Supplementary Appendix

A1. Content Suppression Modules

In order to improve detection of the subtle forensic features and suppress the spatial content of the image, we add additional modules to the encoder's first layer that extract noise level features. For this purpose, we introduced four modules – i) the `SRMConv` [12] layer, ii) the `BayarConv` [1] layer, iii) the classic convolution layer termed as `RGBConv`, and iv) our proposed Error Level Analysis (ELA) Module. Fig. 1 shows the output of applying SRM and ELA on a tampered image.

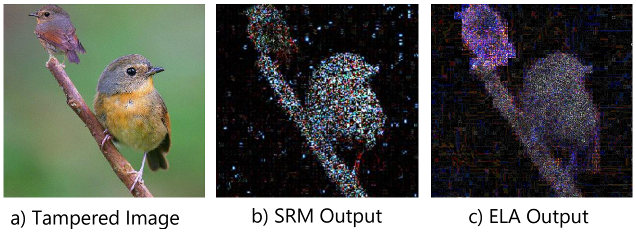

Figure 1. Result of SRM filters and ELA on a tampered image.

ELA has previously been used for localizing compression artefacts from JPEG images [9]. It works by comparing the pixel-wise difference between an image and its compressed copy. If an image contains pixels from a different source, then the pixels of the two sources would produce different levels of compression noise. We propose to use this ELA output as a feature for the encoder. We take an input image and compress it with a reduced 90% compression factor. Then we calculate the difference between the original and the compressed image to generate the ELA output. This output ELA image is then passed through a series of convolution layers before applying activation to produce the ELA feature map.

To evaluate the effect of these modules on the encoder, we compare the detection accuracy on the CASIAv2 validation set in Table 1. We can see that the choice of the first layer affects model performance to a large amount. The proposed ELA module has a notable effect as it improves encoder accuracy by a factor of more than 3%. So, for our final encoder, we select a combination of the four layers. The input images pass through all of them simultaneously, then the outputs are concatenated and sent to the backbone. This additional compression and steganalysis feature helps the network to detect the traces of the boundary regions. Moreover, the encoder becomes more robust to post-processing operations as it learns to detect and correlate the multi-domain artifacts with other spatial features.

1st Conv Layer	#Filters, Kernel Size	Encoder Accuracy (%)
`RGBConv`	16, k=(3,3), p=1, x2	84.61
`SRMConv`	3, k=(5,5)	86.77
`BayarConv`	3, k=(5,5)	85.25
ELA Module	32, k=(3,3), p=1, x2	87.03
Combined	54, -	**88.25**

Table 1. Results of using additional feature extraction layers for the 1st encoder layer with an EfficientNet-B4 backbone. The results compare only the encoder detection accuracy for an image-level binary classification test on the CASIAv2 validation set.

A2. Additional Ablation Experiments

A2.1 Block Positions

In Section 4.3 we had talked about the effects of placing the GCA block in different positions within the network. Fig. 2 shows these placement positions. The blue squares represent the encoder layer, green circles are the decoder nodes, and the red rectangles denote the GCA block.

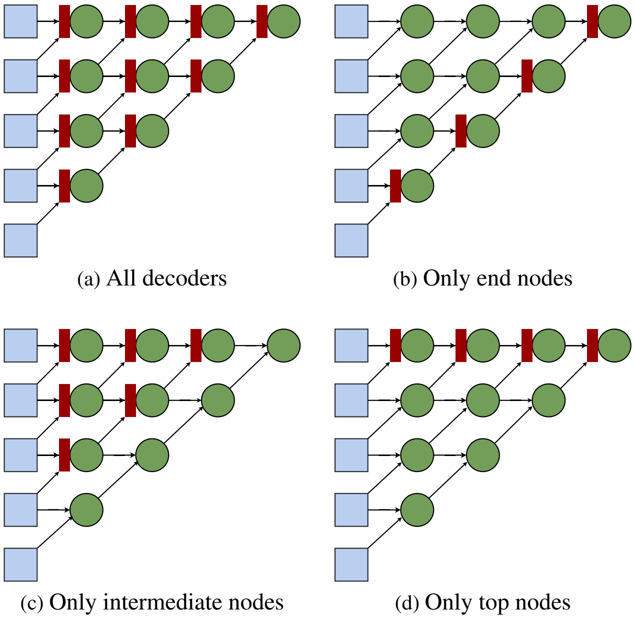

Figure 2. Different positions of placing the GCA block. Blue squares are encoder layers, Green circles are decoder nodes, and Red rectangles represent the GCA block.

A2.2 Backbone Choice

There are no dominant network architectures proven to be useful for IFLD tasks. XceptionNet has been shown

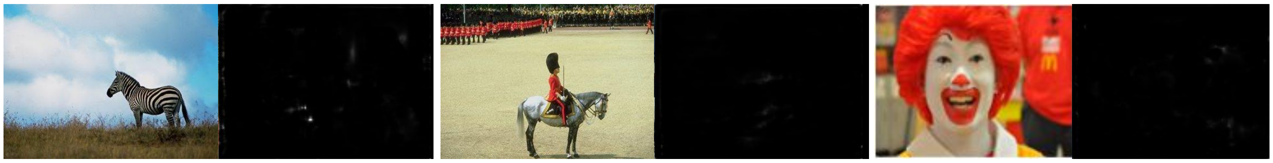

Figure 3. Localization for the three authentic images previously shown in Fig. 2 of the paper. Since groundtruth masks for pristine images are blank, they are not shown here. GCA-Net predicts almost blank masks for authentic images with minimum false positives.

