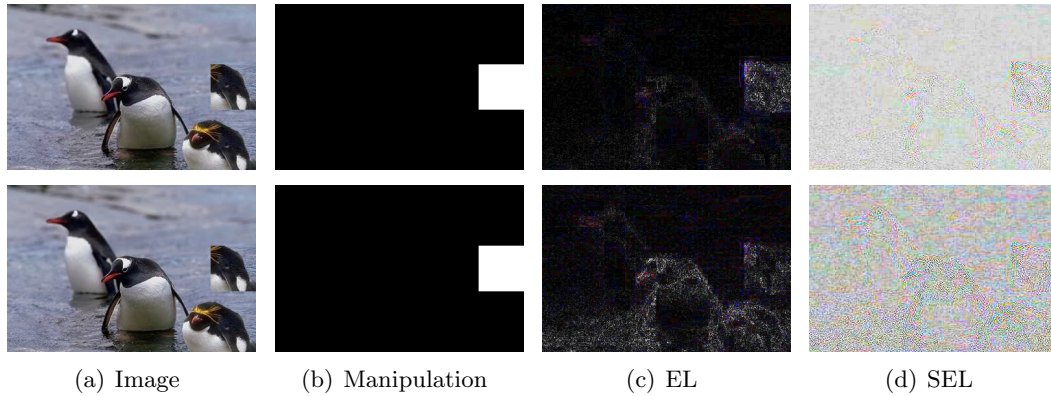


Figure 3.4: The comparison of manipulated images that have different relative compression factors, and their normal Error-level maps (ELA) and proposed Signed-value Error-level (SEL); Top row shows the normal condition that forged regions has lower image quality while the bottom row is the opposite condition; The error-level maps are enhanced for visualization.

Image manipulation detection based on traditional **Error Level** seeks to maximize the discrepancy between the manipulated region ($\mathbf{x}_m$) and the pristine background region ($\mathbf{x}_p$). Typically, manipulated regions undergo additional image editing and lossy JPEG compression, which results in a higher overall level of subtracted errors in the manipulated region $E(\hat{\mathbf{x}}_m)$ compared to the background region $E(\hat{\mathbf{x}}_p)$. This assumption, however, fails in specific cases where the error level artifacts are concentrated more prominently in the pristine background than in the manipulated region. Such situations lead to contradictory feedback for binary classification models, complicating the accurate detection of manipulation. For instance, as illustrated in Figure 3.4, it manually constructs a manipulated image example where the same region is copied-and-pasted, and the background undergoes JPEG compression with varying compression factors. In two distinct scenarios, the absolute error levels between manipulated region x_m and compressed manipulated region x'_m highlight opposite regions. In one case, higher error levels are observed in the pristine background, contrary to the expectation that the tampered region should have higher error levels. This discrepancy demonstrates that traditional EL methods, which assume larger error levels in the manipulated region, do not account for scenarios where the background experiences significant compression artifacts, leading to misleading or inverse feedback in manipulation detection.

To address these limitations, this study proposes the **Signed-value Error Level Analysis (SEL)** method. Unlike conventional EL, SEL retains the **polarity of error** by computing and preserving the signed difference between the original image $\mathbf{x}$ and the compressed version $\mathbf{x}'$. This preserves relative error discrepancies across regions, allowing the method to effectively differentiate between manipulated and background areas even in complex compression scenarios. The signed error difference between the original image $\mathbf{x}$ and its compressed version $\mathbf{x}'$ is calculated as:

$$S(\mathbf{x}) = \mathbf{x} - \mathbf{x}'. \quad (3.3)$$

This operation preserves both positive and negative error polarities (from -255 to 255), ensuring that discrepancies between manipulated and pristine regions are correctly identified, even in situations where the background might exhibit greater errors than the tampered region. Unlike traditional EL, which operates on the absolute value of

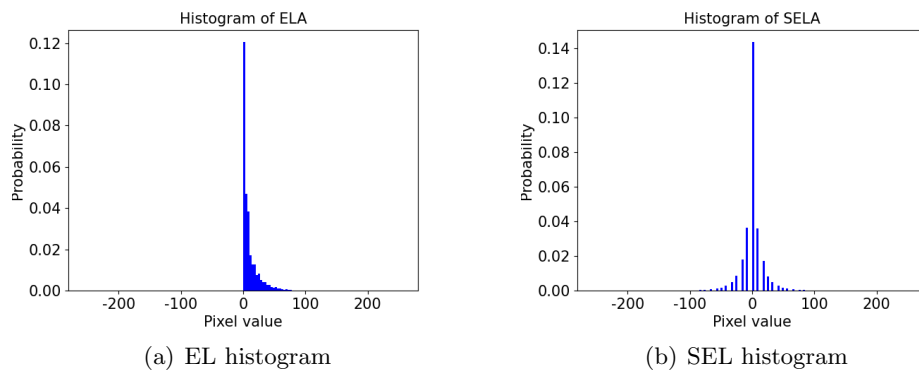


Figure 3.5: The histograms of normal EL and proposed SEL, extracted from the same image.

error differences, SEL retains the error polarity, enabling better handling of cases where background regions show higher error levels than manipulated areas due to compression artifacts. The signed error differences are then normalized to the range of 0 to 255, making them suitable for pixel-level representation, as expressed by the following equation:

$$S(\hat{\mathbf{x}}) = \left(\frac{S(\mathbf{x}) \times 255}{\max(|S(\mathbf{x})|)} + 255 \right) / 2 \quad (3.4)$$

This normalization process ensures that the polarity of error levels is preserved and facilitates the distinction between tampered and untouched areas by enhancing error contrast. Notably, this method resolves scenarios where the background exhibits higher error levels, preventing the misclassification of the background as manipulated.

The SEL method effectively addresses the limitations of traditional ELA by retaining the polarity of error levels, allowing for the accurate detection of manipulated regions even when the background experiences greater compression artifacts. SEL maps $S(\hat{\mathbf{x}})$ preserves both positive and negative error values and normalizes them, ensuring better differentiation between manipulated and unaltered regions. As shown in Figure 3.4, SEL offers robust performance, even under challenging conditions where the background compression artifacts are more prominent than in the manipulated regions. Furthermore, the pixel value distribution of SEL, as illustrated in Figure 3.5, reveals an asymmetry that highlights the importance of preserving the polarity of error levels. Unlike traditional EL, which leads to a symmetric distribution of error values, the SEL representation maintains both positive and negative polarities, indicating that these values carry significant analytical information. This asymmetry is particularly valuable in identifying inconsistencies and distinguishing between manipulated and pristine regions. Additionally, SEL exhibits a broader range of pixel values compared to conventional error level methods, enhancing the separability between tampered regions with higher error levels and unaltered areas with lower levels. This extended error value range facilitates more precise differentiation, thereby improving the overall effectiveness of forgery detection.

3.3.2 Learned Feature Extractor

The pre-processed SEL result, denoted as $S(\hat{\mathbf{x}})$, is then fed into a learned feature extractor to capture multi-dimensional information from the SEL maps. Traditional Convolutional Neural Networks typically organize their convolutional layers into five blocks, each designed to process different kernel sizes and receptive fields (such as ResNet50 [138]). In

the proposed model, ResNet50, with its five convolutional blocks, is used as the feature extractor. The output from each block, denoted as $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5$, corresponds to feature maps of various shapes: $256 \times 256 \times 64$, $128 \times 128 \times 256$, $64 \times 64 \times 512$, $32 \times 32 \times 1024$, and $16 \times 16 \times 2048$. These outputs represent different levels of detailed and global information extracted from the image. Specifically, the feature map $\mathbf{y}_i$ from convolutional block i of the CNN feature extractor is computed as shown in Equation (3.5), where $C(\cdot)$ represents the feature extractor function:

$$\mathbf{y}_i = C\left(S(\hat{\mathbf{x}}), i\right), \quad (3.5)$$

This approach enables the extraction of a rich set of features from varying spatial resolutions, allowing the model to leverage both detailed and global information in the manipulated images.

3.3.3 Peak-shaped feature reshaping

Next, the height and weight of convolutional block features $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5$ need to be unified in order to be acceptable by concatenation layer of any self-attention algorithm. The common reshaping process in most learning methods like UNet and FCN is achieved by a U-shaped workflow, which decreases the feature shape by convolutional blocks and increases it by block-by-block interpolation or transposed convolution T . The size-increasing process of U-shaped structure is operated as:

$$\mathbf{y}'_i = \begin{cases} T(\mathbf{y}_{i+1}) + \mathbf{y}_i, & \text{if } i = 4, \\ T(\mathbf{y}'_{i+1}) + \mathbf{y}_i, & \text{if } i = 1, 2, 3. \end{cases} \quad (3.6)$$

The advantages of this repeated reshaping process, as used in traditional U-shaped structures, are that: 1) it generates the feature map $\mathbf{y}'_1$ with the same size as the original image, which suits the next-step pixel-level classification, and 2) it avoids the pooling operation from destroying the image tampering traces. However, for different image manipulation types or datasets, the importance of each information level varies, resulting in the requirement for different perceptual domains. This highlights a limitation of the U-shaped structure, which applies a uniform reshaping operation across all layers and considers only a single situation. To address it, this study proposes an adaptive structure that different reshaping algorithms, instead of repeated transposed-convolution and addition, are performed on each convolution block individually (like a 'peak') to construct the feature at the target size. The comparison between two structures are shown in Figure 3.6.

To achieve this, one of the convolution blocks is selected as 'peak' block $\mathbf{y}_p$ while the rest of features are reshaped to the its size by dense layers or transposed convolution T , where r represent the ratio of reshaped feature size to the target feature size, which defines the neuron number of dense layer and kernel size of transposed convolution. The peak-shaped reshaping process is described as:

$$\mathbf{y}'_i = \begin{cases} \text{Dense}(\mathbf{y}_i, r), & \text{if } i < p, \\ T(\mathbf{y}_i, r), & \text{if } i > p, \\ \mathbf{y}_i, & \text{if } i = p. \end{cases} \quad (3.7)$$

The selection of peak block number $peak$ is adjustable due to different manipulation types or image characteristics and decides whether to increase feature size without loss of detail information or decrease feature size with more delicate receptive field. Moreover, in order

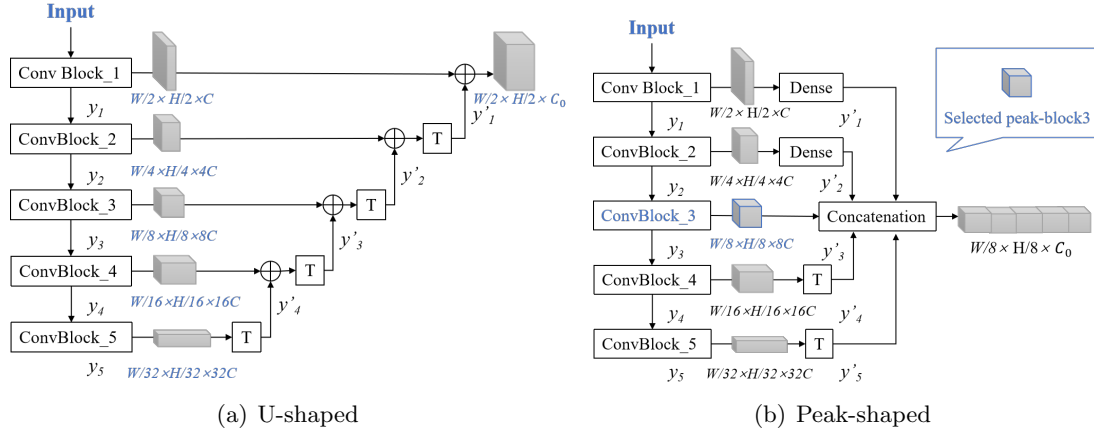


Figure 3.6: The comparison between normal U-shaped (which refers to Peak1 peaked-shaped reshaping) and proposed peak-shaped feature reshaping

to avoid the large gaps in the number of channels of different block output features, the channels are respectively adjusted to the same order of magnitude through the dense layers. Then the combined feature $\mathbf{Y}$ is obtained by concatenating the independently reshaped features:

$$\mathbf{Y} = \text{Concat}(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{y}'_3, \mathbf{y}'_4, \mathbf{y}'_5). \quad (3.8)$$

The proposed peak-shaped feature reshaping structure provides a more flexible alternative to the traditional U-shaped structure by addressing its limitation of relying on a single perceptual domain. One limitation of the proposed method is that the choice of peak requires manual selection through ablation studies. A detailed discussion on how different peak configurations affect model performance (including the case where the U-shaped structure corresponds to *Peak1*) is presented in Section 3.4.5.

3.3.4 Cross Convolutional-blocks Weighting Module

The concatenated feature $\mathbf{Y}$ is processed by a Cross Convolutional-Blocks Weighting Module (CBWM), which enables the model to adjust the weights of convolutional blocks according to various manipulation types and digital characteristics of input images. As shown in Figure 3.3, CBWM consists of two core components: (1) a self-attention mechanism along channels to generate channel-wise weights for each convolutional block, and (2) two fully-connected layers to capture inter-relationships between the blocks. The detailed workflow of CBWM is as follows:

- The feature map $\mathbf{Y}$ of size $W_i \times H_i \times C_0$, which is concatenated from reshaped block features of size $W_i \times H_i$ (i -th convolutional block), is globally pooled along the height and width to produce a vector $\mathbf{A}$ of length equal to the number of channels in $\mathbf{Y}$;
- The vector $\mathbf{A}$, of size $1 \times 1 \times C_0$, is passed through a ReLU activation function to generate trainable weights for all convolutional block channels;
- Two consecutive dense layers with different numbers of neurons are applied to compute the channel-wise inter-relationship, thus enabling the combination of multi-level weighted information across all convolutional blocks. The use of two dense layers allows for the creation of non-linear connections among channels, as opposed to a single linear dense layer;

- Finally, the element-wise multiplication of the inter-related block weights $\mathbf{A}$ and the input feature $\mathbf{Y}$ produces the output feature map $\mathbf{Y}'$, where the vector weights are assigned to the corresponding block output features across channels. The parameters in CBWM are updated during training to adapt the model's use of different convolutional block outputs based on the properties and manipulation types within the training dataset.

In summary, CBWM effectively allows the model to dynamically adjust the contribution of each convolutional block, enhancing the model's adaptability to various image manipulation types.

3.3.5 Binary Classification Module

After the Cross Convolutional-Blocks Weighting Module, the weighted feature $\mathbf{Y}'$ obtained from multi-view convolutional blocks requires to be mapped into the binary classes 'pristine' or 'manipulated.' First, a global average layer is applied to flatten the positional dimensions of the high-level feature $\mathbf{Y}$, which has a shape of $W_i \times H_i \times C_0$, down to $1 \times 1 \times C_0$. This is followed by a dense layer that fully connects all C_0 channels across the convolutional blocks into a single neuron. The output is processed by a Sigmoid activation layer, resulting in a value that ranges from 0 to 1. This value represents the likelihood that the predicted image belongs to the positive class (manipulated, in this case). Based on the most probable predicted class and the corresponding ground-truth image-level label, the binary cross-entropy loss is calculated batch by batch and backpropagated to update the model parameters' weights in the direction of loss minimization.

3.4 Experimental Results and Discussions

In this section, the experimental setup and the applied image manipulation datasets are introduced first, followed by the ablation study and performance evaluation of proposed WCBnet.

3.4.1 Experimental Setup

Hardware The model is built and implemented by TensorFlow2.5, on a PC with NVIDIA GeForce RTX 3090 Ti, 12th Gen Intel(R) Core(TM) i9-12900K 3.20 GHz processor and 32.0 GB RAM.

Datasets Six commonly-used image forgery datasets are applied for training and evaluating proposed WCBnet. The details of their manipulation types, image size and image number are summarized in Table 3.1.

Training details The SEL of each dataset is extracted first and manually split into training set, validation set and test set as the ratio of 7:2:1. The training and ablation experiments of proposed WCBnet are based on a large-scale dataset CASIA2.0, and the other image manipulation datasets are used for comprehensive performance evaluation. The model is trained on the split training set of copy-move or splicing subsets and pristine subsets for 200 epochs and evaluated on the unseen split testing images. For the experiments, ResNet50 with five convolutional blocks, is selected to be the learned feature extractor. The JPEG compression factor of EL and SEL is set to 90 as an empirical choice.

Table 3.1: *This thesis reports results regarding six commonly used image manipulation datasets.*

Manipulation	Dataset	Image size	Image number (forged/pristine)
Copy-move, splicing	CASIA1.0 [25]	400×400 to 300×400	921/800
	CASIA2.0 [25]	900×600 to 160×240	5063/7491
	Defactos [139]	640×426 to 240×320	124000/
Splicing	Columbia [23]	1152×768 to 757×568	183/180
Copy-move	Coverage [36]	752×472 to 253×340	100/100
	NC2016 [30]	4329×3240 to 640×480	1124/847

Evaluation metrics In the pristine-or-manipulated binary classification tasks, the performance of WCBnet is evaluated using several key metrics that offer complementary insights into its predictive capabilities. In the context of this binary classification model, this study defines the positive class as **manipulated images** and the negative class as **pristine images**. Consequently, the performance of proposed manipulation detection model is evaluated using commonly-adopted metrics such as accuracy, precision, recall, F1 score, and AUC based on how well the model distinguishes between these two categories [140]. Specifically, these metrics are calculated based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values which are defined as follows:

- TP: Correctly classifying manipulated images;
- TN: Correctly classifying pristine images;
- FP: Incorrectly classifying pristine images as manipulated;
- FN: Incorrectly classifying manipulated images as pristine.

Accuracy is one of the most straightforward metrics, measuring the proportion of correctly classified instances (both manipulated and pristine images) out of the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (3.9)$$

While accuracy provides an overall success rate, it can be misleading in the presence of class imbalance, as it does not differentiate between the types of errors made. To address this limitation, the F1 score, a harmonic mean of precision and recall, is often employed. Precision measures the proportion of images classified as manipulated that are actually manipulated:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3.10)$$

While Recall reflects the model’s ability to correctly identify manipulated images:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (3.11)$$

The F1 score balances these two aspects, offering a robust measure when both false positives and false negatives are critical to consider:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3.12)$$

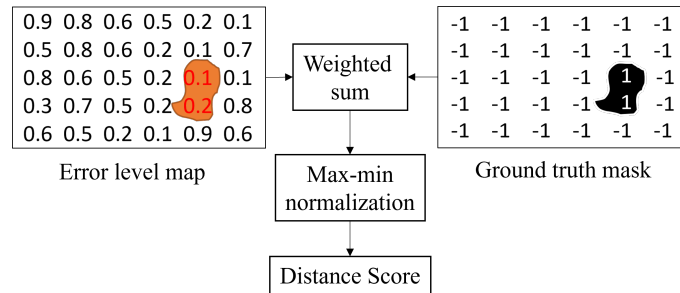


Figure 3.7: Self-defined pixel-level Distance Score for evaluating the similarity between ground truth masks and error level maps.

Another important metric in binary classification is the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. AUC, in my case, provides a summary of the model’s ability to distinguish between manipulated and pristine images across different decision thresholds. The ROC curve draws the relationship between TP rate and FP rate regarding all possible thresholds, and the AUC measures the region under the ROC curve representing the possibility of making positive decisions. As ROC curve approaches the boundaries of TP=1 or FP=0, AUC becomes higher and closer to the maximum value 1 which indicates perfect model classification performance. AUC serves as a better metric than accuracy in imbalanced datasets, in this thesis, represents the model’s ability to distinguish between manipulated (positive class) and pristine (negative class) images across different threshold levels. AUC, of ROC curve is defined as:

$$AUC = \int_0^1 TPR(FPR) dFPR. \quad (3.13)$$

Where the True Positive Rate (TPR) and False Positive Rate (FPR) are given by:

$$TPR = \frac{TP}{TP + FN}, \quad (3.14)$$

$$FPR = \frac{FP}{FP + TN}. \quad (3.15)$$

By using these metrics, including accuracy for overall performance, F1 score for precision-recall trade-offs, and AUC for threshold-independent evaluation, a comprehensive assessment of our binary classification model’s effectiveness can be achieved.

In addition to standard evaluation metrics, a customized metric called the **Distance Score** is defined to measure the pixel-level discrepancy between the error levels and the ground truth (GT) mask. This score provides a preliminary quantitative assessment of how well the error levels correspond to the actual manipulated regions. The calculation of the Distance Score is based on a weighted sum of the pixel values in the error levels, with the ground truth serving as the weight matrix. Specifically, the ground truth mask assigns a weight of +1 to pixels in the positive (manipulated) region and -1 to pixels in the negative (pristine) region. This enables the Distance Score to capture the alignment between the detected artifacts (from error levels) and the true manipulation regions, while penalizing false positives (FP). Mathematically, the Distance Score is computed in three steps:

1. **Weighted Sum:** Each pixel in the error level map $E(x, y)$ is multiplied by the

corresponding pixel weight from the ground truth $G(x, y)$, where:

$$G(x, y) = \begin{cases} +1 & \text{if pixel } (x, y) \text{ belongs to the manipulated region (Positive),} \\ -1 & \text{if pixel } (x, y) \text{ belongs to the pristine region (Negative).} \end{cases} \quad (3.16)$$

The weighted sum across all pixels is then calculated as:

$$S = \sum_{x,y} E(x, y) \cdot G(x, y). \quad (3.17)$$

2. **Normalization:** To normalize S , the theoretical maximum and minimum values that correspond to perfect and completely incorrect predictions are computed:

- **Maximum value (perfect prediction):** This is obtained by simulating the case where the error level map perfectly matches the ground truth. It is computed as:

$$S_{\max} = \sum_{x,y} G(x, y) \cdot G(x, y). \quad (3.18)$$

which is equivalent to $G \cdot G$, representing the dot product of the ground truth with itself.

- **Minimum value (completely incorrect prediction):** This is obtained by simulating the case where the predictions are entirely wrong, meaning the predicted manipulated regions are pristine and vice versa. It is computed as:

$$S_{\min} = \sum_{x,y} G(x, y) \cdot (-G(x, y)). \quad (3.19)$$

which is equivalent to $G \cdot (-G)$, representing the dot product of the ground truth with the negation of itself.

3. **Final Normalized Distance Score:** Using these maximum and minimum values, this study applies a max-min normalization to the weighted sum S :

$$\text{Distance Score} = \frac{S - S_{\min}}{S_{\max} - S_{\min}}. \quad (3.20)$$

This ensures that the Distance Score is normalized between 0 and 1, where a value of 1 indicates a perfect match between the error levels and ground truth, and a value of 0 indicates a completely incorrect prediction.

This scoring mechanism effectively reduces the contribution of false positives by assigning a negative weight to the background regions, thus providing a more accurate measure of how well the error level map identifies manipulated areas. By penalizing errors in pristine regions, the Distance Score offers a more nuanced evaluation of the detection model's performance, particularly in terms of its ability to distinguish between manipulated and unmanipulated areas.

Table 3.2: Performance of the proposed method and the state-of-art models (The best and the second-best methods for each dataset are shown in *red* and *blue*, respectively).

Model	Testing set									
	CASIA1.0		Defactos		Columbia		NC2016		Coverage	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
ManTra-Net[141]	0.297	0.141	0.000	0.543	0.621	0.681	-	-	0.484	0.491
CR-CNN[142]	0.361	0.766	0.397	0.567	0.392	0.783	0.939	-	0.131	0.566
FCN-16[143]	0.775	0.796	0.458	0.551	0.481	0.762	-	-	0.180	0.541
MVSS-Net[13]	0.752	0.839	0.404	0.573	0.802	0.980	-	-	0.244	0.731
umUNET[144]	0.565	-	-	-	0.761	-	-	-	0.423	-
U2-Net[62]	0.557	-	-	-	0.756	-	0.911	-	0.400	-
WCBnet	1.000	1.000	0.801	0.818	0.863	0.950	0.751	0.560	0.646	0.503

3.4.2 Performance Comparison with State-of-the-art Methods

The performance of the proposed WCBnet is compared to several state-of-the-art methods, including MantraNet [141], CR-CNN [142], FCN-16 [143], MvssNet [13], Modified-CNN [144], and U2-Net [62], using two evaluation metrics: AUC and F1-score. The result is shown in Table 3.2. The proposed WCBnet achieves the best performance on CASIA1.0, with both the highest AUC and F1-score of 1.000, and on Defactos, with the highest AUC of 0.818 and F1-score of 0.801. It also achieves the highest F1-score of 0.646 on the Coverage dataset, though its AUC ranks fourth. For the Columbia dataset, WCBnet obtains the highest F1-score of 0.863 and the second-best AUC of 0.950 among all state-of-the-art models. On the NC2016 dataset, it ranks second-best with an F1-score of 0.751. These results indicate that, when facing multiple datasets with different characteristics and manipulation types, the proposed model can effectively adapt its structure to distinguish forged images from pristine ones.

These results indicate that WCBnet can effectively adapt to datasets with diverse characteristics and manipulation strategies. For example, CASIA1.0 and Defactos contain relatively large and clearly manipulated regions, which allow the model to fully leverage image-level classification and low-level trace preservation through its adaptive reshaping strategy. Columbia and NC2016 include more subtle splicing manipulations, where the proposed model still performs competitively due to its ability to balance semantic and fine-grained features. Although Coverage has more complex and varied manipulation boundaries, WCBnet still outperforms others in F1-score, showing robustness in classifying more challenging manipulations. Overall, the results suggest that the proposed model can dynamically adjust to different feature domains, leading to consistent and strong performance across various manipulation scenarios.

3.4.3 Comparing image-level performance with other error-modelling methods

In addition to JPEG compression, various other methods have also been utilized to model the error levels of manipulation traces. The proposed JPEG compression-based SEL is compared with conventional Wavelet-based methods and learning-based Comprint [106].

- **Wavelet-based error levels:** The wavelet transform efficiently converts an image from the spatial domain to the wavelet domain, separating it into distinct frequency components for enhanced data representation [145]. It could be applied for image manipulation detection, namely, extracting energy features of images manip-

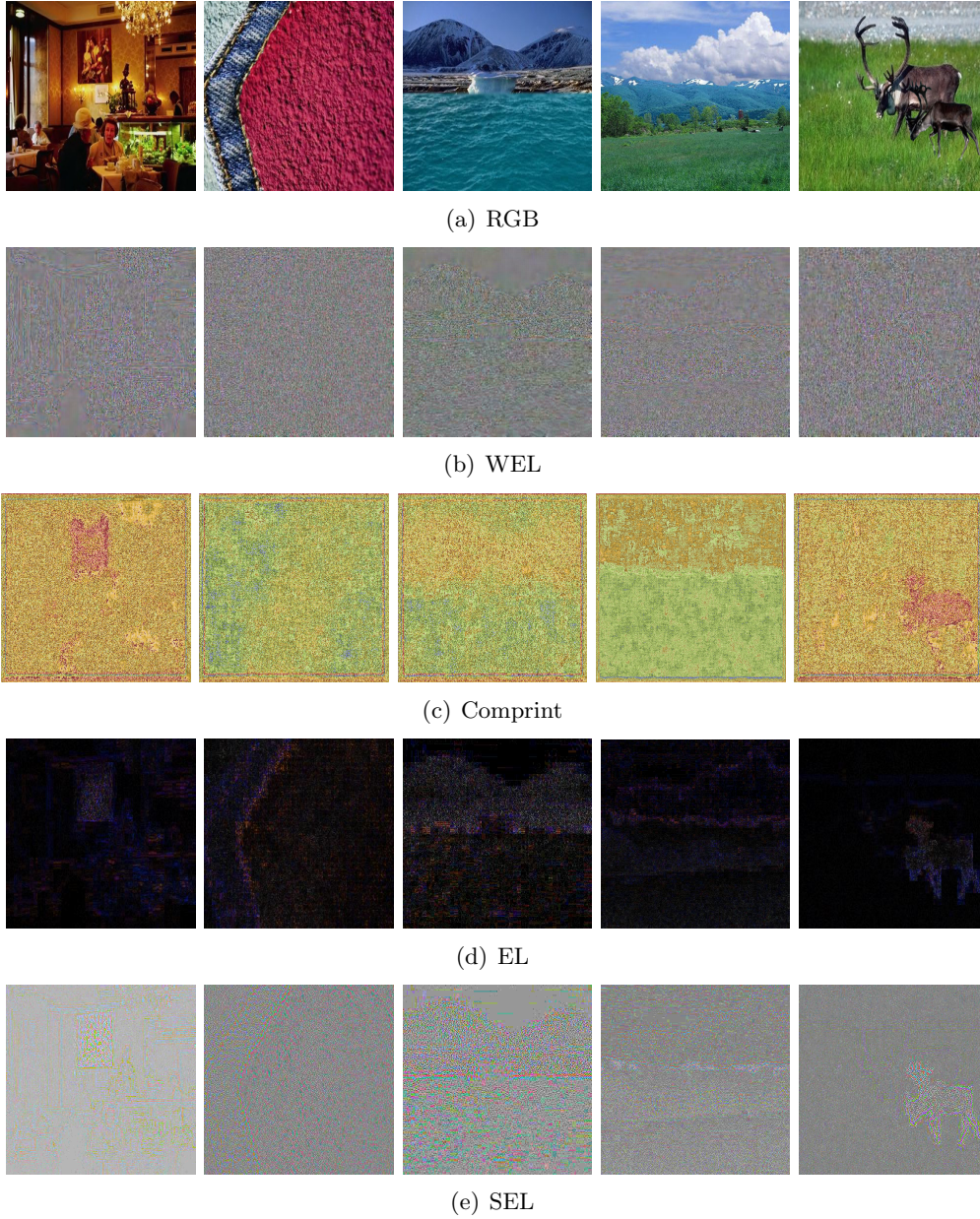


Figure 3.8: Comparisons between different error levels or noises, extracted from manipulated images.

Table 3.3: *The image-level performance of proposed method and backbone on different error levels or noise levels of manipulated images.*

Error type	ResNet50			WCBnet		
	Accuracy	F1	AUC	Accuracy	F1	AUC
WEL	89.8%	0.862	0.958	91.8%	0.884	0.980
Comprint	80.5%	0.701	0.898	89.3%	0.834	0.959
ELA	91.7%	0.887	0.968	92.8%	0.925	0.988
SEL	91.8%	0.915	0.987	95.1%	0.933	0.994

ulated by contrast enhancement [146]. However in this chapter, this study utilizes wavelet-based image compression to extract wavelet-based error levels (WEL) as it plays a crucial role in image compression techniques like JPEG2000. The wavelet-based image compression leverages the process of wavelet transform, decomposition, quantization, and encoding to discard least significant high-frequency image components. The discrepancy in the compression processes between tampered regions and the original background likely leads to inconsistencies. Therefore, by subtracting wavelet-compressed results from the original image, wavelet-based error levels are obtained to highlight the differences between tampered and authentic regions. Following the general framework in [147], wavelet compression with a quantization factor $Q = 2^6$ was employed to control compression strength, where the specific value of $N = 6$ was empirically determined based on preliminary experiments. Subsequently, differences between the original images and their wavelet-compressed counterparts were computed using the same approach as SEL, aiming to extract wavelet-based error levels (WEL) for comparison with SEL results.

