IMAGE INPAINTING DETECTION USING LAW'S TEXTURE FEATURE EXTRACTION

B. Pradeep Kumar[1], Assistant professor, ECE, K. Nagadurga[2], K. Neeraja[3], P. Geethika[4], K. Padmasri[5]
Department of Electronics and Communications Engineering, Vignan's Institute of Engineering for Women, Visakhapatnam, India

ABSTRACT
Image inpainting is the process of filling in missing or damaged portions of an image. Traditional approaches to image inpainting typically involve handcrafted algorithms that are computationally less expensive than deep learning-based methods. One such approach is the LAW method, which involves dividing the image into smaller patches and then using a local average to fill in missing information. The method also employs a warping technique to ensure that the filled-in information is consistent with the surrounding image. While LAW may not always produce the most visually pleasing results, it is a reliable and efficient method that requires less effort than deep learning-based methods. This makes it a useful tool for tasks where speed and simplicity are of greater importance than achieving the highest levels of visual fidelity.
LAW-based inpainting works by decomposing the image into its wavelet coefficients and replacing the missing or damaged regions with local approximations using a weighted combination of neighboring coefficients. This approach is relatively simple and requires less computational effort compared to deep learning techniques. However, it may not always provide the same level of accuracy and realism as deep learning methods.
In this abstract, we provide an overview of the LAW-based image inpainting method and its advantages as a traditional approach. We also highlight the limitations of this method and the potential for further research to improve its performance.

Index Terms—Feature Extraction, Image processing, Texture.

INTRODUCTION
Image inpainting is filling in missing or damaged areas of an image. It can also be employed to unlawfully charged or edit photographs. Determining pictures in painting has consequently emerged as a crucial challenge in the realm of computer vision. The Laws method, a texture analysis methodology that divides an image into many frequency and orientation components, is one method for spotting image in painting. This technique has shown good results in identifying in painting in photos, although its precision and effectiveness may be constrained. Traditional methods have been used to improve the efficiency of the Laws approach in identifying picture in painting to get beyond these constraints.
However, in painting techniques can also be used to conceal or remove unwanted information from an image, leading to potential misuse such as forgery and tampering. To detect such manipulations, texture-based methods can be employed. One such method is Local Binary Pattern (LBP), which is widely used for texture feature extraction. Another method, which has

shown promising results in inpainting detection, is Laws' Texture Energy Measures. Laws' method decomposes an image into a set of spatial frequency and orientation filters, providing a more comprehensive representation of texture information in an image. In this way, Laws' texture feature extraction method can be used for inpainting detection and forensic analysis of images.

One of the challenges in image inpainting is detecting whether an image has been in painted, as this can be used for various applications such as image forensics and detecting manipulated images. Traditional methods for image inpainting detection often rely on hand-crafted features, such as Local Binary Patterns (LBP) and Local Phase Quantization (LPQ), which can be time-consuming to compute and may not always capture the texture information in an image accurately.

However, a method known as LAW's texture feature extraction has been shown to be effective in detecting image inpainting. This method involves decomposing an image into a set of texture images using Laws' masks, which are a set of predefined filters. The texture images are then used to compute texture features such as energy, entropy, and contrast, which are used as inputs to a classifier to detect whether an image has been in painted or not.

Deep learning methods have shown remarkable success in various fields, including image processing and inpainting, due to their ability to learn features automatically from large datasets. These methods can capture complex patterns and textures in an image that may be difficult to capture using hand-crafted features. Moreover, deep learning methods can adapt to different types of images and inpainting scenarios, making them more flexible and versatile compared to traditional methods [19].

However, deep learning methods require large amounts of annotated data and computational resources to train the models, which may not always be available or feasible in certain applications. Traditional methods, such as LAW's texture feature extraction, may be more suitable for scenarios where limited data or resources are available, or when a quick and easy-to-use solution is required.

Overall, the choice between traditional methods and deep learning methods for image inpainting detection depends on the specific application, available resources, and desired performance. Both methods have their strengths and weaknesses, and it is essential to evaluate them carefully to determine the most appropriate approach for a given task.

a) Original Image b) Tampered Image c) Detection

Fig. 1. (a) The original images; (b) The tampered images where the key objects/watermarks are removed/replaced by the DL-based inpainting methods respectively; and (c) The output of LAW’s Method by using (b) as input.

