International Journal of AI & Data Science Review
Volume1, Issue 4, 2025

DEEP FORGERY DETECT: USING DEEP LEARNING-BASED FORGERY DETECTION TO IMPROVE SOCIAL MEDIA SECURITY

R. Abadnai[1], L. Alikha[2]
Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

Article Info

Received: 29-07-2025 ***Revised:28-08-2025*** ***Accepted:20-10-2025*** ***Published:20-11-2025***

Abstract – These days, surveillance cameras play a major role in both legal and security applications. Nonetheless, the images and video recordings may be readily altered using a variety of video editing programs. The recorded data may be used as evidence in a crime scene investigation. The result of a court action or inquiry may be greatly impacted by the integrity of the picture and video. Therefore, before employing surveillance images and videos as evidence in any judicial proceeding, it is essential to verify their legitimacy. Because they can not effectively use high-resolution encoded information, current deep learning-based methods for image copy-move fraud detection are unable to discern the borders of tiny modified objects. The recovered frames in this proposed article improve the accuracy of identification for tiny items by making use of the spatial information found in the feature maps. A customized CNN-LSTM layer that employs transfer learning to differentiate between authentic and modified frames is shown in the study. Ensemble Learning is used to evaluate the model. The algorithm's capacity to identify copy move forgeries is improved by using sophisticated neural network designs, which enable the model to efficiently learn and extract intricate information from both photos and videos. The model achieves 95.00% accuracy in detecting forged films with static backgrounds, 98.67% accuracy in detecting forged videos with moving backgrounds, and 95.08% accuracy in detecting and localizing the forged image. A variety of social media photos and videos are used to evaluate the model. The experimental findings show that the proposed model outperforms other current methods.

Keywords *– Images, Videos, Social media networks, Custom CNN, LSTM*

1. INTRODUCTION

Large volumes of multimedia material may be distributed more easily thanks to the growing usage of social networks. Simultaneously, the development of digital modification tools such as Photoshop, Gimp, and smartphone applications has made it easier for people to alter photographs and videos on these platforms. As a result, there has been a noticeable rise in the number of fake photos and films that are circulating on social networks, making it possible to use them for fraudulent purposes. A kind of digital picture fraud known as "copy-move forgery" occurs when a section of an image is copied and then relocated inside the same image, usually to hide, duplicate, or change the look of a component.



Figure 1. (a) Actual Image



Figure 1. (b) Fake Image

It involves selecting a specific region or object within the image and replicating it, then placing the duplicate in a

different location within the same image. This technique is commonly used to tamper or manipulate the content of an image without leaving obvious traces of alteration. Detecting copy-move forgery often requires specialized algorithms and forensic techniques designed to identify duplicated regions and inconsistencies within the image. Forgery is a prevalent form of spreading misinformation with forged images and videos being prominent tools for broadcasting fake information. Forged images gather more attention than text [1]. n order to prevent the proliferation of fraudulent photos and videos circulating on the internet, several multinational corporations are presently investing in the development of artificial intelligence applications.Fig.1(a) and (b) showcase image sourced from CASIA 2.0 dataset [30]

Due to processing constraints, a lack of generalization, and low performance accuracy, real-time identification of forgery detection of large videos with varying frame speeds is not feasible. Through the use of deep learning, this study looked into the issue of forgery detection of videos running at various frame rates with static and moving backgrounds as well as detection of forged images with smooth edges. These fraudulent pictures are disseminated all over the world after going viral on social media. Some forged photos are harmless, but some may harm a person's reputation or make people afraid [3]. Therefore, developing tools to identify forged multi-media sites is necessary.

2. RELATED WORK

Multimedia forgery attacks are divided into: (i) temporal ii) spatial and (iii) spatiotemporal [2]. In temporal type of attacks, manipulation occurs at the frame level, such as adding, removing, or replacing frames in a video, on the other hand, spatial attack involves the modification of objects within a single video frame or a series of frames including the addition, removal or replacement of distinct objects [4]. The intersection of the geographical and temporal regions is known as the spatiotemporal region.