- **Learning-based Comprint:** Comprint-method applies a Siamese Network to learn the difference between image patches with different compression processes, in order to detect the compression-related manipulations. The pre-trained Siamese Network is applied to generate Comprints of CASIA2.0 datasets which is compared with other non-learning error levels.

The CASIA 2.0 dataset was processed using different noise extraction methods, and several error-level examples of manipulated images regarding WEL, Comprint, EL and SEL are visually compared and statistically evaluated with both the backbone and proposed model. Figure 3.8 presents a visual comparison of various error level maps. In the case of WEL, there is no visually significant distinction in error levels between the tampered regions and the background, as the error levels appear to be uniformly distributed across the entire image. Comprint demonstrates the ability to localize tampered areas when they are small and semantically distinct, such as the painting on the wall in 1st sample or the moose in 5th sample, but fails to effectively identify large-scale tampering in other examples. In contrast, both EL and SEL maps exhibit a clear distinction in error levels between manipulated and non-manipulated regions, regardless of whether the tampered areas are semantically meaningful objects or extensive regions like mountains and lakes. The impact of using these different error level maps as inputs to the manipulation detection network is further evaluated in Table 3.3.

The image-level manipulation detection performance regarding these error levels is shown in the Table 3.3. It reveals that SEL is the most effective error-level extraction method for detecting image manipulations, achieving the highest metrics across all scenarios, including an outstanding AUC of 0.994 and an Accuracy of 95.1% when combined with WCBnet. This indicates SEL’s exceptional ability to highlight manipulated traces. Ad-

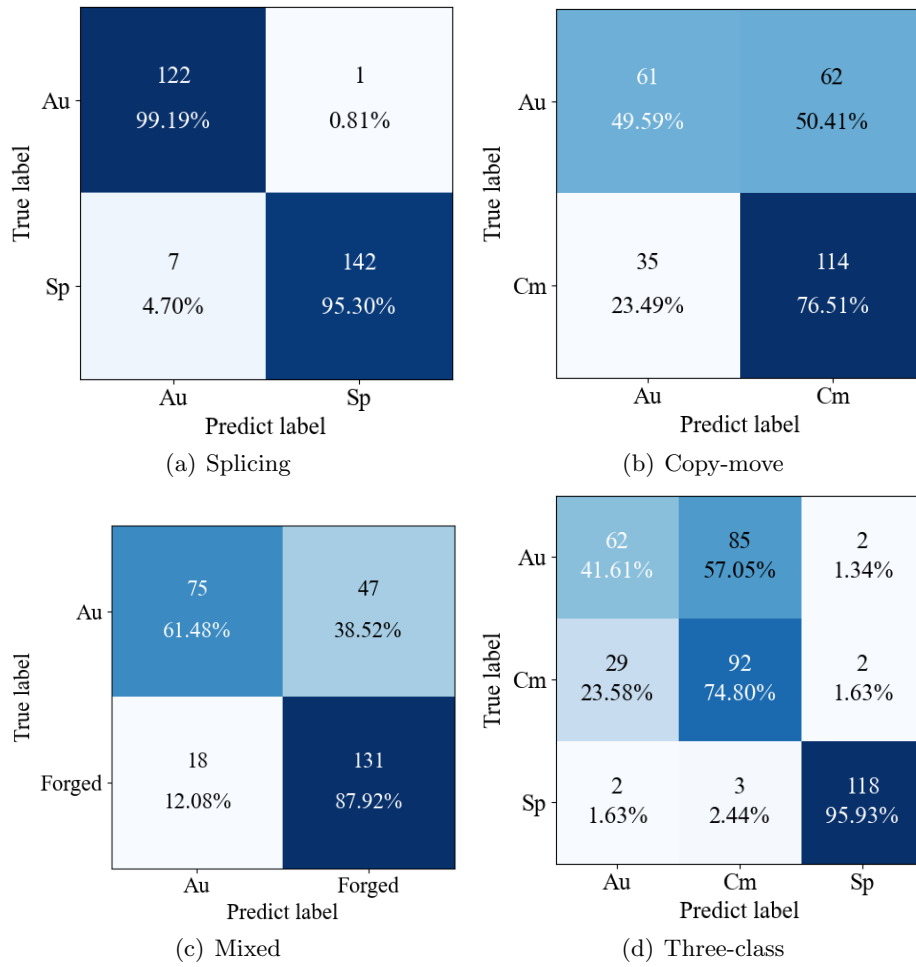


Figure 3.9: Three-class manipulation detection results of WCBnet on subsets of Defacto for identifying different manipulation types, represented as confusion matrix.

ditionally, WCBnet consistently demonstrates robust performance across all error types, significantly surpassing the baseline ResNet50 in Accuracy, F1, and AUC. These findings underline WCBnet’s superior capability in extracting and modelling error-level features.

3.4.4 Evaluation of WCBnet for Identifying Manipulation Types

Having shown the adaptability of proposed WCBnet on classifying pristine or forged images of different datasets, the WCBnet was modified for three-class manipulation detection to distinguish between splicing, copy-move and pristine images. The performance of binary-class and three-class WCBnet with optimal peak block and customized channels was evaluated on the 250 manipulated and pristine images that are randomly selected from Defactos dataset. Three-class WCBnet was trained on SEL maps of splicing, copy-move and pristine images of those two datasets, then the results were compared with binary-class WCBnet which was separately trained on splicing, copy-move and mixed subsets. The confusion matrix of three-class WCBnet is compared to binary-class confusion matrix in Figure 3.9. Three-class WCBnet still achieved a good performance that only 7 out of 364 forged images were mis-classified as authentic images, and more than 92% authentic images were correctly identified. Except for real-authentic rate, the WCBnet achieved

slightly better result when considering splicing and copy-move separately in three-class task or single forgery detection, compared to the scenario that copy-move and splicing are all considered as forged. For the performance of WCBnet on Defactos, it is obvious that authentic images are more likely to be classified as copy-move according to Figure 3.9(b) and Figure 3.9(d). However, when splicing images could be separately considered in the three-class task or splicing-only binary-class task, the classification accuracy was increased to 95.93% and 95.30% respectively. From those two initial experiments, copy-move and splicing manipulations need to be leaned individually by the model, in order to extract features that are more suitable for a certain manipulation detection.

Overall, the result demonstrates WCBnet’s capability to detect manipulation across various scenarios, including single-type, mixed-type, and multi-type manipulation. However, several limitations in its performance remain: 1) In the copy-move only detection task, WCBnet misclassified around 49% of authentic images as copy-move. A potential reason for this is that WCBnet, when trained solely on copy-move subset, is optimized to identify highly similar regions within the same image. In complex authentic scenes, such as those containing repeated patterns like clusters of similar trees, naturally occurring self-similarity can mislead the model; 2) This limitation still exists in the mixed manipulation result but the true negative (authentic) rate improves notably by 12%. This improvement can be attributed to WCBnet learning proper error-level features that are common to both splicing and copy-move manipulations but distinct from authentic images, thereby enhancing its ability to differentiate between actual manipulations and naturally occurring similarities in authentic content.

3.4.5 Ablation Studies

Extensive ablation studies are conducted to optimize each component of the proposed method. The initial phase involves determining the optimal JPEG compression quality factor, which is closely associated with the error levels extracted from manipulated images. This process starts with computing the similarity between the ground truth masks of the CASIA2.0 image manipulation dataset and their corresponding error-level results across various compression factors. The similarity evaluation metrics employed include cosine similarity, pixel-level area under the curve (AUC) scores and a self-defined similarity score. Subsequently, different error levels (ELs) and signed error levels (SELs) corresponding to various compression factors are fed into multiple image-level backbone networks to assess their performance in image manipulation detection tasks and to establish the optimal compression factor.

Additionally, another aspect of the ablation experiments focused on validating the effectiveness of each component of the proposed WCBnet architecture. This involved evaluating the impact of different convolutional blocks on image manipulation tasks, followed by an assessment of the pre-processed model input signed-value error levels, peak-shaped feature reshaping, and the structural variations within the cross convolutional blocks module. Since all these ablation experiments are image-level, the common image-level evaluation metrics such as Accuracy, F1-score and AUC are applied to evaluate the model performance on binary classification tasks.

Ablation experiments of error-level compression factors In the process of calculating the error levels, a key parameter called the compression factor plays a critical role in determining the quality of the re-compressed image. This factor directly influences the overall quantity of extracted errors, as it controls the level of compression applied. A

Table 3.4: The similarity score between ground truth masks and error levels of **manipulated** images, of which the compression quality factor ranging from 10 to 90; where Cos is the Cosine Distance between two masks while Dis is the self-defined Distance Score; The optimal performance is marked as **red** and the sub-optimal performance is **blue**.

	RGB	10		20		30		40		50		60		70		80		90	
		EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL
IOU	0.017	0.008	0.003	0.004	0.007	0.018	0.016	0.003	0.006	0.019	0.002	0.003	0.009	0.004	0.003	0.001	0.002	0.009	0.000
Cos	0.530	0.379	0.181	0.321	0.250	0.540	0.122	0.098	0.197	0.502	0.234	0.188	0.377	0.272	0.352	0.212	0.171	0.436	0.075
Dis	0.530	0.733	0.884	0.841	0.885	0.466	0.914	0.906	0.831	0.518	0.931	0.871	0.654	0.861	0.604	0.810	0.894	0.812	0.941

Table 3.5: The Distance Score between GT and error levels of **pristine** images; The optimal performance is marked as **red** and the sub-optimal performance is **blue**.

	RGB	10		20		30		40		50		60		70		80		90	
		EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL	EL	SEL
Dis	0.929	0.897	0.932	0.904	0.924	0.927	0.903	0.957	0.917	0.902	0.946	0.938	0.928	0.930	0.957	0.942	0.901	0.966	0.921

lower compression factor leads to higher image quality and fewer detectable errors, while a higher compression factor results in greater degradation and an increased number of errors, which can be crucial for identifying manipulation traces. Most EL-based image manipulation detection methods typically set the compression factor to a default value of 90. However, for datasets of varying sizes, resolutions, and tampering operations, the optimal compression factor for error-level analysis can differ significantly. To determine the most suitable compression factor, a direct evaluation method is to compare the similarity between EL and SEL with the corresponding ground truth masks. There are three metrics to check the similarity between error levels and ground truth masks (GT), including IoU, Cosine similarity and self-defined Distance Score. All the images were loaded as gray-scale images with one channel and converted to numpy array with int32 format. The first two methods are widely applied to compute the similarity of two arrays, but the applied GT with only 1s and 0s cannot well reflect the wrongly-classified background decisions. So a self-defined evaluation metric (as introduced in the section 3.4.1) is applied, which considers the negative effect of wrongly-classified background pixels. The result of similarity calculation between GT and inputs (RGB, EL and SEL), regarding three evaluation metrics is shown in Table 3.4 and Table 3.5, which demonstrate the superiority of the EL and SEL compared to the standard RGB inputs. In both manipulated and pristine image scenarios, EL and SEL consistently outperform RGB across various compression factors. This is particularly evident at a compression factor of 90, where EL achieves a Distance Score of 0.941 for manipulated images and 0.966 for pristine images, indicating a remarkable capability to identify and differentiate image characteristics accurately. Similarly, SEL also benefits from this configuration, achieving a commendable score that emphasizes its effectiveness. However, the results of this preliminary experiment exhibit significant variability and do not demonstrate an absolute advantage for any particular compression factor, nor do they lead to a conclusion about the optimal compression factor suitable for WCBnet.

Another experiment on compression factors is feeding error levels computed with all possible JPEG compression factors (ranging from 0 to 100) into the proposed WCBnet architecture. The extent to which EL-based manipulation traces are exposed varies depending on the compression factor applied. Ablation experiments were conducted on the CASIA2.0 dataset to identify the optimal compression factor for this specific dataset, ensuring the highest detection accuracy by maximizing the visibility of tampering artifacts.

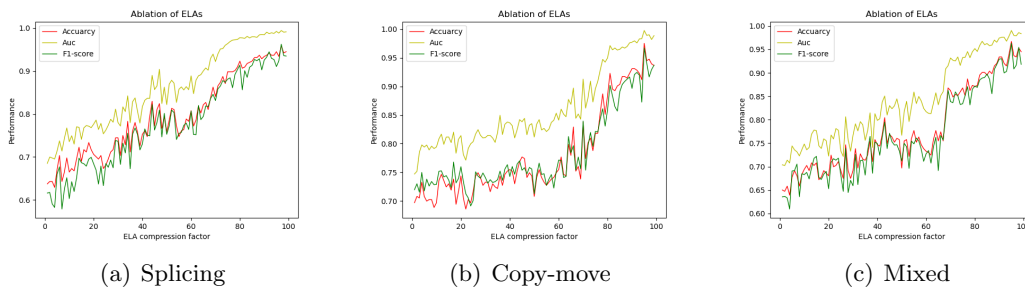


Figure 3.10: The performance of WCBnet on three subsets of CASIA2.0, regarding 100 compression factors

Table 3.6: Performance of WCBnet on three subsets of CASIA2.0, regarding three different input types

Subset Type	RGB			EL			SEL		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1-score	AUC
Splicing	64.84%	0.702	0.811	92.58%	0.891	0.982	95.33%	0.928	0.991
Copymove	71.15%	0.616	0.794	92.31%	0.867	0.983	95.33%	0.913	0.989
Mixed	82.69%	0.695	0.913	94.51%	0.912	0.987	95.88%	0.935	0.992

From the Figure 3.10, compression factor 95 is optimal for copy-move and mixed subsets, which could achieve accuracy of 97.5% and 96.7%, f1-score of 0.967 and 0.963, AUC of 0.997 and 0.990 respectively. For splicing subset, the optimal compression factor is 97 which achieves accuracy 95.9% accuracy, 0.962 f1-score and 0.994 AUC, while compression factor 95 is the 2nd best choice.

Effectiveness of signed-value error levels The effectiveness of SEL is evaluated by comparing the performance of WCBnet on RGB, EL and SEL inputs of three forged subsets of CASIA2.0. As shown in Table 3.6, the most obvious difference is that all error-level-based inputs are superior to RGB input. More precisely, the classification accuracy of model with EL input on splicing, copy-move, and mixed subsets is increased by around 28%, 21%, and 12% respectively, and the F1-score is also increased by 0.21, 0.25 and 0.22, compared to the performance of same models with RGB input. The model with SEL input achieves even better results, compared to the number regarding EL input, which increased the accuracy of the three subsets by 3% on average, and reached an average accuracy of about 95.5%, average f1-score of 0.925 and average AUC score of 0.990, on all three subsets.

Effectiveness of peak-shaped feature reshaping Since the shape and channel-number of target feature in peak-shaped reshaping module are flexible, several ablation experiments are conducted on three subsets of CASIA2.0, regarding SEL inputs only. Each convolutional block is selected as peak-block in turns (from *Peak1* to *Peak5*), in which *Peak1* is equivalent to classic U-shaped reshaping structure. The channel number of these features is also be unified into one of 15, 21, 30, and 45 in turn. As shown in Table 3.7 and 3.8, the performance of networks combining different peak blocks and channel numbers varies. Considering AUC, the network based on *Peak1*, *Peak2* or *Peak3* can achieve a value of about 0.990 under all conditions, and the average difference between the three is only 0.006. When considering F1-score, the network based on *Peak3* and customized channel numbers achieves the most stable and high performance on the three

Table 3.7: Ablation study of peak-shaped feature reshaping, where the data in the table represents F1-score; The top performance is marked as *red*.

	Channel 15			Channel 21			Channel 30		
	Sp	CM	Mixed	Sp	CM	Mixed	Sp	CM	Mixed
Peak1	0.941	0.922	0.893	0.931	0.890	0.929	0.931	0.924	0.939
Peak2	0.943	0.930	0.936	0.946	0.929	0.909	0.948	0.926	0.939
Peak3	0.968	0.934	0.771	0.952	0.945	0.941	0.953	0.918	0.905
Peak4	0.951	0.933	0.941	0.951	0.879	0.912	0.955	0.910	0.947
Peak5	0.959	0.938	0.936	0.972	0.916	0.952	0.955	0.957	0.936

	Channel 45			Customized Channel		
	Sp	CM	Mixed	Sp	CM	Mixed
Peak1	0.926	0.914	0.933	0.929	0.920	0.940
Peak2	0.938	0.926	0.928	0.940	0.920	0.910
Peak3	0.924	0.930	0.937	0.961	0.960	0.949
Peak4	0.940	0.926	0.951	0.953	0.939	0.908
Peak5	0.956	0.944	0.941	0.962	0.946	0.912

Table 3.8: Ablation study of peak-shaped feature reshaping, where the data represents AUC; The top performance is marked as *red*.

	Channel 15			Channel 21			Channel 30		
	Sp	CM	Mixed	Sp	CM	Mixed	Sp	CM	Mixed
Peak1	0.991	0.989	0.982	0.993	0.987	0.989	0.989	0.989	0.992
Peak2	0.995	0.993	0.994	0.995	0.991	0.991	0.993	0.988	0.994
Peak3	0.990	0.993	0.920	0.991	0.986	0.993	0.990	0.985	0.989
Peak4	0.983	0.985	0.980	0.988	0.971	0.978	0.984	0.982	0.981
Peak5	0.981	0.986	0.985	0.987	0.980	0.983	0.989	0.982	0.970

	Channel 45			Customized Channel		
	Sp	CM	Mixed	Sp	CM	Mixed
Peak1	0.990	0.987	0.991	0.991	0.989	0.990
Peak2	0.991	0.991	0.992	0.989	0.992	0.985
Peak3	0.989	0.990	0.993	0.983	0.991	0.989
Peak4	0.980	0.989	0.981	0.983	0.985	0.986
Peak5	0.984	0.987	0.977	0.989	0.994	0.981

subsets which are 0.961, 0.960, 0.949. The customized channels {21, 15, 15, 30, 21} are defined according to the optimal channel selection of different peak-blocks.

Effectiveness of CBWM In order to evaluate the effectiveness of the proposed CBWM, which incorporates self-generated weights and two different dense layers, three reference networks, referred to as **RefNet-1**, **RefNet-2**, and **RefNet-3**, were defined. As shown in Table 3.9, these reference networks utilize different partial structures from CBWM to fuse the concatenated features of convolution blocks, while keeping the remaining components of the model unchanged. Specifically, **RefNet-1** employs an element-wise multiplication of the concatenated features followed by an activation function (ReLU) for its feature fusion module. In **RefNet-2**, the self-generated weight is computed by applying global pooling to the input feature and multiplying it by an activation layer. **RefNet-3** extends **RefNet-2** by adding a dense layer to fully connect the channels and establish a linear channel-wise relationship. The proposed CBWM improves upon **RefNet-3** by replacing the single dense layer with two consecutive dense layers, each with a different number of neurons, thereby enabling a more complex and adaptive block-wise attention mechanism.

Table 3.9: Comparison of RefNet with Partial and Full CBWM on CASIA2.0 subsets; RefNet-*i* models are equipped with partial CBWM structures.

Model	CBWM Structure	Key Components	Accuracy	F1-Score	AUC
RefNet-1	Partial CBWM	Single dense layer, ReLU	94.51%	0.894	0.981
RefNet-2	Partial CBWM	Global pooling, activation layer	90.38%	0.845	0.969
RefNet-3	Partial CBWM	Dense layer + feature fusion	90.11%	0.848	0.976
WCBnet	Full CBWM	Dual dense layers, weighted features	95.05%	0.912	0.987

Table 3.10: Ablation study of CBWM on SEL results across multiple testing datasets; RefNet-*i* models are equipped with partial CBWM structures; The top performance is marked in *red*.

	CASIA1.0			Defactos			CMFDdb		
	Mixed			Mixed			Cpymove		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
RefNet-1	100%	1.000	1.000	73.80%	0.745	0.807	12.50%	0.125	0.156
RefNet-2	100%	1.000	1.000	74.91%	0.681	0.797	25.00%	0.333	0.281
RefNet-3	100%	0.976	1.000	74.54%	0.726	0.802	31.25%	0.478	0.344
WCBnet	100%	1.000	1.000	75.65%	0.750	0.819	43.75%	0.490	0.391

	Coverage			Columbia			Korus		
	Cpymove			Splicing			Splicing		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
RefNet-1	25.00%	0.286	0.160	89.19%	0.895	0.921	39.53%	0.435	0.366
RefNet-2	35.00%	0.435	0.270	83.78%	0.842	0.904	51.16%	0.488	0.498
RefNet-3	50.00%	0.482	0.300	83.78%	0.842	0.892	55.81%	0.596	0.500
WCBnet	35.00%	0.381	0.320	86.49%	0.857	0.892	60.47%	0.514	0.645

Since the performance of these models fluctuated greatly on different subsets of CASIA2.0, the effectiveness of CBWM is evaluated by counting the average classification accuracy and F1-score on all nine subsets of CASIA2.0, and compared to the average performance of other RefNets. The Table 3.9 illustrates that RefNet-1, RefNet-2, RefNet-3, and WCBnet respectively achieved an accuracy of 94.51%, 90.38%, 90.11%, and 95.05%, and F1-scores of 0.894, 0.845, 0.848, and 0.912 on CASIA2.0 SEL dataset. While the proposed WCBnet achieved the best performance in both accuracy and F1-score, it demonstrated a slightly higher F1-score than RefNet-1, showing that it is more effective at distinguishing between pristine and forged images. In contrast, RefNet-1, which achieves the best accuracy, performs better at recognizing pristine images.

The results of further experiments on six more testing datasets are shown in Table 3.10. On these testing datasets, WCBnet consistently outperforms other models, particularly in terms of F1-score. On CASIA1.0, it achieved a perfect F1-score of 1.000, outperforming RefNet-1, RefNet-2, and RefNet-3 in most subsets. In the Defactos dataset, WCBnet achieved the highest accuracy (75.65%) and F1-score (0.750), surpassing the other RefNets. On the more challenging CMFDdb dataset, WCBnet achieved a significantly higher performance with an F1-score of 0.490 and accuracy of 43.75%, while RefNet-1 struggled with very low performance. In the Coverage and Columbia datasets, WCBnet showed stronger performance in distinguishing between forged and pristine images compared to RefNet-1, particularly excelling in F1-score. These results indicate that the full CBWM structure in WCBnet enhances its ability to differentiate between subtle image forgeries, contributing to its superior performance across diverse datasets.

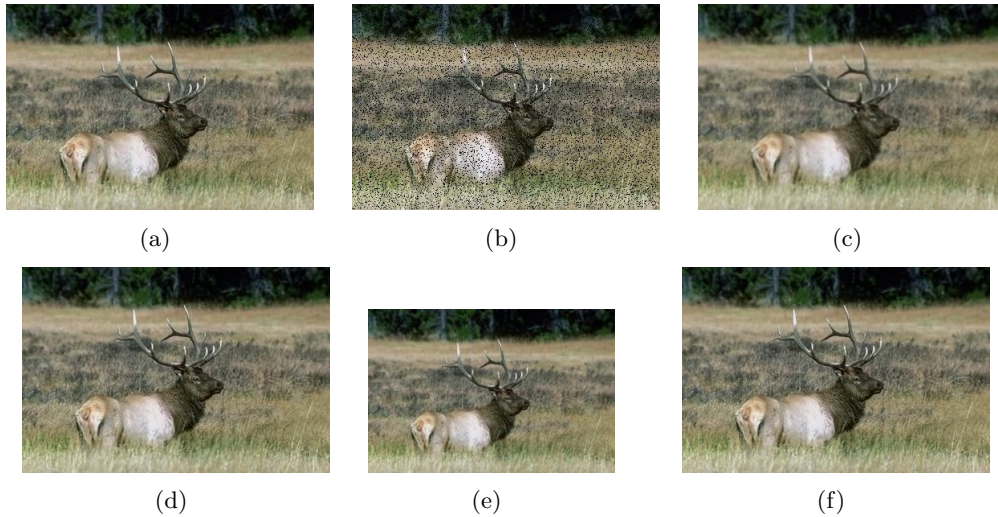


Figure 3.11: Different perturbation methods are applied on the CASIA2.0 image manipulation dataset, which are (a) Gaussian noise, (b) Salt&Pepper noise, (c) blurring, (d) contrast, (e) scaling and (f) JPEG compression. The detailed parameters are illustrated in the TABLE 3.11.

3.4.6 Robustness Test Against Perturbations

In real-world scenarios, images often undergo various post-processing operations (such as compression, resizing, or enhancement) either intentionally to refine manipulations or unintentionally during data transmission and storage. These perturbations may destroy subtle manipulation traces, thereby negatively affect the task of image manipulation detection [13, 148]. To assess the robustness of our proposed WCBnet against such common perturbations, this study applied a series of post-processing techniques to the manipulated images in the CASIA2.0 dataset. A sample image with different perturbation methods is shown in Figure 3.11. These perturbations include Gaussian noise (with standard deviation $\sigma = 3$), Gaussian blur (with kernel size 5×5), contrast adjustment (scaling factor 1.5), image scaling (resizing to $0.5\times$ and then back to original dimensions), and JPEG compression (quality factor $q = 0.5$). Salt& pepper noise is also introduced during evaluation, as it effectively simulates bit errors and sudden signal drop-outs, and represents a sparse and discrete form of impulsive noise[149] (2% of pixels are randomly set to 0 or 255). Each of these operations simulates typical distortions encountered during image acquisition, editing, or transmission [22, 11].

The manipulated images of CASIA2.0 are post-processed using these perturbations respectively, and the backbone ResNet50 and WCBnet is evaluated on seven different post-processed versions of CASIA2.0. As shown in Table 3.11, WCBnet consistently outperformed the baseline ResNet50 across all perturbation scenarios. For example, under Gaussian noise, WCBnet achieves a significant improvement in accuracy (79.9% and 50.0%), F1 score (0.766 and 0.654), and AUC (0.877 and 0.648) compared to the backbone ResNet. Similar trends are observed under scaling (81.3% and 76.1% accuracy) and JPEG compression (79.4% and 77.5% accuracy). Although the results show a slight decline compared to perturbation-free scenario, it is evident that WCBnet consistently improves robustness against perturbations compared to the backbone ResNet50. Overall, the results underscore WCBnet’s superior ability to handle image manipulations, particularly in the presence of various distortions, which makes it more reliable in real-world

Table 3.11: *The image-wise performance of the backbone and WCBnet on the CASIA2.0 dataset with different perturbations; Signal-to-noise ratio (SNR) of the examples in Figure 3.11 is also included.*

Perturbations	SNR (dB)	ResNet50			WCBnet		
		Acc	F1	AUC	Acc	F1	AUC
Gaussian noise ($\sigma = 3$)	32.14	50.0%	0.654	0.648	79.9%	0.766	0.877
Salt&pepper noise (0.02)	15.80	71.4%	0.702	0.797	68.7%	0.681	0.763
Blurring ($ksize = (3,3)$)	25.54	79.4%	0.757	0.862	80.2%	0.747	0.879
Contrast adjustment ($\alpha = 1.0$)	141.66	83.8%	0.805	0.886	81.3%	0.776	0.877
Scaling (0.8)	26.32	76.1%	0.728	0.845	81.3%	0.788	0.889
JPEG compression (0.9)	42.44	75.3%	0.748	0.844	78.6%	0.734	0.853
JPEG compression (0.5)	25.23	77.5%	0.733	0.830	79.4%	0.764	0.862

applications.

To better characterize the effect of each perturbation, the corresponding signal-to-noise ratio (SNR) was estimated, as shown in Table 3.11. While SNR alone does not fully explain detection performance, it provides meaningful context when considered alongside the nature of the distortion. Salt&pepper noise, with the lowest SNR (15.80dB), introduces localized pixel disruptions that significantly degrade detection accuracy. Gaussian noise (32.14dB), blurring (25.54dB), scaling (26.32dB), and JPEG compression (25.23dB at Q=50; 42.44dB at Q=90) cause moderate structural degradation and varying levels of impact. In contrast, contrast adjustment yields a high SNR (141.66dB) and minimal influence on detection. These results indicate that both the intensity and spatial distribution of perturbations are critical factors influencing model robustness.

3.4.7 Extended Experiment for Generative AI Detection

As illustrated in the Section 2.2.3, Generative AI (Gen-AI) tools could automatically learn the characteristics of real-world images and generate fake visual contents as realistic as possible. With the iterations of learning-based image manipulation technology, Gen-AI images have become not only difficult for the human eye to distinguish but can even confuse powerful learning-based detection models. To address the last research question from Section 2.5, the performance of WCBnet is evaluated over such challenging Gen-AI images.

Generative image dataset DeepfakeArt

Despite the considerable challenges in accurately detecting Gen-AI (also called deepfake images) images, this study has selected several state-of-the-art deepfake datasets to rigorously evaluate the performance of the proposed image manipulation detection network, WCBnet. Given that WCBnet’s detection principle relies on extracting multi-dimensional information to distinguish between manipulated and original regions, evaluating it on entirely AI-generated images would lack relevance. Therefore, the DeepfakeArt [150] dataset is selected, as it contains various types of image manipulations created with different Gen-AI tools, including inpainting, style transfer, and adversarial data poisoning. Among these, the inpainting subset, as shown in the Figure 3.12, leverages Stable Diffusion 2 [151] to generate manipulated content in specific regions of the image with characteristics closely matching the surrounding areas. For instance, in the first figure, the bottom region has been replaced with an image of a dog playing on grass, seamlessly blending with the surrounding background. The DeepfakeArt dataset exhibits significant

Table 3.12: *The image-level performance of WCBnet on Deepfake datasets regarding different reshaping peaks; The best performance is marked as red.*

	DeepfakeArt		
	Accuracy	F1-score	AUC
Best mentioned in the paper(DINOv2 [152])	82.75 %	0.800	-
WCBnet_peak1	50.1%	0.651	0.742
WCBnet_peak2	66.1%	0.710	0.851
WCBnet_peak3	88.2%	0.844	0.963
WCBnet_peak4	92.2%	0.882	0.971
WCBnet_peak5	93.9%	0.912	0.981

variations in image size, shape, and resolution. The image dimensions range from a minimum of 533x47 pixels to a maximum of 6215x3445 pixels, with file sizes spanning from 27 KB to 13,908 KB. The Inpainting subset, specifically, contains 5,063 manipulated images along with their corresponding pristine (original) images.