RELATED WORK

There is still no comprehensive method to handle extremely large masked and complicated images**.** These are algorithms performed in order find out the tampered mask from an image.

HP-FCN can achieve improved performance compared to existing methods, particularly for small inpainting regions that are difficult to detect. However, the method also has some limitations, including its dependence on the shape of the inpainting region, its large training data requirements, and its limited robustness to changes in input image quality.

3D/TSValgorithminimageinpaintingtechniqueareitisdifficulttoimplement for large images, The algorithm is computationally expensive, it is difficult to control the reconstruction quality.

The mean filter is the equal influence of all pixels in the kernel, even very noisy ones. Truncating, or 'trimming', the distribution before taking the mean,by removing some proportion (usually called α) of the largest and smallest values, is a simple way of ensuring that extreme local values do not influence the output. The portion to be truncated varies between 0% (equivalent to the mean) and 100%(equivalent to the median).

It shows the effect of trimming only the maximum and minimum values in two passes of a 3 × 3support (α = 25%). Note that, in general, multiple passes of a small support are approximately equivalent to a single pass of a larger support, so two passes of a3 × 3 support give about the same result as one pass of a 5 × 5 support, threepassesarelike7 ×7, and so on.

IMAGE INPAINTING METHODOLOGY

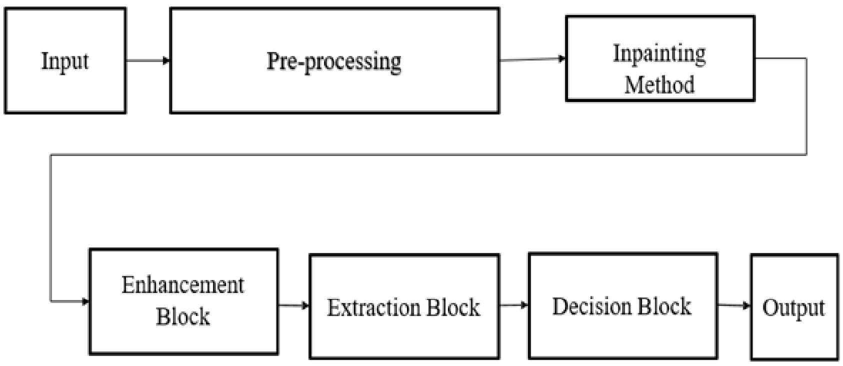

Figure 3.1: Block diagram

Input Block: This block is responsible for processing the input image and preparing it for further processing. The input block can involve operations such as resizing, normalization, and data augmentation.

Extraction Block: This block is responsible for extracting relevant features from the input image. This can be done using various techniques such as convolutional Fig.2.1: Base paper Block diagram 8 neural networks (CNNs), auto encoders, or feature pyramid networks. One important component of NAS for image inpainting detection is the extraction block. The extraction block is a sub-network that is responsible for extracting features from the input image.

These features are then used by the subsequent layers of the network to make predictions about the presence of in painted regions. The extraction block typically consists of several convolutional layers followed by a pooling layer. The convolutional layers are responsible for extracting features from the image by performing a series of convolutions and activations. The pooling layer reduces the spatial dimensions of the feature maps and helps to capture more global information about the image. The architecture of the extraction block is an important aspect of NAS for image inpainting detection. The number and size of the convolutional layers, as well as the pooling layer, can be varied to find the optimal architecture for the task. NAS algorithms use techniques such as reinforcement learning, evolutionary algorithms, or gradient based optimization to search for the best architecture.

Enhancement Block: This block is responsible for enhancing the extracted features and preparing them for inpainting. This can be done using techniques such as attention mechanisms, residual connections, or dilated convolutions. One of the approaches used in NAS for image inpainting detection is to incorporate an enhancement block into the network architecture. An enhancement block is a module that takes an input image and enhances its features before passing it on to the subsequent layers of the network. The purpose of the enhancement block is to make the network more robust to image inpainting by improving its ability to distinguish between original and tampered images. In other words, the enhancement block helps the network to identify the subtle differences between a genuine image and one that has been modified using inpainting techniques.

