Detecting Image Manipulation in Lossy Compression: A Multi-modality Deep-Learning Framework

Vijayakumar Kadha
Dept. of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, India
vijayakumar_kadha@nitrkl.ac.in

Santos Kumar Das
Dept. of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, India
dassk@nitrkl.ac.in

***Abstract*—Due to the advancement of photo editing techniques, it has become easier to create fake photos that look incredibly realistic and are edited in a way that leaves no visible signs of manipulation, making them ideal for synthesis. However, Instagram, WeChat, and TikTok are some of the popular social media platforms where the images have been lossy compressed before uploading them. As a result, learning to spot forged images in their compressed form is crucial. As part of this, some forensic detection techniques have made great strides in uncompressed scenarios, but there is still much to learn about the forensics of lossy compressed images. Therefore, this research proposes a hybrid deep learning framework by dissecting compressed and manipulated images at the preprocessing and feature extraction levels. The suggested noise stream progressively prunes the texture information to prevent the model from fitting the compression noise. Hence, a noise stream is employed to extract temporal correlation characteristics to address the potential problem of ignoring temporal consistency in lossy compressed images. Further, residuals from two streams are fed to custom ResNet blocks to enhance the clues of manipulation and pooled to concatenate the enhanced fingerprints. Finally, the proposed method outperforms state-of-the-art techniques in identifying manipulation in lossy compressed images.**

***Index Terms*—Digital image forensics, Double compression, Manipulation detection, lossy compression**

I. Introduction

Nowadays, people have long had access to global news through various social networks, and more time is spent online by individuals than in actual social interactions due to rapidly advancing communication and computing. However, the data collected in the social network environment is unreliable. As Artificial intelligence advances, expect to see more and more examples of manipulated media content shared across social media platforms, some for entertainment and some for impaired individuals [1], [2]. Generally, manipulated images that have been compressed are now commonly used on social media. This is because high-quality picture transmission speeds suffer without sufficient network bandwidth. Therefore, when a user uploads an image to a social media platform, it automatically reduces the image's file size. Moreover, users of social media apps like WeChat and Instagram are forced to repeatedly compress and reupload images because of the apps' size limits. Hence, forgers distribute bogus images using compression, reducing forensic investigators' ability to identify them as forgeries [3], [4]. To tackle this issue, forensic analysis of compressed forged images is crucial for addressing the ”seeing is not believing” problem. Hence, it is important to consider real-world instances where a forensic investigator lacks background knowledge regarding the system's operating settings.

Towards this goal, many handcrafted features are widely used for picture modification detection [5]–[7]. However, these characteristics are able to identify specific types of manipulation. On the other hand, researchers have explored using Convolutional Neural Networks (CNNs) to learn alteration attributes due to their robust learning capability and generalizability [8], [9]. Contrary to popular belief, CNNs are not designed to learn manipulation features (such as tamper artefacts) but rather image content information. Hence, some solutions have been reported to address this issue, such as the Steganalysis Rich Model (SRM) [5] and constrained convolutional layers [10] that remove image texture information while keeping manipulation traces. However, without additional image semantic information like compression artefacts or noise, these methods struggle to detect manipulation regions accurately in compressed scenarios. Therefore, bridging the gap for manipulation identification in the presence of lossy compression is challenging due to the fact that lossy compression will erase manipulation clues, deteriorating further detection performance. To address the issue mentioned earlier, we developed a multi-modality framework to identify manipulation in lossy compression scenarios. The key contributions of the article are put briefly.

- We propose a novel multi-modality framework for detecting image manipulation, which is robust against different kinds of manipulation.
- To improve the performance of highly lossy compressed images, we introduced Discrete Cosine Transform (DCT) residuals in the proposed framework.
- Finally, state-of-the-art performance is achieved on JPEG and non-JPEG manually manipulated photos using our two-stream system, as shown by substantial testing findings.