DeepfakeArt detection results

To evaluate the proposed WCBnet for Deepfake detection, this study adopt the same experimental setup as in previous experiments. The 5,063 Deepfake splicing images and corresponding 5,063 pristine images are divided into training, validation, and testing subsets with a ratio of 8:1:1. Additionally, the manipulation traces (error levels) previously applied become meaningless in this case, as there is no image compression involved during the direct generation of Deepfake images. The input to WCBnet is the original three-channel RGB images, with the focus on evaluating the model’s ability to weight features at different levels of information extracted from Deepfake images. Furthermore, WCBnet’s receptive field is constrained by the reshaping process. *WCBnet_peak i* with different reshaping peak i from 1 to 5 is evaluated to account for the varying image sizes in the Deepfake dataset. The image-level performance of *WCBnet_peak i* is presented in Table 3.12, where it is compared to the best-performing model mentioned in DeepfakeArt paper.

The experimental results in Table 3.12 show that WCBnet performs increasingly well in detecting Deepfake manipulations as reshaping peaks increase. WCBnet_peak1 starts with a modest accuracy of 50.1%, an F1-score of 0.651, and an AUC of 0.742. Each subsequent peak improves on these metrics: WCBnet_peak2 reaches 66.1% accuracy with an F1-score of 0.710 and AUC of 0.851, while WCBnet_peak3 outperforms the baseline model, DINOv2 [152], achieving 88.2% accuracy, an F1-score of 0.844, and AUC of 0.963. WCBnet_peak4 and WCBnet_peak5 show further improvements, with WCBnet_peak5 reaching the highest performance at 93.9% accuracy, an F1-score of 0.912, and an AUC of 0.981. These results highlight the model’s enhanced manipulation detection capacity as peak increases, suggesting that broader receptive fields allow WCBnet to better capture manipulation features in Deepfake images for more precise classification.

3.4.8 JPEG-AI and Its Impact on Error Level

JPEG-AI is an advanced image compression technique that utilizes AI tools to enhance the quality of images while maintaining a smaller file size. Unlike traditional JPEG compression, which relies on discrete cosine transforms (DCT) and quantization, JPEG-AI uses machine learning algorithms to learn optimal compression strategies. This enables

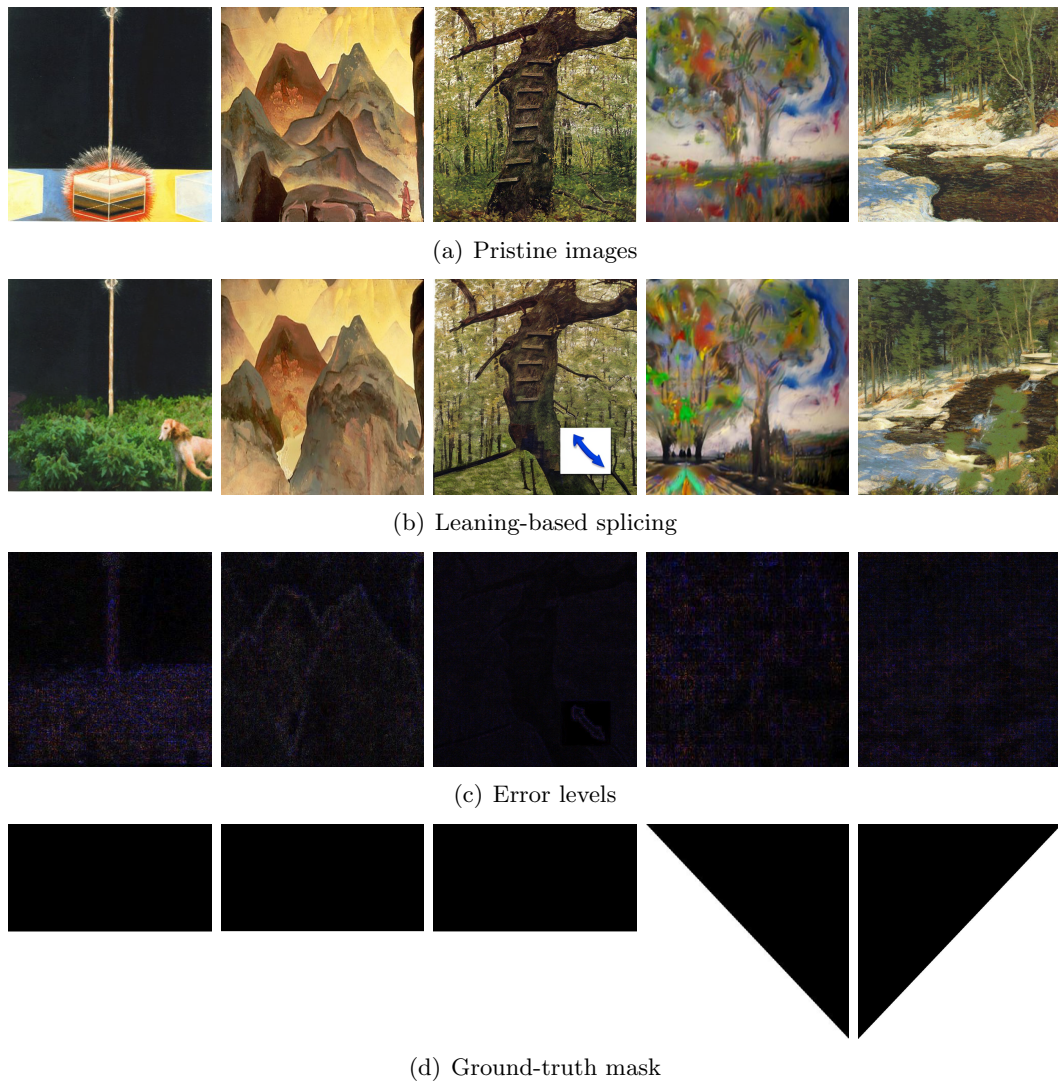


Figure 3.12: *Inpainting subset of DeepfakeArt challenge, which contains leaning-based splicing fake contents.*

JPEG-AI to produce images with fewer compression artifacts, such as blocking, color banding, and ringing. As a result, JPEG-AI generally outperforms traditional JPEG compression in terms of visual quality and compression efficiency, making it an attractive option for modern image storage and transmission applications [153, 154].

JPEG-AI's Effect on Error Level

EL is a widely used technique for detecting image manipulations. It works by comparing an original image with its recompressed version to identify discrepancies. These discrepancies, typically manifested as error levels, are used to detect manipulated areas, as the recompressed regions often exhibit more errors due to the manipulation process. However, the effectiveness of ELA depends heavily on the visible compression artifacts, which are utilized to distinguish between manipulated and unmodified areas. When a manipulated image is compressed using JPEG-AI, the resulting error level map will show less pronounced differences compared to traditional JPEG compression. This occurs because JPEG-AI's superior compression capabilities reduce the error levels across both manipulated and non-manipulated regions of the image. As a result, error levels in manipulated regions may not differ significantly from the background, making it harder for EL to detect manipulations.

- **Diminished Error Level Discrepancies:** Since JPEG-AI reduces visible artifacts, error level maps will show smaller discrepancies between manipulated and unmodified regions. This reduction in visible discrepancies could weaken the effectiveness of ELA in detecting tampered areas [155].
- **Loss of Diagnostic Artifacts:** Traditional JPEG compression produces visible artifacts that ELA relies on to identify tampered areas. With JPEG-AI's smoother compression, the absence of these artifacts makes it harder to discern alterations.
- **Challenges in Detecting Covert Manipulations:** When heavily compressed, JPEG-AI may smooth out any discrepancies, making it difficult to detect covert manipulations. The reduced error levels in manipulated regions may prevent clear identification of alterations.

In contrast, traditional JPEG compression produces more noticeable discrepancies between manipulated and original regions, resulting in error level maps with higher contrast. This contrast makes it easier for forensic tools to identify manipulated areas.

JPEG-AI, with its reduced compression artifacts and improved image quality, presents a significant challenge for traditional Error Level Analysis (ELA). While JPEG-AI offers notable advantages in terms of image quality and compression efficiency, its ability to smooth compression artifacts weakens the capacity of ELA to detect manipulated regions. This makes JPEG-AI less suitable for forensic analysis relying on error level maps, particularly when the compression artifacts play a critical role in identifying image manipulation.

3.5 Conclusions

In this chapter, this study introduces **WCBnet**, a novel architecture designed for detecting manipulated images with high precision and robustness. The model integrates

several innovative components: the *Signed-Value Error Level* (SEL) map as input, *multi-level learned feature extraction*, *peak-shaped feature reshaping*, and a *Cross Convolutional-Blocks Weighting Module* (CBWM). These components respectively address the limitations of conventional RGB/EL inputs, fixed feature reshaping strategies, and standard feature fusion mechanisms.

Comprehensive evaluations on the CASIA2.0 dataset demonstrate that WCBnet consistently outperforms state-of-the-art manipulation detection models. Specifically, the proposed CBWM enables the model to adaptively fuse features extracted at various network depths, offering a dynamic weighting mechanism that reflects the significance of detailed and global information. Compared to baselines, WCBnet yields significant improvements in F1-score (ranging from 6.1% to 34.3%), with SEL alone contributing a performance increase of 23% over RGB inputs and 4% over conventional EL maps. On CASIA2.0, WCBnet achieves an average accuracy of 87.07% and an F1-score of 0.879 across all subsets. Additionally, when applied to the learning-based DeepfakeArt inpainting dataset, WCBnet with a larger receptive field (reshaping to peak 5) attains exceptional results (93.9% accuracy, 0.912 F1-score, and 0.981 AUC), demonstrating its robustness to complex generative manipulations and its ability to leverage multi-level features in more abstract or diverse manipulation contexts.

Despite these advancements, WCBnet encounters limitations when applied to small-scale or finely post-processed image manipulation datasets such as CMFDdb and Coverage, where limited sample diversity constrains effective feature learning. While these datasets pose challenges for many state-of-the-art models, WCBnet still achieves the highest reported performance (F1-score of 0.646 on Coverage, surpassing ManTra-Net by 0.162), though results remain below ideal expectations. Nevertheless, the results highlight a broader challenge faced by deep learning-based manipulation detectors: generalizing across data-sparse and real-world noisy conditions. Moreover, while the CBWM offers a compelling mechanism for feature fusion, its explainability remains limited as a black-box classification model. The lack of explicit visualization or attention maps hinders our understanding of how information from manipulated images is being prioritized or suppressed across scales. Similarly, while peak-shaped feature reshaping enhances performance on datasets like CASIA2.0, its fixed design constrains the model’s receptive field to a specific configuration, potentially limiting adaptability across diverse image manipulations with varying spatial distributions. Looking forward, several promising directions can be pursued. First, applying attention-based explainability tools (e.g., Grad-CAM) could help visualize the decision-making process and working principles of proposed modules. Second, a more flexible reshaping strategy that dynamically adjusts receptive fields to accommodate multi-scale manipulations could enhance manipulation detection ability across various datasets.

In summary, WCBnet represents a novel advancement in image manipulation detection models, offering a well-rounded solution with improved robustness, accuracy, and adaptability as a fast fact-checking tool. With additional refinement in explainability and scalability, it holds strong potential for deployment in real-world forensics, journalism and visual content authenticity verification.

Chapter 4

Explainability of Image-wise Manipulation Detection methods and generating dense-view manipulation heatmaps (DenseWCBnet)

Although the WCBnet introduced in Chapter 3 has been shown to accurately distinguish between manipulated and non-manipulated images, it lacks the ability to provide insight into the reasoning behind its decisions. Furthermore, it is constrained by the manually selected peak-shaped structure, offering only a limited receptive field. To address such limitation and 1st to 3rd research questions proposed in Section 2.5, an enhanced model named DenseWCBnet (Densely Weighting Convolutional Blocks for Modelling Error Levels of Manipulated images) will be introduced in this chapter, which expands the receptive field by fully considering all convolutional blocks' and fusing the multi-level features at each of their receptive fields. Besides by leveraging Grad-CAM to visualize the feature maps at different stages, this study provides an intuitive explanation of the proposed structures' functionality, thus achieving explainability.

4.1 Introduction

In an era of explosive information, where image manipulation tools can easily create imperceptible fake visual content, the rapid spread of visual dis/misinformation on social media makes it crucial to develop AI-based tools for the verification of image authenticity. Deep-learning models, capable of extracting abstract high-level features through successive convolution layers, have demonstrated remarkable accuracy in classifying manipulated image. For instance, the WCBnet model proposed in Chapter 3 achieves superior performance across various image manipulation datasets, consistently with F1-score reaching 0.9. However, despite the high accuracy achieved by these image-level approaches, they often struggle to provide a clear explanation for their classification decisions, which can undermine their interpretability and reliability. A network that accurately identifies a vehicle, for example, only has the ability of recognizing specific, unrelated features such as the license plate number, rather than the overall structure of the vehicle itself, so that it cannot be technically introduced as a vehicle recognizing model [156, 157]. This lack

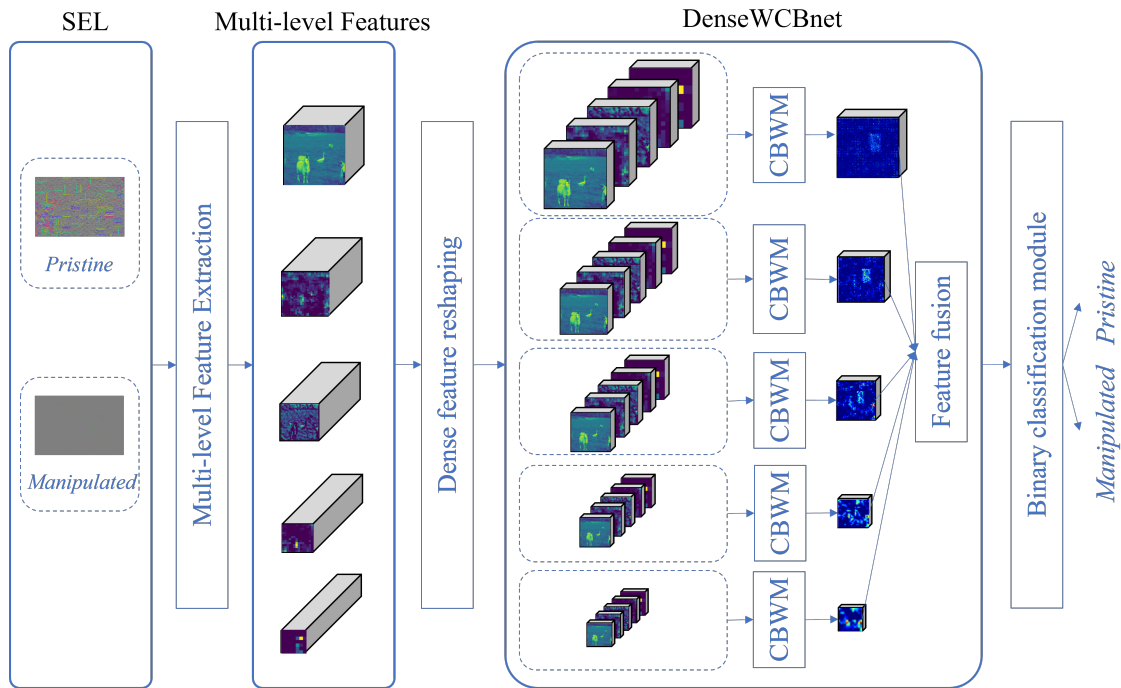


Figure 4.1: Overview of proposed DenseWCBnet, which models manipulation-related error level by densely weighting and fusing multi-level features of hierarchical convolutional blocks.

of transparency in decision-making limits the model’s persuasiveness, especially in applications where understanding the reasons behind identifying the manipulations is critical. Additionally, these deep-learning methods address the challenges of image manipulation detection by fusing high-level features with detailed features from shallow convolutional layers, thereby reusing the delicate manipulation traces from them. However, such feature fusion methods with a fixed reshaping process ignore the different roles played by each level of information. This necessitates an adaptive architecture that can tailor any CNNs for different image manipulations regarding various sizes, resolutions or other image properties. In our previous work WCBnet, a cross-block weighting module CBWM was proposed to adaptively fuse multi-level features extracted by five convolutional blocks. This module computes channel-wise attention across these blocks and optimizes the reshaping process through ablation experiments. While the single-stage feature fusion structure of WCBnet achieve extremely high accuracy over other state-of-art methods, they lack consideration for feature fusion across different receptive fields and perform poorly on small-target manipulations. In a word, how to improve the performance of these image manipulation detection methods in adapting to the complexity of image manipulation datasets, and proving the feasibility of learning-based models, remains a challenge.

Building upon the image-level WCBnet introduced earlier, this chapter presents an advanced model named DenseWCBnet. As shown in the Figure 4.1, the key innovation in DenseWCBnet is the comprehensive consideration of the receptive fields of all extracted features. Rather than relying on the single-view peak-shaped reshaping module used in the original framework, DenseWCBnet fuses the features from all five convolutional blocks at different levels, effectively integrating multi-view information to enhance the manipulation-related feature representation. In addition to this structural enhancement, Grad-CAMs are incorporated to visualize the attention feature maps across various re-

ceptive fields of the original backbone, the feature fusion process across different levels, and the final output in the form of manipulation heatmaps. This combination not only improves feature extraction and fusion but also enhances the interpretability of the model by offering visual insights into how different levels of features contribute to the detection of manipulated regions. DenseWCBnet serves as an advanced image-level manipulation detection method for fact-checking, sharing a similar structure with WCBnet. Both models begin by extracting hierarchical features from Signed-value Error Level (SEL) maps of manipulated images through a series of convolutional blocks. This process is followed by the Cross Convolutional-blocks Weighting Module (CBWM) and a Dense Feature Weighting and Fusing (DFWF). These modules apply trainable weights to convolutional blocks based on their interactions and densely fuse multi-level features across each block’s dimensions. The performance of DenseWCBnet is evaluated on multiple benchmark image manipulation datasets and compared with state-of-the-art methods. Moreover, by visualizing attention heatmaps generated from various convolutional layers, DenseWCBnet enhances interpretability, offering insights into the processes of feature weighting and fusion within key components such as CBWM and DFWF. There are several differences from previous work: 1) densely involving multi-stage feature weighting over all levels of convolutional blocks instead of single-stage peak-shaped reshaping with restricted view; 2) including explainability by generating various-stage feature maps using gradCAM which visualizes the hypothesis dense multi-level fusion and shows the rough manipulation regions; 3) improving the image-level performance of WCBnet over various image manipulation datasets and generating more accurate heatmaps; 4) thoroughly evaluating the effectiveness of SEL using more metrics (like distance between background and foreground histograms) .

To sum up, the main contributions of proposed DenseWCBnet include the following:

- A CNN-based model called DenseWCBnet is designed and proposed for fast fact-checking of image manipulations. This model, compared to previous WCBnet, effectively utilizes multi-level features from hierarchical convolution blocks at dense views of these blocks. It further improves model’s capabilities of modelling JPEG compression-based error levels and achieves superior image-level classification accuracy across multiple datasets;
- An advanced feature fusion module called DFWF is devised which consists of reshaping the convolutional block outputs to each size of them and individual weighted fusion (CBWMs) at each level. Subsequently, these densely-weighted features are fused across multiple receptive domains, resulting in richly dimensioned weighted features;
- The working principle of DenseWCBnet’s components, such as CBWM and DFWF is interpolated by producing CAM-based feature maps of features at different stages, which visualizes the model’s attention on manipulated images, the distinct roles of convolutional blocks, and the feasibility of the proposed feature fusion module.

This chapter about explain-ability and multi-view feature fusion is structured as follows: Section 4.1 begins with a concise overview of related work and then a brief introduction of the proposed work. The details of the proposed methodology are illustrated in Section 4.2, followed by a thorough evaluation of performance in Section 4.3 and conclusions in Section 4.4.

4.2 Methodology

4.2.1 Overview

This work is based on our previous work WCBnet [18] (Chapter 3), and improved by involving the feature fusion over various receptive field and involves the interpretability of each component of the proposed DenseWCBnet. Following the same goal of image-level fast fact-checking to detect manipulated images without ground-truth masks as WCBnet’s, DenseWCBnet shares the similar model structure. As shown in the Figure 4.2, the former structure of DenseWCBnet is the same as WCBnet, namely SEL and CNN-based feature extractor, which extract basic local or global information from error-level manipulation traces. For next feature weighting and fusing stage, the single-view CBWM is replaced by a three-step dense-view feature fusion structure. Considering the varying receptive fields of different convolutional blocks, the proposed model 1) rescales the features to each feature’s dimension as a unified size, 2) applies CBWMs individually at different views for feature weighting and 3) concatenates the five weighted features as final dense-view output. Through this approach, densely weighted-and-fused features are obtained, containing rich information of manipulation traces across multiple levels. The proposed model is evaluated on two common image manipulation types with shared and independent traceable manipulation traces, namely copy-move and splicing, in order to verify its adaptation ability. Furthermore, by computing attention heatmaps via grad-CAM [17] between different convolutional layers of DenseWCBnet, the process of multi-level feature weighting and dense feature fusion is visualized for explainability. The details of each component are illustrated in the following sections.

4.2.2 The SEL Processing and Multi-level Feature Extraction

The proposed DenseWCBnet first computes signed-value error levels (SEL) to expose the compression-related error-level inconsistency between manipulated and pristine regions. SEL involves both positive and negative values of error-level differences considering the abnormal situation mentioned in Section 3.3.1, and enhances those minimal error-level values by proper normalization. The input of DenseWCBnet is still RGB image $\mathbf{x} \in \mathbb{R}^{384 \times 384 \times 3}$, from which signed-value error level $S(\hat{\mathbf{x}})$ is extracted to reveal the JPEG compression-based manipulation traces. To differentiate between the signed-value error levels of the manipulated region $S(\hat{\mathbf{x}}_{\mathbf{m}})$ and the pristine region $S(\hat{\mathbf{x}}_{\mathbf{p}})$, the result of the signed-value error level computation $S(\hat{\mathbf{x}})$ is inputted into a learned feature extractor, such as ResNet50. Both detailed information regarding manipulation traces and abstract information among regions, extracted by shallow-to-deep layers, are essential for image manipulation detection. Incorporating all possible dimensions from the five convolutional blocks of ResNet50, each with specific and consistent dimensions, ensures comprehensive feature extraction across various levels of abstraction. The outputs of all five shallow-to-deep convolutional blocks of ResNet50, denoted as $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$, and $\mathbf{y}_5$, with dimensions 192×192 , 96×96 , 48×48 , 24×24 , and 12×12 , respectively, are utilized. $C(\mathbf{x}, i)$ represents the feature extraction operation from layer i of the Convolutional Neural Network given the input SEL $S(\hat{\mathbf{x}})$ results:

$$\mathbf{y}_i = C\left(S(\hat{\mathbf{x}}), i\right). \quad (4.1)$$

where $\mathbf{y}_i$ denotes the output feature map of convolutional block i of the backbone network. It contains the various levels of features from extracted signed-value error levels.

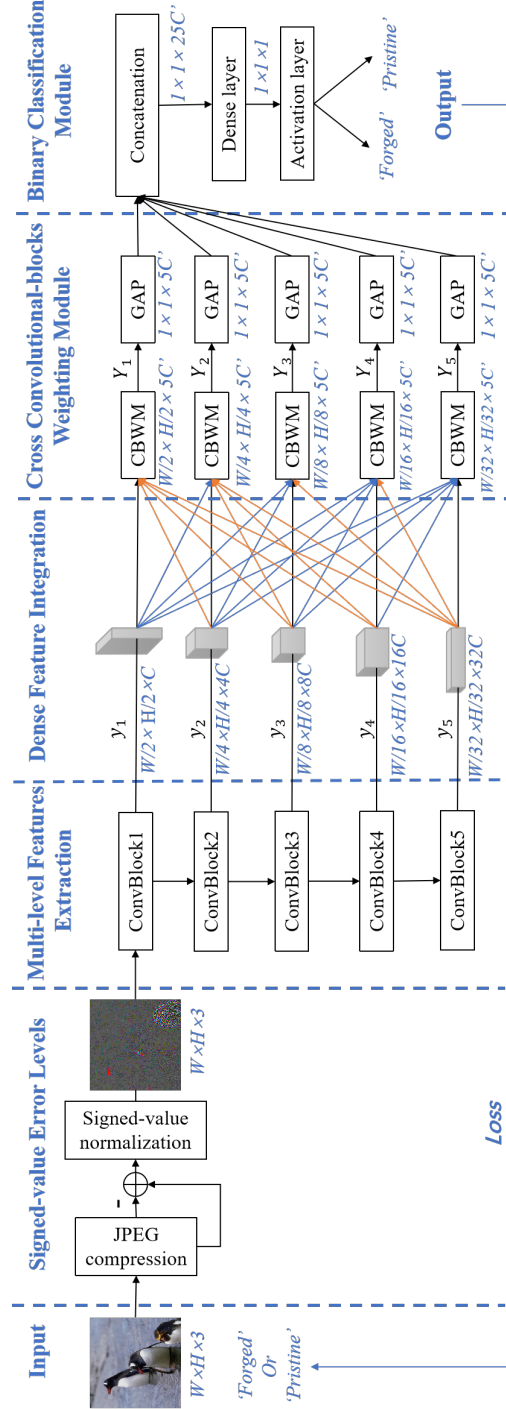


Figure 4.2: Workflow of proposed image-level manipulation detection method DenseWCBnet which consist of five core components; The blue lines and orange lines in Dense Feature Fusion represent down-scaling and up-scaling; In the figure, **ConvBlock_i** is the convolutional block *i* of backbone networks, **CBWM** is the Cross Convolutional-block Weighting Module (as shown in the Figure 4.3), **GAP** is Global Average Pooling layer.

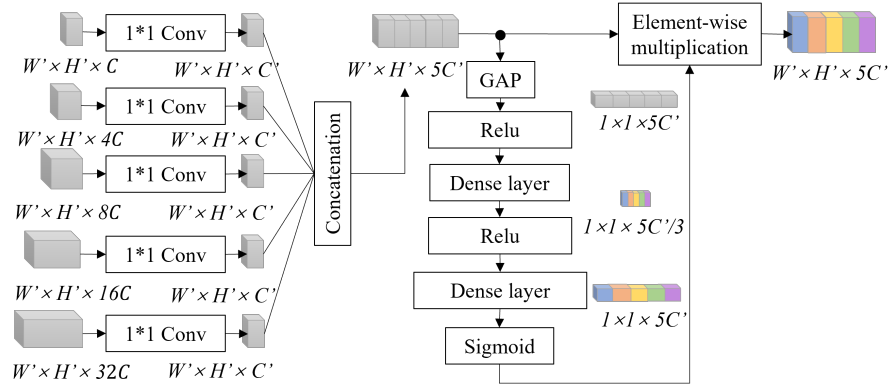


Figure 4.3: Single-view CBWM at the dimension of one convolutional block, which is similar to the one illustrated in Chapter 3 while such CBWM of different scales is included for multiple times.

4.2.3 Single-view CBWM

The next step is to fuse these feature maps $\mathbf{y}_i$ extracted from the convolutional blocks $\mathbf{i}$. The conventional method of multi-dimensional feature fusion involves repetitively up-sampling and concatenating/adding the outputs of adjacent convolutional blocks [144, 62]. However, as discussed earlier, features extracted by shallow or deep layers of CNNs play distinct roles for various image manipulation detection tasks. Accordingly, single-view CBWM, as illustrated in Section 3.3.4, is applied to assign weights for these multi-level features $\mathbf{y}_i$ at the feature scale of convolutional blocks. The advancement in the Chapter is that five CBWMs with each feature's scale is involved, which means that the inner-relationship among convolutional blocks is individually computed at their own scales (each scale is the shape of $\mathbf{y}_j$ while $\mathbf{j}$ is selected from 1 to 5). As shown in Figure 4.3, the single-view CBWM has the same workflow as WCBnet's and one of five CBWMs which fuses the five feature maps $\mathbf{y}_i$ at the level of feature $\mathbf{y}_j$'s view is described as follows:

Initially, each feature $\mathbf{y}_i$ is reshaped to a unified size $\mathbf{y}'_{ij}$, where i denotes the convolutional block and j represents the selected block with the target shape. In the Cross Convolutional-blocks Weight Module (CBWM), the channel number of all reshaped features is standardized to C' using 1×1 convolutional layers, ensuring consistent weight ranges across blocks. An attention mechanism is then applied uniformly across all reshaped blocks $\mathbf{Y}'_j$ to compute feature interactions, with global pooling producing a vector $\mathbf{W}_j$ from $\mathbf{Y}'_j$. This vector passes through a ReLU activation layer, creating trainable channel weights. Dense layers connect all channels of $\mathbf{W}_j$, establishing cross-block channel relationships and resulting in the weight $\mathbf{W}'_j$. Finally, element-wise multiplication of $\mathbf{W}'_j$ with the concatenated feature $\mathbf{Y}'_j$ assigns channel-specific weights to block outputs, thus weighting the convolutional block outputs $\mathbf{y}_i$ accordingly.

4.2.4 Dense Feature Integration

When employing CBWM or other feature fusion methods, it is inevitably to standardize features of varying sizes by upscaling or downscaling, which results in loss or inaccuracies in information related to manipulation traces. Additionally, such feature reshaping process with unified sizes restricts the CNNs' perspective to a singular view, limiting its ability to capture the comprehensive information. In previous work, the rescaled feature shape is optimized by massive ablation experiments, but it still cannot avoid the prob-

lems caused by the single-view CBWM. To address these limitations, this study proposes a Dense Feature Integration approach by thoroughly considering all possible rescaling procedures. As shown in the Figure 4.2, each feature extracted from convolutional blocks are individually standardized to match the dimensions of these blocks, facilitating feature integration across multiple views. Subsequently, employing independent CBWM on each view facilitates the weighting and fusion of features at different levels. This dense integration process ensures that the resulting features encapsulate a rich and diverse representation of information from all blocks.

