This is a repository copy of *WCBnet: Weighted convolutional block modelling of signed-value error levels for image-wise copy-move and splicing detection*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/206916/

Version: Accepted Version

Proceedings Paper:
Wang, Z. and Abhayaratne, C. orcid.org/0000-0002-2799-7395 (2023) WCBnet: Weighted convolutional block modelling of signed-value error levels for image-wise copy-move and splicing detection. In: 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP) Proceedings. 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP), 27-29 Sep 2023, Poitiers, France. Institute of Electrical and Electronics Engineers (IEEE) . ISBN 9798350338942

https://doi.org/10.1109/mmsp59012.2023.10337680

© 2023 The Authors. Except as otherwise noted, this author-accepted version of a paper published in 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP) Proceedings is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:
https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

WCBnet: Weighted Convolutional Block Modelling of Signed-value Error Levels for Image-wise Copy-move and Splicing Detection

Ziyong Wang and Charith Abhayaratne
Department of Electronic and Electrical Engineering, The University of Sheffield
Sheffield, S1 3JD, United Kingdom
zwang172@sheffield.ac.uk, c.abhayaratne@sheffield.ac.uk

***Abstract*—Image manipulation which can easily generate hard-to-perceive fake information by image editing tools has become a threat of spreading visual mis/disinformation. With the speed and growth of such visual information presence in social media with respect to the current geopolitical affairs, tools for highly accurate verification of the authenticity of images are vital for AI-based fact checking. This work presents an efficient convolutional neural network (CNN) based approach for image manipulation detection. Our method, called WCBnet, starts with extracting learned features from the signed-value error levels (SEL) of compressed 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. We evaluate the performance of the proposed approach with respect to common manipulation datasets and compare 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.**

***Index Terms*—mis/disinformation, fact checking, manipulated image detection, self-attention, error-level analysis, feature reshaping**

I. 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]. Efficient and highly accurate methods for detecting such manipulated images and verifying the authentic images from large sets of social media images are vital for automated fact-checking purposes. The existing methods focus on image-level [2] and/or pixel level detection [3]. Our focus in this paper is on image-level classification of manipulated / authentic images considering an application domain of fact-checking of large sets of images, where quick verification of the authenticity of individual images is required.

Learning-based methods as well as hand-crafted feature-based methods have been widely designed and refined to expose the visually un-noticeable tampering trajectory of specific manipulation types or the dataset properties [4]. Although those data-driven approaches can achieve extremely high classification accuracy regarding the training dataset, the way they have learned to extract specific features for the training task becomes less effective in the face of other types of tampering. How to guarantee the performance of these image manipulation detection methods to adapt to the complexity of the image forgery dataset is still a challenge.

Several hand-crafted features that globally exist in different image manipulation have been used for image forgery detection. Because of the complex and unique imaging process of the pictures, any image manipulation operations, namely copy-move, splicing and removal, inevitably leave tractable traces due to the inconsistency of the forged area and the background area [1]. Error-level Analysis (ELA) is known to track the artifacts that are caused by inconsistent compression factors or compression times of tampered regions and background regions. An image has to be saved in certain fixed formats after being manipulated, so the tampered area is compressed and saved several times or compressed with different factors from background which results in unnatural artifacts around forged areas forming varying error levels. This JPEG compression-based artifacts have widely been used as hand-crafted features in image manipulation detection [5], [6].

Meanwhile in learning-based methods, CNNs are equipped with the ability to fuse different levels of information at different depths of the convolutional layers to cope with the diversity and complexity of image manipulations. Those multi-scale features that contain global and local information are commonly combined by Feature Pyramids in which the features maps from different layers are fused by repeated up-sample and feature addition [7]. However, this feature combination method with fixed reshaping parameters ignores the different roles played by multiple levels of information, which requires an adaptive architecture that can tailor convolutional neural networks for different types of image manipulation.

