U-Net Approach for Image Manipulation Localization

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Image manipulation localization is not the same as semantic object segmentation which pays more attention to the image content, whereas localization of manipulation has to concentrate more on manipulated artifacts in the image. Local distribution of image pixels will be different near manipulated boundaries compared to non-manipulated regions. The Coefficient of Variation (COV) is used to enhance the manipulated artifacts by exploiting the local distribution feature. COV value is calculated for a 3x3 sliding window applied on the manipulated image which results in COV-image. Only COV-image will not suffice for manipulation localization so COV-image is superimposed with the manipulated image which results in the modified image. We propose a U-Net architecture with ResNet-34 blocks initialized to encoder part is trained end to end to localize the manipulated region given a modified image as input. With the modified image rather than the plane manipulated image, our model is able to see manipulated artifacts better and localize manipulated regions. Our model produces promising results on four standard image manipulation datasets and better than state-of-the-art performance with robustness to jpeg compression.**

***Index Terms*—U-Net, Coefficient of variation, ResNet-34, COV-image, Modified image.**

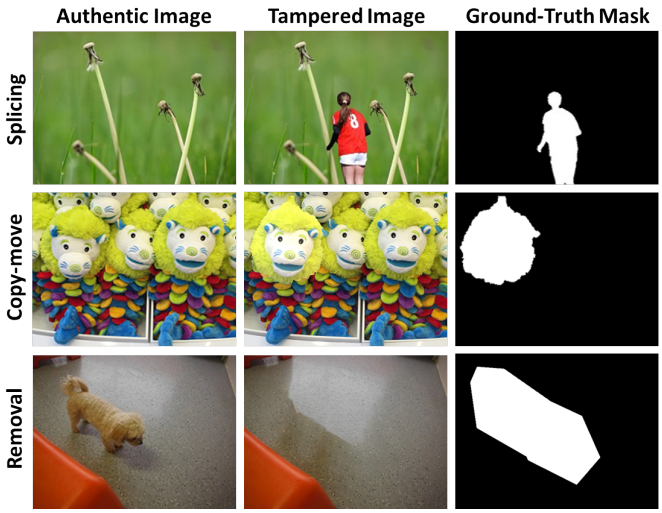

Fig. 1. Different tampering techniques.

I. Introduction

In the era of digitalization, visual information is carried by digital images. In a day to day life images are found everywhere. Images are involved in various fields like medicine, crime, sports, journalism, etc., where authenticity is most important. Many image editing tools are available at low-cost or free of cost for editing images. Manipulation of images from some tools can be done to such an extent where it is very difficult to identify if that image is manipulated or not by human eyes. So image manipulation detection is a very challenging research area. Image manipulation can be classified into two different categories, one is content preserving and another is content changing. Content preserving manipulation does not change the semantic meaning of image content. Post-processing is the main reason for content preserving manipulation. Some of the examples are *compression*, *blurring*, etc, and considered less harmful as it preserves the image content. Content changing manipulation techniques, shown in Fig 1, involves *splicing*, *copy-move*, *removal* are the most common manipulation techniques. These techniques reshape image content arbitrarily and alter the semantic meaning significantly.

Image splicing splices the desired object from an authentic image and pastes it on to other images. Copy-move splices the desired object and pastes it to the same image. Removal removes the region from the original image and then cover it by inpainting. Previous works [1], [2] on image localization needs large computing resources and trained on huge data that takes a lot of time to train the model. Without proper resources, it becomes difficult to address this problem and to come up with a general model that localizes all different types of manipulation. The goal of our work is to show that even with less computing power and less data we are able to perform pixel-level localization of all different types of manipulations. Our model structure is also not complex, which takes less time to train and also gives better results than the existing models. We propose a simple and efficient U-Net [3] architecture with ResNet-34 [4] blocks initialized to encoder