The outputs of the Convolution blocks, denoted as $\mathbf{y}_i$, undergo individual resizing to different shapes $\mathbf{y}'_{ij}$ with dimensions $W' \times H'$, which correspond to the shape of each $\mathbf{y}_j$. These resized outputs are then concatenated to form five all-block features $\mathbf{Y}'_1, \mathbf{Y}'_2, \mathbf{Y}'_3, \mathbf{Y}'_4, \mathbf{Y}'_5$. By computing CBWM individually over all five feature levels, the input features $\mathbf{Y}'_i$ are weighted based on the inter-relationship among all convolutional blocks, resulting in $\mathbf{Y}_i$. With the objective of image-wise fact-checking, each weighted cross-block feature $\mathbf{Y}_i$ is globally pooled into an image-wise vector of shape $1 \times 1 \times 5C'$ and concatenated along channels as the final multi-view features $\mathbf{Y}$ of shape $1 \times 1 \times 25C'$.

4.2.5 Binary Classification Module

The all-block feature $\mathbf{Y}$ from the last step that has densely weighted and fused features of convolutional blocks is downscaled into a scalar representing the probability of belonging to either forged or pristine ($1 \times 1 \times 1$) using a dense layer with Sigmoid activation. Subsequently, based on the predicted class with the highest probability and the given image-level label, the loss is calculated batch by batch and back-propagated to update the model parameters' weights through gradient descent. During training, the parameters in dense-view CBWM are updated to adjust the model's utilization of different convolutional blocks based on the properties and manipulation types present in the training dataset.

4.2.6 Explainability: Feature Maps and Dense-view Heatmap

In addition to achieving high classification accuracy, image-level deep-learning methods often lack explainability, offering limited insight into decision-making processes and the reasons behind performance improvements. To address such limitation, this study incorporates gradient-weighted class activation mappings (grad-CAMs) at various stages of the proposed workflow, enabling visualization of the feature maps that indicate the model's focus areas when classifying manipulated images. Then by integrating activation levels from the dense layer with multi-view feature maps from the weighted convolutional blocks $\mathbf{Y}_i$, this research defines Dense-CAM, a novel explainability solution generating detailed heatmaps that highlight regions of image manipulation.

Feature maps $\mathbf{f}_i$ of original convolutional blocks $\mathbf{y}_i$ To obtain the regions where the deep-learning model focuses, Grad-CAM calculates the gradient of the target class score with respect to the feature maps from any convolutional layer, producing 'importance' weights for each feature map. Since Grad-CAM is applicable between any two convolutional layers, this study computes several Grad-CAMs between the features before& after CBWMs to show how DenseWCBnet's focus changes during classifying manipulations. As shown in the Figure 4.4, Grad-CAMs are computed in the two stages of the proposed workflow. The features $\mathbf{y}_1$ to $\mathbf{y}_5$ extracted from 5 convolutional blocks and the manipulated class score extracted from the final Dense layer are first used to compute first-stage feature maps $f = \{f_1, f_2, \dots, f_5\}$ which visualize where these convolutional blocks focus

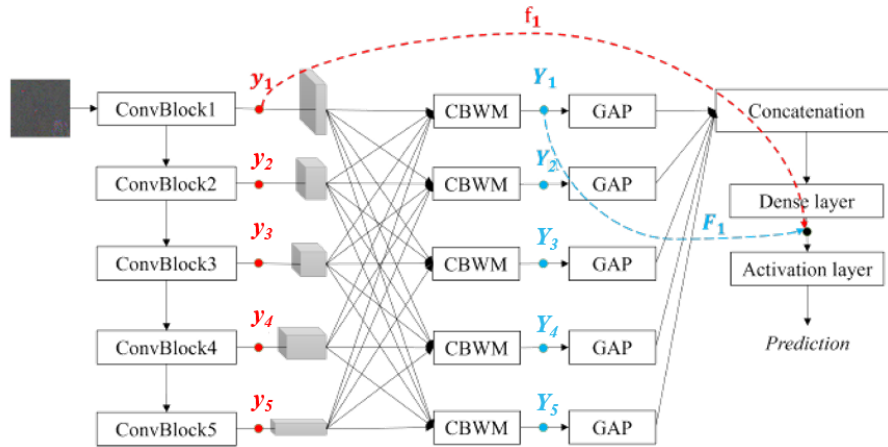


Figure 4.4: The generation of feature maps for explainability; The dotted lines represent the grad-CAMs, while different colors mean the feature maps in two stages.

(the process is marked as **red** in the Figure 4.4). These basic feature maps are considered as the backbone and are then used for comparison with weighted features and final dense-view heatmaps.

Feature maps F_i of CBWM-weighted features Y_i The feature maps in the next stage are obtained by Grad-CAM between the convolutional layer Y_i after each CBWM and the last neuron of the model output layer (as shown in the **blue** dotted lines of the Figure 4.4). As introduced in the last Chapter, CBWM sets weights for each convolutional block output and fuses them to jointly involve all levels of information. Moreover, the novelty of DenseWCBnet is that CBWMs at all five feature scales are involved which individually achieves weighted feature fusion processes thoroughly. The feature maps (defined as $F = \{F_1, F_2, \dots, F_5\}$) of Y_i are visualized to show how CBWMs affect the model's focus during image manipulation detection regarding different receptive fields.

Dense-view manipulation heatmap H The final step is to combine multi-view feature maps F , in order to obtain a comprehensive region of interests of DenseWCBnet when classifying input images as manipulated. The first and most straightforward combination method is to compute the average of the n feature maps ($n = 5$ in this study), which is defined as follows:

$$H = \frac{\sum_i F_i}{n}. \quad (4.2)$$

This approach enhances manipulated regions by accumulating strong activations from all views, enabling fine details from high-resolution maps to complement broader context from lower-resolution ones. However, it also has a key drawback that errors or noise from each feature map are equally aggregated, which can result in false activations in the final heatmap.

To address this issue, an alternative method is to compute their geometric mean, which involves normalizing the values of the feature maps to a range of 0 to 1 and then combining them by taking the cumulative square root. The geometric mean of F_1 to F_n

is defined as ($n = 5$ in this study):

$$H = \sqrt[n]{\prod_{i=1}^n \frac{F_i}{\max(F_i)}}. \quad (4.3)$$

The geometric mean suppresses low-confidence or noisy activations more effectively, since any low value in the product tends to decrease the final result, leading to more discriminative heatmaps. This noise-suppressing property has been successfully applied in other areas such as image fusion [158], and more recently, in multi-scale manipulation detection frameworks [159, 160]. Given its advantages in reducing noise accumulation while maintaining the spatial consistency of manipulated features, this study selects the geometric mean to generate the final heatmap H . This choice improves the localization precision of the manipulation class and leads to clearer and more reliable predictions.

The proposed DenseCAM enhances the explainability of DenseWCBnet by mapping dense-layer activations directly to manipulation-sensitive regions, thereby providing a comprehensive multi-dimensional view of image manipulation across various feature scales. The effectiveness of DenseWCBnet will be evaluated by its image-level performance over multiple datasets and pixel-wise multi-view feature maps and dense-view heatmap.

4.3 Experimental Results and Discussions

This section begins by presenting the experimental setups and the datasets used for image manipulation detection. Subsequently, the experimental results demonstrate the image-wise performance of DenseWCBnet on commonly used image manipulation datasets, comparing it with other state-of-the-art methods. Furthermore, the effectiveness of each component of DenseWCBnet is discussed based on additional image-wise classification experimental results and dense-view heatmaps generated by explainability method (Grad-CAM [161]).

4.3.1 Experimental Setup

Most of the experimental setup is the same as descriptions in the Section 3.4.

Hardware The model is built in the environment as WCBnet’s depending on TensorFlow version 2.7 and Python version 3.7. Specific functionalities are only executable under TensorFlow’s graph mode within this environment. The experiments are run on a PC with NVIDIA GeForce RTX 3090 Ti, 12th Gen Intel(R) Core(TM) i9-12900K 3.20 GHz processor and 32.0 GB RAM.

Datasets To make the experimental results comparable with WCBnet, the DenseWCBnet is trained and evaluated on the same image manipulation datasets. These images are manipulated by splicing (which splices objects from sources into a new background) , or copy-move (which copies the region and pasting it on the same image). The details of their manipulation types, image size and image number are summarized in Table 3.1.

Model complexity The training parameters of backbone (ResNet50), WCBnet, and DenseWCBent are 23.6 million, 24.1 million (with 2.3% increase) and 24.8 million respectively. The Floating Point Operations Per Second (FLOPS) is also applied to evaluate

the complexity of the models. The total FLOPs of three models are 22.78 GFLOPs, 23.06 GFLOPs and 25.77 GFLOPs respectively.

Training details In order to study the impact of different manipulation methods on the proposed method, each dataset is divided into splicing and copy-move subsets to separately evaluate the performance of the model. Then, these subsets are split into training sets, validation sets and test sets as the ratio of 7:2:1 and they will not change in subsequent experiments once split, in order to ensure training and testing on the same data. The large-scale CASIA2.0 dataset is used for optimizing the model, in which approximately 2000 images are selected from each subset, namely, splicing, copy-move and authentic. When evaluating on other datasets, transfer-learning is applied to transfer the features learned from CASIA2.0 into other image manipulation datasets. For each subset, normal ELA and proposed SEL are extracted from images and saved as PNG files as preprocessing. In case one of manipulated or pristine classes has imbalanced number of images, this study applies binary-class F1-loss as the image-level loss function.

Evaluation metrics For the purpose of fast fact-check, the model is trained and evaluated using image-level metrics for manipulated-or-pristine binary classification tasks, such as accuracy, F1-score, and AUC (area under curve).

Moreover, there is a pixel-wise evaluation metric to directly measure the similarity between error levels and ground truth, which is called Bhattacharyya distance. This metric is well-suited for binary image manipulation detection, as it effectively quantifies the similarity between two distributions by comparing their feature histograms. In the context of image manipulation detection, where manipulated regions are contrasted with ground truth masks, a smaller Bhattacharyya distance indicates greater similarity between the predicted manipulation map and the actual manipulated areas. Given the normalized histograms of error levels S and ground-truth masks G , the Bhattacharyya coefficient $BC(S, G)$ is calculated as:

$$BC(S, G) = \sum_i \sqrt{S(i)G(i)}. \quad (4.4)$$

The Bhattacharyya distance D is then defined by:

$$D(P, Q) = -\ln(BC(S, G)). \quad (4.5)$$

This metric is widely used in image processing tasks, such as object tracking and edge detection [162, 163], and is particularly effective in assessing the spatial alignment between the manipulated and original regions in binary classification problems [164, 165]. Its application provides an efficient method for evaluating the similarity between manipulated and ground-truth images, particularly in the context of detecting image manipulation localization tasks.

4.3.2 Comparison with State-of-art Methods Regarding Image-level Performance

The image-wise performance of proposed DenseWCBnet is compared to several state-of-the-art methods, including MantraNet [141], CR-CNN [142], FCN-16 [143], MVSS-Net [13], MVSS-Net++ [13], Modified-CNN [144], U2-Net [62] and our previous work WCBnet, via two evaluation metrics that are AUC and F1-scores. The results in TABLE 4.1 highlight the superior performance of DenseWCBnet and WCBnet compared to other

Table 4.1: Image-level performance of the proposed method and the state-of-art models; The best and the second-best methods for each dataset are shown in red and blue, respectively); This table is extended from Table 3.2 of Chapter 3

Model	Testing set									
	CASIA1.0		Defactos		Columbia		NC2016		Coverage	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
ManTra-Net[141]	0.297	0.141	0.000	0.543	0.621	0.681	-	-	0.484	0.491
CR-CNN[142]	0.361	0.766	0.397	0.567	0.392	0.783	0.939	-	0.131	0.566
FCN-16[143]	0.775	0.796	0.458	0.551	0.481	0.762	-	-	0.180	0.541
MVSS-Net[13]	0.752	0.839	0.404	0.573	0.802	0.980	-	-	0.244	0.731
MVSS-Net++[13]	0.694	0.862	0.478	0.531	0.930	0.984	-	-	0.895	0.726
umUNET[144]	0.565	-	-	-	0.761	-	-	-	0.423	-
U2-Net[62]	0.557	-	-	-	0.756	-	0.911	-	0.400	-
WCBnet [18] (Chapter 3)	1.000	1.000	0.801	0.818	0.863	0.950	0.751	0.560	0.646	0.503
DenseWCBnet	1.000	0.994	0.914	0.977	0.816	0.923	0.639	0.653	0.594	0.790

competing models in detecting image manipulations. DenseWCBnet achieves perfect F1-scores and AUCs of 1.000 and 0.994 on the CASIA1.0 dataset, and outstanding scores of 0.914 and 0.977 on the Defactos dataset, significantly outperforming MVSS-Net and other models. Similarly, on the Columbia dataset, DenseWCBnet maintains high F1 and AUC scores of 0.816 and 0.923, respectively, showcasing its robust detection capabilities across diverse datasets. WCBnet, while slightly less performant than DenseWCBnet, still demonstrates substantial improvements over traditional models. It achieves perfect scores on CASIA1.0 and competitive results on Defactos with an F1-score of 0.801 and an AUC of 0.818. Additionally, WCBnet’s consistent performance on the Columbia and NC2016 datasets further underscores its reliability. Overall, DenseWCBnet and WCBnet exhibit clear advantages in detecting image manipulations, with DenseWCBnet’s dense connectivity architecture providing a marked improvement in capturing intricate manipulation artifacts.

4.3.3 Effectiveness of Signed-value Error Level

Signed-value error-level is the first core component of the proposed method, which distinguishes forged and pristine regions by the inconsistent levels of JPEG compression-based errors. Several experiments are conducted to evaluate the effectiveness of SEL application compared to other types of inputs, regarding histogram difference, image-wise performance and SEL ablation study.

Comparing Foreground and Background in RGB, EL and SEL maps

Since error levels could distinguish the data discard patterns between manipulated region and authentic region, this research examines the separability of manipulated regions from pristine background, histograms were computed over the manipulated foreground, pristine foreground, and shared background areas, using the RGB image, its corresponding ELA, and the proposed SEL map. An example is shown in the Figure 4.5 which extracts the manipulated foreground and intact region from the pristine images with the same background.

From the histograms of manipulated pristine regions in Figure 4.6, the distribution or separability of the same image regarding RGB, EL and SEL maps varies. In the RGB domain, pixel distributions of all regions heavily overlap, with manipulated foreground showing slightly broader intensity range but limited distinction from background and pristine regions. ELA histograms, by contrast, reveal clearer separation: manipulated regions

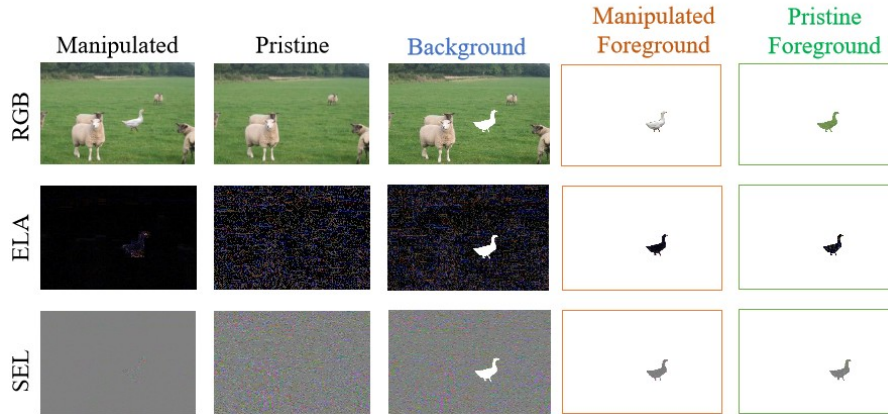


Figure 4.5: Comparison between the foreground and background of pristine and manipulated image, regarding RGB, EL and SEL maps.

Table 4.2: The comparison of histogram similarity, **Distance_S** represents the distance between background and manipulated foreground while **Distance_A** is between background and authentic foreground.

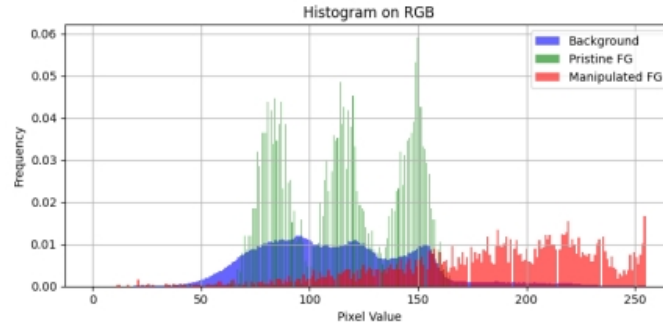
	RGB	ELA	SEL
Distance_A	0.841 ± 0.012	1.138 ± 0.078	0.251 ± 0.065
Distance_S	0.697 ± 0.024	0.718 ± 0.097	0.536 ± 0.181

exhibit concentrated error patterns in mid-low intensity bins (5–20), while pristine and background pixels remain largely near zero. SEL further enhances this separability by centering all pixel values around 125 and expanding manipulated region variance symmetrically across a wider range, forming a near-Gaussian distribution. This structured divergence in SEL histograms effectively shows differences between manipulated and non-manipulated regions, which means that the pixels belonging to those two image classes are easier to distinguish.

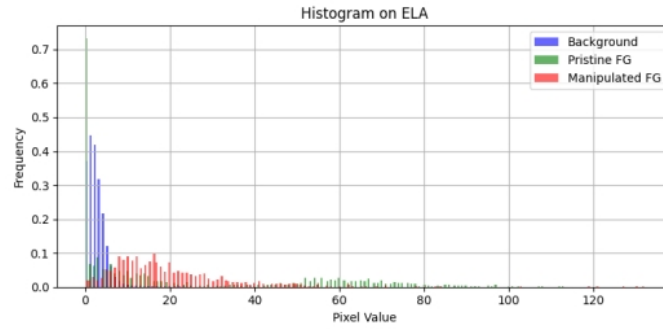
In order to further analyse these differences numerically, Bhattacharyya distance is used to measure the similarity between two histograms belong to manipulated/pristine regions (probability distributions). Bhattacharyya distance between manipulated foreground and background in manipulated images **Distance_S** and the distance between authentic foreground in the corresponding region and pristine background in pristine image **Distance_A** are calculated. As shown in the TABLE 4.2, the average **Distance_A** and average **Distance_S** in SEL have the largest difference which are 0.251 and 0.536 respectively, which means that manipulated region and pristine region are comparatively distinguishable in SELs.

Comparing Image-level Performance Regarding RGB, ELA and SEL

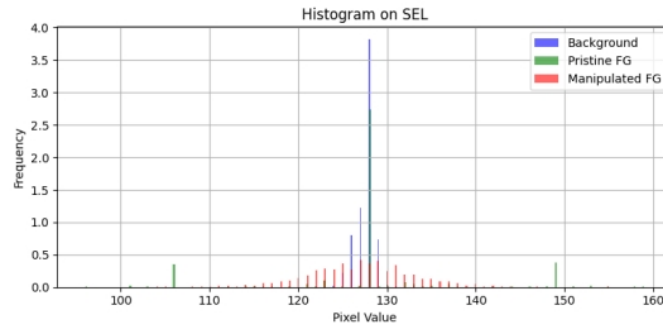
The effectiveness of SEL is evaluated by comparing the performance of DenseWCBnet on RGB, ELA and SEL inputs of three forged subsets of CASIA2.0. The difference between WCBnet and DenseWCBnet is also shown in the TABLE 4.3. Regarding those three input types, error-based results have a significant improvement compared to RGB, increasing the F1-score of ResNet, WCBnet and DenseWCBnet by 14%, 21.9% and 7.2% respectively. On the basis of normal ELA, the application of SEL increases F1-scores by 1.7 to 3.5 percentage points. It can be observed that DenseWCBnet except for modelling



(a) RGB



(b) EL



(c) SEL

Figure 4.6: Comparison of Foreground (FG) and Background (BG) Histograms in RGB, ELA, and SEL Formats, with Pristine Foreground in *Green*, Shared Background (BG) in *Blue*, and Splicing Foreground in *Red*.

Table 4.3: The image-level F1-score of backbone network, WCBnet and DenseWCBnet, on three CASIA2.0 subsets regarding RGB, ELA and SEL respectively; The best performance is marked as *red*

	RGB			ELA			SEL		
	Sp	CM	Mixed	Sp	CM	Mixed	Sp	CM	Mixed
ResNet50	0.634	0.803	0.798	0.905	0.856	0.894	0.911	0.909	0.925
WCBnet	0.702	0.616	0.695	0.891	0.867	0.912	0.928	0.913	0.935
DenseWCBnet	0.875	0.845	0.815	0.931	0.912	0.908	0.934	0.926	0.943

Table 4.4: The image-level performance of proposed method and backbone on different error levels or noise levels of manipulated images; The best performance is marked as *red*; This table is extended from Table 3.3 of Chapter 3.

Error type	ResNet50			WCBnet			DenseWCBnet		
	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
WEL	89.8%	0.862	0.958	91.8%	0.884	0.980	92.0%	0.893	0.959
Comprint	80.5%	0.701	0.898	89.3%	0.834	0.959	88.5%	0.859	0.948
ELA	91.7%	0.887	0.968	92.8%	0.925	0.988	94.8%	0.931	0.975
SEL	91.8%	0.915	0.987	95.1%	0.933	0.994	95.0%	0.943	0.978

errors, also performs well on RGB inputs, reaching an average F1-score of 0.85.

Comparing Image-level Performance Regarding Other Error-modelling Methods

Similar to previous experiments in Section 3.4.3, there are some other methods to extract the error levels exposing image manipulation traces, such as Wavelet-based error levels (WEL) and Learning-based Comprint. Some error-level samples have been shown in the Figure 3.8. The image-wise performance of DenseWCBnet is included as additional results to Table 3.3, so the comparison between backbone, WCBnet and DenseWCBnet regarding these different types of error levels is presented in Table 4.4. The results reveal distinct advantages of proposed DenseWCBnet over the conventional ResNet50. Specifically, DenseWCBnet consistently achieves the highest accuracy and F1 scores, underscoring its superior capability in handling various manipulated image noise and error levels. For instance, DenseWCBnet attains an accuracy of 92.0% and an F1 score of 0.893 for wavelet-based error levels (WEL), outperforming both ResNet50 and WCBnet. Similarly, in Error Level Analysis (ELA), DenseWCBnet’s accuracy of 94.8% and F1 score of 0.931 markedly exceed those of ResNet50. Moreover, while WCBnet often trails DenseWCBnet in accuracy and F1 scores, it consistently achieves the highest AUC values across all error types, with a notable AUC of 0.994 in SEL. To sum up, WCBnet excels in distinguishing between classes with a higher degree of certainty, as indicated by its superior AUC scores while DenseWCBnet provides the most robust overall performance. Consequently, the integration of these advanced network architectures into image manipulation detection frameworks can substantially enhance detection accuracy and reliability, particularly in complex scenarios with varying error characteristics.

4.3.4 Effectiveness of CBWM

The next core component of DenseWCBnet is Cross Convolutional-blocks Weighting Module, which sets trainable weights for each convolutional block make adaptive use of extracted multi-level features. To evaluate the effectiveness of CBWM on those features, this study compares the performance of backbone network and single-stage CBWM (which is

proposed in WCBnet paper) on different image manipulation detection tasks. The experimental results regarding image-wise performance, explainability of feature weighting and CBWM structure ablation are shown as follows.

Image-wise performance using single-stage CBWM

To evaluate the effectiveness of CBWM, the classification performances of backbone network (ResNet50) and WCBnet with single-stage CBWM are compared, regarding different image types, manipulation types and manipulation datasets. As shown in the TABLE 4.3, image-wise F1-scores of ResNet50 and WCBnet (with single stage CBWM) on CASIA2.0 subsets are compared. ResNet50, that has been pre-trained by ImageNet object detection datasets, is better at distinguish between objects from RGB images. So F1-score of ResNet50 on RGB CASIA2.0 subsets is larger than WCBnet by around 7% on average. Since WCBnet is designed to model the error levels, its performance on ELA or SEL subsets is improved from 0.88 and 0.91 to 0.92 on average respectively. More comparison experimental results are shown in the TABLE 3.3, CBWM from WCBnet achieves better modelling on error levels or noise-level for detecting manipulated images which improves the overall performance on noise-inputs by 3.8% accuracy, 5.3% f1-score and 3.3% AUC.

Explainability of CBWM

As described in Section 4.2, the objective of the Convolutional Block Weighting Module (CBWM) is to assign weights to features extracted by five convolutional blocks $\mathbf{y}_i$. Initially, feature maps are computed for each block output layer $\mathbf{y}_i$ in relation to the image-level output of the trained WCBnet using Gradient-weighted Class Activation Mapping (gradCAMs). This step highlights the differences in the model’s attention across various convolutional layers. Subsequently, the manipulation heatmap is derived from the weighted block that is generated by single-stage CBWM and the model output of WCBnet. This heatmap is single-view, as the convolutional blocks are weighted and fused in CBWM with a fixed shape (Y_4). Figure 4.7 presents the multi-level feature maps and the single-view heatmap of WCBnet. The heatmaps reveal that shallow convolutional blocks focus on detailed information, capturing rich manipulation details, as seen in block 2 of the first two examples. However, these blocks also introduce considerable redundant background information. In contrast, the attention of deeper convolutional blocks is more diffused, concentrating on global abstract information. For instance, in block 5 of the third example, the maximum heat points appear in the manipulated region, indicating focused attention, yet some valid information within the manipulated area is lost. The final column demonstrates that CBWM effectively integrates multi-level information from different convolutional blocks, thereby enhancing the detection of suspicious manipulations within the image.

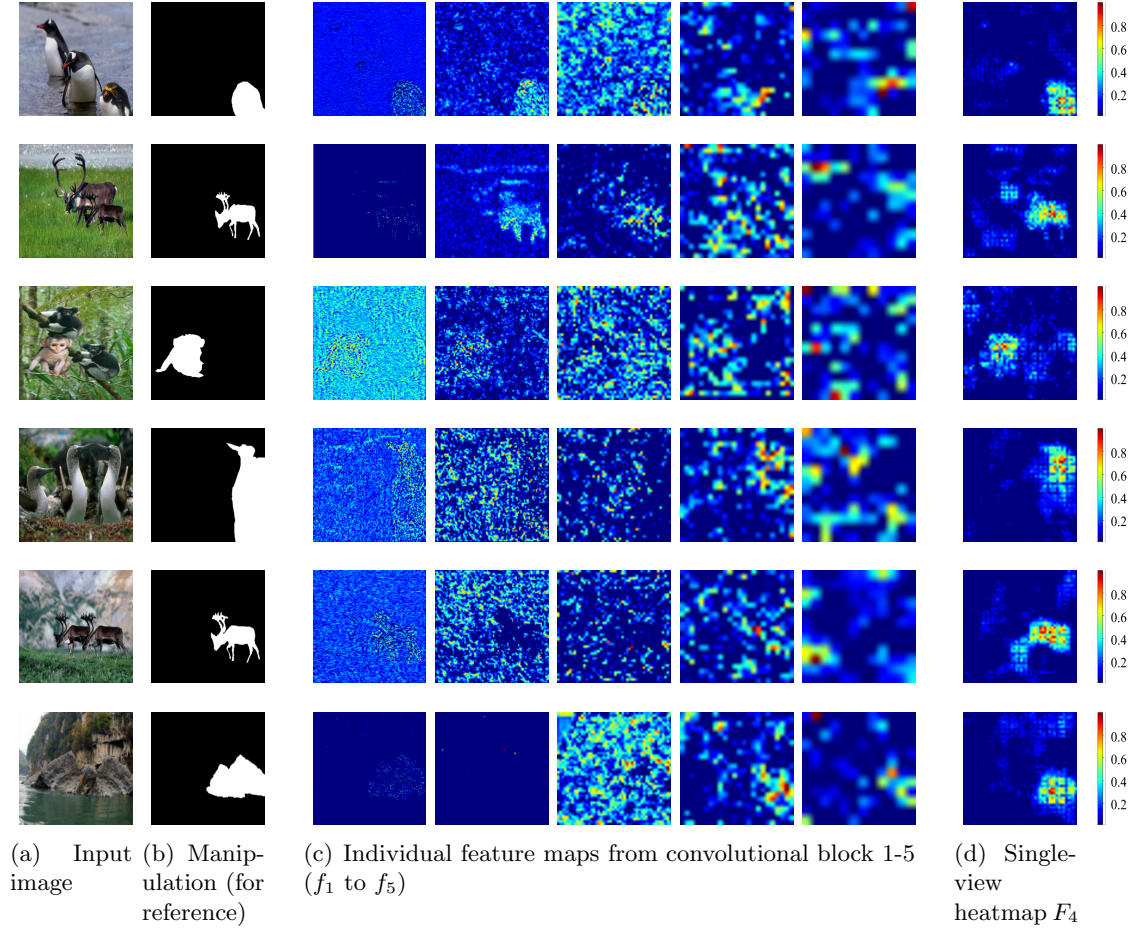


Figure 4.7: Various-stage feature maps generated from image-level WCBnet using gradCAM; The feature maps in middle columns show the attention of five convolutional blocks (shown as f_i in Figure 4.2), while the last column shows the single-view heatmaps of weighted block (F_4 in Figure 4.2) which is generated using single-stage CBWM with fixed feature shape; The first three rows show the results of splicing images, and the last three are copy-move maps.

4.3.5 Effectiveness of Dense Feature Weighting and Fusion

Although CBWM could weight and fuse the features, the single-stage CBWM in WCBnet only considers the interaction of the convolution blocks in single dimension. The largest difference between WCBnet and DenseWCBnet is that CBWMs are individually computed on features of all scales that densely weight and fuse multi-level features. The effectiveness of Dense feature weighting and fusion is evaluated by image-wise classification performance and visualized by grad-CAM maps of densely-weighted features.

Image-wise performance using Dense Feature Weighting and Fusing (DFWF)

The comparison between WCBnet with single-stage CBWM, and DenseWCBnet with dense CBWM is shown in Table 4.1, Table 4.3 and Table 4.4. DenseWCBnet achieves best or second-best performance over most of the testing sets, among which it significantly improves the Defactos-based performance to 0.914 and 0.977 regarding F1-score and AUC. For the ability of modelling error levels, the F1-score performance of DenseWCBnet is higher than WCBnet by 2.3% and 2.9% regarding splicing and copy-move subsets of

CASIA2.0. As shown in the Table 4.4, DenseWCBnet performs better than WCBnet on modelling different noise-level inputs, which improves the F1-score by 0.9% and 2.5% on Wavelet-based errors and Comprint compared to WCBnet.