The remainder of the article is organized as follows. Recent literature on image manipulation detection is discussed in Section II. Next, the proposed method using a multi-modality framework is represented in Section III. In Section IV, experimental results and ablation studies are reported. Finally, concluding remarks are discussed in Section V.

II. Related work

In this section, we look back at recent related work for detecting and pinpointing tampering using CNNs. For several image manipulation operations, general-purpose detection techniques have been developed [10]–[12]. These methods have proven they can autonomously learn picture editing features from data. To identify the numerous image manipulation operations while hiding the image texture information, [10] proposes a unique restricted convolutional layer-based CNN, and [13] refines and improves upon this network. In addition, a densely connected CNN is reported for use in general-purpose visual forensics [11]. As part of this, high-pass filtering with the isotropic convolutional layer makes the artefacts of image processing procedures stand out. Further, in [14], a method for detecting picture modification is provided based on [10] and employs a deep siamese CNN network. Instead, they focused on determining whether or not a given set of input patches (two photos) had been similarly processed. On the other hand, classifying numerous image processing tasks while considering small-sized images is the focus of [15], where the Xception architecture is used. However, the present general-purpose forensic methods are limited to identifying manipulations in uncompressed scenarios. Recently, in [16], the authors offer a generalized manipulation detection method that relies on the multi-scale residual module CNN and considers most of the manipulation operations, including a wide range of anti-forensic methods. On the other hand, most of the above methods are considered in un-compressed scenarios, and further, their performance degrades significantly in lossy compression which is the most realistic scenario. In addition, detailed recent studies on manipulation detection are also mentioned in Table. I.

TABLE I: Recent related work on several manipulations detection techniques

Method	Clue/Feature	Manipulation parameters					Remarks
		Blurring		Noise	Resampling	Compression	
		MF	GB	AWGN	RS	JPEG	
[13]	Constrain CNN	✓	✓	✓	✓	✓	L_3
[12]	RGB + SRM				✓	✓	L_2, L_3
[17]	Visual + Compression	✓	✓	✓	✓	✓	L_2, L_2
[15]	Magnified layer	✓	✓	✓	✓	✓	L_1, L_3
[16]	Multi-scale residual module	✓	✓	✓	✓	✓	L_1, L_3
[18]	Noise residual	✓	✓	✓	✓	✓	L_1, L_2

Remarks: L_1: Computationally expensive, L_2: Performance to detect multiple manipulations is less, L_3: Difficult to detect manipulation in compressed scenario.

Several restrictions are visible, as reported in the literature: First, most forensic methods only detect one kind of manipulation, and second, most detectors were designed for uncompressed images, thus their performance suffers greatly in the lossy compressed scenario. Hence, we proposed an end-to-end network to classify the various manipulation and operator chains as illustrated in Table II in light of the aforementioned drawbacks.

III. Proposed Method

The proposed network includes enhancing manipulation traces, learning features with CNN, and further manipulation is predicted using the classification stage.

TABLE II: Total type of manipulations considered in this article

Alteration type	Notation	Parameters
Median filtering	MF	$3 \times 3, 5 \times 5, 7 \times 7$
Gaussian bluring	GB	$3 \times 3, 5 \times 5, 7 \times 7$
Additive White Gaussian Noise	AWGN	$\sigma = 0.5, 1.5, 2.0$
Resampling (Bilinear)	RS	0.6, 0.8, 1.2, 1.5

A. Overview

An end-to-end framework is proposed to detect manipulations in re-compressed images and the detailed architecture and its blocks with multi-modalities utilized in pre-processing steps are illustrated in Fig. 1. The proposed method uses three distinct modules in sequence: initially, a pre-processing module is utilized with SRM filters [5], constrain CNN layer [10], and DCT residuals are concatenated to improve tampering traces. Secondly, five Residual blocks are utilized to learn discriminating features to enhance manipulation clues rather than image content. Finally, a classification module is used to achieve image-wise prediction based on the probability of each class achieved in the last fully connected layer.