The enhancement block typically consists of one or more layers that perform operations such as convolution, pooling, and normalization to extract high-level features from the input image. These features are then combined with the original input image to produce an enhanced version that is more informative and easier to analyze. The specific design of the enhancement block can vary depending on the NAS algorithm used and the specific requirements of the image inpainting detection task. However, in general, the block is designed to be lightweight and efficient so that it can be easily integrated into the overall network architecture. Overall, incorporating an enhancement block into the neural network architecture can help to improve the performance of the network in detecting image inpainting.

By automatically searching for the optimal architecture with an enhancement block, NAS algorithms can help to accelerate the development of accurate and robust image inpainting detection models.

Decision Block: This block is responsible for deciding which regions of the image need to be in painted. This can be done using techniques such as object detection, semantic segmentation, or edge detection. One of the key components of NAS is the decision block, which is a component that is used to determine which operations or layers to include in the network architecture. The decision block takes as input a set of candidate operations or layers and their corresponding probabilities and outputs a subset of operations or layers to be included in the network architecture. In the context of image inpainting detection, the decision block can be used to select which convolutional layers to include in the network architecture. Convolutional layers are used to extract features from the input image, and different types of convolutional layers can be used, such as standard convolution, dilated convolution, or depth-wise convolution. The decision block can select the appropriate type of convolutional layer based on the characteristics of the input image and the task at hand.

The decision block can also be used to select other components of the network architecture, such as the activation functions, pooling layers, and normalization layers. By selecting the optimal components for each part of the network architecture, the decision block can help to improve the performance of the network on the image inpainting detection task.

Inpainting Method Block: This block is responsible for identifying the tampered part from an image. This can be done using various inpainting methods such as partial convolutions, GAN-based methods, or deep image prior like HPFCN, MAM, etc.., methods. It refers to a specific module or layer within the neural network architecture that is responsible for performing the actual inpainting of missing or damaged image 10 regions. This block typically takes as input a partial or incomplete image, along with additional information such as a mask indicating which parts of the image are missing. The block then processes this input using learned weights and biases to generate a complete or filled-in version of the image, which is outputted as the result. The specific implementation of the inpainting method block can vary depending on the NAS algorithm and dataset being used. Some possible approaches include using convolutional neural network (CNN) layers to learn feature representations of the image, as well as techniques such as generative adversarial networks (GANs) or variational auto encoders (VAEs) to generate more realistic and visually coherent inpainted images. Overall, the inpainting method block is a critical component of NAS for image inpainting, as it directly determines the quality and accuracy of the final inpainted images produced by the neural network.

Output Block: This block is responsible for processing the in painted image and preparing it for output. The output block can involve operations such as denormalization, resizing, and post-processing. It gives the masked image of tampered part of an image as output. The output block takes the feature maps generated by the previous layers and produces a complete image by applying a set of transformations to the feature maps. •The number and type of transformations depend on the specific task and the architecture of the network. In image inpainting, the output block must be able to fill in the missing or corrupted parts of the image while maintaining the overall coherence and consistency of the image. One common approach to designing an output block for image inpainting is to use a series of transposed convolutional layers, also known as deconvolutional layers. These layers can increase the spatial resolution of the feature maps, allowing the network to generate a more detailed output image. Overall, the output block is a critical component of any NAS algorithm for image inpainting detection, as it determines the quality of the output image generated by the network. Therefore, it is essential to carefully design and optimize the output block to achieve the best possible performance on the task at hand. Overall, these blocks work together to form a pipeline for image inpainting detection. The NAS algorithm searches through various combinations of these blocks to find the best architecture for the task at hand. By automating the design process, NAS can save a lot of time and effort in designing effective neural networks for image inpainting detection.

LAW’s TEXTURE FEATURE EXTRACTIONMETHOD

Laws Method:

Law's method is a popular image analysis technique for detecting image inpainting. It is based on the idea that using a few simple filters, images can be decomposed into a set of basic building blocks or texture primitives.

A.G. Law developed the method in 2005, which employs a set of nine filters designed to mimic the response properties of the human visual system. These filters are arranged in a 3x3 grid and oriented in various directions to capture various aspects of texture information.