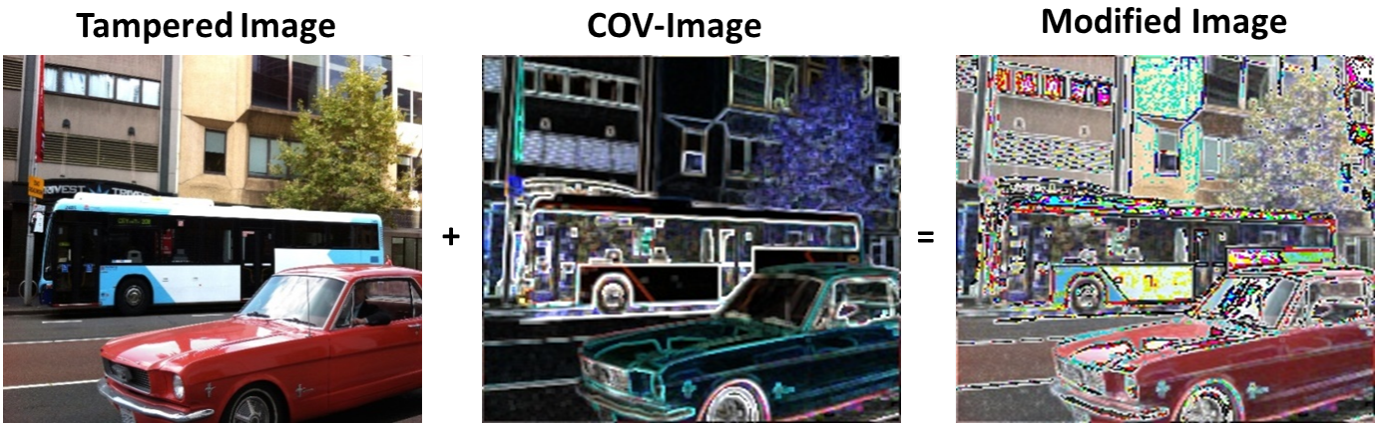

Fig. 2. Creation of modified image

part localizes various types of manipulation given a modified image as input.

II. Related Work

Image forensics research shows that there are many different approaches. Error level analysis (ELA)[5] tries to find the difference between jpeg compression of manipulated images and authentic images to find the compression error as evidence of tampering. Noise inconsistency (NOI)[6] is evident which is used by the high pass wavelet coefficient.

In past years we were able to solve many difficult tasks such as object detection, object segmentation, etc. using deep learning techniques. Even in image manipulation detection and localization, there are many contributions. Copy-move and splicing localization using Sobel filter attached to mask regional convolutional neural network[7] which acts as task auxiliary to encourage predicted masks. Many works focused on building a general model that is capable of localizing all types of tampering. Stacked LSTM[8] included between convolution layers taking image patches as input to find manipulation artifacts over the boundaries of manipulated patches and non-manipulated patches. The frequency-domain correlation between boundaries of manipulation is analyzed by Hybrid LSTM [9] encoder and decoder network. Resampling features from LSTM combined with encoder-decoder architecture, which has Skip pooling [10], is designed to identify the manipulation across boundaries by taking image patch as input. All the [8]–[10] works concentrate more on manipulated boundaries by analyzing images at the patch level. RGB-N [1] is a two-stream network that detects the region of manipulation using both RGB features as well as noise features that are given to the Faster R-CNN network. MT-Net[2] has two sub-networks, one for manipulation feature extraction another for localizing, and also claims about localizing various types of manipulation.

III. Proposed Model

We propose the U-net architecture [3], that is trained end to end, the U-net architecture is composed of two parts encoder which results in lower dimension feature maps, and the decoder results in masks of the same size as input. The blocks of ResNet-34[4] are used in the encoder.

The modified image is the result of the manipulated image superimposed on the COV-image and used as an input. To find the COV-image sliding window of 3x3 is applied to the manipulated image. Finally, the model is trained end to end with modified images to localize the manipulated regions in the image.