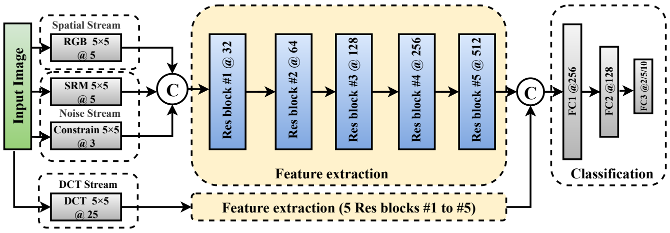

Fig. 1: Archiecture of the proposed framework.

B. Enhancing manipulation traces

From a forensics perspective, the local reliance of pixels with their neighbours is the most crucial detail extracted by this pre-processing layer. To refine an incoming image, we devise a pre-filtering module that uses 5 SRM filters specified in Fig. 2, constrain CNN layer [10], and RGB residuals. Specifically, in [10], the k filter weights individually specified as

$$\omega_k^1(0,0) = -1 \quad and \sum_{x_1,x_2 \neq 0} \omega_k^1(x_1,x_2) = 1 \tag{1}$$

where, $\omega_k^1(x_1,x_2)$ denotes the weight at position (x_1,x_2) of the k th filter and $\omega_k^1(0,0)$ indicates the weight at the middle of the corresponding filter kernel. The process is repeated for each pixel in the patch by shifting the kernels over the image patch. As part of manipulation traces extraction, the input image of size $m \times n \times c$ passes through pre-processing step and generates the features fm_r, fm_c, and fm_s with a size of $m \times n$ with 5, 3, and 5 feature maps respectively. Then each feature is concatenated to represent an enhanced manipulated trace output feature (fm_{o1}) represented as:

$$fm_{o1} = |fm_r; fm_c; fm_s| \tag{2}$$

The final output fm_{o1}, i.e., has a $m \times n$ size with 13 feature maps. Further, it is noted that the filter kernels are made trainable during training to modify their parameters using gradient descent. The noise residual features from

0	0	0	0	0
0	-1	2	-1	0
0	2	-4	2	0
0	-1	2	-1	0
0	0	0	0	0

(i)

-1	2	-2	2	-1
2	-6	8	-6	2
-2	8	-12	8	-2
2	-6	8	-6	2
-1	2	-2	2	-1

(ii)

0	0	0	0	0
0	0	1	0	0
0	0	-2	0	0
0	0	1	0	0
0	0	0	0	0

(iii)

0	0	0	0	0
0	0	0	0	0
0	1	-2	1	0
0	0	0	0	0
0	0	0	0	0

(iv)

0	0	0	0	0
0	0	0	1	0
0	0	-2	0	0
0	1	0	0	0
0	0	0	0	0

(v)

Fig. 2: Five SRM kernels are utilized in noise stream

the concatenated block are given as input for the feature extraction stage, as shown in Fig. 1.

However, the purpose of the proposed approach is to detect manipulation in the JPEG domain. Hence, as a preliminary step in extracting transform features, we devise DCT residuals with a size of 5×5, which helps to exact the clues left by lossy compression. In the JPEG realm, these residuals and their variation [19] are effective characteristics for JPEG picture steganalysis. We believe that these residuals can effectively identify clues left by manipulations. Therefore, these DCT basis patterns are a pre-processing convolutional stage that outputs twenty-five residual maps from decompressed JPEG pictures.

$$B_{pq}^{(i,j)} = \frac{w_i w_j}{5} \cos \frac{\pi k(2p+1)}{10} \cos \frac{\pi l(2q+1)}{10} \tag{3}$$

where $0 \leq i,\ j \leq 4$, $0 \leq p,\ q \leq 4$, w_x is defined as follows

$$w_x = \begin{cases} 1, & x = 0 \\ \sqrt{2}, & 1 \leq x \leq 4 \end{cases} \tag{4}$$

Decompressing a JPEG input image of size $m \times n$ yields I_{mn} in spatial domain. After convolving I_{mn} with $B(i,j)$, we obtain twenty-five residual maps $R(i,j)$ as shown below.

$$R(i,j) = I * B(i,j) \tag{5}$$

Finally, 25 feature maps (fm_{o2}) are generated and given to the second stream with a similar feature extraction stage.