This research paper examines the detection of spatial type of forgery, which involves two tampering methods: copy-move and splicing. Copy-move involves duplicating the image or video frame and affixing it to a various segment within the identical image or video, while removal of objects is achieved by filling the region with adjacent pixels sourced from the same frame. Splicing, on the other hand, is a technique where objects and content from different videos are combined or merged together.

An instance of spatial type of forgery is depicted in the figure below. Authentic video frame sequence and tampered video sequence are depicted in Fig 2(a) and Fig 2(b).

Fig 2(a) shows the actual frame sequences and Fig.2 (b) illustrates an example where a segment of the same video is duplicated and put into other part of the same video. This leads to the misguidance of viewers through misinformation. This type of tampering is harmful and has an adverse effect on society since it hides the facts for purposes of criminal intent rather than just editing or changing the format. Based on the type of features, different methods for detecting spatial tampering are identified. Kobayashi et al. [5] used noise characteristics. In [6], authors used SIFT characteristics, noise residual, and pixel correlation for identifying forgery.

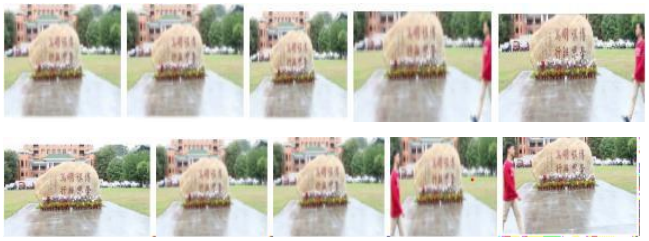


Figure 2. (a) Actual video frames, **(b)** Fake video frames

In [7], the authors discussed sensor pattern noise to identify forgeries. Authors of [8] also utilized quantization features, distortion residual, and their changes to identify forgery. Most of the image tamper classification methods is based on frequency domain characteristics or statistical features of an image [9],[10],[11],[12]. The mean gradient of each color channel and additional statistical features of wavelet Coefficients are gathered and entered as vectors of attributes into the support vector machine (SVM) in [13] in order to distinguish between real objects and fake ones.

Lichao Su [14] used statistical correlation for detecting forgery of videos subjected to mirroring. Singh and Aggarwal [15] used discrete fractional Fourier transformation, pixels correlation, noise inconsistency. Singh and Singh [16] calculate the region similarities to detect the replicated regions within the videos. Analysis of multiples JPEG compression's is used in [17] to identify altered images, which are specific to the JPEG formats.

Images that have been manipulated and circulated on social media platforms experience a series of alterations, leading to further degradation in quality through the addition of noise. The researchers in the study [18] identify instances of inter-frame copy-move forgery through the use of optical flow (OF). The initial phase of detection involves scrutinizing the OF two flag areas that could potentially be indicative of tampering. Subsequently, the precise location of the forgery is identified through a more meticulous detection process. However, a drawback lies in its inability to uncover forgery within extensively altered videos depicting static scenes. In another work [19], the authors introduce optical flow inconsistencies and employ using DTW matches find occurrences of copy-move forgeries in videos.

The histogram features of an image are used in [20]. These HOG features use gradient orientation, which is ineffective in accurately describing the patterns and variations related to local texture of an image. In [22], Su et al. employed improved MISIFT to identify the forged areas in later frames, spatio-temporal training was used to initially detect picture copy-move forgery in the initial frame of a the media.. Meanwhile, Su et al. [23] utilized K-SVD algorithm and K-means to detect forgeries. Additionally, the technique presented in [24] was able to detect forged ballistics motion in video sequences. In [25], Yao et al. employed a CNN Layer to extract features with high-dimensionality. To eliminate temporal redundancy, they employed the absolute

difference between succeeding frames. Max pooling as well as high pass filtering layers were included by the authors in order to decrease computing complexity and increase the quantity of residual information throughout the tampering method, accordingly.