Model	#Params (M)	Encoder Accuracy (%)
XceptionNet [2]	22.86	78.03
DenseNet-161 [5]	28.68	83.56
ResNeXt-50 [11]	30.42	82.29
SEResNeXt-50 [4]	27.56	85.81
EfficientNet-B4 [8]	19.34	**87.65**

Table 2. Baseline detection accuracy of different architectures for image-level binary classification on CASIA validation set.

to perform well for DeepFake detection, and media forgeries [7]. DenseNet also showed promise in detecting camera model features [6], which has relevant implications for manipulation identification. We test multiple such backbone networks to test their efficacy for manipulation detection. We trained and tested these baseline models using the CASIAv2 [3] dataset. Since we are evaluating the encoder performance only, we perform these tests as a classification task without the decoder and compare the image-level detection performance. From Table 2 we see that EfficientNet performs the best. Additionally, it uses an inverse bottleneck convolution with channel attention making it the lightest of all the networks with only 19.34 million parameters.

A3. Implementation Details

In order to tackle the challenge of low data and improve generalizability, all images were augmented using Flipping, Random Rotations, Optical and Grid Distortions, and Gaussian Blur, each with a probability of 30% - 50%. We trained the model with the encoder pre-loaded with Imagenet weights, using Adam optimizer with a learning rate of 0.00001 and a weight decay of 0.00005. Learning rate scheduling was done using Reduction on Plateau by a factor of 0.25. All models were trained for 60 epochs and with Early-Stopping patience of 20 epochs. The model was implemented using PyTorch. For the EfficinetNet backbone we used the implementation from Timm models [10].

References

[1] 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*, 2018. 1

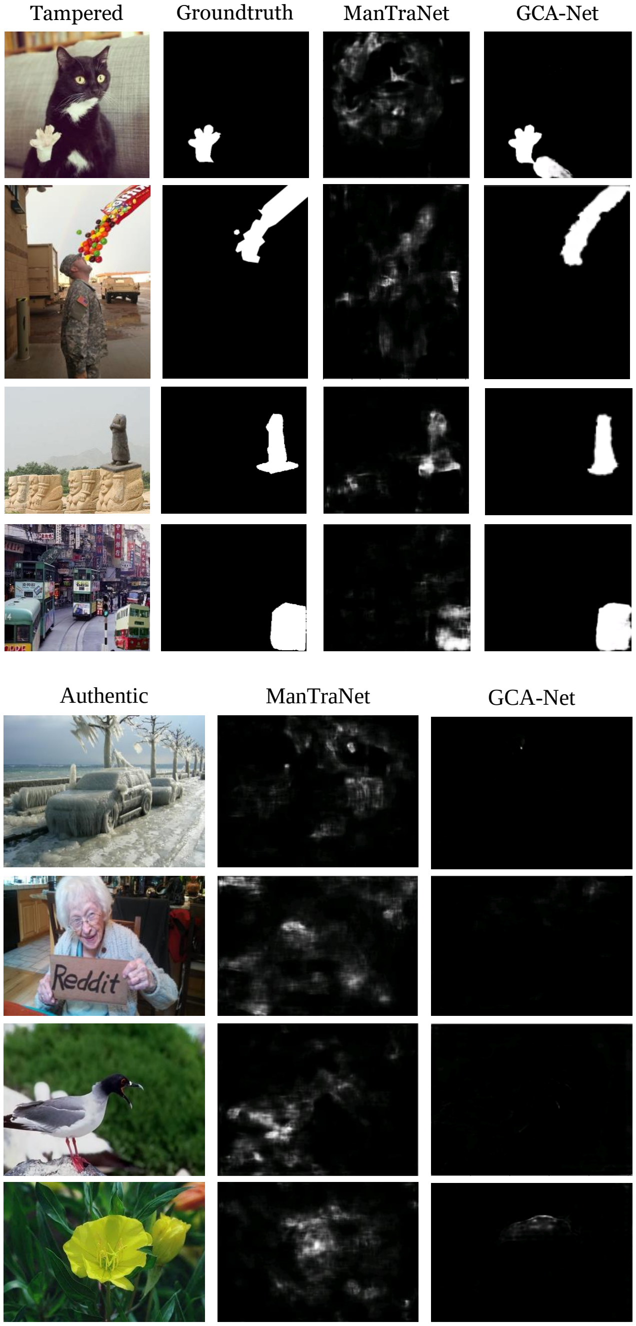

Figure 4. Qualitative comparison of GCA-Net and ManTraNet for various tampered and authentic images.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017. 2

[3] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. pages 422–426, 07 2013. 2

[4] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019. 2

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. 2

[6] Abdul Muntakim Rafi, Uday Kamal, Rakibul Hoque, Abid Abrar, Sowmitra Das, Robert Laganière, and Md. Kamrul Hasan. Application of densenet in camera model identification and post-processing detection, 2019. 2

[7] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images, 2019. 2

[8] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 2

[9] N. B. A. Warif, M. Y. I. Idris, A. W. A. Wahab, and R. Salleh. An evaluation of error level analysis in image forensics. In *2015 5th IEEE International Conference on System Engineering and Technology (ICSET)*, pages 23–28, 2015. 1

[10] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 2

[11] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017. 2

[12] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection, 2018. 1