C. Learning feature with CNN

Once the pre-processing stage collects efficient and discriminative features from the images to identify altered from unaltered images. The five ResNet blocks that comprise the planned feature extraction module are divided into two ”bottleneck” branches. Every bottleneck consists of a batch normalisation and ReLu activation step, followed by three consecutive convolutional layers and an identity skip connection. Three convolutional layers have kernel sizes of 1×1, $3{\times}3$, and $1{\times}1$, respectively and stride is 1, excluding the final layer in second branch of each block, which has a stride of 2 for pooling and decreasing size in the spatial domain specified in Fig. 3. In keeping with ResNet's default configuration, we set the channel depth of the first two convolutional layers to equal one-fourth of the depth of the final convolutional layer (output depth). We begin with a depth of 128 for the first block's output and increase it by 128 for each successive block. The 13-channel and 25-channel residuals are fed into a feature extraction module, where 512 feature maps are learned with a spatial and transform domain of the original image concatenated to form 1024 feature maps and fed to the classification stage.

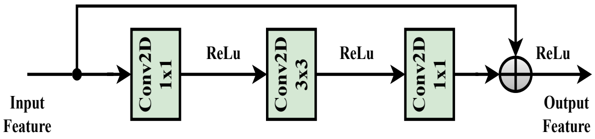

Fig. 3: Residual block used in proposed network.

D. Performing manipulation prediction

Finally, to predict various manipulations, three fully connected layers (f_{c1}, f_{c2}, f_{c3}) are utilized in the classification stage. For better prediction, 256 and 128 nodes in f_{c1} and f_{c2}, respectively and final layer nodes are taken based on the experiment, such as 2 for single manipulation, 4 for manipulation classification, and 10 for operator chain classification. The proposed model trained for 50 epochs with an initial learning rate (lr) of 0.001, and it is decreased with a factor =0.5 for every 5000 iterations. Also, for the single manipulation experiment, we used binary cross-entropy as a loss function; for the other studies, we used categorical cross-entropy. All experiments use the Adam optimizer [20] as their optimisation method to get the most stable and quick convergence.

IV. Experimental Results & Discussions

Several experiments are performed to validate the proposed approach's ability to identify manipulations, considering various alterations in lossy JPEG compression scenarios.

A. Experimental Setup

All the experiments of the proposed network are built in the Pytorch framework using GPU as the backend. This has been accomplished using a desktop computer outfitted with an NVIDIA Quadro P6000 24 GB GPU. The preprocessing was done in Python using the OpenCV package. All the experiments were done by utilizing the RAISE [21] dataset with 8,156 images of various sizes utilized for training and DRESDEN [22] with 25,137 colour images of various sizes used for testing. In addition, images in both datasets are in TIFF format. Finally, the manipulated synthetic dataset is created with different manipulations as shown in Table II.

B. Performance of the proposed framework

Since filtering modification operators can alter an image's aesthetic appearance while preserving its overall structure and content, they are frequently utilised in the creation of forged photographs. Hence, the generality of the proposed framework was accessed by conducting the experiments with

TABLE III: Detection performance (%) of the proposed method for single manipulation in lossy compressed images with primary quality factor $QF_1 = 90$ and secondary quality factor randomly chosen $QF_2 \in (50, 60, 70, 80, 90)$