A. COV-Image

Every image has its own distribution of pixels. There are various types of manipulations like copy-move, splicing, and removal. If we employ any type of manipulation to the image, there will be some distortion or variation to the original distribution of pixels. These manipulation artifacts can be found in the manipulated image at the boundaries of manipulated regions which is not visually apparent. Coefficient of Variation (COV) is used to enhance such manipulated artifacts which are not perceptible.

$$COV = \frac{\sigma}{\mu} \tag{1}$$

where, σ= standard deviation μ= mean. COV value given by the sliding window of size 3x3 on the manipulated image gives the degree of variation in that particular window. The values are stacked horizontally as well as vertically to get the COV-image. COV-image will represent the degree of variation around neighboring pixels within the window.

B. Modified Image

For the model to localize image manipulation at the pixel level, COV-image alone will not be sufficient. Information to model has to be rich enough to see through manipulation artifacts in the image. This can be achieved by superimposing the manipulated image with the COV-image to get the modified image. As shown in Fig. 2, the manipulated image is added to the COV-image, and the resulting values are clipped to 0-255 range because image pixel value ranges from 0-255.

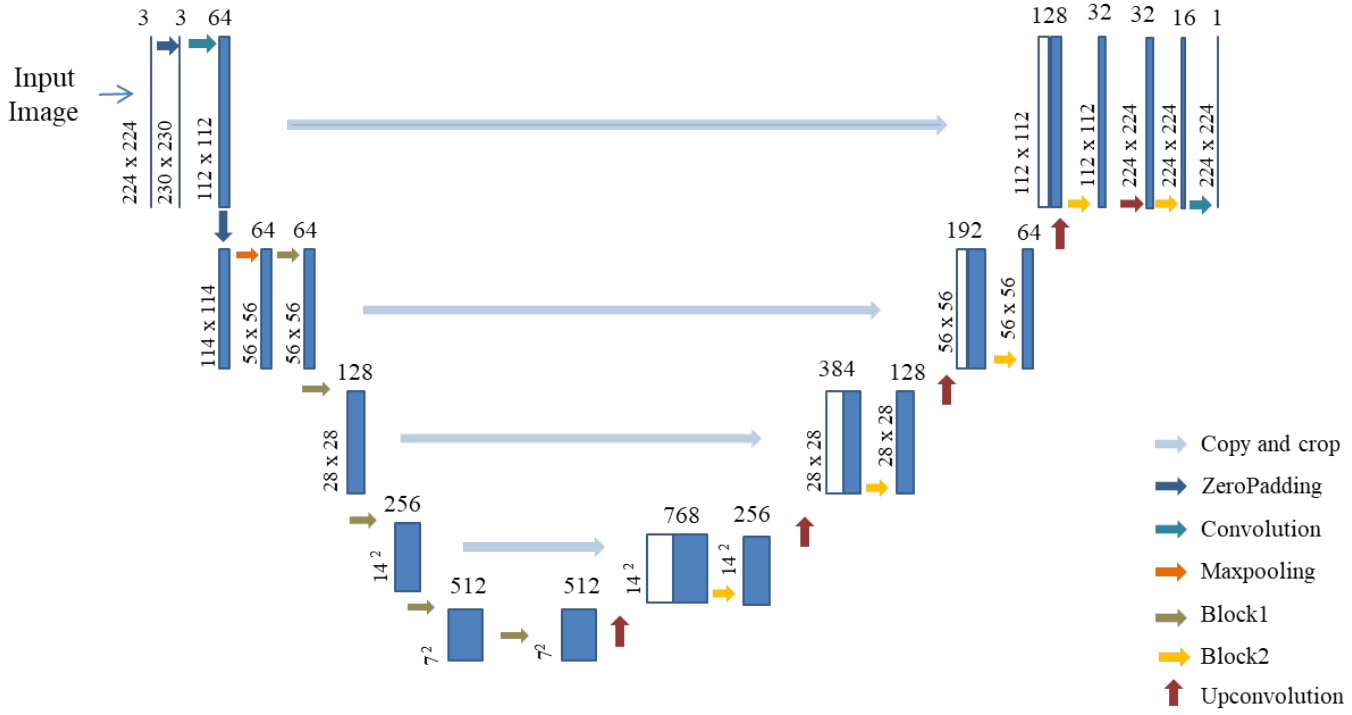

Fig. 3. Network Architecture

C. Network Architecture

The standard U-Net[3] is used mainly for segmentation applications. Here we are posing the image manipulation localization as a single manipulated segmentation problem where we encode each non-manipulated pixel as 0 and the manipulated pixel as 1. The U-Net architecture can be viewed as two parts Encoder and Decoder as shown in the Fig. 3.

D. Encoder

The encoder is the contracting path that follows the typical convolutional neural network architecture that takes the images of size 224×224 as input and results in a feature map of 7×7×512. We have initialized the layers of standard ResNet-34[4] architecture in the encoder part of our model. The skip connections of the