To address this, this paper proposes a weighted convolutional block model (WCBnet) to fuse various levels of features extracted from different convolutional blocks, by assigning different weights to those block outputs and computing their

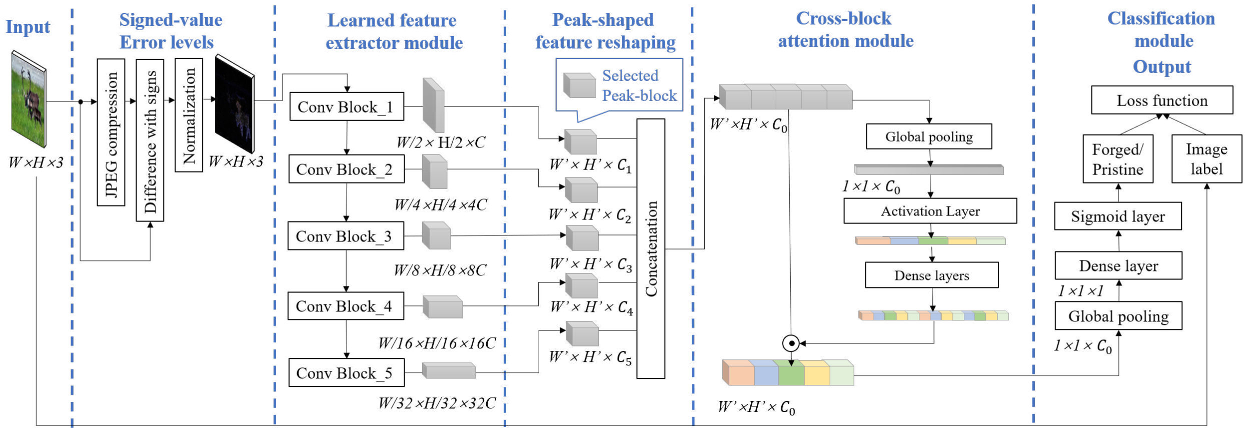


Fig. 1. The five stages of proposed WCBnet, and different colors of features in the figure represent different weights

co-relationship. Before that, individual peak-shaped reshaping is performed on each convolutional block, which could adapt the shape and channel number of combined features according to the dataset properties. As opposed to using an RGB as the input to the learned feature extracting backbone, we compute signed-value error levels (SEL) to use the corresponding error level data as the input to the backbone, which emphasizes the artifacts caused by inconsistent compression process between forged and background areas. The proposed model is evaluated on two common image manipulation types with shared and independent traceable tampering traces, namely copy-move and splicing, in order to verify its adaptation ability.

The main contributions of this paper include:

- 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-block attention 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 those multi-level features according to the needs of different image manipulation detection tasks;
- Proposal of 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 we call signed-value error levels (SEL).

The rest of the paper is organised as follows: The related work is briefly discussed in Sec. II. The details of the proposed method are presented in Sec. III, followed by a thorough evaluation of performance in Sec. IV and the concluding remarks in Sec. V.

II. Related work

The related work includes methods based on handcrafted or learned features, which have shown to be superior in performance compared to that of the former. In order to combine features extracted by different convolutional layers, a U-shaped process was applied to reshape those features and generate a prediction map in umUNET [8] and U-2NET [4] for pixel-level manipulation detection and localization.

Self-attention modules have been applied in CNN structures to detect or locate image manipulation, as a tool to strengthen the interrelationship among separated information located in different locations or channels of extracted high-level features. Dense position attention modules (PAM) and channel attention modules (CAM) were applied serially on each convolutional block, in order to generate the interaction between image points of the feature map and improve the performance of Xception on face image forgery datasets [9]. CAM and PAM were conducted in parallel as dual-attention module to refine the distance between ground-truth mask and generated manipulated feature mask in MVSSnet [10]. However, those attention modules are applied as self-adaptive feature enhancement on each feature map of convolutional block or on single combined feature. In the proposed method, CAM is extended on five reshaped features of convolutional blocks to adaptively generate relationship among different level of information contained in those features.

Error-level analysis (ELA) has been used as a hand-crafted method in detecting different types of image manipulations. A dense layer and flatten layer are used as a classifier for ELA results of forged face images [11]. Input images were pre-processed by ELA and classified by a simple two-layer or three-layer convolutional neural network, regarding splicing and copy-move respectively [12]. ELA was applied on UNet to achieve pixel-level image forgery detection, and improved the pixel-level F1-score to 0.686 [13]. For above methods, CNNs were used as a classifier for ELA-based input images. In the proposed method, a pre-processing method called signed-value error levels (SEL) is applied to expose image manipulation traces, the local or global features which are then learned and adatptively used by the proposed WCBnet.