	Image Blurring						Noise addition			Resampling (Bicubic)			
	MF @ 3	MF @ 5	MF @ 7	GB @ 3	GB @ 5	GB @ 7	AWGN @ 0.5	AWGN @ 1.5	AWGN @ 2	RS @0.6	RS @0.8	RS @1.2	RS @1.5
MISLNet [13]	97.48	95.75	96.72	96.05	97.16	96.75	99.55	99.36	99.85	96.16	97.58	98.35	**98.99**
MCNet [17]	96.58	98.32	**98.96**	95.24	94.00	95.25	97.25	98.56	98.24	95.28	94.95	97.36	97.64
MDRNet [18]	98.24	98.05	98.64	98.58	98.12	98.95	**99.99**	**99.99**	**99.99**	97.14	**98.15**	98.25	97.96
Proposed	**98.95**	**98.84**	98.86	**99.05**	**98.96**	**98.98**	**99.99**	**99.99**	**99.99**	**98.96**	98.12	**98.95**	98.10

various filter parameters specified in Table II in three scenarios: single manipulated, multiple manipulated and series of manipulations in the JPEG compression scenario.

1) Single manipulation detection: Initially, the significance of the proposed framework is accessed to identify whether the image is altered or not. In this experiment, four different manipulations are considered with different manipulation parameters and a total of 13 experiments are conducted. Specifically, 13 databases were created, each consisting of 50000 image patches, including 25000 altered and 25000 unaltered patches. To create each patch; initially, 512 × 512 size un-processed patch is taken from RAISE [21] and compressed with a quality factor ($QF_1 = 90$). Further, it is altered with one of the manipulation types followed by a randomly chosen secondary quality factor ($QF_2 = 50, 60, 70, 80, 90$). Finally, a centrally cropped patch with a size of 256 × 256 is generated to form an altered patch. Similarly, an unaltered patch with a size of 256 × 256 is generated to form a second class. Later, the proposed model is trained with an 80:20 split ratio (i.e. 40000 for training and 10000 for validation) for 50 epochs as specified in Section III-D.

Once the model is trained, to test the effectiveness of the proposed model, we took 10000 patches from the DRESDEN dataset [22] with 5000 altered and 5000 unaltered. Further, the performance is noted in terms of accuracy metric and present in Table III. Similarly, for all other types of manipulations, corresponding performance is measured with the same experiment settings. For comparative analysis, we considered MISLNet [13], MCNet [17], and MDRNet [18] with the same settings, and the corresponding mean accuracy is noted in Table. III. It is to be noted that the proposed method outperforms the early reported techniques in the lossy compression scenario.

2) Multiple manipulation detection: In this experiment, multiple manipulations are considered, where most of the social networking platforms are utilized such as Gaussian blur, median filtering, and resampling. Hence, these processes can leave artefacts or abnormalities that can be noticed and analysed to determine image integrity using the proposed method. To conduct this, the proposed framework with 4 neurons in the f_{c3} layer is assumed and corresponds to 3 altered and one unaltered class. Following the lead of the first experiment, we extracted 100000 patches from the RAISE dataset [21] and used them to train our model with the output layer activated using the softmax activation function. Later, the proposed model is trained with an 80:20 split ratio (i.e. 80000 for training and 20000 for validation) for 50 epochs as specified in Section III-D. For testing, we used 5000 patches from the DRESDEN [22] for each class and tested on a total of 20000 patches to determine how well the proposed network performed in a mismatch scenario. As can be shown in Fig. 4, the suggested network achieves a greater level of accuracy when it comes to GB (94.60%), MF (96.48%), original (97.10%), and RS (96.94%) alterations in lossy compressed images. In addition, the loss curve for detecting multiple manipulations with proposed framework is depicted in Fig. 6.