Explainability of DFWF

As illustrated in experimental results 4.3.4, the single-view heatmap of WCBnet roughly highlights regions in multi-level feature maps that respond intensely to manipulated image classification. However, these single-view heatmaps are constrained by the limited receptive field of reshaped features. To generate dense-view heatmaps and visualize the effectiveness of Dense Feature Weighting and Fusing (DFWF), different heatmaps are computed between each weighted feature $\mathbf{Y}_i$ of DenseWCBnet and its image-level output. As depicted in the Figure 4.8, the heatmaps in the initial dimensions clearly delineate the shape of the manipulated objects while reducing redundant interference points in the background, as compared to the results in Figure 4.7. In the deeper convolutional block dimensions, the weighted blocks capture more pertinent information in manipulated regions, generating higher intensity points in the maps. This indicates stronger responses in these regions when classifying manipulated images. Because the five convolution blocks are weighted and fused across each receptive field, the densely-weighted blocks 1-5 retain the global attention heat points and detailed features of the manipulated areas. By computing the geometric mean of these dense-view heatmaps, the manipulation heatmap focuses on the manipulation regions and adjacent edge pixels. Thus, DenseWCBnet effectively extracts features from manipulated images by densely weighting and fusing convolutional blocks, integrating global information from deep features and local information from shallow features.

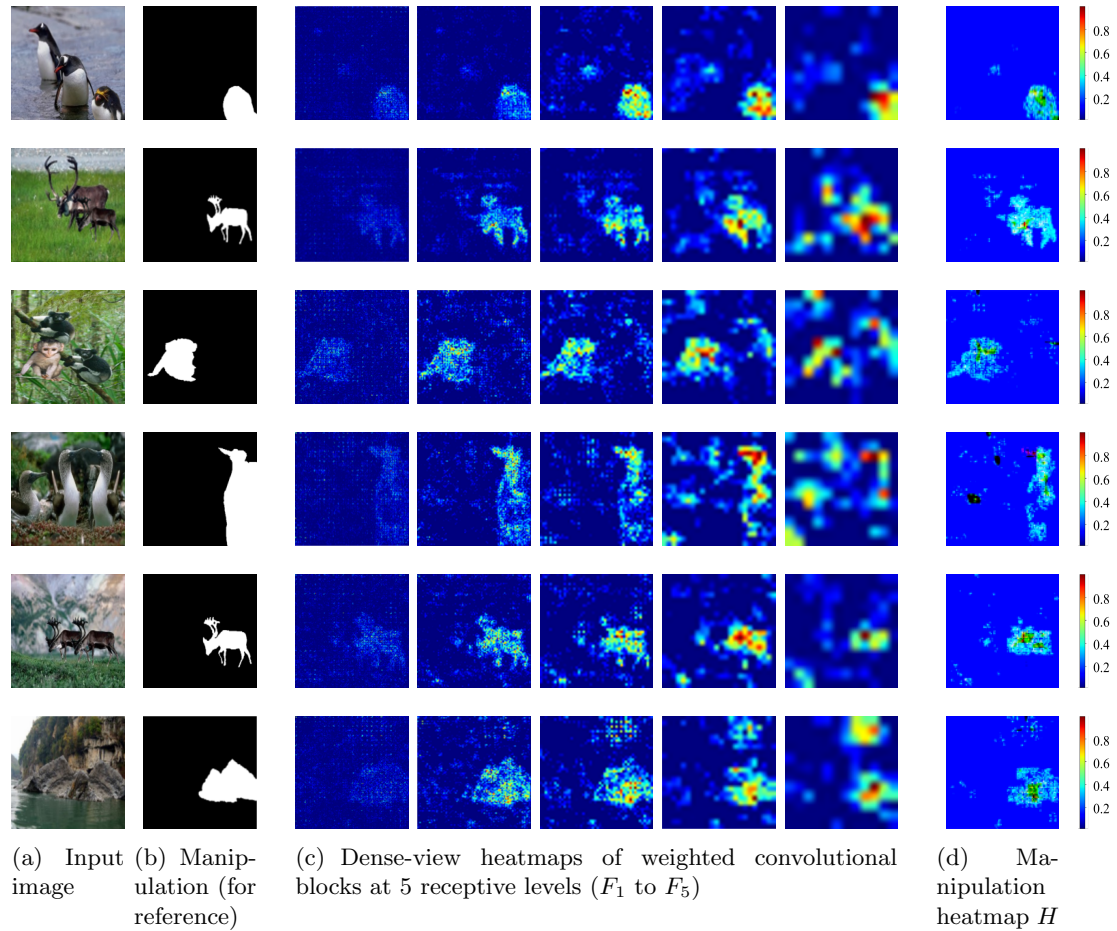


Figure 4.8: Dense-view heatmaps generated from image-level DenseWCBnet using gradCAM; Heatmaps in the middle columns show the attention of densely weighted features (F_1 to F_5 in Figure. 4.2), which are processed by different CBWMs at the level of 5 weighted blocks; The last column shows manipulation heatmap H which fuses the dense-view heatmaps with weights, obtained from the activation values of the dense layer; The first three rows show the results of splicing images, and the last three are copy-move.

4.3.6 Robustness Test Against Perturbations

As previously mentioned in Section 3.4.6, the manipulated images are vulnerable to various types of post-processing in order to refine the manipulation traces. The performance of DenseWCBnet is evaluated on the CASIA2.0 with 7 types of common perturbations, and compared with the results of backbone and WCBnet (the details of perturbations could be found in Figure 3.11 and Table 3.11). The robustness tests presented in the Table 4.5 clearly illustrate the superior performance of DenseWCBnet across multiple perturbation scenarios. Notably, DenseWCBnet achieves the highest or near-highest Accuracy, F1-score, and AUC metrics in most cases. For instance, under Gaussian noise perturbation, DenseWCBnet outperforms both ResNet50 and WCBnet with an AUC of 0.887 and an F1-score of 0.779. Similarly, for blurring perturbation, DenseWCBnet achieves an Accuracy of 83.0%, an F1-score of 0.803, and an AUC of 0.892, all of which are the highest among the tested models. DenseWCBnet also demonstrates robust performance under Salt & Pepper noise and JPEG compression, particularly excelling in the

Table 4.5: The image-wise performance of the backbone, WCBnet and DenseWCBnet on the CASIA2.0 with different perturbations; The best performance for each perturbation method is marked as **red**; This table is extended from Table 3.11 of Chapter 3.

Perturbations	ResNet50			WCBnet			DenseWCBnet		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
Gaussian noise ($\sigma = 3$)	50.0%	0.654	0.648	79.9%	0.766	0.877	79.4%	0.779	0.887
Salt&pepper noise (0.02)	71.4%	0.702	0.797	68.7%	0.681	0.763	72.5%	0.723	0.793
Blurring ($ksize = (3,3)$)	79.4%	0.757	0.862	80.2%	0.747	0.879	83.0%	0.803	0.892
Contrast adjustment ($\alpha = 1.0$)	83.8%	0.805	0.886	81.3%	0.776	0.877	79.9%	0.777	0.857
Scaling (0.8)	76.1%	0.728	0.845	81.3%	0.788	0.889	79.1%	0.785	0.877
JPEG compression (0.9)	75.3%	0.748	0.844	78.6%	0.734	0.853	79.7%	0.768	0.877
JPEG compression (0.5)	77.5%	0.733	0.830	79.4%	0.764	0.862	79.4%	0.775	0.872

Table 4.6: The image-level performance of DenseWCBnet on Deepfake datasets, compared with the best model in DeepfakeArt challenge and optimal WCBnet (shown in the Table 3.12 of Chapter 3); The best performance is marked as **red**.

	DeepfakeArt		
	Accuracy	F1-score	AUC
Best mentioned in the paper (DINOv2 [152])	82.75 %	0.800	-
WCBnet	93.9%	0.912	0.981
DenseWCBnet	97.0%	0.951	0.993

F1-score and AUC metrics. This indicates its ability to maintain high-quality predictions even when images are subjected to substantial distortions. Although ResNet50 performs well under contrast adjustment, DenseWCBnet generally surpasses or matches its performance in other perturbation tests. These findings underscore the enhanced robustness and generalization capabilities of DenseWCBnet, making it a more reliable choice for handling manipulated images in real-world scenarios where images are often subject to various types of noise and distortions.

4.3.7 Extended Experiments for Generative AI Detection

To address the 5th research question from Section 2.5 as WCBnet’s, this study also evaluates the performance of DenseWCBnet over Gen-AI images and compare it with WCBnet’s results.

Deepfake dataset The DenseWCBnet is evaluated the same Gen-AI dataset named DeepfakeArt which applies multiple learning-based models to achieve various manipulation types. As the same experimental setup in Section 3.4.7, the in-paining subset of DeepfakeArt which replaces part of images with fake contents via learning from the background, is applied for the image-level performance evaluation and pixel-level explainability visualization.

Image-level Deepfake detection result The image-level performance of DenseWCBnet is evaluated via three common evaluation metrics, namely, accuracy, F1-score and AUC. DenseWCBnet is compared to the best model mentioned in DeepfakeArt challenge, and WCBnet with the optimal reshaping peak. As shown in the Table 4.6, DenseWCBnet outperforms single-view WCBnet, achieving an additional 3.1% increase in accuracy,

a 3.9% boost in F1-score, and a 1.2% improvement in AUC. Specifically, on the DeepfakeArt in-painting dataset, DenseWCBnet attains an exceptionally high accuracy of 97%, an F1-score of 0.951, and an AUC of 0.993. These results highlight the effectiveness of DenseWCBnet’s dense-scale feature weighting and fusion capability in identifying challenging deepfake images.

4.4 Conclusions

In this chapter, DenseWCBnet, a novel approach designed to address conventional image manipulations, is proposed by modelling compression-based error levels and densely fusing multi-level features. Initially, the Signed-value Error Level (SEL) technique is applied, which preserves relative image differences to expose manipulation artifacts resulting from inconsistent JPEG compression processes between foreground and background regions. Subsequently, SEL maps are fed into a CNN-based backbone to extract multi-level features. Throughout this process, this study employs the Cross Convolutional-blocks Weighting Module (CBWM) and Dense Feature Integration structure which enable the setting of trainable weights for convolutional blocks that play distinct roles in image manipulation tasks, and facilitate the fusion of these weighted features across various scales of feature dimensions. Through comprehensive experiments evaluating image-level performance and visualizing explainability by heatmaps of different convolutional layers, DenseWCBnet demonstrates its adaptiveness in manipulation detection accuracy across multiple image manipulation datasets, as well as its ability to combine global and local information extracted by convolutional blocks. DenseWCBnet proves to be a valuable tool for applications requiring high accuracy and robustness to image perturbations, such as digital forensics and media authentication. Moreover, the extended experiments on DeepfakeArt dataset show that DenseWCBnet is capable of distinguishing deepfake images in a superb accuracy over state-of-art methods.

Although DenseWCBnet shows superior image-level performance and provides explainability for reasons behind image manipulation classification, the generated manipulation heatmaps with relatively unclear boundaries is not accurate enough for manipulation localization tasks. It is required to further refine these coarse manipulation heatmaps in order to precisely locate manipulated regions.

Chapter 5

Weakly-supervised Pixel-wise Image Manipulation Localization Using Multi-resolution Learned Features (WSWCBnet)

The proposed image-level manipulation detection models, namely WCBnet of Chapter 3 and DenseWCBnet of Chapter 4 have been proven as effective fast fact-checking tools and DenseWCBnet has provided manipulation heatmaps to explain the model’s region of interest when classifying manipulated images. However, the effectiveness of the heatmaps generated by DenseWCBnet depends heavily on the accuracy of the Grad-CAM feature maps, which provide only a coarse representation of the model’s attention. To address such limitation and 4th research question proposed in Section 2.5, this chapter extends previous approach (DenseCAM) by proposing a weakly supervised image manipulation localization network, WSWCBnet (Weakly-supervised WCBnet), designed to produce more precise manipulation heatmaps without the need for manually labeled ground-truth masks. This is achieved by combining non-semantic image edges, generated by segmentation networks, with heatmaps from previous WCBnets. This integration helps clarify manipulated region boundaries and eliminates false-positive areas distant from the manipulated regions, thereby enhancing the performance of pixel-level image detection.

5.1 Introduction

With the rapid growth of digital images and the widespread availability of powerful image editing tools, detecting manipulated content has become increasingly critical. Manipulated images are now produced at an alarming rate, posing significant risks to cybersecurity and the integrity of visual information across domains like journalism, legal evidence, and social media [11, 3, 4, 1]. Current deep learning-based manipulation detection methods excel at achieving high image-level classification accuracy [156, 166], yet they often fall short in terms of interpretability and precise localization of manipulated regions [167, 168]. Additionally, the absence of pixel-wise annotations in real-world scenarios limits the effectiveness of existing fully-supervised manipulation localization techniques [169, 170, 171].

To overcome these challenges, several weakly-supervised image manipulation localization methods have been proposed in recent years. The feature maps of image-wise

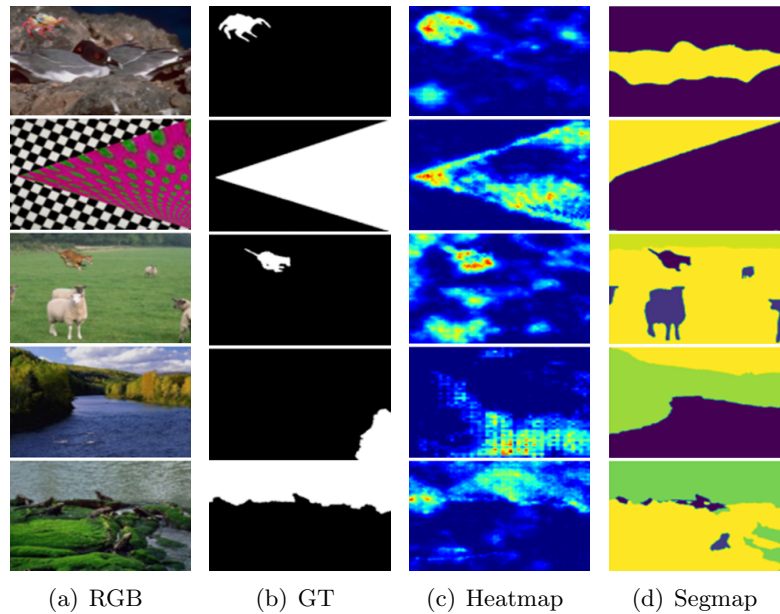


Figure 5.1: Several examples of heatmaps from image-level WCBnet and segmented maps from non-semantic segmentation models, which show the challenges combining them.

manipulation model, extracted by grad-CAM, are directly applied as pixel-wise predictions and compared with semi- or fully-supervised models [172]. Object edges extracted by clustering super-pixels are applied to enhance the grad-CAM result in order to localize the manipulated regions [173], while another method applies a three-source stream (RGB, SRM and Bayar) to generate pseudo pixel-wise labels leveraging the image-wise model for manipulation localization [174]. However, enhancement through image segmentation is more closely aligned with the nature of image manipulation itself, as it involves the insertion of meaningful objects or regions to create misinformation [175]. Segmentation-based approaches [176, 177, 178] can provide pixel-level details but typically require extensive labelled data, which is often unavailable. Consequently, the segmentation models require collaboration with image manipulation detection methods for distinguishing the manipulated class. Several challenging conditions when combining image manipulation heatmaps and segmented maps are shown in the Figure 5.1. In the examples shown in the figure, the non-semantic segmentation maps fail to accurately segment the target manipulated regions. For instance, these segmented maps only include a portion of the manipulated area, ignore small manipulated regions while focusing on larger ones, or treat different regions as the same category. Therefore, in the problem of image manipulation detection where target regions are diverse, the effective combination method is required to keep the heatmap as the primary reference while effectively refining the heatmap using segmentation maps.

This chapter tackles the challenge of localizing image manipulations without relying on pixel-level annotations by proposing a novel method that combines multi-view feature maps from a classification network and fine-grained object regions from pre-trained segmentation models. Our method builds upon our image-level manipulation detection model WCBnet and DenseWCBnet (introduced in Chapter 3 and Chapter 4), applying different Cross Convolutional-blocks Weighting Module (CBWM) structures to produce multi-view feature maps for initial, coarse localization. These feature maps, generated by

Grad-CAMs [17] on the differently fused features, visualize the image-level models’ attention when classifying manipulated regions with multiple resolutions and comprehensive receptive fields. The feature maps, served as coarse localization maps, are further refined by pre-trained non-semantic segmentation models such as DeepLab [179], SAM [180], and PSPNet [181] in order to identify potentially manipulated regions. Ultimately, this study integrates these heatmaps with the segmentation maps using Bayesian inference to achieve more accurate manipulation localization results.

The primary contributions of this work are summarized as follow:

- Multi-view attention map generation by leveraging CBWMs within DenseWCBnet to generate diverse Grad-CAM-based feature maps that capture manipulation clues from multiple receptive fields.
- Annotation-free region refinement by employing pre-trained non-semantic segmentation models to extract candidate regions without the need for pixel-level labels.
- Proposing a principled Bayesian inference method to integrate feature maps and segmentation masks for improved image manipulation localization accuracy.
- Annotation-efficient localization framework which demonstrates the feasibility of high-quality manipulation localization through the combination of classification heatmap and generic segmentation, eliminating the need for pixel-level annotations.

The rest of this chapter introducing the proposed weakly-supervised WSWCBnet is structured as follows: Section 5.2 introduces work that is related to the proposed model, such as class activation maps and non-semantic image segmentation. After that, the several steps of the proposed methodology are described in Section 5.3, followed by experimental results of weakly-supervised manipulation localization performance in Section 5.4 and summarized conclusions in Section 5.4.

5.2 Related Work

The main contribution of the proposed WSWCBnet is the combination between manipulation heatmap generated from image-level manipulation detection model and non-semantic object boundaries from segmentation models. These two core components during such combination that achieves weakly-supervised image manipulation localization, has been individually applied on other researching field.

5.2.1 Gradient-weighted Class Activation Mapping (Grad-CAM)

Gradient-weighted Class Activation Mapping (Grad-CAM) is a bridge that connects image-level Convolutional Neural Networks (CNNs) to pixel-level heatmaps that highlight important regions in an image relevant to CNNs’ prediction. Specifically, Grad-CAM computes the gradients of the predicted image class to the any convolutional layers which reveals the extracted feature regions that react the most during making this decision. By aggregating these gradients, Grad-CAM generates a coarse localization heatmap indicating regions of models’ attention, thus providing explainability of image-level models’ decision-making process for specific classes [17].

Gradients can be computed at any intermediate layer within CNNs, allowing Grad-CAM to have broader applicability compared to other CAM methods constrained by specific network output structures. Building on Grad-CAM, advanced gradient-based

techniques have been developed to further enhance localization accuracy and explainability. Guided Grad-CAM combines Grad-CAM with guided back-propagation, providing more details for heatmap generation by involving the local features from shallow convolutional layers. This approach has been particularly useful in applications requiring pixel-level precision, such as medical imaging [17]. Grad-CAM++ is another extended version of Grad-CAM by incorporating the influence of each pixel’s contribution across multiple classes, enabling more accurate localization for complex scenes with multiple objects [182]. Layer-CAM (Layer-wise Class Activation Mapping) considers spatially varying weights for feature maps, which obtains more detailed spatial location and fine-grained object details for the feature maps [183].

These advancements in the Grad-CAM framework have enhanced the ability to analyze and interpret CNNs, making them valuable tools regarding explainable AI (XAI), medical diagnosis, and image manipulation detection. Although the role of shallow convolutional layers has been considered in several Grad-CAM methods such as Layer-CAM and Guided Grad-CAM, the importance of different dimensional information in the field of image manipulation detection varies and requires advanced weighting and fusion. Moreover, the heatmaps generated by Grad-CAM techniques are normally applied for explainability but need to be further refined for accurate localization purposes.

5.2.2 Non-semantic Image Segmentation

Non-semantic segmentation models are designed to divide images into regions based on perceptual characteristics, such as edges, textures and color gradients, without assigning specific semantic labels to each segment. Unlike traditional semantic segmentation, where regions are labelled by classes, non-semantic segmentation aims to distinguish distinct structures within an image to facilitate further analysis or processing.

There are several state-of-art learning-based segmentation models excelling in capturing structural and regional details without requiring class-specific labels. DeepLab leverages Atrous convolution and Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale visual contents for boundary-focused tasks [179]. Segment Anything Model (SAM), developed by Meta AI, introduces prompt-based segmentation, enabling flexible, generalizable segmentation across diverse image domains [180]. PSPNet incorporates a pyramid pooling module that captures both global and local information from images, enhancing boundary delineation and spatial awareness [181]. These models collectively support non-semantic segmentation tasks which give emphasized object boundaries and differentiated region.

Even with highly precise object boundary extraction, segmentation models alone cannot differentiate between manipulated objects and other suspicious segments, rendering segmented maps ineffective for image manipulation detection. Distinction between manipulated and non-manipulated regions requires a reliable manipulation classification network that focuses on manipulation-related image features, thus assigning meaningful labels to the segmented regions.

5.3 Methodology

5.3.1 Overview

In this section, the study describes the methodology employed to detect image manipulation and enhance the localization of manipulated regions without using pixel-wise ground-truth labels. As shown in the Figure 5.2, our approach consists of four main

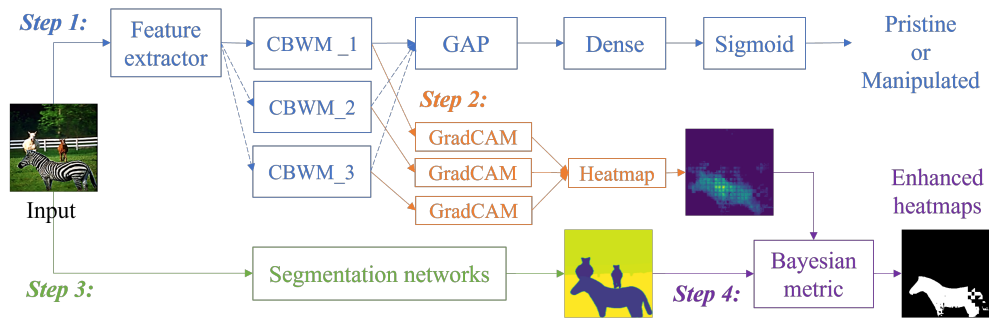


Figure 5.2: The work-flow of proposed weakly-supervised manipulation localization model; Different colors represent four steps of proposed WSWCBnet.

steps: 1) image-wise manipulation classification, 2) manipulation heatmap generation, 3) segmentation map extraction using a pre-trained network, and 4) combination to produce enhanced heatmaps.

5.3.2 Image-wise Manipulation Classification

The first step in our method involves classifying whether an image has been manipulated or pristine. According to our previous chapters (Chapter 3 and 4), WCBnet and DenseWCBnet which have shown the prior image-level performance on multiple datasets are applied as the backbone. The workflow of these two models is first applying a CNN-based feature extractor to extract the hierarchical features from input images which contain global and local information of manipulation traces. Since both types of information are equally important in image manipulation detection, single-view or dense-view Cross Convolutional-blocks Weighting Module (CBWM) are applied to assign adaptive weights for these high-level and low-level features and fuse them. Specifically, the input image $x \in \mathbb{R}^{384 \times 384 \times 3}$ is first processed by a CNN-based feature extractor (ResNet50) to produce multi-scale feature maps $\{y_i\}_{i=1}^5$, where each y_i represents features extracted by 5 convolutional blocks at different scales. $y_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ that represents the feature map at scale i is defined as:

$$y_i = \text{CNN}(x). \quad (5.1)$$

These feature maps are then fed into the CBWM structure, which rescales the features to a consistent dimensionality $\mathbb{R}^{H \times W \times C}$ and applies self-attention mechanisms to model the relationships between different convolutional blocks. This process assigns trainable weights according to the inner relationships between features, enhancing the model's ability to capture and utilize complex feature dependencies. For single-view CBWM of WCBnet, the features are weighted and concatenated as:

$$Y = \text{CBWM}(y_1, y_2, \dots, y_n), \quad (5.2)$$

and both single-view CBWM and dense-view CBWM are considered and compared for the weighted feature fusion (the details of these two structures are introduced in the Section 3.3.4 and Section 4.2.6). As for dense-view CBWM of DenseWCBnet, it fuses various levels of features extracted from different convolutional blocks by assigning different weights to those block outputs and computing their channel-wise co-relationship. Considering the multiple conceptional fields of convolutional blocks and the effect of up-

or down-sampling operations in image manipulation detection, this model densely involves all conceptional fields by integrating those features at each level of view. These densely weighted multi-level features are concatenated and pooled for binary classification pristine-or-manipulated tasks.

The processed features Y are then passed through a global average pooling (GAP) layer to obtain a fixed-size feature vector $v \in \mathbb{R}^D$. This vector is subsequently input to a dense layer, producing single logit $z \in \mathbb{R}$. Finally, a sigmoid activation function $\sigma(z)$ is applied to the logits to yield binary classification probabilities $p \in [0, 1]$ belonging the manipulated class. The first stage of image manipulation detection is the fundamental base of the following steps, which determines the quality of manipulation heatmaps. Consequently the selection of WCBnet and DenseWCBnet, achieving superior image-level performance, is necessary for the whole weakly-supervised image manipulation localization workflow.

5.3.3 Feature Map Generation

Upon classifying the image as manipulated, Gradient-weighted Class Activation Mapping (Grad-CAM [17]) and several advanced versions are applied to identify and visualize the model's region of interest that responds actively when classifying manipulated images.

Grad-CAM Grad-CAM applied in this chapter generates a heatmap highlighting the most suspicious regions in an input image for manipulation class. The transferring from image-level prediction to pixel-level localization is achieved by computing the gradients of the prediction class score with respect to feature maps from a chosen convolutional layer, and using these gradients as weights for each feature map. The feature map generation between weighted features Y of image-level WCBnet (or DenseWCBnet) and the prediction score for manipulation class p is computed in the following steps:

1) Computing gradients of the prediction score to features: The gradients $\frac{\partial p}{\partial Y}$ are computed with respect to the weighted features Y in the single-view CBWM layer. These gradients indicate how much the selected features influence the prediction score p .

2) Global average pooling to get importance weights: After calculating the gradients, this algorithm performs a global average pooling operation across the spatial dimensions to obtain the average values for each feature channel, as the weight for the feature Y . The weight α_k for feature at channel k is defined as:

$$\alpha_k = \frac{1}{N} \sum_i \sum_j \frac{\partial p}{\partial Y_k^{ij}}, \quad (5.3)$$

where N is the total number of pixels in the feature, while i and j are the spatial axis.

3) Weighting the features by the importance: The feature Y is then weighted by its respective importance weight α , and combined to form the class activation map.

$$H = \sum_k \alpha_k \cdot Y_k^{ij}. \quad (5.4)$$

4) Normalizing the feature map: Moreover only the positive values of the results are reserved, and normalized between 0 and 1 for clearer visualization:

$$H = \frac{\max(0, H)}{\max(H)}. \quad (5.5)$$

This process is based on Grad-CAM, which could generate a manipulation heatmap from any convolutional layer that highlights the most relevant regions in the input image for manipulation classification. It is the basic method used to obtain heatmap of WCBnet which has only one weighted feature output.

LayerCAM Layer-CAM method is an extension of the Grad-CAM that improves the accuracy of localization by directly using the gradients of the prediction class to each spatial location in the features, rather than globally averaging them (as the second step in Grad-CAM). The pixel-level weight α^{ij} for the feature Y changes to:

$$\alpha_k^{ij} = \frac{\partial p}{\partial Y_k^{ij}}, \quad (5.6)$$

where the shape of α_k^{ij} is the same as feature's, instead of single-value global weight for the whole feature. Then the feature weighting is computed by the dot multiplication between two matrixes with the same shape, followed by aggregation along channel-wise:

$$H = \sum_k \alpha_k^{ij} \cdot Y_k^{ij}. \quad (5.7)$$

Layer-CAM computes the pixel-wise activation at the spatial location for the features Y , which are computed as gradient-based weights, improving the accuracy of identifying regions of interest for the image manipulation detection.

Grad-CAM++ Grad-CAM++ is another advanced version of Grad-CAM, which uses a more fine-grained weighting approach including more complicated gradient computation. The biggest difference is that Grad-CAM++ involves second-order and third order gradients of prediction score with respective to features, to measure the more sensitive level of the score to the change of features. The first and second step in Grad-CAM are replaced by the following equation:

$$\alpha^{ij} = \frac{\frac{\partial^2 p}{\partial (Y^{ij})^2}}{2 \frac{\partial^2 p}{\partial (Y^{ij})^2} + \sum_{a,b} Y^{ab} \frac{\partial^3 p}{\partial (Y^{ij})^3}}, \quad (5.8)$$

where α^{ij} is the pixel-level weights at each location in the features (similar to Layer-CAM), rather than the single weight per feature. $\sum_{a,b} Y^{ab} \frac{\partial^3 p}{\partial (Y^{ij})^3}$ aggregates the contributions across all spatial locations in the feature, weighted by the third-order partial derivatives. Then Grad-CAM++ follows the same combination step between features Y and importance weights α^{ij} to generate the manipulation heatmap H .

Grad-CAM++ is more sensitive to multiple occurrences or small objects since it includes pixel-level weights, which often result in sharper and more localized heatmaps.

Proposed DenseCAM In order to involve the DenseWCBnet's multi-view attention at different feature scales when classifying manipulated images, this study proposes a DenseCAM that combines the feature maps from five convolutional blocks using their convolutional block-wise activation values from final dense layer. As illustrated in Section 4.2, Dense Feature Integration of DenseWCBnet combines features weighted by different CBWMS IN three steps which are pooling across geometric dimensions, concatenating features and connecting through a dense layer. Each neuron within this dense layer exhibits varied activation levels during image manipulation detection, reflecting the degree

of each feature channel’s contribution to the detection process. To construct DenseCAM, this study utilizes the activation values of neurons within this dense layer to perform weighted combinations of corresponding Y_i feature maps. This weighting approach leverages the unique activation levels as indicators of each feature map’s importance, enabling the generation of multi-view heatmaps that effectively localize and highlight manipulated regions within the detected images. This process involves transforming the activation vector, computing weights, and combining weighted feature maps as follows:

1) Input setup: Grad-CAMs are first computed as previously mentioned to generate a set of feature maps $F = \{F_1, F_2, \dots, F_5\}$, generated from the weighted features $Y = \{Y_1, Y_2, \dots, Y_5\}$ of DenseWCBnet. The dense layer activation vector A with a length of $5C$ is also extracted, where C corresponds to the dimensionality of each feature map.