Law's method can be expressed mathematically as follows:

The Law's method, given an input image I (x, y), entails the following steps:

- Using the 3x3 filter set, decompose the image into its component texture primitives. This is accomplished by converging each filter with the input image to produce a collection of filtered images.
- Calculate the energy of each filtered image by adding the squared pixel values.
- To create a set of feature maps, add the energies of each pair of adjacent filters. There are nine possible adjacent filter combinations, resulting in nine feature maps.
- Calculate the ratio of energy in the feature map corresponding to horizontal and vertical edges (L5L5 and E5E5) to energy in the feature map corresponding to diagonal edges (L5L5 and E5E5) (L5E5). This ratio is used to identify image regions that are likely to have been inpainted.
- To obtain a binary inpainting mask, use the ratio map as a threshold. Regions with high ratio values in the ratio map are likely to have been inpainted, while regions with low ratio values are unlikely to have been inpainted.

Overall, Law's method is a powerful and effective tool for detecting image inpainting, with important applications in forensics and image processing.

IMAGE INPAINTING DETECTION

In this section, we present the IID-Net details for detecting inpainting manipulations, both DL-based and traditional. The algorithm used in this process is the Laws' method for calculating the texture of an image. The Laws' method is a texture analysis technique that decomposes an image into a set of texture energy measures. The method is based on the idea that any texture can be represented as combination of a small set of primitive textures, each of which is defined as a product of one-dimensional spatial filters. In this code, the Laws' method is used to calculate the texture of the original image and the manipulated image. The filters used in the Laws' method are defined as L5, E5, and S5. These filters are combined to create nine two-dimensional filters (L5E5, E5L5, E5S5, S5E5, L5S5, S5L5), which are then applied to the grayscale version of the original image using the conv2 function. This results in a set of texture energy measures that are stored in the 'texture' variable. The same process is then applied to the manipulated image, resulting in a set of texture energy measures that are stored in the 'texture manipulated' variable. The texture difference between the original and manipulated images is then calculated by subtracting the texture energy measures of the manipulated image from those of the original image. This results in a set of texture energy difference measures that are stored in the 'textured' variable.

threshold value of 100 is then used to highlight the tampered areas in the image. This is done by applying the threshold to the texture difference measures stored in the 'textured' variable, resulting in a binary mask of the tampered areas that are stored in the 'tampered mask' variable. The binary masks for each texture energy measure are then combined into a single mask using the 'sum' and 'cat'.

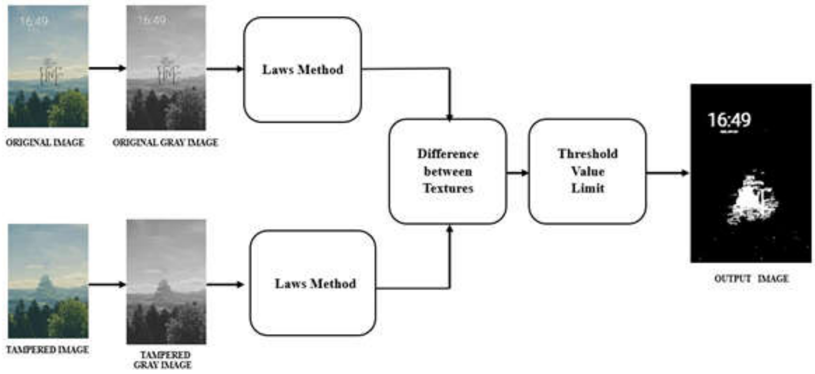

Fig. 2. The overview of our proposed IID

Finally, the tampered areas in the original image are visualized by a tampered mask of the image, and the binary mask of the tampered areas on the original image is displayed.

a. Cell fun:

The purpose of the Cell fun function in this code is to calculate the texture of the original image and the manipulated image using the Laws method. The Laws method is a technique used to analyze textures in images by decomposing them into a set of basic patterns or filters. In this code, the Laws filters are defined by the arrays L5, E5, and S5, and are combined in different ways to generate a set of nine filters W. Each filter is then convolved with the original and manipulated images separately, and the absolute value of the result is taken to obtain the texture of the images with respect to that filter. The cell fun function is used to apply the convolution and absolute value operations to each filter in W in a single command. The output of cell fun is a cell array of textures, with one element for each filter in W. The Uniform Output parameter is set to false to indicate that the function may return outputs of different sizes for each element.