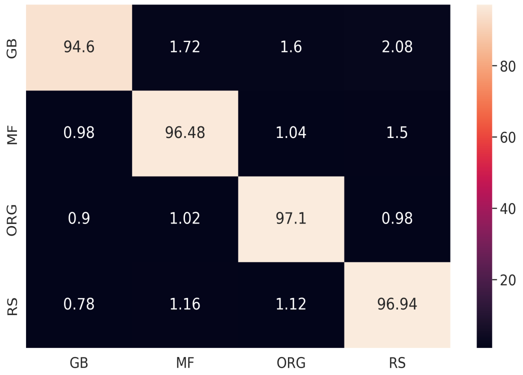

Fig. 4: Confusion matrix to identify multiple manipulations (i.e JPEG-manipulation-JPEG) using the proposed framework.

3) Operator chain classification: Finally, an unavoidable scenario is considered, when multiple modifications have been made to the same image, many services try to act like social networking software by re-compressing the final product before sharing it. To demonstrate this, a third set of experiments are conducted to see how a well-proposed network can spot the image's altered history after JPEG lossy compression. Prior to being lossy compressed with a quality factor of 90, each image patch experienced a sequence of up to two distinct modifications. The last fully connected layer comprises ten nodes describing nine different transformations and one original category. For training, we used 25,000 original grayscale image patches created the same way as before from the RAISE dataset [21]. Following this, the image patches are blurred using a variety of techniques, such as MF, GB, and bicubic interpolation to resize (RS) specified in Table II. Each manipulation could have up to two such operations which are denoted by $"1^{st} manipulation \rightarrow 2^{nd} manipulation"$. Further, 250000 patches with 25000 patches of each class are created from RAISE [21] to train the proposed framework. Later, the proposed model is trained with an 80:20 split ratio (i.e. 200000 for training and 50000 for validation) for 50 epochs as specified in the previous experiment. Once the model is trained, for testing, 50 TIFF photos are chosen randomly from the DRESDEN dataset [22], and 10 patches are collected with a size of 512 × 512 to form 5000 unaltered patches for this analysis. Next, images are manipulated using the same data

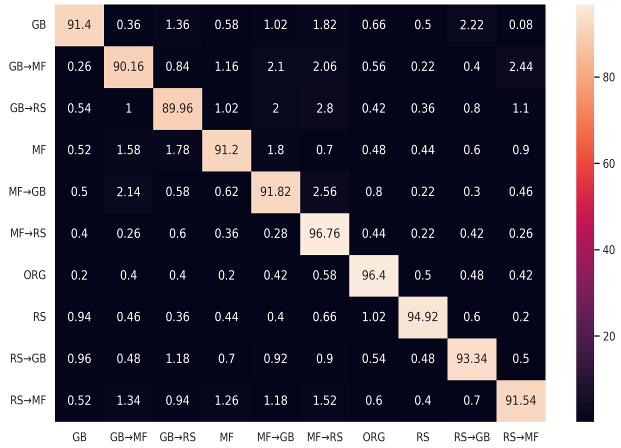

Fig. 5: Confusion matrix to identify operator chain detection (i.e. JPEG-series of manipulations-JPEG) with the proposed framework.

preparation processes mentioned in the previous experiment. For each class, we collected centrally cropped 5000 image sets with a size of 256×256. Finally, the model is tested and a test accuracy of more than 89.96% was achieved by the model across all series of manipulations using photos that were not viewed during training. The performance was checked across various manipulations for a more in-depth examination. In addition, the performance is measured and the corresponding confusion matrix is depicted in Fig 5 and noted that the multi-modality framework achieves better performance, specifically MF followed by RS. In addition, the loss curve for detecting operator chain with proposed framework is shown in Fig. 6.