2) Splitting the activation vector: To assign specific weights to each feature map, the activation vector A is divided into five sub-vectors $A_1, A_2, \dots, A_5$, each of length C . Each sub-vector A_i corresponds directly to one of the feature maps F_i , allowing for individual weighting of each map.

3) Averaging for weight calculation: For each sub-vector A_i , this study computes its average value, denoted as α_i . These averages represent the overall contribution or importance of each feature map F_i to the manipulated region detection task. By using these averages as weights, the influence of each map is captured and applied in the heatmap generation.

4) Applying weights and accumulating the heatmap: For each feature map F_i , this study applies the corresponding weight α_i to create a weighted feature map $W_i = \alpha_i \cdot F_i$. Each weighted feature map W_i is then added to the heatmap H , effectively layering the weighted contributions from all feature maps, the process is defined as:

$$H = \sum_i \alpha_i \cdot F_i \quad (5.9)$$

This module effectively weights and fuses feature maps F_i across multiple scales, enhancing the feature representation of manipulation-related traces. Moreover, some other map combination methods are included, for example, the mean of these fused feature maps is computed:

$$H = \frac{1}{n} \sum_i N(F_i), \quad (5.10)$$

where $N(\cdot)$ is the normalization function to prevent pixel values from becoming too large or too small during accumulation or summation, ensuring they remain within the range of 0 to 1. Similarly, multi-scale feature maps could be aggregated by geometric mean as:

$$H = \left(\prod_{i \in \{1,2,3,4,5\}} N(F_i) \right)^{\frac{1}{n}}, \quad (5.11)$$

where $n = 5$ is the number of scales used for the geometric mean computation. These three approaches effectively combine the detailed feature maps with global context, enhancing the detection and visualization of manipulated regions.

5.3.4 Generating Segmentation Maps Using Pre-trained Models

To further enhance the precision of the heatmaps H generated by the image-wise manipulation classification model, this study applies several pre-trained segmentation models

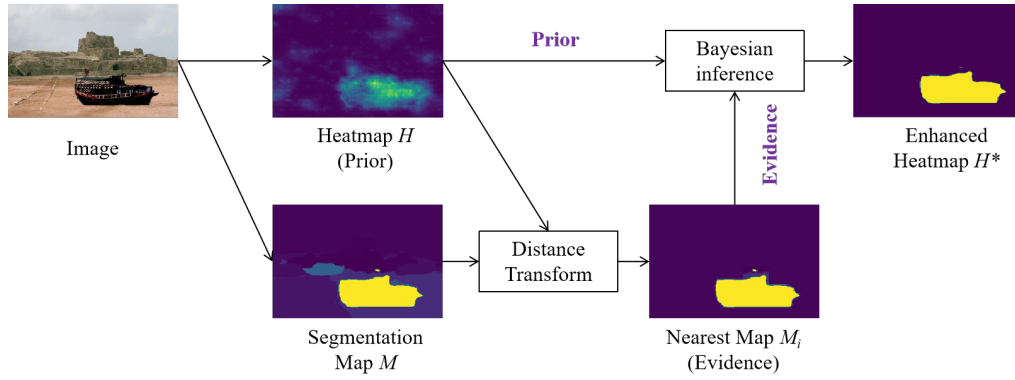


Figure 5.3: The workflow of applying Bayesian inference to enhance heatmap H generated from DenseWCBnet.

to divide the image into distinct regions that potentially contain manipulated areas. The learning-based instance segmentation models employed, namely DeepLab, the Segmentation Anything Model (SAM) and PSPnet, have been pre-trained on large-scale datasets to detect and segment normal objects. These models offer pixel-level accuracy in identifying potential manipulation regions with higher precision. The resulting segmentation maps M , which consist of various masks M_i corresponding to each detected object i , are defined as follows:

$$M = \bigcup_{i=1}^n M_i. \quad (5.12)$$

However, the presence of numerous detectable objects within an image makes it difficult for segmentation networks to distinguish manipulated regions from the array of detected objects.

5.3.5 Combining Manipulation Heatmap and Segmentation Maps

In the final step, the manipulation heatmaps H , which indicate coarse regions of interest for image-wise manipulation classification, are refined using segmentation maps M that delineate fine-grained object regions without semantic labels. These segmentation maps function as structure-aware masks providing detailed spatial cues to improve localization accuracy.

As shown in Figure 5.3, a similarity score between the multi-scale feature map H and each binary object mask $M_i \in M$ is computed using a distance-based weighting strategy. Specifically, this study applies a distance transform $D(M_i)$ to quantify the distance of each pixel to the nearest boundary of M_i . The resulting similarity function, denoted as $\text{Sim}(M_i, H)$, is defined as:

$$\text{Sim}(M_i, H) = \frac{1}{\sum M_i} \sum (D(M_i) \cdot H), \quad (5.13)$$

where normalization by the region size $\sum M_i$ ensures fair contribution from regions of varying scales. This metric captures the degree of spatial alignment between the coarse manipulation response and potential object boundaries.

The segmentation mask M_{i^*} with the highest similarity score is selected, and the heatmap is subsequently refined as:

$$H^* = B(H, M_{i^*}), \quad \text{where } i^* = \arg \max_i \text{Sim}(M_i, H). \quad (5.14)$$

The enhancement function $B(\cdot)$ is guided by a Bayesian-inspired formulation, in which the coarse heatmap H serves as a prior indicating semantic confidence in manipulation, while the selected segmentation mask M_{i^*} provides structural evidence. This concept is motivated by Bayes' rule:

$$P(H \mid M_{i^*}) = \frac{P(M_{i^*} \mid H) \cdot P(H)}{P(M_{i^*})}. \quad (5.15)$$

Although no explicit probabilistic modeling is performed, this framework informs the design of $E(H, M_{i^*})$, enabling integration of high-level classification signals with low-level spatial structure. This strategy facilitates coarse-to-fine manipulation localization by combining activation maps from manipulation classification networks with object-level shape information derived from pre-trained segmentation models, without relying on semantic or pixel-wise annotations.

The methodology presented combines image manipulation classification, weakly-supervised localization, and segmentation techniques to achieve the detection and localization of manipulated regions in the absence of pixel-wise labels. The integration of multi-view activation maps and pre-trained segmentation networks via Bayesian interference, combining the semantic information from the manipulation classification network and the fine-grained regional information from segmentation networks into enhanced manipulation localization results.

5.4 Experimental Results and Discussions

In this section, this study presents a series of experiments designed to evaluate the effectiveness of our proposed multi-view heatmap approach, assess different segmentation networks, and examine the performance of enhanced heatmaps generated by combining heatmaps with segmentation masks. Before that, the experimental setup of evaluating our proposed model on the task of generating enhanced heatmap of image manipulation at the absence pixel-wise labels is illustrated.

5.4.1 Dataset

The experiments are conducted mainly on CASIA2.0 image manipulation dataset [25], and select approximately 1800 splicing, 1800 copy-move and 1800 authentic images to train the image-wise feature extractor (WCBnet, DenseWCBnet) and generate multi-view heatmaps. Each image is resized to 384x384 pixels for consistency. All images are pre-processed using signed-value error levels [184] following the experimental setup of WCBnet, to extract JPEG compression-based artifacts of manipulation traces.

5.4.2 Experimental Setup

The experiments were conducted on a server with NVIDIA GeForce RTX 3090 Ti, 12th Gen Intel (R) Core (TM) i9-12900K 3.20 GHz processor and 32.0 GB RAM, and the model is constructed based on Python 3.7 and TensorFlow 2.7. For the image-wise classification step, our model is based on the ResNet-50 architecture, with additional Cross Convolutional-blocks Weighting module for feature weighting and fusion on the image manipulation tasks. The image-wise model was trained using the SGD optimizer with an initial learning rate of 0.001, a batch size of 12, and for 200 epochs. For the segmentation models, several state-of-art image segmentation models, namely DeepLab2, Segmentation

Anything Model (SAM) and PSPnet, that have been trained on object detection datasets are used to generate pixel-wise masks for each region. To claim, only the non-semantic segmentation masks from DeepLab were used in the study although it could be a semantic segmentation model. The image-wise performance is evaluated using accuracy and F1-score to account for the class imbalance in the dataset, while the pixel-wise performance is evaluated using AUC and F1-score with fixed threshold. This research conducts experiments to visually and statistically prove the effectiveness of multi-view heatmaps and the combination with segmentation masks.

5.4.3 Evaluation of Heatmaps Generated by DenseCAM

Intuitive Comparison of single- and dense- view heatmaps

Initially, the manipulation heatmaps of different image-level manipulation models (including the backbone network, WCBnet with different CBWM scale) were computed, by applying the same Grad-CAM. Specifically, for the image-wise manipulation detection model WCBnet, it employed single-view CBWM structure to weight and fuse convolutional blocks within a fixed receptive field, achieving outstanding image-level manipulation classification accuracy across multiple datasets. Building upon this, WSWCBnet investigates the impact of multiple CBWM structures regarding massive receptive fields and the resulting heatmaps on manipulation localization.

This study trains several structures of WCBnet, namely, WCBnet₂, WCBnet₃, and WCBnet₄, each incorporating different CBWM configurations. These models achieve commendable F1-scores of 0.912, 0.933, and 0.935, respectively, for image-level manipulation classification tasks on CASIA2.0 dataset. Additionally, by computing class heatmaps between the weighted feature layers of these WCBnet_{*i*} models and their output layers, feature maps are extracted that highlight the most active pixels during classification. As illustrated in Figure 5.4, the backbone ResNet50 exhibits limited focus on the manipulated regions, despite correctly classifying the image type. In contrast, the feature maps from WCBnet clearly delineate the boundaries between manipulated and background regions. However, the feature maps produced by shallow CBWM exhibit excessive highlighting of background regions, while those from deeper CBWM lack detailed information nearby region edges. The geometric mean of these feature maps provides a more accurate representation of the manipulation regions, which combines the detailed information of the former and the regional information of the latter. DenseCAM further improves the accuracy in localizing the manipulated regions, as it combines feature maps at different scales with weights.

Intuitive Comparison Between Different Versions of Grad-CAMs

To further improve the accuracy of manipulation heatmaps, the Grad-CAM of weighted feature from WCBnet is compared with some other advanced versions, namely Grad-CAM++ and Layer-CAM. The manipulation heatmaps of different scales from five weighted feature maps WCBnet₁ to WCBnet₅, generated by Grad-CAM, Layer-CAM and Grad-CAM++ are compared. As shown in the two examples in the Figure 5.5 and Figure 5.6, three versions of Grad-CAMs perform approximately the same at different heatmap scales. In heatmaps of five sizes, Layer-CAM and Grad-CAM++ do not have obvious improvements on localizing the manipulated regions. Considering the computational cost, Grad-CAM with easiest computation and approximately accurate heatmap is selected as the basic for the following experiments.

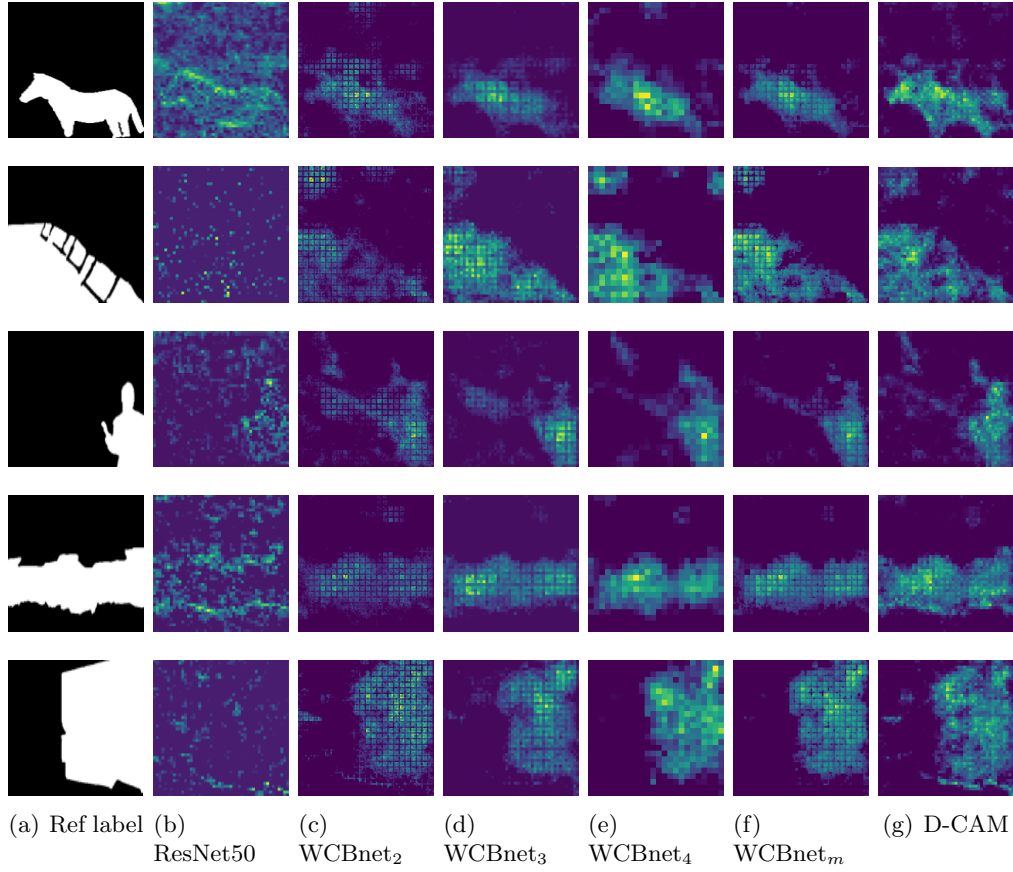


Figure 5.4: The features maps of the backbone *ResNet50*, *WCBnets* and *DenseWCBnet*; The *WCBnet_i* means the *CBWM* with block *i*'s receptive field, while *WCBnet_m* is their geometric mean; *D-CAM* is the proposed *DenseCAM*; *Ref label* is the pixel-wise labels just for reference.

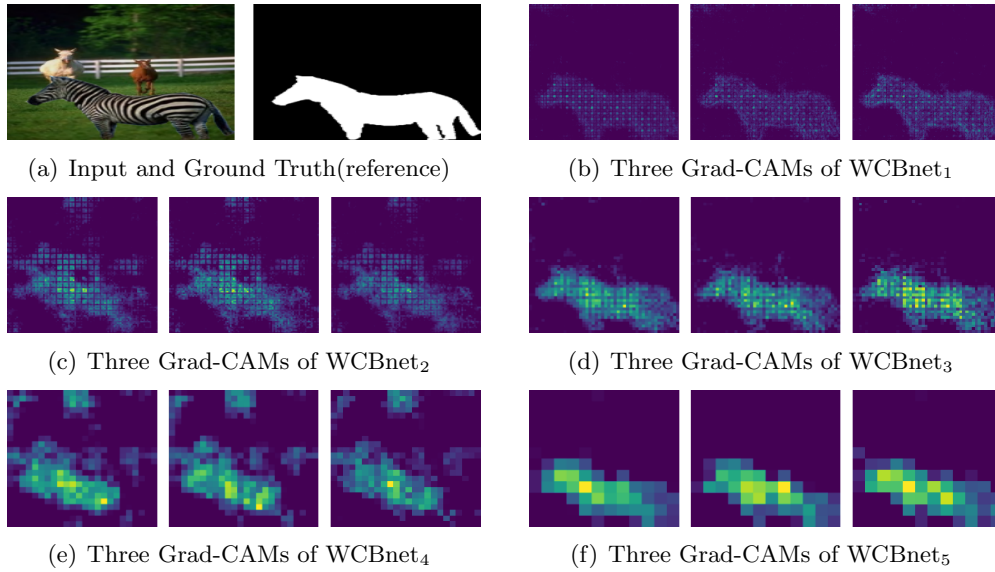


Figure 5.5: Comparison between *Grad-CAM*, *Layer-CAM*, *Grad-CAM++* on manipulated images, regarding different views of *WCBnet*.

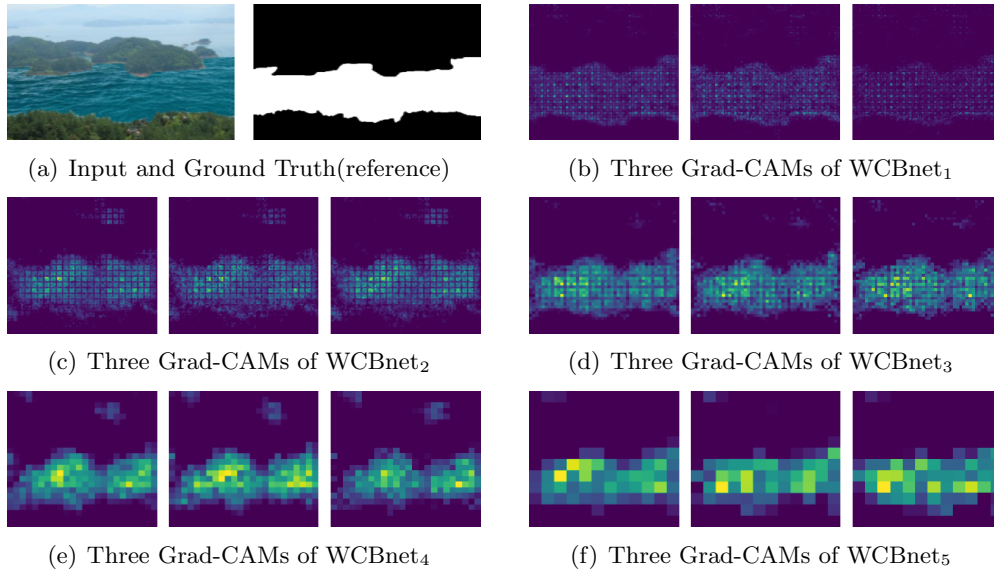


Figure 5.6: Another example of comparison between Grad-CAM, Layer-CAM, Grad-CAM++ on manipulated images, regarding different views of WCBnet.

5.4.4 Intuitive Comparison of Segmentation Networks on Manipulated Images

As previously discussed, three state-of-the-art image segmentation models with pre-trained weights (DeepLab, SAM, and PSPNet) were utilized to detect and localize potentially manipulated regions within the images. Several samples of manipulated images along with their corresponding segmentation maps produced by these models are visualized for comparison. As shown in the Figure 5.7, DeepLab is effective for segmenting large areas and images with a limited number of categories. In contrast, PSPNet and SAM excel at segmenting a broader range of smaller regions within the image. However, DeepLab may struggle to distinguish between manipulated and adjacent regions, while PSPNet and SAM may detect overly small areas, potentially missing parts of the manipulated object. These segmentation maps require to be combined with class activation maps to distinguish the manipulated regions.

5.4.5 Evaluating the Combination of Multi-view Heatmaps and Segmented Masks

This study integrates the multi-view feature map H with the single-class mask M_{i^*} derived from segmentation maps using Bayesian interference. Several samples of resulting enhanced heatmaps H are presented in Figure 5.8 for intuitive comparison, and more heatmaps are illustrated in Appendix A. Visually, the segmented regions produced by DeepLab more accurately approximate the size of the manipulations in the testing images of the dataset, making the enhanced heatmap WCBnet more aligned with the reference label compared to the other heatmaps. In contrast, the finer segmented regions from PSPNet and SAM lead to excessive detail, causing parts of the manipulated region to be excluded by the heatmap. Notably, while ResNet50 achieves high image-wise classification accuracy, its pixel-wise localization results misclassify manipulated regions or produce inaccurate maps. For the small portion of such ineffective heatmap enhance-



Figure 5.7: Comparison of Segmentation Networks on Manipulated Images, generated by three state-of-art pre-trained image segmentation methods.

ment results, there are three scenarios that negatively affect the combination that: 1) image manipulation heatmaps generated from image-level classification models highlight wrong regions, 2) segmented maps from non-semantic segmentation models fail to segment complete boundaries of target manipulated regions for the following combination, 3) the applied combination metric, Bayesian interference, becomes invalid when dealing with complicated visual contents. More results of ineffective combination between heatmaps and segmaps are shown in Appendix A.2.

Table 5.1: The pixel-wise performance comparison between the backbone network, proposed method and a fully-supervised ManTraNet; In which the weakly supervised method combines image-wise models and segmentation models.

	ResNet(backbone)			WCBnet			DenseCAM	ManTraNet (fully-supervised)
	DeepLab	SAM	PSPnet	DeepLab	SAM	PSPnet	DeepLab	
AUC	0.535	0.545	0.524	0.704	0.608	0.703	0.656	0.653
F1	0.296	0.257	0.259	0.682	0.310	0.365	0.694	0.238

To further analyse pixel-wise manipulation localization statistically, the enhanced heatmaps are evaluated using several pixel-wise metrics. The AUC and F1-score are computed to measure the similarity between the generated manipulation localization maps and the ground-truth labels. For this evaluation, 40 images were manually selected that

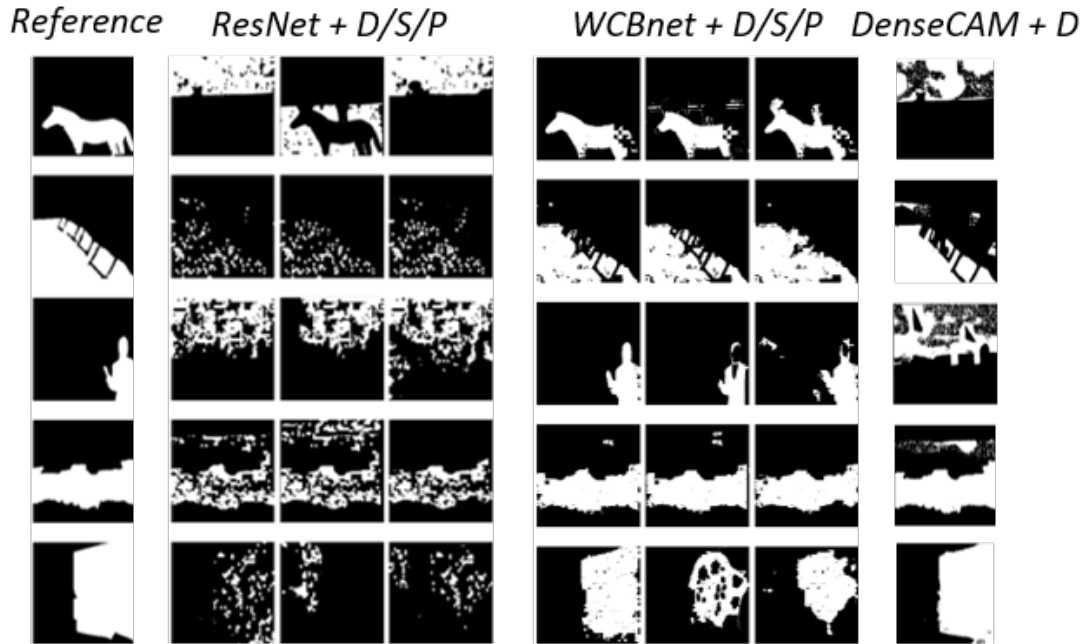


Figure 5.8: Refined manipulation localization results using combinations of different classification models and segmentation models. *ResNet*, *WCBnet* and *DenseCAM* denote different classification models for generating Grad-CAM-based heatmaps. *D*, *S*, and *P* represent segmentation masks from DeepLab, SAM (Segment Anything Model), and PSPNet, respectively.

are correctly identified as manipulated and for which the heatmaps approximately highlight the target region. The performance of weakly-supervised WCBnet and DenseWCBnet with different pre-trained segmentation models is also compared to a fully-supervised model name ManTraNet [185].

Table 5.1 compares the pixel-level localization performance of several **weakly-supervised** methods such as ResNet, WCBnet, and DenseCAM, each of which is combined with different segmentation models (DeepLab, SAM, and PSPNet). Among them, WCBnet shows a clear advantage over the ResNet backbone across all segmentation models. For example, with DeepLab, WCBnet improves AUC from 0.535 to 0.704 (a gain of 0.169) and F1-score from 0.296 to 0.682 (a gain of 0.386). Similarly, SAM yields a gain of 0.063 in AUC and 0.053 in F1-score. The best overall performance is achieved by DenseCAM which is built upon DenseWCBnet and Grad-CAM, with an AUC of 0.656 and F1-score of 0.694.

Finally, these weakly-supervised approaches are compared with a fully-supervised model, ManTraNet [185]. Notably, both WCBnet and DenseCAM outperform ManTraNet in terms of F1-score (0.682 and 0.694 against 0.238), while achieving comparable or superior AUC scores (0.704 and 0.656 against 0.653). These results demonstrate the effectiveness of the proposed pipeline, showing that pixel-level annotations are not strictly necessary for high-quality manipulation localization when combining multi-view classification feature maps with pre-trained segmentation priors.

5.5 Conclusions

This chapter has introduced a novel weakly-supervised framework for localizing image manipulations without relying on pixel-level annotations. The proposed approach lever-

ages heatmaps generated from image-level manipulation detection models (e.g., WCBnet and DenseCAM) and refines them using pre-trained segmentation models (e.g., DeepLab, SAM, and PSPNet). First, multi-view class activation maps are generated, by incorporating different feature fusion structures named CBWM across various receptive fields and computing weighted mean of these multi-view activation to obtain heatmaps. These heatmaps are subsequently integrated with the segmentation maps using Bayesian inference to yield enhanced manipulation localization maps. This framework presents an effective solution to the long-standing challenge of obtaining precise localization without pixel-level annotations.

The evaluation demonstrates both qualitative and quantitative effectiveness of the proposed method. Qualitatively, it can be observed that heatmaps generated by WCBnet and DenseCAM exhibit improved spatial focus on manipulated regions compared to the ResNet backbone, achieving a better balance between precision and recall. Furthermore, the incorporation of segmentation masks significantly improves localization accuracy with clearer boundaries, and the Bayesian inference which relies on prior (multi-view heatmap) helps avoid errors from inaccurate segmentation by falling back on these reliable prior.

Quantitatively, the proposed framework achieves significant improvements in both AUC and F1-score metrics. For example, when combined with DeepLab, WCBnet achieves an AUC of 0.704 and an F1-score of 0.682, marking an absolute increase of 16.9% and 38.6% respectively over the ResNet backbone. DenseCAM performs similarly well, reaching an F1-score of 0.694. Compared to the fully-supervised ManTraNet, which achieves an F1-score of only 0.238, the proposed weakly-supervised methods maintain a substantial advantage in precision-recall performance. The significance of these results lies in demonstrating the feasibility of localizing image manipulations without relying on pixel-level labels, which is a departure from existing manipulation localization models.

While the method performs well in most cases, a few instances still show suboptimal outcomes due to weak heatmap responses, segmentation inaccuracies, or challenging scene compositions. Examples of such failure cases are discussed in Appendix A. Future work will explore more robust segmentation-guided fusion strategies, investigate performance under more complex manipulations, and extend the system to multi-object and small-scale forgery detection. These efforts aim to further enhance the applicability and generalizability of weakly-supervised manipulation localization in real-world settings.

While this chapter demonstrates numerous successful examples of combining heatmaps and segmentation maps for manipulation localization, a minority of cases still exhibit suboptimal results due to issues such as invalid heatmap generation, region segmentation errors, or unsuitable combinations. Additional examples illustrating these challenges can be found in the Appendix A. Future work in the domain of explainable manipulation detection will focus on enhancing the model’s performance in more challenging scenarios, including detecting small-scale manipulations, handling complex backgrounds, and improving multi-target detection capabilities. These advancements aim to further refine the robustness and applicability of the proposed approaches against various manipulation scenarios in real-world contexts

Chapter 6

Conclusions

6.1 Research Summary and Objectives

In conclusion, this PhD research has systematically explored the application of deep learning-based models to address the complex challenges in image manipulation detection. The work was motivated by the increasing fineness and accessibility of image manipulation techniques, including traditional splicing/copy-move and recent generative AI-based modifications. To tackle this, five key research questions were formulated as outlined in Section 2.5, each addressing a specific aspect of detection performance, explainability, and supervision constraints.

To address these challenges, three learning-based models were proposed:

- WCBnet which enhances image-level manipulation detection by modelling error levels with hierarchical feature weighting and fusing(Chapter 3);
- DenseWCBnet which progresses WCBnet by improving multi-level feature integration and achieving explainability during classification process(Chapter 4);
- WSWCBnet which enables precise manipulation localization without pixel-level labels, by refining attention maps from proposed classification models(Chapter 5).

These models collectively aimed to deliver fast image-level fact-checking, improve interpretability, and enable weakly supervised localization capabilities.

6.2 Addressing the Five Research Questions: Contributions and Insights

The proposed models were designed with the specific aim of addressing the distinct challenges posed by each research question in Section 2.5. In the following, each research question is revisited, detailing the corresponding approach and key findings.

6.2.1 Key Contributions Addressing Five Research Questions

1. **Research Question 1:** How can this study identify a type of hand-crafted features that are widely present across all manipulation traces, thereby enhancing the adaptiveness of detection methods for diverse image manipulation tasks?

Proposed Method:

To address this question, this research focused on identifying and leveraging JPEG compression-based error levels that generally exist in any image compression and re-saving process during manipulation operations. These error-level features exposed disparity between manipulated and authentic regions, consequently served as a robust foundation for improving the applicability of detection methods across various manipulation scenarios. By designing Signed-value Error Levels (SEL) retaining the sign information of the pixel values in the image subtraction result and applying as the inputs to the proposed WCBnet, DenseWCBnet and WSWCBnet, it has achieved significant improvement over conventional RGB or other error levels inputs.