ResNet-34[4] help the model to have a better gradient flow. In the encoder, we use 4 identical blocks of "Blocks 1" which is shown in Fig 4. Block 1 is a series of layers arranged in such a way that first we concatenate the output of the Block 1.1, a 3×3 Conv layer, and then pass it as input to Block 1.2 plus a skip connection. Block 1 ends with applying the activation function of the Rectified Linear Unit (ReLU). All the convolution operations in Block 1 have 3×3 kernel with stride 1 which is the same as ResNet-34[4].

E. Decoder

The decoder is an expansive path that consists of four successive operations of up-convolutions followed by Block 2. The decoder gets the output generated by encoder 7×7×512, and each up-convolution operation doubles the width and height of the feature map, keeping the same number of filters. The output of the up-convolutions is concatenated with the corresponding block of the encoder, i.e. the first up-convolutions is concatenated with the output of the 4th "Block 1" and the output of the 2nd up-convolutions is concatenated with the 3rd "Block 1" and so on. The concatenated output is then passed to the "Block 2" in which we apply the sequence conv-bn-relu two times, the output generated from the block 2 will have the same height and width as input except for the number of filters that gets reduced. The last layer of the decoder is the convolution layer with sigmoid activation whereas the rest of the network has ReLU activation function. The output of the decoder will have the same dimensions as the input image with one filter i.e. greyscale image.

F. Training loss

The loss of the network is calculated by both focal loss and dice loss. Total loss = dice_loss + focal_loss.

$$focal\_loss = gt\alpha(1-pr)^{\gamma}\log(pr) - (1-gt)\alpha pr^{\gamma}\log(1-pr) \tag{2}$$

where, gt - ground truth, pr - prediction, alpha (α) - The weighting factor set to 0.25, gamma (γ) - focusing parameter for modulating factor (1 - p), set to 2.0.

$$dice\_loss = \frac{(1+\beta^2)tp}{(1+\beta^2)fp + \beta^2 fn + fp} \tag{3}$$

where, tp - true positives, fp - false positives, fn - false negatives, beta (β) - integer coefficient set to 1.