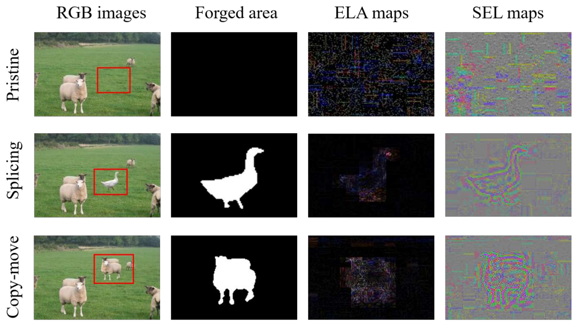


Fig. 2. Three examples of pristine, splicing and copy-move images, and their corresponding ground truth, ELA maps and SEL maps; Forged areas in similar positions are marked with red boxes and zoomed in

III. Proposed method

As shown in Fig. 1, the proposed model consists of five stages. They include 1) Signed-value error-level analysis (SEL) image pre-processing; 2) Leaned feature extractor module; 3) peak-based adjustable feature reshaping process; 4) block-wise feature attention module; 5) classification module. The details of each stage will be introduced in this section.

A. Signed-value error levels (SEL)

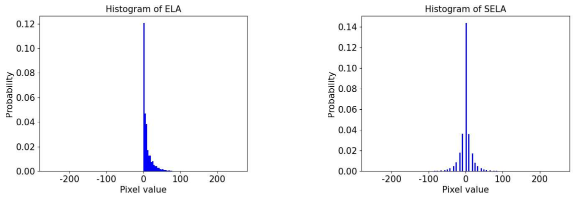
Instead of using RGB images with excessive image features and background distractions as network input, error-level analysis (ELA) is applied as pre-processing to concentrate on the artifacts generated by inconsistent compression process between forged area and background. So the ELA result, extracted from a given RGB input $\mathbf{x}$ of size $W \times H \times 3$, is fed into the learnt feature extractor. Conventionally the ELA $\mathbf{x}'$ of an RGB image $\mathbf{x}$ is computed by taking the absolute difference between the original $\mathbf{x}$ and its corresponding JPEG encoded and decoded image with respect to the compression factor f as follows:

$$\mathbf{x}' = \left| \mathbf{x} - \mathrm{JPEG}(\mathbf{x}, f) \right| . \tag{1}$$

However, the sign information of image difference which might contribute to important features corresponding to manipulations, is ignored in the above conventional ELA calculation. In order to preserve the sign information and enhance those small values, we propose a normalisation approach as follows leading to signed-value error levels:

$$\mathbf{x}' = 255 \times \frac{\mathbf{x} - \mathrm{JPEG}(\mathbf{x}, f)}{\max(|\mathbf{x} - \mathrm{JPEG}(\mathbf{x}, f)|)} . \tag{2}$$

This allows those decimal values that represent manipulation artifacts to be normalized to the maximum range that can be stored in a pixel (-255 to +255). The difference of value distribution between ELA and SEL is shown in Fig. 3(a). Three example images of CASIA2.0, regarding pristine, splicing and copy-move respectively, are compared in Fig. 2.



(a) ELA histogram (b) SEL histogram

Fig. 3. The histograms of ELA and SEL, extracted from the same image

B. Learned feature extractor module

The pre-processed signed-value error levels result $\mathbf{x}'$ is then fed into the any learned feature extractors, in order to extract multi-dimensional information from SEL maps. The common CNNs divide its convolutional layer into several convolutional blocks according to different kernels sizes and conceptional field. In the proposed model, ResNet50 with five convolutional blocks is selected as the learned feature extractor, and the output of each convolutional block $\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, \mathbf{y_4}, \mathbf{y_5}$ with different feature shape $256 \times 256 \times 64$, $128 \times 128 \times 256$, $64 \times 64 \times 512$, $32 \times 32 \times 1024$, $16 \times 16 \times 2048$ are extracted in order to obtain different levels of detailed information and global information in the image. The extracted feature $\mathbf{y_i}$ of convolutional block i of CNN extractor is determined as Equation(3):

$$\mathbf{y_i} = \mathrm{CNN}(\mathbf{x}', i). \tag{3}$$

C. 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:

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

The aim of this repeated reshaping process is to obtain the feature map $\mathbf{y}'_\mathbf{1}$ with the same size as the original image for pixel-level classification and avoid the pooling operation from destroying the image tampering traces. However, for different image tampering types or datasets, the importance of each information level is different resulting in the requirement of different perception domains. We propose an adaptive structure that different reshaping algorithms, instead of repeated transposed-convolution and addition, are performed on each convolution block outputs individually to construct the feature of the target size, as shown in Fig. 4. 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