The effectiveness of proposed SEL has been evaluated using multiple metrics in the Section 3.4.3 and Section 3.4.5 regarding WCBnet’s performance, and Section 4.3.3 regarding DenseWCBnet. In Table 3.4, this study compared the custom Distance Score of RGB, normal error level (EL), and the proposed SEL with the provided pixel-level ground truth. The SEL value for manipulated images was 0.941, higher than EL’s 0.906 and RGB’s 0.530 which shows better precision in manipulated regions. After which in Table 4.2, this research compared the Bhattacharyya distance between manipulated foreground and pristine background ($Distance_S$) and pristine foreground and pristine background ($Distance_A$). SEL showed the largest differences of $Distance_S - Distance_A$ with values of 0.285, indicating that the manipulated regions and pristine regions in SEL are most distinct and easiest to differentiate compared to RGB and EL. Furthermore, Tables 4.3 and 4.4 demonstrate the superiority of SEL as an input to learning-based networks over other input types, such as RGB, normal EL, Wavelet-based EL, and learning-based Comprint. SEL improved the accuracy, F1-score, and AUC of ResNet, WCBnet, and DenseWCBnet to their highest values. For example, in terms of F1, SEL outperformed the second-best input method by 3.2%, 0.8%, and 1.2% across the three models. These experimental evaluations collectively demonstrate the consistent advantages of using SEL across multiple models and datasets. SEL significantly improves manipulation region precision compared to traditional RGB and EL inputs, showing advancement in applying SEL which highlights the importance of such universal feature representations in enhancing the scope and reliability of manipulation detection methods.

2. **Research Question 2:** How to make full use of both high- and low- level information from hierarchical features that play different roles in diverse image manipulation detection tasks?

Proposed Method:

In response to this research question about modelling multi-level information from SEL manipulation traces, this study has proposed Cross Convolutional-block Weighting Module (CBWM) which sets weights for multi-level features extracted by all convolutional blocks of the backbone CNN model. The single-view CBWM of WCBnet is introduced in Section 3.3.4, which assigns weights for five convolutional blocks according to their interrelationship by unifying the shape of their multi-level hierarchical features and computing channel-wise self-attention module. Moreover, the dense-view CBWM of DenseWCBnet, as illustrated in Section 3.3.4, advancing the process of unifying features in single-view CBWM by involving more target reshaping sizes to weight and fuse features at dense receptive fields of all convolutional blocks. The effectiveness of these CBWM structures on utilizing high- and low-level information and modelling SEL results for image manipulation detection has

been evaluated in Section 3.4.5 and Section 4.3.6 individually. The evaluation of WCBnet shows its clear benefit in modelling hierarchical features for manipulation detection. Specifically, Table 3.9 and Table 3.10 show the image-level performance of WCBnet with or without CBWM (referred as RefNet-1) that single-view CBWM improves the backbone’s performance in 4 out of 6 image manipulation datasets regarding accuracy, F1-score and AUC while achieves second-best performance in the rest results. The biggest improvement appears in the CMFDdb copy-move detection, which increases by about 31% accuracy, 0.37 F1-score and 0.24 AUC. As for the experimental results of the dense-view DenseWCBnet, as shown in Tables 4.1, 4.5, and 4.6, it outperforms the single-view WCBnet by improving the majority of image-level F1-scores across most commonly-used manipulation datasets, under various perturbations introduced during typical image transmission processes, and even in the presence of generative image manipulations. Furthermore, Dense-view CBWM enables DenseWCBnet better modelling different error levels compared to the models with single-view CBWM or without CBWM. It achieves F1-scores of 0.893, 0.859 and 0.943 regarding WEL, Comprint and SEL respectively, which improves the second-best data of single-view CBWM by around 0.01, 0.025 and 0.008.

WCBnet and DenseWCBnet were proposed to address Research Question 1 by leveraging SEL-based manipulation traces, and Research Question 2 by introducing a dense-view Cross Block Weighting Module (CBWM) to effectively integrate hierarchical features. Having addressed these two questions and achieved superb image-level manipulation detection performance across diverse scenarios, the focus then shifted to enhancing the explainability of such image-level decisions, aiming to reveal how these black-box methods distinguish manipulated regions from pristine content.

3. **Research Question 3:** How to explain the reasons behind decisions made by learning-based manipulation detection method that operates like black box?

Proposed Method:

To address this research question, this study bridged the gap between image-level manipulation detection and pixel-level manipulation localization by applying Grad-CAMs at different layers of the proposed network. By further developing a novel explainability method named DenseCAM (Section 5.3.3), which leverages the dense receptive fields in the DenseWCBnet architecture, it generated superior manipulation heatmaps that balance precision and recall regarding manipulated regions. As visualized in Section 4.3.4, the process of multi-dimensional feature weighting and fusion in single-view CBWM highlights the complementary roles of shallow and deep backbone features: shallow layers capture fine-grained details from error levels but lack global context, while deep layers excel at distinguishing manipulated and pristine regions based on abstract information but miss finer details. Examples in Figure 4.7 demonstrate how single-view CBWM effectively weights and fuses these diverse features, elucidating how WCBnet identifies manipulated regions by focusing on the target objects. Additionally, Figure 4.8 illustrates the enhanced feature fusion capabilities of DenseWCBnet. This model performs feature weighting across receptive fields of five convolutional blocks and consolidates these initial heatmaps using activation values from dense layers. By visualizing the heatmap generation process within DenseCAM, it could be explained about how DenseWCBnet’s broader receptive fields enable it to better localize manipulated regions with higher precision. This enhanced focus on manipulated areas translates into improved performance of

DenseWCBnet in image-level manipulation classification compared to WCBnet.

These findings demonstrate that the proposed networks with layer-wise Grad-CAM visualizations and the design of single-view and dense-view heatmaps, successfully enhance model’s explainability, thereby addressing Research Question 3. The use of multi-level feature representations allows the networks not only to provide accurate image-level decisions but also to localize manipulation evidence, enabling a transition from image-level classification to weakly-supervised pixel-level localization. This justifies the reliability of the generated heatmaps, as they are based on semantically meaningful manipulative traces rather than spurious correlations, making them suitable as proxies for ground truth in the absence of pixel-level annotations. However, these manipulation heatmaps are insufficient for precise pixel-level manipulation localization. Therefore a novel weakly supervised network, WSWCBnet, was proposed to address the next research question.

4. **Research Question 4:** How to achieve pixel-level image manipulation localization without pixel-level manual annotations?

Proposed Method:

To address the question of localization without pixel-level labels and refine the manipulation heatmaps generated from previous models, WSWCBnet, a weakly-supervised model that leverages DenseWCBnet’s focus heatmap with object segmentation maps from non-semantic networks was proposed. The coarse localization is achieved by the proposed DenseCAM that integrates multi-scale feature maps from DenseWCBnet with weights obtained from the final dense layer, which is then refined by instance segmentation maps. From the visualization results in Figures 5.4, 5.5, and 5.6, this study compared different Grad-CAM methods and selected DenseCAM, based on Grad-CAM, as the optimal choice in terms of accuracy and computational cost. Additionally, Figure 5.7 illustrates the performance of three different semantic-free image segmentation models on manipulated images, showing that each model is suited for specific manipulation scenarios. Finally, Figure 5.8 and Table 5.1 validate the best combination of manipulation heatmap and segmentation 5.3.4 for integrating heatmaps and segmentation maps. These results demonstrate that WSWCBnet can accurately localize manipulation regions while providing clear region boundaries. Furthermore, in comparison to fully supervised image manipulation detection methods, WSWCBnet achieves superior F1-scores without requiring pixel-level annotations, with a maximum F1-score of 0.417 compared to 0.238 for fully supervised methods.

In summary, the fourth research question is addressed by developing a weakly-supervised manipulation localization system that bridges image-level detection and pixel-level explainability. Crucially, WSWCBnet is built upon the two previously proposed image-level manipulation detection models (WCBnet and DenseWCBnet) whose decisions are made interpretable through explainability techniques such as Grad-CAM and the proposed DenseCAM. These visualization techniques extract class-discriminative attention maps that reflect how the networks distinguish between pristine and manipulated regions. The resulting heatmaps serve as the foundational input for subsequent refinement through segmentation masks. This integration not only enables localization without pixel-level labels, but also confirms that the detection models are grounded in manipulation-relevant features, allowing the transition from classification to reliable region-level localization. The overall frame-

work thus provides an end-to-end interpretable solution for manipulation detection and localization.

5. **Research Question 5:** How to tackle the challenging learning-based image manipulations, called Generative images or Deepfake images?

Proposed Method:

To address the final research question concerning learning-based manipulations such as Generative-AI images, the DeepfakeArt challenge was adopted as a representative benchmark to evaluate the proposed WCBnet and DenseWCBnet. This dataset differs from conventional splicing datasets in that the manipulated content is generated entirely by learning-based models trained to mimic the visual context of the image, often leaving negligible artifact traces detectable by traditional detection methods. To evaluate the robustness of the proposed models, WCBnet and DenseWCBnet were trained and tested on the learning-based splicing subset of DeepfakeArt. Experimental results demonstrated that both models retained high discriminative capability even under these challenging conditions. As shown in Tables 3.12 and 4.6, WCBnet achieved a classification accuracy of 93.9% and an F1-score of 0.912, while DenseWCBnet achieved 97.0% accuracy and an F1-score of 0.951, both surpassing the best-reported results in the challenge benchmark. This performance can be attributed to the models' design, particularly their use of SEL-based error-level manipulation cues and multi-view convolutional block weighting, which facilitate the extraction of subtle manipulation patterns even in generative contexts. These findings validate the effectiveness of the proposed framework in handling modern AI-based image manipulations and extend its applicability to a broader range of real-world scenarios.

The ability to detect and localize such highly realistic fake images highlights the strength of the developed methods in advancing multimedia security and authenticity, especially considering the popularity of generative content across digital platforms.

6.2.2 Summary of Findings

This thesis investigated five critical research questions in the domain of image manipulation detection and localization, and proposed three deep learning-based frameworks (WCBnet, DenseWCBnet, and WSWCBnet) to address challenges in classification performance, model explainability, and weakly-supervised pixel-level localization. Key contributions of this research is summarized:

Robust image-level manipulation classification: The proposed WCBnet and DenseWCBnet achieved state-of-the-art image-level classification performance across six widely used manipulation datasets. Additionally, both models demonstrated strong robustness under various image perturbation scenarios and surpassed existing methods on the DeepfakeArt challenge's learning-based splicing subset, with DenseWCBnet achieving an F1-score of 0.951.

Image manipulation detection network explainability: To address the black-box nature of deep learning models, the DenseCAM method was introduced. By leveraging multi-scale receptive fields within DenseWCBnet, DenseCAM produced reliable manipulation heatmaps, enabling interpretation of the model's decision-making process and addressing the third research question regarding model explainability.

Weakly-supervised pixel-level localization: To enable manipulation localization without the need for pixel-level annotations, the WSWCBnet framework was developed. This model refines the DenseCAM-generated heatmaps by integrating them with non-semantic segmentation maps. The resulting localization maps not only provided intuitive and accurate boundaries but also outperformed a fully supervised method (ManTraNet) on a 40-image test set by 17.9 percentage points in F1-score.

Potentiality to generative manipulations: The proposed models were also evaluated on challenging generative-AI manipulations. On the DeepfakeArt dataset, both WCBnet and DenseWCBnet maintained high detection performance, achieving classification accuracies of 93.9% and 97.0%, respectively. These results highlight their strong potential capabilities to unseen and complex manipulation techniques.

In summary, the contributions presented in this thesis demonstrate the capability of the proposed frameworks to achieve reliable, explainable, and robust image manipulation detection and localization across diverse real-world scenarios, offering practical tools for enhancing multimedia integrity and trustworthiness.

6.3 Future Work

While the methods proposed in this thesis have made significant strides in image manipulation detection and localization, they still exhibit certain limitations, and there remain several challenges to be addressed in the future. Building on the foundations of the proposed WCBnet, DenseWCBnet, and WSWCBnet models, future work will aim to expand the generalization capabilities and real-world applicability of image manipulation detection systems.

6.3.1 Multi-Branch Hybrid Feature Representation

In particular, one promising direction is the development of a multi-branch hybrid feature network, capable of simultaneously addressing various manipulation scenarios, compression formats, and image domains. As illustrated in Figure 6.1, a future architecture could consist of three parallel sub-branches, each tailored to extract features from different image representations: 1) **RGB branch:** responsible for extracting sufficient and original information, such as color features or texture features from the raw image; 2) **SEL branch:** designed to detect error-level manipulation artifacts, as used in the proposed models which have shown superb performance regarding various manipulation datasets; 3) **Edge branch:** focuses on high-frequency boundary artifacts, helpful for identifying clear boundaries of manipulated regions.

Each branch can employ custom feature extractors suited for their input domain, potentially with varying receptive fields. For example, RGB branch could be equipped with a high-dimensional feature extractor and corresponding CBWM for semantic feature extraction. The outputs of multiple branches are then fused using a cross-feature CBWM, forming a rich hybrid feature representation. This enables the network to leverage multi-perspective features, improving robustness in highly post-processed, generative, and format-diverse image manipulation tasks. Specifically, this architecture shows great potential in:

- **Post-processed datasets:** by capturing more diverse visual clues from different image context, the model could improve performance on highly polished datasets such as Coverage and NC2016.

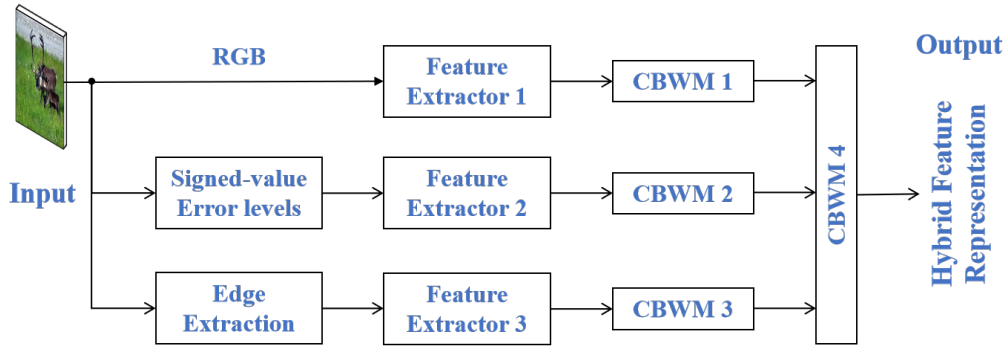


Figure 6.1: Conceptual design of a future multi-branch hybrid feature network for challenging image manipulation detection.

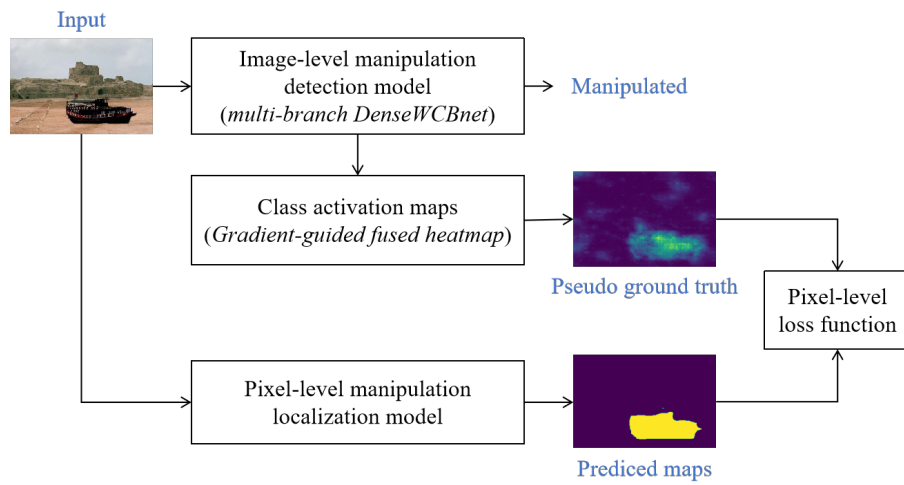


Figure 6.2: Conceptual design of advanced semi-supervised image manipulation localization framework, which also includes multi-branch manipulation detection model and gradient-guided fused heatmap.

- **Non-JPEG formats:** SEL features are format-sensitive as it focuses on the error-level inconsistency in JPEG compression processes; combining them with edge-based and semantic features enables detection in PNG, JPEG-AI, or raw image formats where error-level artifacts are not available.
- **Generative manipulations:** hybrid feature fusion may help detect subtle generation inconsistencies, even when global structure appears coherent, offering a way to counter advanced generative image manipulation tools.

6.3.2 Advancing Explainability and Localization

Even WSWCBnet has shown potential capacity of localizing manipulated regions without annotations, the weakly-supervised results are still less competitive to fully-supervised models. To further enhance manipulation localization and model explainability, three key directions are proposed for extending the DenseCAM framework (as shown in Figure 6.2):

- First, future efforts will aim to further progress image-level manipulation classification networks (as illustrated in the last future work of progressing WCBnet and

DenseWCBnet) to **improve the reliability and consistency** of generated attention maps. By enhancing classification accuracy across a broader range of manipulation datasets, these models can yield more trustworthy class attention responses, providing a stronger foundation for further image manipulation localization.

- Second, the DenseCAM method, while capable of generating multi-view heatmaps from diverse receptive fields, currently fuses them using fixed weights derived from global activation magnitudes. This approach assigns a uniform weight across all pixels for each heatmap, which limits its ability to capture localized differences. A more advanced weighting mechanism is proposed, whereby the gradient of the predicted class with respect to each CBWM output before feature fusion is utilized to derive spatially varying pixel-level weights. This **gradient-guided heatmap fusion** would allow the model to emphasize manipulation regions more adaptively, especially in images with small-scale modifications or complex backgrounds.
- Third, the initial attention maps generated by DenseCAM could serve as **pseudo ground-truth masks** for training a fully-supervised manipulation localization network. By leveraging these automatically derived pseudo masks, it becomes feasible to train high-capacity segmentation models without requiring pixel-level annotations. This semi-supervised learning framework bridges the gap between weakly-supervised explainability-driven localization and fully-supervised segmentation accuracy, further extending the competitiveness to state-of-the-art fully-supervised image manipulation localization methods.

Together, these directions offer a path toward more accurate, interpretable, and annotation-efficient manipulation localization frameworks capable of operating in real-world scenarios with limited supervision.

6.3.3 Toward Practical, Adaptive, and Scalable Manipulation Detection

As the demand for trustworthy visual content increases in digital media, journalism, and daily communication, image manipulation detection must evolve from academic models into deployable, adaptive systems. Future work should prioritize three interlinked goals: real-time performance, consumer accessibility, and continuous learning in dynamic environments.

- First, real-time detection is essential for practical use in social platforms and fact-checking pipelines. The model's accessibility to the users necessitates the development of lightweight architectures through techniques such as model distillation [186, 187], enabling deployment on mobile or embedded devices.
- Second, with manipulated images constantly being created, models must acquire incremental learning capabilities, allowing them to adapt to unseen manipulation types without full retraining. It could be achieved by three main directions in the incremental learning, namely regulation, replay and parameter isolation [188, 186, 189].
- Third, to bridge the gap between research and public accessibility, manipulation detection systems should integrate with cloud-based infrastructures to be applicable in users' daily life. For example, it could be achieved by Model as a Service (MaaS) which enables that pre-trained models can be uploaded to cloud platforms

and offered as APIs for efficient online inference [190]. This approach allows for complicated learning-based model structure and real-time processing of large datasets, making the proposed image manipulation detection model adaptable to evolving requirements.

In summary, advancing toward practical deployment requires models that are not only accurate and interpretable but also lightweight, adaptive, and accessible which closes the loop between deep learning research and real-world multimedia authentication.

Bibliography

- [1] S. Tyagi and D. Yadav, “A detailed analysis of image and video forgery detection techniques,” *The Visual Computer*, vol. 39, no. 3, pp. 813–833, 2023.
- [2] J. Langguth, K. Pogorelov, S. Brenner, P. Filkuková, and D. T. Schroeder, “Don’t trust your eyes: image manipulation in the age of deepfakes,” *Frontiers in Communication*, vol. 6, p. 632317, 2021.
- [3] H. R. Chennamma and B. Madhushree, “A comprehensive survey on image authentication for tamper detection with localization,” *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 1873–1904, 01 2023.
- [4] M. Zanardelli, F. Guerrini, R. Leonardi, and N. Adami, “Image forgery detection: a survey of recent deep-learning approaches,” *Multimedia Tools and Applications*, vol. 82, no. 12, pp. 17 521–17 566, 2023.
- [5] L. Verdoliva, “Media forensics and deepfakes: an overview,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.
- [6] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.
- [7] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proceedings of Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.
- [8] S. Walia, K. Kumar, M. Kumar, and X.-Z. Gao, “Fusion of handcrafted and deep features for forgery detection in digital images,” *IEEE Access*, vol. 9, pp. 99 742–99 755, 2021.
- [9] Z. Zhang, Y. Qian, Y. Zhao, X. Zhang, L. Zhu, J. Wang, and J. Zhao, “Noise and edge based dual branch image manipulation detection,” in *Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things*, 2023, pp. 963–968.
- [10] F. Matern, C. Riess, and M. Stamminger, “Gradient-based illumination description for image forgery detection,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1303–1317, 2020.
- [11] R. Thakur and R. Rohilla, “Recent advances in digital image manipulation detection techniques: A brief review,” *Forensic Science International*, vol. 312, p. 110311, 2020.

- [12] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, “AugFPN: Improving multi-scale feature learning for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 595–12 604.
- [13] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, “MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3539–3553, 2022.
- [14] X. Liu, Y. Liu, J. Chen, and X. Liu, “PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 1–1, 11 2022.
- [15] A. Chinthia, B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright, and R. Ptucha, “Recurrent convolutional structures for audio spoof and video deepfake detection,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1024–1037, 2020.
- [16] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” in *Proceedings of 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.
- [17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: visual explanations from deep networks via gradient-based localization,” *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2020.
- [18] Z. Wang and C. Abhayaratne, “WCBnet: Weighted convolutional block modelling of signed-value error levels for image-wise copy-move and splicing detection,” in *Proceedings of 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2023, pp. 1–6.
- [19] P. Duszejko, T. Walczyna, and Z. Piotrowski, “Detection of manipulations in digital images: A review of passive and active methods utilizing deep learning,” *Applied Sciences*, vol. 15, no. 2, p. 881, 2025.
- [20] M. Mehralian and B. Karasfi, “RDCGAN: Unsupervised representation learning with regularized deep convolutional generative adversarial networks,” in *Proceedings of 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*, 2018, pp. 31–38.
- [21] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, “Alias-free generative adversarial networks,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 852–863, 2021.
- [22] V. K. Kadha, S. Bakshi, and S. K. Das, “Unravelling digital forgeries: A systematic survey on image manipulation detection and localization,” *ACM Computing Surveys*, 2025.
- [23] Y.-F. Hsu and S.-F. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency,” in *Proceedings of International Conference on Multimedia and Expo*, 2006.
- [24] M. Hussain, G. Muhammad, and G. Bebis, “Image splicing detection using chroma and saturation features,” *International Journal of Artificial Intelligence and Tools*, vol. 23, no. 02, p. 1460002, 2014.

- [25] J. Dong, W. Wang, and T. Tan, “CASIA image tampering detection evaluation database,” in *Proceedings of 2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.
- [26] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. de Rezende Rocha, “Exposing digital image forgeries by illumination color classification,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, 2013.
- [27] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in *Proceedings of 2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.
- [28] P. Korus and J. Huang, “Multi-scale analysis strategies in PRNU-based tampering localization,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, 2016.
- [29] —, “Evaluation of random field models in multi-scale tampering localization,” in *Proceedings of 2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.
- [30] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus, “MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation,” in *Proceedings of 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 63–72.
- [31] D. Shullani, M. Fontani, M. Iuliani, A. Piva, and M. Barni, “VISION: A video and image dataset for source identification,” *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, 2017.
- [32] D. Cozzolino, G. Poggi, and L. Verdoliva, “Copy-move forgery detection based on patchMatch,” in *Proceedings of 2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5312–5316.
- [33] J. Wang, Q. Zhao, H. Guo, and Z. Yang, “A hybrid copy-move forgery detection based on SIFT and Hu moments,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, p. 67, 2022.
- [34] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, “CoMoFoD — new database for copy-move forgery detection,” in *Proceedings of ELMAR-2013*, 2013, pp. 49–54.
- [35] P. Korus and J. Huang, “Comofod—new database for copy-move forgery detection,” in *Proceedings of 2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2043–2047.
- [36] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “COVERAGE — a novel database for copy-move forgery detection,” in *Proceedings of 2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 161–165.
- [37] J. Li, R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian, “Universal perturbation attack against image retrieval,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4899–4908.
- [38] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, “Exploiting spatial structure for localizing manipulated image regions,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4970–4979.

- [39] 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.
- [40] M. Huh, A. Liu, A. Owens, and A. A. Efros, “Fighting fake news: Image splice detection via learned self-consistency,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.
- [41] J. Fridrich, M. Goljan, and R. Du, “Detection of double-compression in JPEG images for applications in steganography,” *IEEE Transactions on Information Forensics and Security*, vol. 53, no. 4, pp. 585–592, 2003.
- [42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [43] —, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [44] E. Gonzalez-Sosa, J. Fierrez, R. Vera-Rodriguez, and F. Alonso-Fernandez, “Facial soft biometrics for recognition in the wild: Recent works, annotation, and COTS evaluation,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2001–2014, 2018.
- [45] I. Korshunova, W. Shi, J. Dambre, and L. Theis, “Fast face-swap using convolutional neural networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3677–3685.
- [46] J. Adler and S. Lunz, “Banach wasserstein GAN,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [47] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *Proceedings of International Conference on Learning Representations*, 2019.
- [48] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
- [49] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.
- [50] T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4570–4580.
- [51] Y. Jiang, S. Chang, and Z. Wang, “TransGAN: Two pure transformers can make one strong gan, and that can scale up,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 745–14 758, 2021.
- [52] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, “On the detection of digital face manipulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5781–5790.

- [53] J. C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proença, and J. Fierrez, “GANprintR: Improved fakes and evaluation of the state of the art in face manipulation detection,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1038–1048, 2020.
- [54] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *International Conference on Learning Representations*, 2018.
- [55] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Face-forensics++: Learning to detect manipulated facial images,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11.
- [56] P. Korshunov and S. Marcel, “Deepfakes: a new threat to face recognition? assessment and detection,” *ArXiv Preprint ArXiv:1812.08685*, 2018, (Not peer-reviewed).
- [57] Y. Li and S. Lyu, “Exposing deepFake videos by detecting face warping artifacts,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
- [58] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Celeb-DF: A large-scale challenging dataset for deepfake forensics,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216.
- [59] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, “Deepfakes evolution: Analysis of facial regions and fake detection performance,” in *Proceedings of International Conference on Pattern Recognition*. Springer, 2021, pp. 442–456.
- [60] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, “The deepfake detection challenge (DFDC) preview dataset,” *ArXiv Preprint ArXiv:1910.08854*, 2019, (Not peer-reviewed).
- [61] Y. Li, M.-C. Chang, and S. Lyu, “In ictu oculi: Exposing AI created fake videos by detecting eye blinking,” in *Proceedings of 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1–7.
- [62] T. Zecheng, W. Xinyuan, Y. Hongli, and Z. Yansong, “U2-Net for image forgery detection and localization,” in *Proceedings of 2021 International Conference on Computer Technology and Media Convergence Design (CTMCD)*, 2021, pp. 166–172.
- [63] A. Parveen, Z. H. Khan, and S. N. Ahmad, “Block-based copy-move image forgery detection using DCT,” *Iran Journal of Computer Science*, vol. 2, pp. 89–99, 2019.
- [64] M. H. Siddiqi, K. Asghar, U. Draz, A. Ali, M. Alruwaili, Y. Alhwaiti, S. Alanazi, and M. Kamruzzaman, “Image splicing-based forgery detection using discrete wavelet transform and edge weighted local binary patterns,” *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.
- [65] E.-S. M. El-Alfy and M. A. Qureshi, “Combining spatial and DCT based markov features for enhanced blind detection of image splicing,” *Pattern Anal. Appl.*, vol. 18, no. 3, p. 713–723, Aug. 2015.

- [66] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on markov features in DCT and DWT domain," *Pattern Recognition*, vol. 45, no. 12, pp. 4292–4299, 2012.
- [67] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proceedings of the 9th Workshop on Multimedia & Security*, 2007, pp. 51–62.
- [68] X. Li, T. Jing, and X. Li, "Image splicing detection based on moment features and hilbert-Huang transform," in *Proceedings of 2010 IEEE International Conference on Information Theory and Information Security*, 2010, pp. 1127–1130.
- [69] Z. Moghaddasi, H. Jalab, R. Md. Noor, and S. Aghabozorgi, "Improving RLRN image splicing detection with the use of PCA and kernel PCA," *The Scientific World Journal*, p. 606570, 2014.
- [70] P. Sutthiwan, Y. Q. Shi, W. Su, and T.-T. Ng, "Rake transform and edge statistics for image forgery detection," in *Proceedings of 2010 IEEE International Conference on Multimedia and Expo*, 2010, pp. 1463–1468.
- [71] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on DCT and local binary pattern," in *Proceedings of 2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 253–256.
- [72] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications*, vol. 25, pp. 985–995, 2014.
- [73] W. Wang, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of markov chain," in *Proceedings of 2010 IEEE International Conference on Image Processing*, 2010, pp. 2101–2104.
- [74] C.-M. Pun, B. Liu, and X.-C. Yuan, "Multi-scale noise estimation for image splicing forgery detection," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 195–206, 2016.
- [75] Q. Zhang, W. Lu, and J. Weng, "Joint image splicing detection in DCT and contourlet transform domain," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 449–458, 2016.
- [76] Y. Zhang, C. Zhao, Y. Pi, S. Li, and S. Wang, "Image-splicing forgery detection based on local binary patterns of DCT coefficients: Image-splicing forgery detection," *Security and Communication Networks*, vol. 8, no. 14, pp. 2386–2395, 2015.
- [77] X. Zhang, Z. H. Sun, S. Karaman, and S. F. Chang, "Discovering image manipulation history by pairwise relation and forensics tools," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1012–1023, 2020.
- [78] O. Mayer and M. C. Stamm, "Exposing fake images with forensic similarity graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1049–1064, 2020.

- [79] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication*, vol. 39, pp. 46–74, 2015.
- [80] R. Dixit, R. Naskar, and A. Sahoo, "Copy-move forgery detection exploiting statistical image features," in *Proceedings of 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 2277–2281.
- [81] N. Kanwal, A. Girdhar, L. Kaur, and J. S. Bhullar, "Detection of digital image forgery using fast fourier transform and local features," in *Proceedings of 2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, 2019, pp. 262–267.
- [82] G. Gani and F. Qadir, "A robust copy-move forgery detection technique based on discrete cosine transform and cellular automata," *Journal of Information Security and Applications*, vol. 54, p. 102510, 2020.
- [83] F. Yang, J. Li, Z. Wei, and J. Weng, "Copy-move forgery detection based on hybrid features," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 73–83, 03 2017.
- [84] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.
- [85] C.-M. Pun, X.-C. Yuan, and X.-L. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1705–1716, 2015.
- [86] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.
- [87] S. Bravo-Solorio and A. K. Nandi, "Exposing duplicated regions affected by reflection, rotation and scaling," in *Proceedings of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1880–1883.
- [88] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block," in *Proceedings of 2009 International Conference on Multimedia Information Networking and Security*, vol. 1, 2009, pp. 25–29.
- [89] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 857 – 867, 01 2011.
- [90] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on SURF," in *Proceedings of 2010 International Conference on Multimedia Information Networking and Security*, 2010, pp. 889–892.