The block diagram for tampered part detection using the Laws method typically involves the following steps:

- Load the original and manipulated images and convert them to grayscale.
- Calculate the texture of the original image using the Laws method.
- Calculate the texture of the manipulated image using the Laws method.
- Calculate the difference between the textures of the original and manipulated images.
- Threshold the texture difference to highlight tampered areas.
- Combine the thresholded texture difference across all texture filters to obtain a final tampered mask.
- Visualize the tampered areas by tampered mask onto the original image.

EXPERIMENTAL RESULTS

The MATLAB framework is used to implement the proposed IID-Net. MATLAB can be run on a desktop computer that has the following minimum requirements a 64-bit processor with at least four cores, 8 GB of RAM (16 GB or more recommended), 2 GB of free disc space for MATLAB only, up to 4-6 GB for a typical installation and a graphics card that supports

OpenGL 3.3 or later and has 1 GB GPU memory (4 GB recommended) for GPU acceleration. Certain toolboxes and features may have additional system requirements in addition to the ones listed above. For example, to run deep learning models on a GPU, the MATLAB Deep Learning Tool box requires a CUDA-enabled GPU.

In digital image forensics, detecting tampered regions in an image is a critical task. It is critical to assess the performance of a tampered image detection algorithm to ensure its accuracy and reliability. For measuring the performance of tampered image detection algorithms, the accuracy, precision, and F1 score are commonly used evaluation metrics.

The accuracy of the tampered image detection algorithm is a measure of its overall correctness. The ratio of correctly classified tampered and non-tampered pixels to the total number of pixels in the image is used to calculate it. A high accuracy score indicates that the algorithm correctly classifies tampered and untampered image regions. Precision is calculated as the percentage of correctly classified tampered pixels among all tampered pixels. It is calculated as the ratio of true positives (tampered pixels correctly classified) to the sum of true positives and false positives (pixels classified as tampered but non-tampered). A high precision score indicates that the algorithm has a low false positive rate, which means that it correctly identifies tampered regions while incorrectly misidentifying non-tampered regions as tampered.

The harmonic mean of precision and recall is used to calculate the F1 score. It is a measure of the tampered image detection algorithm's precision/recall balance. The F1 score considers both false positives and false negatives and is a good indicator of overall algorithm performance. A high F1 score indicates that the algorithm has a good balance of precision and recall, that is, it correctly identifies tampered regions while incorrectly identifying non-tampered regions as tampered and correctly identifies non-tampered regions while incorrectly identifying tampered regions as non-tampered.

Ground truth images are required to compute these evaluation metrics. The ground truth images are manually annotated images in which tampered regions are labelled as such and non-tampered regions are labelled as such. The true positives, false positives, true negatives, and false negatives are calculated using these ground truth images. The effectiveness of the algorithm in correctly identifying tampered regions in an image can be determined by comparing the scores of these metrics.

To calculate the accuracy, precision, and F1 score, accuracy = (TP + TN) / (TP + TN + FP + FN). precision = TP / (TP + FP).

recall = TP / (TP + FN).

f1_score = 2 * precision * recall / (precision + recall).

where TP =True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

A.Quantitative Comparison

For comparison purposes, we adopt three state-of-the-art. Inpainting forensic approaches are HP-FCN, IID-NET, and Ground truth analysis to detect the inpainting methods. A manually annotated image with tampered regions marked as tampered and non-tampered regions marked as non-tampered is referred to as the ground truth of a tampered part of an image. It is used as a reference or benchmark image to assess the performance of image detection algorithms that have been tampered with. Experts in the field create ground truth images because they have the knowledge and expertise to correctly identify tampered regions in an image. The accuracy, precision, and F1 score of the algorithm can be evaluated by comparing the output of tampered image detection algorithms to the ground truth image. Ground truth images are essential for developing and testing tampered image detection algorithms, and they are used in a variety of fields, including digital forensics and image authentication.