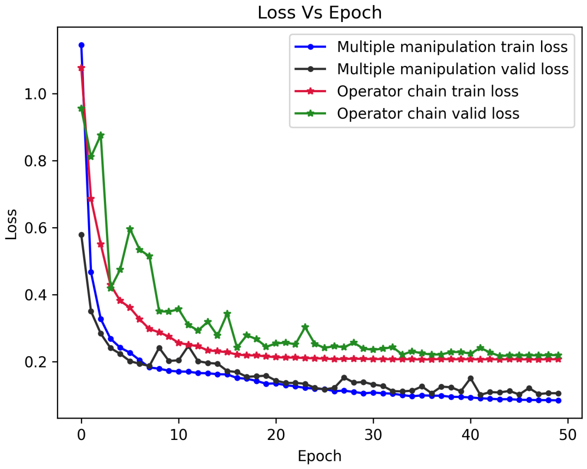

Fig. 6: Loss curve of multiple manipulations (5) and operator chain classification (10) followed by re-compression with $QF_2 \in (50, 60, 70, 80, 90)$.

C. Ablation study

Since the suggested architecture comprises several blocks, including Residual blocks with multi-modality input streams, ablation research is done to determine how each block affects the architecture's performance. Table IV shows several experimental findings for the two datasets such as RAISE [21] & DRESDEN [22]. As part of this, three streams are considered such as Constrain layer, SRM, and DCT residual stream and conducted experiments to find the effectiveness of each stream. It is noted from Table IV that the suggested architecture greatly raised the mean accuracy of all experiments which included the DCT residual stream as compared to other combinations. Hence, it is showing that the DCT module has improved proposed method's performance as demonstrated by the accuracy.

TABLE IV: Ablation study for effectively comparing the proposed network with different modules on two datasets.

	RAISE [21]				DRESDEN [22]			
SRM	✓		✓	✓	✓		✓	✓
Constrain		✓	✓	✓		✓	✓	✓
DCT	✓	✓		✓	✓	✓		✓
Mean accuarcy (%)	94.65	95.54	96.56	97.85	95.48	96.85	97.56	98.95

D. Comparative analysis of the proposed framework with state-of-the-art techniques

As a last test, we used identical experimental circumstances as previously described to determine the mean accuracy of multiple manipulations & operator chain identification in lossy compressed images with state-of-the-art. Fig. 7 reports the comparative analysis of the proposed method with early reported techniques like MISLnet [13], MCNet [17], and MDRNet [18]. In comparison to MISLnet (88.25%), MCNet (88.68%), and MDRNet (86.98%), the suggested method achieves a mean accuracy of 89.48% for heavy lossy compression ($QF_2 = 50$). Similar to the previous experiment with the same settings operator chain mean accuracy is also measured and depicted in Fig. 8. Therefore, the better accuracy will confirm that the proposed network is superior to several existing approaches, especially with regard to lossy compression scenarios.

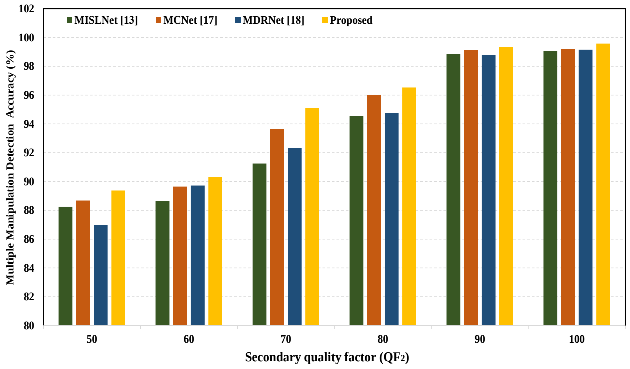

Fig. 7: Compartive analysis to detect multiple manipulations with early reported techniques for different secondary quality factors $QF_2 \in (50, 60, 70, 80, 90, 100)$.