- [91] J.-C. Lee, C.-P. Chang, and W.-K. Chen, "Detection of copy-move image forgery using histogram of orientated edients," *Information Sciences*, vol. 321, pp. 250–262, 2015.
- [92] J. Zhong and Y. Gan, "Detection of copy-move forgery using discrete analytical Fourier–Mellin transform," *Nonlinear Dynamics*, vol. 84, no. 1, pp. 189–202, 2016.
- [93] G. Jin and X. Wan, "An improved method for SIFT-based copy-move forgery detection using non-maximum value suppression and optimized J-linkage," *Signal Processing: Image Communication*, vol. 57, pp. 113–125, 2017.
- [94] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2, pp. 180–189, 2006.
- [95] I. T. Ahmed, B. T. Hammad, and N. Jamil, "Image copy-move forgery detection algorithms based on spatial feature domain," in *Proceedings of 2021 IEEE 17th International Colloquium on Signal Processing Its Applications (CSPA)*, 2021, pp. 92–96.
- [96] 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, pp. 2691–2706, 2018.
- [97] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, pp. 1–74, 2021.
- [98] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *IEEE Access*, vol. 8, pp. 133 488–133 502, 2020.
- [99] A. Abhishek and N. Jindal, "Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3571–3599, 2020.
- [100] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proceedings of 2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1–6.
- [101] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 1–12, 2016.
- [102] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, "Aligned and non-aligned double JPEG detection using convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 153–163, 2017.
- [103] J. Bakas, S. Ramachandra, and R. Naskar, "Double and triple compression-based forgery detection in JPEG images using deep convolutional neural network," *Journal of Electronic Imaging*, vol. 29, no. 2, pp. 1 – 25, 2020.

- [104] R. Salloum, Y. Ren, and C.-C. Jay Kuo, “Image splicing localization using a multi-task fully convolutional network (MFCN),” *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201–209, 2018.
- [105] B. Chen, X. Qi, Y. Zhou, G. Yang, Y. Zheng, and B. Xiao, “Image splicing localization using residual image and residual-based fully convolutional network,” *Journal of Visual Communication and Image Representation*, vol. 73, p. 102967, 2020.
- [106] H. Mareen, D. Vanden Bussche, F. Guillaro, D. Cozzolino, G. Van Wallendael, P. Lambert, and L. Verdoliva, “Comprint: Image forgery detection and localization using compression fingerprints,” in *Proceedings of International Conference on Pattern Recognition*. Springer, 2022, pp. 281–299.
- [107] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-forensics: Using capsule networks to detect forged images and videos,” in *Proceedings of ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2307–2311.
- [108] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” in *Proceedings of 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.
- [109] X. Zhang, S. Karaman, and S.-F. Chang, “Detecting and simulating artifacts in GAN fake images,” in *Proceedings of 2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2019, pp. 1–6.
- [110] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, “Recurrent convolutional strategies for face manipulation detection in videos,” *Interfaces (GUI)*, vol. 3, no. 1, pp. 80–87, 2019.
- [111] D. Güera and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in *Proceedings of 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.
- [112] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, “Incremental learning for the detection and classification of GAN-generated images,” in *Proceedings of 2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019, pp. 1–6.
- [113] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, “Detecting deep-fake videos from appearance and behavior,” in *Proceedings of 2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.
- [114] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” *Information Fusion*, vol. 64, pp. 131–148, 2020.
- [115] T. Brooks, A. Holynski, and A. A. Efros, “InstructPix2Pix: Learning to follow image editing instructions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 18 392–18 402.
- [116] Z. Zhang, C. Wang, and X. Zhou, “A survey on passive image copy-move forgery detection,” *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 6–31, 2018.

- [117] B. Chen, X. Qi, Y. Zhou, G. Yang, Y. Zheng, and B. Xiao, "Image splicing localization using residual image and residual-based fully convolutional network," *Journal of Visual Communication and Image Representation*, vol. 73, p. 102967, 2020.
- [118] X. Bi, Z. Zhang, and B. Xiao, "Reality transform adversarial generators for image splicing forgery detection and localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 294–14 303.
- [119] A. Islam, C. Long, A. Basharat, and A. Hoogs, "Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4676–4685.
- [120] T. Bianchi and A. Piva, "Detection of double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2012.
- [121] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "Splicing forgeries localization through the use of first digit features," in *Proceedings of 2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 143–148.
- [122] W. Wang and H. Farid, "Detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 1137–1148, 2009.
- [123] D. Bhowmik, A. Natu, T. Ishikawa, T. Feng, and C. Abhayaratne, "The Jpeg-Blockchain framework for glam services," in *Proceedings of 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2018, pp. 1–6.
- [124] C. Abhayaratne and D. Bhowmik, "Scalable watermark extraction for real-time authentication of JPEG 2000 images," *Journal of Real-time Image Processing*, vol. 8, pp. 307–325, 2013.
- [125] N. Nida, A. Irtaza, N. Ilyas *et al.*, "Forged face detection using ELA and deep learning techniques," in *Proceedings of 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*. IEEE, 2021, pp. 271–275.
- [126] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, "Learning JPEG compression artifacts for image manipulation detection and localization," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1875–1895, 2022.
- [127] D. Agrawal, H. Makwana, S. S. Dave, S. Degadwala, and V. Desai, "Error level analysis and deep learning for detecting image forgeries," in *Proceedings of 2023 7th International Conference on Computing Methodologies and Communication (IC-CMC)*, 2023, pp. 114–117.
- [128] N. Vijayalakshmi K, J. Sasikala, and C. Shanmuganathan, "Copy-paste forgery detection using deep learning with error level analysis," *Multimedia Tools and Applications*, vol. 83, no. 2, pp. 3425–3449, 2024.
- [129] A. B. Singh and K. M. Singh, "Application of error level analysis in image spam classification using deep learning model," *PLOS ONE*, vol. 18, no. 12, pp. 1–17, 2023.

- [130] X. Lin, S. Wang, J. Deng, Y. Fu, X. Bai, X. Chen, X. Qu, and W. Tang, “Image manipulation detection by multiple tampering traces and edge artifact enhancement,” *Pattern Recognition*, vol. 133, p. 109026, 2023.
- [131] Z. Shi, H. Chen, and D. Zhang, “Transformer-auxiliary neural networks for image manipulation localization by operator inductions,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4907–4920, 2023.
- [132] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, and X. Liu, “Hierarchical fine-grained image forgery detection and localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 3155–3165.
- [133] G. Singh and P. Goyal, “SDCN2: A shallow densely connected CNN for multi-purpose image manipulation detection,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 18, no. 3s, pp. 1–22, 2022.
- [134] G. Singh and K. Singh, “Counter JPEG anti-forensic approach based on the second-order statistical analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1194–1209, 2019.
- [135] Y. Niu, B. Tondi, Y. Zhao, R. Ni, and M. Barni, “Image splicing detection, localization and attribution via JPEG primary quantization matrix estimation and clustering,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5397–5412, 2021.
- [136] I. B. K. Sudiatmika, F. Rahman, T. Trisno, and S. Suyoto, “Image forgery detection using error level analysis and deep learning,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, pp. 653–659, 2019.
- [137] M. S, S. M D, R. S, A. K. Subramanian, V. K. V, and M. S. S, “Convolutional neural network-based image tamper detection with error level analysis,” in *Proceedings of 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 2024, pp. 1–7.
- [138] 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.
- [139] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. Marc, “DEFACTO: Image and face manipulation dataset,” in *Proceedings of 2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [140] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [141] 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9535–9544.
- [142] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, “Constrained R-CNN: A general image manipulation detection model,” in *Proceedings of 2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

- [143] 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.
- [144] T. Le-Tien, D. Ho-Van, N. Pham-Ng-Quynh, H. Phan-Xuan, and T. Nguyen-Thanh, “Modified CNN model-based forgery detection applied to Multiple-Resolution tampered images,” in *Proceedings of 2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, 2021, pp. 126–131.
- [145] D. Mishra, S. K. Singh, and R. K. Singh, “Wavelet-based deep auto encoder-decoder (WDAED)-based image compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1452–1462, 2020.
- [146] G. K. Birajdar and V. H. Mankar, “Passive image manipulation detection using wavelet transform and support vector machine classifier,” in *Proceedings of International Conference on ICT for Sustainable Development: ICT4SD 2015 Volume 1*. Springer, 2016, pp. 447–455.
- [147] D. Bhowmik and C. Abhayaratne, “The effect of quality scalable image compression on robust watermarking,” in *Proceedings of 2009 16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–8.
- [148] H. H. Nguyen, M. Kuribayashi, J. Yamagishi, and I. Echizen, “Effects of image processing operations on adversarial noise and their use in detecting and correcting adversarial images,” *IEICE TRANSACTIONS on Information and Systems*, vol. 105, no. 1, pp. 65–77, 2022.
- [149] J. Azzeh, B. Zahran, and Z. Alqadi, “Salt and pepper noise: Effects and removal,” *JOIV: International Journal on Informatics Visualization*, vol. 2, no. 4, pp. 252–256, 2018.
- [150] H. Aboutalebi, D. Mao, R. Fan, C. Xu, C. He, and A. Wong, “DeepfakeArt challenge: A benchmark dataset for generative AI art forgery and data poisoning detection,” *ArXiv Preprint ArXiv:2306.01272*, 2023, not peer-reviewed.
- [151] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [152] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, “DINOv2: Learning robust visual features without supervision,” *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.
- [153] J. Ascenso, E. Alshina, and T. Ebrahimi, “The JPEG AI standard: Providing efficient human and machine visual data consumption,” *IEEE MultiMedia*, vol. 30, no. 1, pp. 100–111, 2023.
- [154] E. Alshina, J. Ascenso, and T. Ebrahimi, “JPEG AI: The first international standard for image coding based on an end-to-end learning-based approach,” *IEEE MultiMedia*, vol. 31, no. 4, pp. 60–69, 2024.
- [155] S. Bergmann, F. Brand, and C. Riess, “Three forensic cues for JPEG AI images,” *ArXiv Preprint ArXiv:2504.03191*, 2025.

- [156] I. C. Camacho and K. Wang, "Convolutional neural network initialization approaches for image manipulation detection," *Digital Signal Processing*, vol. 122, p. 103376, 2022.
- [157] Z. Guo, G. Yang, J. Chen, and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," *Computer Vision and Image Understanding*, vol. 204, p. 103170, 2021.
- [158] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2864–2875, 2013.
- [159] Y. Zhang and G. Feng, "Efficient multi-scale feature fusion for image manipulation detection," *IEICE TRANSACTIONS on Information and Systems*, vol. 105, no. 5, pp. 1107–1111, 2022.
- [160] M. Roy and S. Mukhopadhyay, "Multi-focus fusion using image matting and geometric mean of DCT-variance," in *Proceedings of Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part I 5*. Springer, 2021, pp. 212–223.
- [161] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
- [162] A. D. Nascimento, M. M. Horta, A. C. Frery, and R. J. Cintra, "Comparing edge detection methods based on stochastic entropies and distances for PolSAR imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 2, pp. 648–663, 2013.
- [163] J. Yoon and C. Lee, "Edge detection using the bhattacharyya distance with adjustable block space," *Electronic Imaging*, vol. 32, pp. 1–7, 2020.
- [164] M. Ugale and J. Midhunchakkaravarthy, "Machine learning-based image forgery detection using combinatorial mapping with hybrid distance measure," in *Proceedings of International Conference on Computational Intelligence*. Springer, 2023, pp. 617–629.
- [165] A. Alharbi, W. Alhakami, S. Bourouis, F. Najar, and N. Bouguila, "Inpainting forgery detection using hybrid generative/discriminative approach based on bounded generalized gaussian mixture model," *Applied Computing and Informatics*, vol. 20, no. 1/2, pp. 89–104, 2024.
- [166] Y. Hamid, S. Elyassami, Y. Gulzar, V. R. Balasaraswathi, T. Habuza, and S. Wani, "An improvised CNN model for fake image detection," *International Journal of Information Technology*, vol. 15, no. 1, pp. 5–15, 2023.
- [167] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, and X. Liu, "Hierarchical fine-grained image forgery detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 3155–3165.
- [168] M. Barni, Q.-T. Phan, and B. Tondi, "Copy-move source-target disambiguation through multi-branch CNNs," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1825–1840, 2021.

- [169] S. Walia, K. Kumar, S. Agarwal, and H. Kim, “Using XAI for deep learning-based image manipulation detection with shapley additive explanation,” *Symmetry*, vol. 14, no. 8, p. 1611, 2022.
- [170] W. H. Abir, F. R. Khanam, K. N. Alam, M. Hadjouni, H. Elmannai, S. Bourouis, R. Dey, and M. M. Khan, “Detecting deepfake images using deep learning techniques and explainable AI methods,” *Intelligent Automation & Soft Computing*, vol. 35, no. 2, pp. 2151–2169, 2023.
- [171] Q. Zeng, H. Wang, Y. Zhou, R. Zhang, and S. Meng, “Semi-supervised image manipulation localization with residual enhancement,” *Expert Systems With Applications*, vol. 252, p. 124171, 2024.
- [172] D.-C. Țânțaru, E. Oneață, and D. Oneață, “Weakly-supervised deepfake localization in diffusion-generated images,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6258–6268.
- [173] Y. Zhou, H. Wang, Q. Zeng, R. Zhang, and S. Meng, “Exploring weakly-supervised image manipulation localization with tampering edge-based class activation map,” *Expert Systems With Applications*, vol. 249, p. 123501, 2024.
- [174] Y. Zhai, T. Luan, D. Doermann, and J. Yuan, “Towards generic image manipulation detection with weakly-supervised self-consistency learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 390–22 400.
- [175] M. Zhang, Y. Zhou, J. Zhao, Y. Man, B. Liu, and R. Yao, “A survey of semi-and weakly supervised semantic segmentation of images,” *Artificial Intelligence Review*, vol. 53, pp. 4259–4288, 2020.
- [176] K. Hacıfendioglu, S. Adanur, and G. Demir, “Automatic landslide segmentation using a combination of grad-CAM visualization and K-means clustering techniques,” *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 48, no. 2, pp. 943–959, 2024.
- [177] S. N. Hasany, C. Petitjean, and F. Mériaudeau, “Seg-XRes-CAM: Explaining spatially local regions in image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 3733–3738.
- [178] H. Kweon, S.-H. Yoon, and K.-J. Yoon, “Weakly supervised semantic segmentation via adversarial learning of classifier and reconstructor,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 329–11 339.
- [179] M. Weber, H. Wang, S. Qiao, J. Xie, M. D. Collins, Y. Zhu, L. Yuan, D. Kim, Q. Yu, D. Cremers, L. Leal-Taixe, A. L. Yuille, F. Schroff, H. Adam, and L.-C. Chen, “DeepLab2: A TensorFlow Library for Deep Labeling,” *ArXiv: 2106.09748*, 2021.
- [180] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

- [181] D. Gupta, “Image segmentation Keras: Implementation of Segnet, FCN, Unet, PSPnet and other models in Keras,” *ArXiv Preprint ArXiv:2307.13215*, 2023.
- [182] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks,” in *Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.
- [183] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, “LayerCAM: Exploring hierarchical class activation maps for localization,” *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021.
- [184] T. S. Gunawan, S. A. M. Hanafiah, M. Kartiwi, N. Ismail, N. F. Za’bah, and A. N. Nordin, “Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7, no. 1, pp. 131–137, 2017.
- [185] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9543–9552.
- [186] Z. Wang, T. Gu, Y. Zhu, D. Li, H. Yang, and W. Du, “FLDNet: Frame-level distilling neural network for EEG emotion recognition,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 7, pp. 2533–2544, 2021.
- [187] F. M. Nardini, C. Rulli, S. Trani, and R. Venturini, “Distilled neural networks for efficient learning to rank,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4695–4712, 2022.
- [188] G. M. Van de Ven, T. Tuytelaars, and A. S. Tolias, “Three types of incremental learning,” *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.
- [189] E. Belouadah, A. Popescu, and I. Kanellos, “A comprehensive study of class incremental learning algorithms for visual tasks,” *Neural Networks*, vol. 135, pp. 38–54, 2021.
- [190] W. Gan, S. Wan, and S. Y. Philip, “Model-as-a-service (MaaS): A survey,” in *Proceedings of 2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 4636–4645.

Appendices

Appendix A

Enhanced heatmaps of WSWCBnet

This chapter is the supplemental materials to the experimental results of Section 5.4.5, which show the enhanced heatmaps combined by manipulation heatmaps and segmented maps.

A.1 Heatmaps with Effective Combination

More enhanced heatmaps with effective combination are shown as follows:

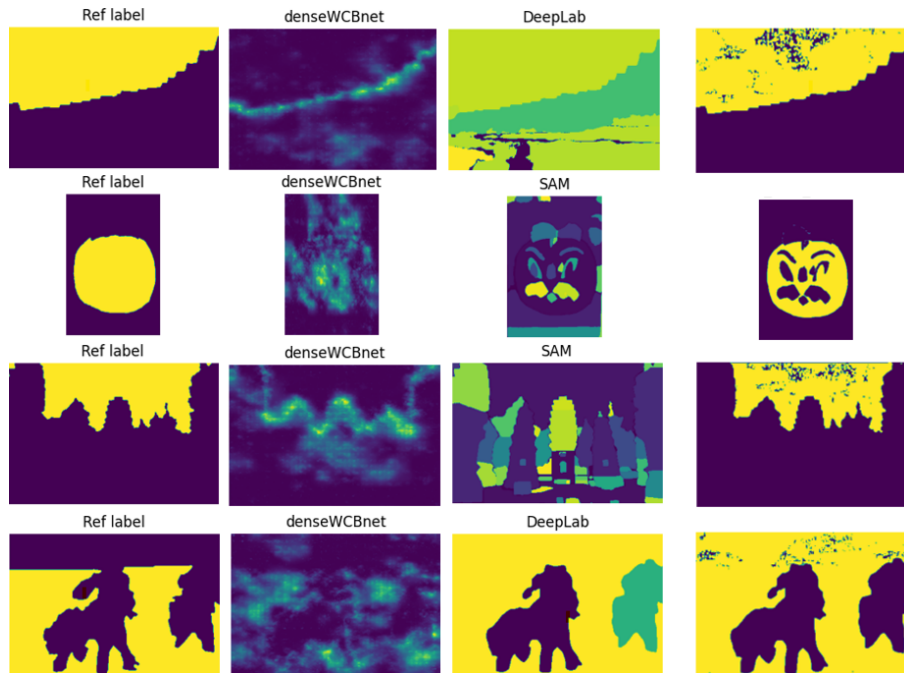


Figure A.1: *Enhanced heatmaps generated by effective combinations of WSWCBnet (part 1); The first column shows the reference labels, second and third columns are the selected models, and the last column is the enhanced heatmaps.*

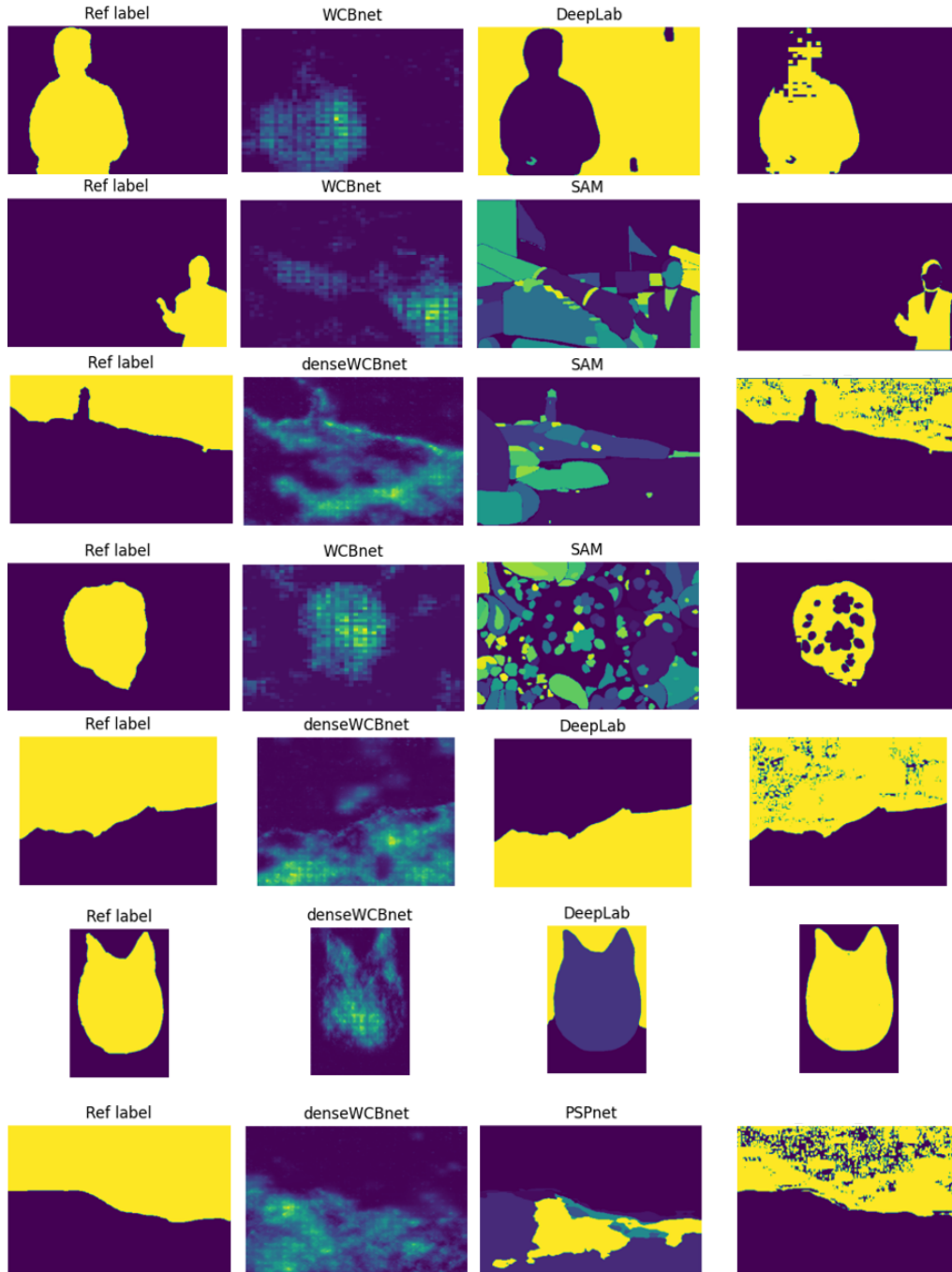


Figure A.2: *Enhanced heatmaps generated by effective combinations of WSWCBnet (part 2).*

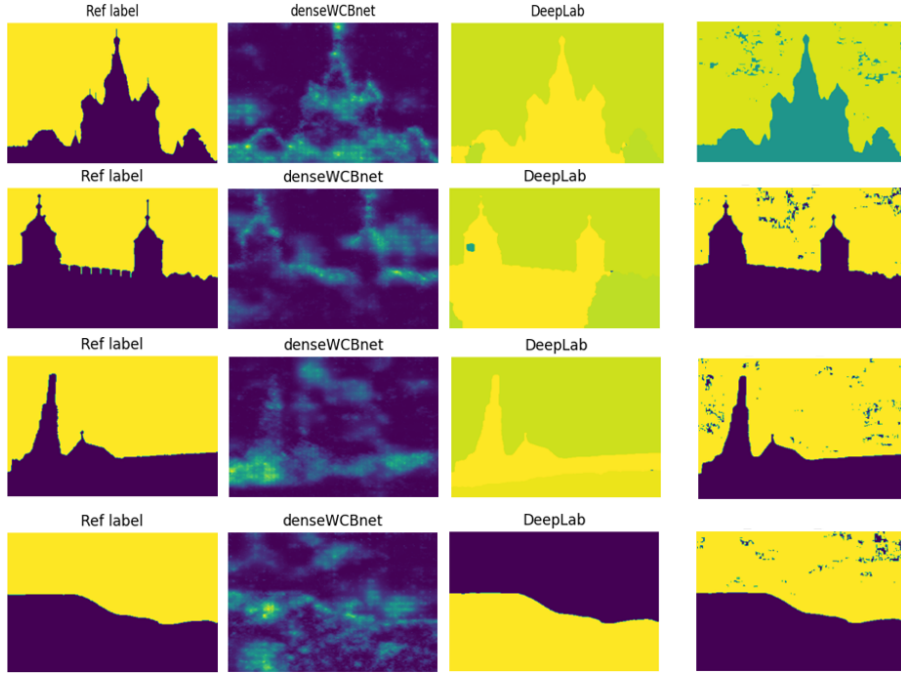


Figure A.3: *Enhanced heatmaps generated by effective combinations of WSWCBnet (part 3).*

A.2 Heatmaps With Ineffective Combination

There are three main conditions that cause the ineffective combination between manipulation heatmap and segmented map. Several examples that either heatmaps or segmaps were ineffective during combination are shown as follows:

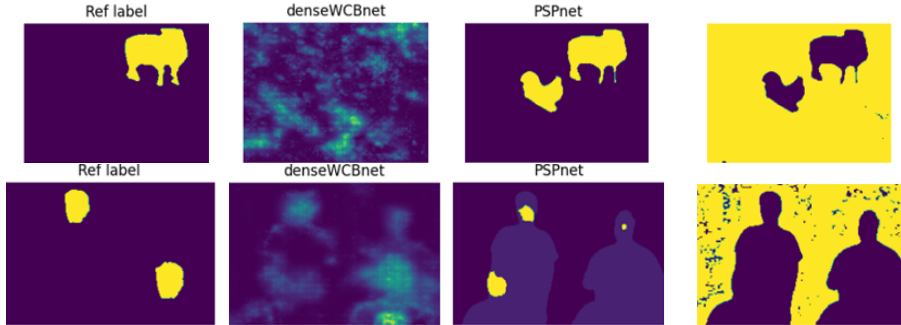


Figure A.4: *Invalid manipulation heatmap that the image-level models focus on wrong regions in the manipulation heatmap.*

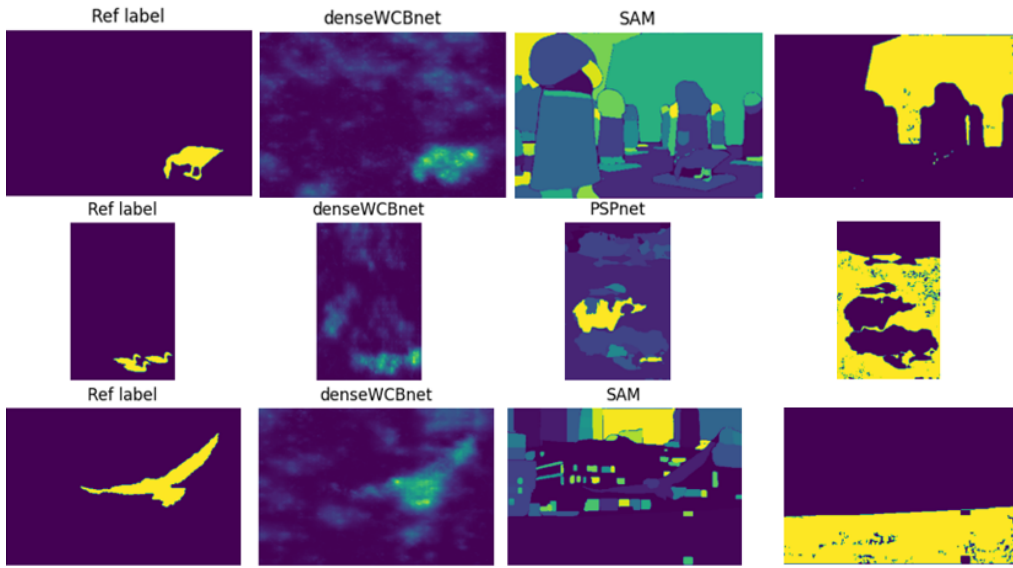


Figure A.5: Invalid segmented maps that the target regions are not clearly extracted by the non-semantic segmentation models.

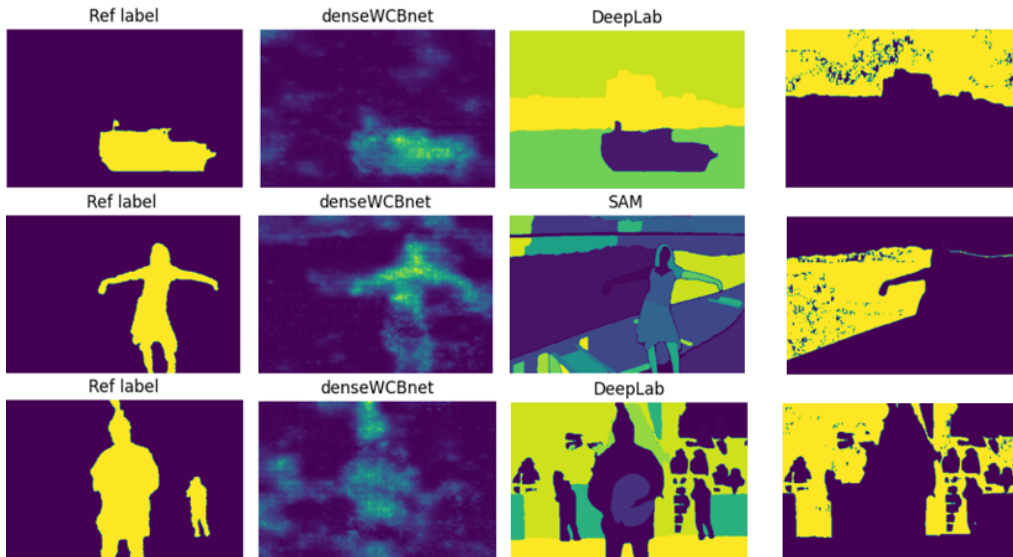


Figure A.6: Invalid combination metric that both heatmaps and segmap are valid while the method to combine them is ineffective.