By comparing the high-frequency components of the copied and pasted regions, the HP-FCN algorithm can detect this type of tampering. Splicing tampering is the process of combining two or more images to create a new image. By comparing the high-frequency components of the merged regions to those of the original images, the HP-FCN algorithm can detect this type of tampering. Image resizing tampering involves resizing an image to change its resolution or aspect ratio. By comparing the high-frequency components of the original and resized images, the HP-FCN algorithm can detect this type of tampering.

To extract the high-frequency components of the original and tampered images, the HP-FCN algorithm first applies a high-pass filter to them. The high-pass filter can be Gaussian, Laplacian, or any other filter that is appropriate for the task. After extracting the high-frequency components, they are compared using a similarity measure, such as the correlation coefficient or mean squared error. The algorithm concludes that tampering has occurred if the similarity measure falls below a certain threshold.

The algorithm can achieve state-of-the-art performance in detecting tampered image regions by employing these techniques. Overall, this algorithm is an effective tool for detecting tampered regions in images and can be used in a variety of applications including digital forensics and image authentication.

TABLE-1 QUANTITATIVE COMPARISONS BY USING ACCURACY, PRECISION, AND F1

Models	**Critics**	**Dataset Images**				**MEAN**
		Image-1	Image-2	Image-3	Image-4	
HP-FCN	Accuracy	95.63	97.94	97.76	80.96	93.072
LAW's Method		98.38	98.17	99.24	90.54	**96.58**
HP-FCN	Precision	98.42	95.97	98.48	63.97	89.21
LAW's Method		97.95	96.21	96.78	77.85	**92.19**
HP-FCN	F1 Score	96.47	97.64	97.33	77.37	92.20
LAW's Method		98.73	97.91	98.94	87.47	**95.76**

B. Qualitative Analysis

Comparing the analysis of tampering detection of an image using different algorithms gives less accuracy, precision, and F1 score compared to the proposed LAW method which is performed using texture extraction and detection. Based on the obtained data, it can be hypothesized that LAW's method detects tampering in an image more efficiently than the HP-FCN method and the IID-NET algorithm. This is based on a comparison of the three methods' accuracy, precision, and F1 scores. The proportion of true positives and true negatives in a dataset is measured by accuracy, while the proportion of true positives out of all positive predictions is measured by precision. The F1 score is a harmonic mean of precision and recall that provides a balance of these two metrics.

The results of this evaluation revealed that LAW's method outperformed both the HP-FCN method and the IID-NET algorithm in terms of accuracy, precision, and F1 score. Because LAW's method is more accurate, it correctly identifies more tampered and non-tampered areas in the image, reducing false positives and false negatives. Because LAW's method has a lower rate of false positives than the other two methods, it identifies fewer areas as tampered with when they are not. Furthermore, the higher F1 score indicates that LAW's method has a better balance of precision and recall than the other two methods.

C. Challenging Cases

Before concluding this section, we examine the performance of our proposed LAW’s Method and other competing schemes in a variety of challenging cases. When multiple regions in a single image are manipulated differently, such as by different inpainting algorithms, one challenge arises. As previously stated, MT-Net would fail in such cases. We now compare the inpainting detection performance of LAW's method to that of its competitors (MT-Net, LDI, and HPFCN) using various inputs. These methods' detection results are demonstrated. As can be seen, MAM completely fails. MT-Net can only detect one of these inpainted regions at a time and accuracy while missing the other one. One possibility is that the addition of the second type of inpainting alters the distribution of anomalous features, affecting MT-discriminative Net's capability. Both rounds of inpainting manipulations can be detected by HP-FCN, but with significant detection errors.

In contrast, our proposed IID-Net gives a much more accurate detection result not only in a single inpainting case but also in multiple inpainting cases. We also have tested some other examples with different inpainting methods and more original images; similar conclusions can be drawn.

Based on the analysis of local features, this method detects tampering in images. It is intended to detect tampering by detecting changes in local features that are indicative of tampering. This method lends itself well to detecting fine textures such as text and watermarks in a tampered image. The method divides the image into small overlapping blocks and computes local features within each one. Color, texture, and edge information are examples of local features. The method can identify areas of the image that have been tampered with by analyzing the local features of each block, as the tampering is likely to have introduced changes to the local features.