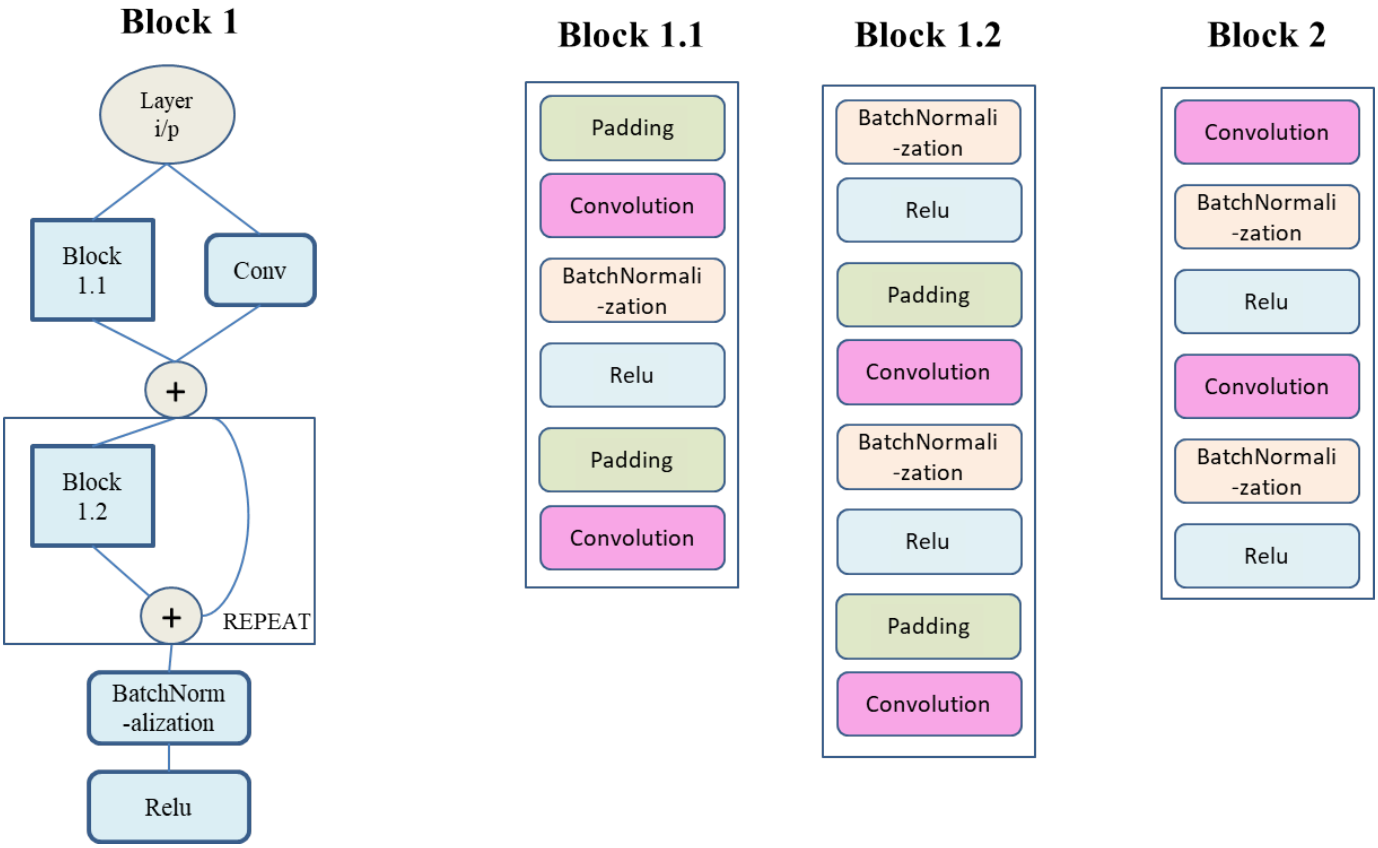

Fig. 4. ResNet-34 blocks

G. Implementation details

To implement the architecture, Tensorflow and Segmentation models are used. The adaptive learning rate (Adam) optimization rule is applied to optimize the model. The learning rate is set to 0.003 which is fixed. Model is trained for 100 epochs in a batch of 8 images using single NVIDIA 1050 card which takes 0.2 seconds per image. Other methods[1] use more advanced computing resources as well as 110K epochs to train their models.

IV. Experiments

The proposed model is demonstrated on four standard image manipulation datasets. The results are compared with existing state-of-the-art methods. We show how our model behaves towards data segmentation and stability of results. The robustness of the model to jpeg compression is also shown.