V. Conclusion

This work introduces a novel end-to-end framework that retains numerous tampering traces from noise residuals and DCT residuals in the JPEG domain. To identify image alteration methods, the proposed model combines data indicative of tampering such as DCT residuals and noise features learned by multi-modality with features indicative of artefacts in the image collected by CNN. The noise stream can be

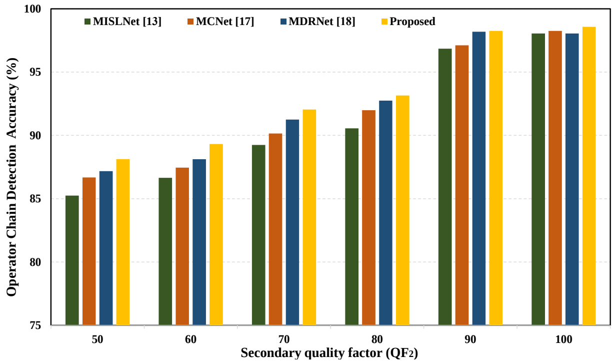

Fig. 8: Compartive analysis to detect operator chains with early reported techniques for different secondary quality factors $QF_2 \in (50, 60, 70, 80, 90, 100)$.

used to restore the integrity of delicate boundary artefacts in fused feature maps that were weakened by post-processing techniques. Extensive experiments show that the proposed network is more effective than early-reported techniques. In addition, our approach is also robust to identifying photos that have been altered through highly compressed or post-processing. In future, it would be ideal to have a global feature extractor that is more effective than pre-processing SRM and contain CNN layers. It will also be interesting to look at new ways of using uncertainty estimation to spot photographs modified outside the norm.

References

[1] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pp. 43–47, 2018.

[2] W. Sun, J. Zhou, R. Lyu, and S. Zhu, "Processing-aware privacy-preserving photo sharing over online social networks," in *Proceedings of the 24th ACM International Conference on Multimedia*, pp. 581–585, 2016.

[3] H. Wu, J. Zhou, J. Tian, and J. Liu, "Robust image forgery detection over online social network shared images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13440–13449, 2022.

[4] 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)*, pp. 101–117, 2018.

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

[6] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2444–2447, IEEE, 2011.

[7] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, "WLD: A robust local image descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2009.

[8] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2016.

[9] N. Huang, J. He, and N. Zhu, "A novel method for detecting image forgery based on convolutional neural network," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, pp. 1702–1705, IEEE, 2018.

[10] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *ACM Workshop on Information Hiding and Multimedia Security*, pp. 5–10, 2016.

[11] Y. Chen, X. Kang, Z. J. Wang, and Q. Zhang, "Densely connected convolutional neural network for multi-purpose image forensics under anti-forensic attacks," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pp. 91–96, 2018.

[12] 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*, pp. 1053–1061, 2018.

[13] 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.

[14] A. Mazumdar, J. Singh, Y. S. Tomar, and P. K. Bora, "Universal image manipulation detection using deep siamese convolutional neural network," *arXiv preprint arXiv:1808.06323*, 2018.

[15] L. Yang, P. Yang, R. Ni, and Y. Zhao, "Xception-based general forensic method on small-size images," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceedings of the 15th International Conference on IIH-MSP*, pp. 361–369, Springer, 2020.

[16] K. Rana, G. Singh, and P. Goyal, "MSRD-CNN: Multi-scale residual deep CNN for general-purpose image manipulation detection," *IEEE Access*, vol. 10, pp. 41267–41275, 2022.

[17] I.-J. Yu, S.-H. Nam, W. Ahn, M.-J. Kwon, and H.-K. Lee, "Manipulation classification for JPEG images using multi-domain features," *IEEE Access*, vol. 8, pp. 210837–210854, 2020.

[18] V. K. Kadha, P. Deshmukh, K. C. Rayasam, and S. Kumar Das, "Robust manipulation detection scheme for post-JPEG compressed images using CNN," in *2022 IEEE 19th India Council International Conference (INDICON)*, pp. 1–6, 2022.

[19] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1200–1214, 2017.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *ACM Multimedia Systems Conference*, MMSys '15, (New York, NY, USA), p. 219–224, Association for Computing Machinery, 2015.

[22] T. Gloe and R. Böhme, "The'dresden image database' for benchmarking digital image forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 1584–1590, 2010.