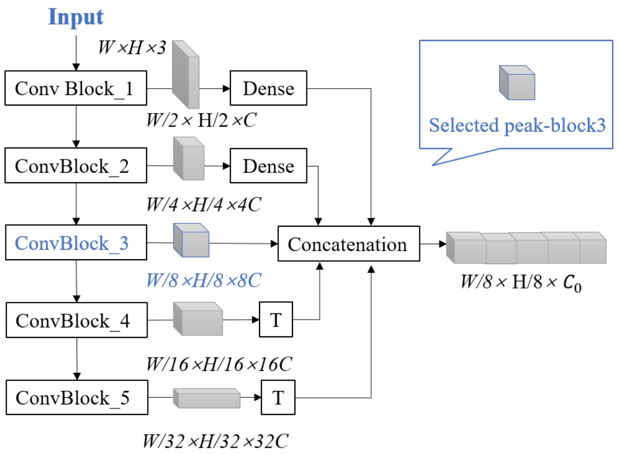


Fig. 4. Proposed Peak-shaped reshaping process

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:

$$\mathbf{y_i'} = \begin{cases} \mathrm{dense}(\mathbf{y_i}, r), & \text{if } i < p, \\ T(\mathbf{y_i}, r), & \text{if } i > p, \\ \mathbf{y_i}, & \text{if } i = p. \end{cases} \tag{5}$$

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 perceptual field. Moreover, in order 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:

$$\mathbf{Y} = \mathrm{Concat}(\mathbf{y_1'}, \mathbf{y_2'}, \mathbf{y_3'}, \mathbf{y_4'}, \mathbf{y_5'}). \tag{6}$$

D. Cross-block attention module (Cross-BAM)

The concatenated feature $\mathbf{Y}$ is then processed by a cross-block attention module, which enables the model adjusting the weights of convolutional blocks according to various manipulation types and digital characteristics of input images. Cross-BAM consists of two core parts, including a self-attention algorithm along channels to generate channel-wise weights for each convolutional block and two fully-connected layers on all weighted channels to obtain the inter-relationship between blocks. The detailed workflow of cross-BAM is as follows:

- The feature $\mathbf{Y}$ of size $W_i \times H_i \times C_0$, concatenated by reshaped block features according to the size of peak block $W_i \times H_i$ (convolutional block i), is globally pooled along the height and width to a vector $\mathbf{A}$with the same length as channel number of $\mathbf{Y}$;
- The vector $\mathbf{A}$ of size $1 \times 1 \times C_0$ is connected to a ReLu activation layer, in order to generate trainable weights for all convolutional block channels;
- Two consecutive dense layers with different neuron factors are then applied to compute the channel-wise inter-relationship, in other words, to comprehensively combine weighted multi-level information of all convolutional blocks. Two different dense layers are used to generate non-linear connection among channels, compared to linear single dense layer.
- Finally the element-wise multiplication of block-wise inter-related weights $\mathbf{A}$ and the input feature $\mathbf{Y}$ is computed as feature $\mathbf{Y'}$, to assign the vector weights to the block output features in corresponding channels. The parameters in cross-BAM are updated during training, in order to change the model's utilization of different convolutional block outputs according to the properties and manipulation types of the training dataset.

TABLE I

THE COMMON IMAGE FORGERY DATASETS USED IN THE EXPERIMENTS.

Manipulation	Dataset	Image size	Image number (forged/pristine)
Copy-move, splicing	CASIA1.0 [14]	400×400 to 300×400	921/800
	CASIA2.0 [14]	900×600 to 160×240	5063/7491
	Defactos [15]	640×426 to 240×320	124000/
Splicing	Columbia [16]	1152×768 to 757×568	183/180
Copy-move	Coverage [17]	752×472 to 253×340	100/100
	NC2016 [18]	4329×3240 to 640×480	1124/847

E. Classification layer

After the block-wise attention module, a global average layer and a dense layer with Sigmoid activation map the weighted feature $\mathbf{Y'}$ to a single value of size $1 \times 1 \times 1$ which indicates the possibility of the input image belonging to the positive class (pristine). Based on the most possible predicted class and given ground-truth image-level label, binary cross-entropy loss is calculated batch by batch and back-propagated to update weights of model parameters along the direction of loss drop.