A. Pre-trained Model

To train the Deep learning models we need an enormous amount of data. Standard datasets do not have enough data to train the network. So synthetic data created by[9] is used. They created synthetic data by extracting objects from MS-COCO[11] to tamper the images of NIST16 [12](non-manipulated images), Dresden [13] datasets. A total of 65k manipulated images were created. A sample of 35k images is used to train our network end-to-end for 100 epochs. 80% of data is used to train and 20% of the data to test the model.

TABLE I
F1-score comparison on various datasets

	NIST16	Coverage	Columbia	CASIA1.0
ELA[5]	0.236	0.222	0.470	0.214
NOI1[6]	0.285	0.269	0.574	0.263
RGB-N[1]	0.722	0.437	0.697	0.408
U-Net	0.765	0.632	0.840	0.454
M-U-Net	**0.833**	**0.610**	**0.869**	**0.602**

B. Datasets

Four Standard image manipulation datasets are considered and compared with existing state-of-the-art methods. NIST16[12], Columbia[14], Coverage[15], CASIA1.0[16]. For all datasets, 75% used for training, and 25% used for testing.

- NIST16[12]: Dataset contains all three types of tampering which are slicing, copy-move, and removal. It is considered as challenging dataset. Ground truths are provided. The total number of images is 564.
- Columbia[14]: Splicing is mainly focused in this dataset. Ground truths are provided. The total number of images is 180.

TABLE II
AUC-SCORE COMPARISON ON VARIOUS DATASETS

	NIST16	Coverage	Columbia	CASIA1.0
ELA[5]	0.429	0.583	0.581	0.613
NOI1[6]	0.487	0.587	0.546	0.612
LSTM-EnDec-Skip[10]	0.857	-	-	0.814
RGB-N[1]	0.937	0.817	0.858	0.795
MT-Net[2]	0.795	0.819	0.824	0.817
U-Net	0.887	0.851	0.928	0.756
M-U-Net	**0.962**	**0.854**	**0.937**	**0.843**

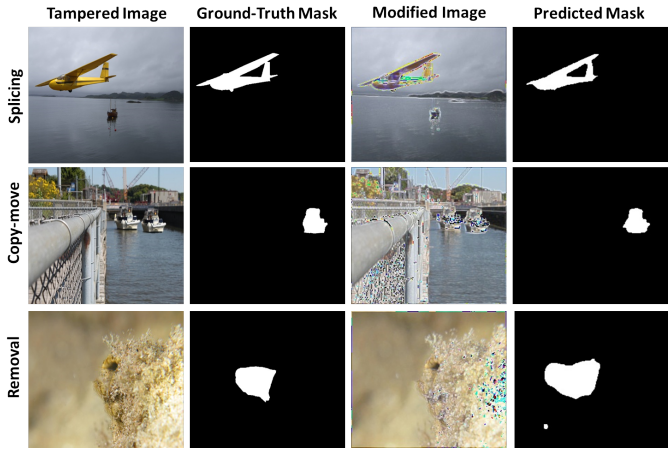

Fig. 5. NIST16 results

- Coverage[15]: Copy move is mainly focused in this dataset. Ground truths are provided. The total number of images is 100.
- CASIA1.0[16]: Both splicing and copy-moved images are present in this dataset. Ground truths generated by namtpham is used. The total number of images is 921.

C. Evaluation metric

Pixel level Area Under the receiver operating characteristic Curve(AUC) and F1 score are used as metrics which is the same as [1]. The maximum resulting F1-score threshold is considered.

We compare our model with various other models which are listed in Table I and Table II. Two variants of our model are listed. One model takes the manipulated image U-Net as input another model M-U-Net which is proposed model takes the modified image as input. Results of ELA, NOI1 is obtained from [1] work. LSTM-EnDec-Skip, RGB-N, MT-Net results are taken from literature. – represents results not available in the literature.

D. Stability of Model

The stability of the model is measured by repeating experiments several times. The results shown in Table I and Table II are obtained by taking an average of ten trials of repeated experiments.