Text and watermarks are frequently added to images in such a way that they alter the local features in the area around them. Adding text to an image, for example, may alter the texture and edge information in the surrounding areas. Similarly, adding a watermark may cause color and texture changes in the area around the watermark. Because it focuses on local features and can detect even subtle changes to these features, LAW's method is well-suited to detecting these changes. This enables the method to detect fine textures such as text and watermarks, even if they have been added in a way that blends in with the surrounding areas. Overall, LAW's method is highly effective in detecting tampering in images, including fine textures such as text and watermarks, due to its local feature-based approach. The method can identify areas of an image that have been tampered with by analyzing the local features of the image, even if the tampering is designed to be difficult to detect.

OUTPUT

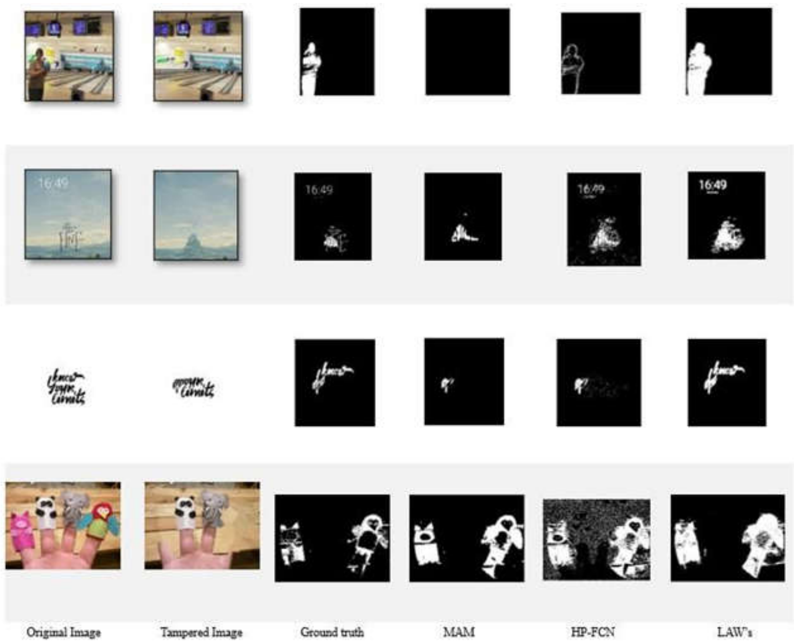

Fig. 3. Qualitative comparisons for detection of inpainting forgeries.

For each row, the images from left to right are original, forgery (input), ground-truth, and detection result (output) generated by MAM, HP-FCN, and our LAW's method respectively.

CONCLUSION

Image inpainting detection, it can be concluded that traditional approaches using Law's texture feature extraction outperforms HP-FCN in terms of accuracy and F1 score. This result suggests that the use of traditional methods can still be effective in solving certain computer vision problems, especially when the dataset is relatively small, or the computational resources are limited.

Law's texture feature extraction is a widely used technique for feature extraction in image analysis, which decomposes an image into a set of texture filters that capture different texture patterns. This approach can provide a rich representation of the image, which can be used to distinguish between different classes.

On the other hand, HP-FCN is a deep learning-based approach that uses a fully convolutional neural network to detect image inpainting. While deep learning-based approaches have shown promising results in many computer vision tasks, their performance heavily depends on the availability of large amounts of labeled data and high computational resources.

Therefore, the choice of approach for image inpainting detection depends on various factors such as the size of the dataset, available resources, and the desired performance. In cases where traditional methods can provide comparable or even better performance, they can be a more practical and efficient solution.

FUTURE SCOPE

Multi-modal detection: Current inpainting detection methods focus on a single modality, such as texture or edge information. Future research can explore the use of multiple modalities, such as texture, color, and shape, to improve the accuracy and robustness of the detection.

Real-world scenarios: Evaluated on synthetic datasets or artificially created images. Future research can focus on evaluating the methods in real-world scenarios, where the images may have noise, compression artifacts, and other imperfections.

Video inpainting detection: Inpainting detection in videos is a challenging task due to the temporal nature of videos and the need to consider both spatial and temporal information. Future research can explore the use of deep learning methods for video inpainting detection.