IV. EXPERIMENTS

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.

A. Experimental setup

a) 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.

b) 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 summarised in Table I.

c) 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 datset CASIA2.0, and the other image manipulation datasets are used for

TABLE II
ABLATION STUDY OF PEAK-SHAPED FEATURE RESHAPING, WHERE THE DATA IN THE TABLE REPRESENTS F1-SCORE

	Channel 15			Channel 21			Channel 30			Channel 45			Customized Channel		
	Spling	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed
Peak1	0.941	0.922	0.893	0.931	0.890	0.929	0.931	0.924	0.939	0.926	0.914	0.933	0.929	0.920	0.940
Peak2	0.943	0.930	0.936	0.946	0.929	0.909	0.948	0.926	0.939	0.938	0.926	0.928	0.940	0.920	0.910
Peak3	**0.968**	0.934	0.771	0.952	**0.945**	0.941	0.953	0.918	0.905	0.924	0.930	0.937	**0.961**	**0.960**	**0.949**
Peak4	0.951	0.933	**0.941**	0.951	0.879	0.912	0.955	0.910	**0.947**	0.940	0.926	**0.951**	0.953	0.939	0.908
Peak5	0.959	**0.938**	0.936	**0.972**	0.916	**0.952**	**0.955**	**0.957**	0.936	**0.956**	**0.944**	0.941	0.962	0.946	0.912

TABLE III
ABLATION STUDY OF PEAK-SHAPED FEATURE RESHAPING, WHERE THE DATA IN THE TABLE REPRESENTS AUC

	Channel 15			Channel 21			Channel 30			Channel 45			Customized Channel		
	Spling	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed	Splicing	Copymove	Mixed
Peak1	0.991	0.989	0.982	0.993	0.987	0.989	0.989	**0.989**	0.992	0.990	0.987	0.991	**0.991**	0.989	**0.990**
Peak2	**0.995**	0.993	**0.994**	**0.995**	**0.991**	0.991	**0.993**	0.988	**0.994**	**0.991**	**0.991**	0.992	0.989	0.992	0.985
Peak3	0.990	**0.993**	0.920	0.991	0.986	**0.993**	0.990	0.985	0.989	0.989	0.990	**0.993**	0.983	0.991	0.989
Peak4	0.983	0.985	0.980	0.988	0.971	0.978	0.984	0.982	0.981	0.980	0.989	0.981	0.983	0.985	0.986
Peak5	0.981	0.986	0.985	0.987	0.980	0.983	0.989	0.982	0.970	0.984	0.987	0.977	0.989	**0.994**	0.981

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 ELA and SEL is set to 90 as an empirical choice.

d) Evaluation metrics: Image-level accuracy, F1-score, and AUC (area under curve) are selected as the evaluation metrics for forged-or-pristine binary classification task.

B. Ablation studies

Ablation studies, including signed-value error levels and peak-shaped feature reshaping, were conducted to show the effectiveness of the core components of WCBnet and optimize its structure.

a) 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 II and III, 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 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.

b) Effectiveness of signed-value error levels: The effectiveness of SEL is evaluated by comparing the performance of WCBnet on RGB, ELA and SEL inputs of three forged subsets of CASIA2.0. As shown in Table IV, the most obvious difference is that all error-level-based inputs are superior than RGB input. More precisely, the classification accuracy of model with ELA 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 ELA 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.

TABLE IV
PERFORMANCE OF WCBNET ON THREE SUBSETS OF CASIA2.0, REGARDING THREE DIFFERENT INPUT TYPES

Subset	RGB			ELA			SEL		
Type	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

TABLE V
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 [19]	0.297	0.141	0.000	0.543	0.621	0.681	-	-	0.484	0.491
CR-CNN [20]	0.361	0.766	0.397	0.567	0.392	0.783	0.939	-	0.131	0.566
FCN-16 [21]	0.775	0.796	0.458	0.551	0.481	0.762	-	-	0.180	0.541
MVSS-Net [10]	0.752	0.839	0.404	0.573	0.802	0.980	-	-	0.244	0.731
umUNET [8]	0.565	-	-	-	0.761	-	-	-	0.423	-
U2-Net [4]	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