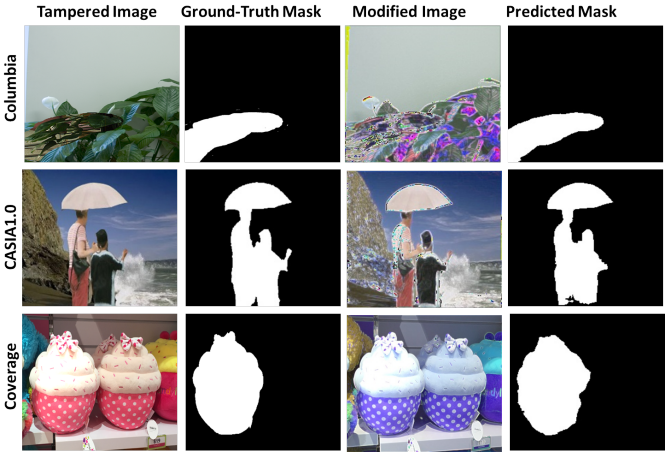

Fig. 6. Columbia, CASIA1.0, Coverage results

E. Qualitative Analysis

Results of experiments is shown in Fig 5 and Fig 6. Fig 5 consists of NIST16 results on various types of manipulations. Results of Columbia, CASIA1.0, Coverage is shown in Fig 6.

F. Data Augmentation

Only the image flipping method considered for data augmentation. The results with flipping and without flipping are shown in Table III

TABLE III
DATA AUGMENTATION RESULTS

F1/AUC	NIST16	Coverage	Columbia	CASIA1.0
None	0.859/0.944	0.621/0.786	0.844/0.919	0.552/0.731
Flipping	0.815/0.962	0.610/0.851	0.869/0.937	0.602/0.843

G. Robustness of Model

Model robustness is measured by reducing the quality (70,50) of images using jpeg compression. NIST16 data is exploited and compared with other method is shown in Table IV.

TABLE IV
ROBUSTNESS RESULTS

F1/AUC	100	70	60
RGB-N	0.722	0.677	0.677
M-Unet	0.833	0.823	0.820

H. Limitations

The proposed model doesn't do well for IEEE IFS-TC data. It produces 0.610 AUC score which is less compared to LSTM-EncDec AUC score 0.757. The model is fine-tuned for each and every standard dataset considered. Whereas MT-Net[2] will not do any fine-tuning, a pre-trained model is

directly used to predict on various datasets. The model cannot handle different shapes and sizes of input because output at various levels of the encoder is concatenated with decoder stage output which cannot happen for all different shapes and sizes of input.

I. Conclusion

We propose a novel M-U-Net that is capable of localizing various types of manipulations. Change in local distribution of pixels is extracted through COV-image. The modified image is formed by superimposing manipulated image with COV-image which is fed to model to get the results. Skip connections in U-Net, as well as ResNet-34, help model to train better. The above experiments show that our model is capable of localizing various manipulations even with less training data and less complex network structure. Thus, it can be said that with more training data and more complex architecture one can attain even better results.

REFERENCES

[1] 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.

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

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[4] 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.

[5] N. Krawetz and H. F. Solutions, "A picture's worth," *Hacker Factor Solutions*, vol. 6, no. 2, p. 2, 2007.

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

[7] X. Wang, H. Wang, S. Niu, and J. Zhang, "Detection and localization of image forgeries using improved mask regional convolutional neural network," *Mathematical biosciences and engineering: MBE*, vol. 16, no. 5, pp. 4581–4593, 2019.

[8] 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.

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

[10] G. Mazaheri, N. C. Mithun, J. H. Bappy, and A. K. Roy-Chowdhury, "A skip connection architecture for localization of image manipulations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 119–129.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[12] N. Nimble, *Datasets*, 2016.

[13] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

[14] T.-T. Ng, J. Hsu, and S.-F. Chang, "Columbia image splicing detection evaluation dataset," *DVMM lab. Columbia Univ CalPhotos Digit Libr*, 2009.

[15] 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)*, IEEE, 2016, pp. 161–165.

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