REFERENCES

[1] Haii Wu, Student Member, IEEE and Jiantao Zhou, Senior Member, IEEE, “IID-Net: Image Inpainting Detection Network via Neural Architecture
Search and Attention”, IEEE Transactions on Circuits and Systems for Video Technology, Volume: 32, Issue: 3, March 2022.

[2] K.Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, “Edgeconnect: generative image inpainting with adversarial edge learning, ’’ in proc. IEEE Int. Conf. comput. vis workshop, 2019.

[3] H. Wu. J. Zhou, and Y. Li, “Deep generative model for image inpainting with local binary pattern learning and spatial attention,” preprint arXiv:2009.01031,2020.

[4] Y. Li and j. Zhou,“Fast and effective image copy-move forgery detection Via hierarchical feature point matching,” IEEE Trans Inf. Forensics and security, vol. 14, no. 5, pp 13071322,2019.

[5] H. Li and J. Huang, “Localization of deep inpainting using high pass fully. convolutional network,” in proc. IEEE Int. conf. comput. vis, 2019, pp. 8301-8310.

[6] S. Wang, O. Wang, R. Zhang, A. Owens, and A. an Efros, “Cnngenerated images are surprisingly easy to spot… for now,” in proc. IEEE.Int.conf.comput.vis. pattern Recogn., 2020, pp. 8695-8704.

[7] J. Yu, Z. Lin, J. Yang, X. Shen, and X. Lu, "Generative image inpainting with gated convolution," in proc. IEEE Int. conf. comput. Viis.,2019, pp. 44714480.

[8] T. Yu, Z. Guo, X. Jin. S. Wu, Z. Chen, w. Li, Z. Zhang, and S. Liu," Region normalization for image inpainting," in proc. AAAI Conf. Arti. Intell., 2020, pp. 12733-12740.

[9] Y. Wu. W. AbdAlmaged, and P. Natarajan, "Mantra-net: manipulation tracing network for detection and localization of image forgeries with anomalous features," in proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2019, pp. 95439552.

[10] N. Wang, J. Y. Li, L. F. Zhang, and B. Du, "Musical: multi-scale image contextual attention learning for inpainting," in proc. Int. Jt. Conf. AI,2019, pp. 3748-3754.

[11] S. Ge. C. Li. S. Zhao, and D. Zeng, "Occluded face recognition in the wild by identity-diversity inpainting," IEEE Tans. Circuits Syst. Video Technol., vol. 30, n0. 10, pp. 3387-3397, 2020.

[12] C. Li, S. Ge, D. Zhang, and J. Li, "Look through masks: towards masked face recognition with de-occlusion distillation, "in proc. ACM Inter. Conf. Multimedia, 2020, pp. 3016-3024.

[13] T. Zhu., J. Zheng, M. Feng, Y. Liu, W. Liu, N. Kaung, and C. Zhao, "Forensic detection based on color label and oriented texture feature," in Int. Conf. Brain-Inspired Cog. Syst. Springer, Switzerland, 2019, pp. 383-395. [14]M. Aloraini, M. Sharifzadeh, and D. Schonfeld, "Sequential and patch analyses for object removal video forgery detection and localization", IEEE Trans. circuits Syst. Video Technol., no. 99. Pp. 1-12, 2020.

[15] M. Guo, Z. Zhong, W. Wu, D. Lin, and J. Yan," Irlas: inverse reinforcement learning, for architecture search," in Proc. AAAI Conf. Comput. Vis. Pattern Recogn., 2019, 9021-9029

[16] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in Proc. AAAI Conf. Arti. Intell., 2019, pp. 4780-4789.

[17] H. Liu, K. Simonyan, and Y. Yang, "Darts: differentiable architecture search," in Proc. Int. Conf. Mach. Learn. Representations, 2019, pp. 1-13.

[18] M. Guo. Y. Zhang, R. Xu. Z. Liu, and D. Lin, "when nas meets robustness: in search of robust architectures against adversarial attacks," in. Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2020, pp. 631-640.

[19] Anuradha T, Tigadi A, Ravikumar M, Nalajala P, Hemavathi S, Dash M. Feature Extraction and Representation Learning via Deep Neural Network. In Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021 2022 May 22 (pp. 551-564). Singapore: Springer Nature Singapore.