C. Performance comparison with the state-of-the-art methods

The performance of proposed WCBnet is compared to several state-of-the-art methods, including MantraNet [19], CR-CNN [20], FCN-16c̃itelong2015fully, MvssNet [10], Modified-CNN [8] and U2-Net [4], via two evaluation metrics that are AUC and F1-scores. The result is shown in the Table V. The proposed WCBnet achieves the best performance on CASIA1.0 with highest AUC of 1.000 and highest F1-score of 1.000, Defactos with highest AUC of 0.818 and highest F1-score of 0.801, Coverage with highest F1-score of 0.646 but fourth highest AUC, Columbia with highest F1-score of 0.863 and second-best AUC of 0.950 among all the state-of-the-art models, and was also ranked second best on the NC2016 dataset achieving 0.751 for F1-score. The results indicate that, in the face of several datasets with different characteristics and manipulation methods, the proposed model can adapt its structure to distinguish well between those forged images and pristine images.

D. Discussion and limitations

The three core components of proposed WCBnet, namely SEL, Peak-based reshaping and Cross-block attention, have been proven to optimize the model performance over different image manipulation types and datasets, compared to results of RGB or ELA inputs, fixed-shape reshaping and block feature-fusion modules. But the limitation of the current WCBnet is that it works comparatively poorly on small-scale and fine post-processed image manipulation datasets, such as CMFDdb and Coverage with only dozens of images. Those specific datasets are also challenging for other state-of-art methods, on which WCBnet still achieves the best results (F1-score of 0.646 on Coverage, higher than the second performance of ManTra-Net by 0.162) but does not achieve enough satisfactory results.

V. Conclusions

In this paper, we propose a novel approach for detection of manipulated images. Main core components of WCBnet include signed-vale error level analysis (SEL), Muti-level learned feature extraction, individual feature reshaping and a cross-block attention module. The novel cross-block attention module in WCBnet adapts to different image manipulation types and dataset variables, by making adaptive use of the multi-level information extracted by convolutional blocks.

The performance of WCBnet shows superior performance on commonly-used image forgery datasets and proven to be resulting the best performance in terms of f1-scores compared to those of the state-of-the-art methods showing improvements ranging from 6.1% to 34.3 % with respect to the second-best methods. The use of SEL showed superior model performance by 23% and 4%, compared to RGB-based and conventional ELA-based results, respectively. The proposed block-wise attention module consisting of self-generated weights and non-linear interrelation achieved the best performance on RGB, ELA, and SEL input modes for all subsets of CASIA2.0 (87.07% on average classification accuracy and F1-score of 0.879).

References

[1] L. Verdoliva, “Media forensics and deepfakes: an overview,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[2] 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.

[3] 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.

[4] T. Zecheng, W. Xinyuan, Y. Hongli, and Z. Yansong, “U2-Net for image forgery detection and localization,” in *2021 International Conference on Computer Technology and Media Convergence Design (CTMCD)*, 2021, pp. 166–172.

[5] D. Bhowmik, A. Natu, T. Ishikawa, T. Feng, and C. Abhayaratne, “The jpeg-blockchain framework for glam services,” in *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2018, pp. 1–6.

[6] 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.

[7] 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.

[8] 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 *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, 2021, pp. 126–131.

[9] H. Lin, W. Luo, K. Wei, and M. Liu, “Improved xception with dual attention mechanism and feature fusion for face forgery detection,” in *2022 4th International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2022, pp. 208–212.

[10] 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*, 2022.

[11] Q. ul ain, N. Nida, A. Irtaza, and N. Ilyas, “Forged face detection using ELA and deep learning techniques,” in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, 2021, pp. 271–275.

[12] W. P. Sari and H. Fahmi, “The effect of error level analysis on the image forgery detection using deep learning,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2021.

[13] N. K. Hebbar and A. S. Kunte, “Image forgery localization using U-Net based architecture and error level analysis,” in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 1992–1996.

[14] J. Dong, W. Wang, and T. Tan, “CASIA image tampering detection evaluation database,” in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.

[15] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. Marc, “DEFACTO: Image and face manipulation dataset,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[16] Y.-F. Hsu and S.-F. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency,” in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 549–552.

[17] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “COVERAGE — a novel database for copy-move forgery detection,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 161–165.

[18] 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 *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 63–72.

[19] Y. Wu, W. AbdAlmageed, and P. Natarajan, “ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9535–9544.

[20] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, “Constrained R-CNN: A general image manipulation detection model,” in *2020 IEEE International conference on multimedia and expo (ICME)*. IEEE, 2020, pp. 1–6.

[21] 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.