Detecting Copy move forgeries present a greater challenge due to the subtler nature of their abnormalities compared to other forms of forgery. Methods [26-28] employed Convolutional Neural Networks (CNNs) to identify forged images through the detection of abnormal traces such as in consistencies in noise and illumination direction throughout the entire image. However, smaller abnormality of images using CMF compared to other types of forgeries hampers its effective detection. Consequently, dedicated methods specifically to detect MF have emerged. In[29] also uses CNN to assess whether an image is tampered or not. Addressing the issue of low accuracy in the detection results of most CNN based copy move forgery is the key objective of the proposed method. The main focus of CNN based approaches for forgery detection lies in image analysis rather than video analysis. The proposed method focuses on video as well as image forgery detection which results in better accuracy.

Compared to previous techniques discussed, the proposed CNN-LSTM model adds a recurrent layer that can capture temporal dependencies in the input data, making it more suitable for sequential data such as video frames. A pure LSTM model can learn temporal dependencies in video data, the CNN-LSTM model takes advantage of the CNN's ability to extract spatial features from each frame, making it a more powerful model for video as well as image forgery detection. In comparison to a simple feed forward neural network (FFNN), the CNN-LSTM model can handle variable-length sequences by using the LSTM layer to learn and remember important information from past frames in the video. Unlike a recurrent neural network (RNN), the CNN-LSTM model can process multiple frames of input data in parallel using the CNN layer, making it faster and more efficient for video analysis tasks In comparison to other advanced models used in video forgery detection, such as 3D CNNs or attention-based architectures, the CNN-LSTM model strikes a beneficial balance between model complexity and performance. This makes it a viable and sensible choice for a wide range of applications.

The rest of the research is structured as subsequent sections. Section III furnishes a detailed explanation of the approach for detecting video and image fraud. Sections IV and V offer extensive experimental details, results, and their analysis. Section VI finishes this study by discussing further research.

3. DEEPFORGERYDETECT: DETECTION NETWORK OF IMAGE AND VIDEO FORGERY

The DeepForgeryDetect is an efficient network architecture for video and Image forgery detection which consists of two primary elements: a convolutional neural network (CNN) and a long short-term memory (LSTM) network. The VGG16 model, a deep neural network architecture has been trained on a large image dataset for image classification, is used in the CNN portion of the design.

The VGG16 network comprises a total of 13 convolutional layers and 3 fully connected layers, and is known for its ability to extract high-level features from images. In the proposed architecture, the VGG16 network is used to extract features from each frame of the input video. The CNN layers are followed by a set of LSTM layers, which are employed to capture the temporal dependencies between the frames in the video sequence. These layers maintain the context of previous frames while processing the current frame. To enhance and optimize the performance of the model, a transfer learning technique is applied, adjusting the parameters of the VGG16 network to suit the specific task of video forgery detection. This involves locking the pre-trained VGG16's weight parameters and training CNN, LSTM layers on a large dataset of videos and image with and without forgeries.

The model's output is a binary classification that categorizes each frame as either genuine or forged. The model achieves high accuracy on a variety of video and Image forgery datasets, demonstrating the effectiveness of combining CNN and LSTM networks for forgery detection.

VGG16 is a pre-trained CNN architecture that has demonstrated exceptional performance in numerous image classification tasks. By using it as a feature extractor, we can obtain a high-dimensional feature representation that captures important information in the video frames. The CNN and LSTM models built on top of the VGG16 features can further refine this representation and model the temporal dependencies in the video, leading to higher accuracy in detecting video forgeries. The architecture of the proposed system is shown in fig 3.

Utilizing a pre-trained VGG16 model for extracting features enables us to take advantage of transfer learning. Transfer learning enables us to use knowledge learned from one task (e.g., object recognition in natural images) to improve performance on another task (e.g., video forgery detection) with less data and computation. This can save time and resources in developing a video forgery detection model. The VGG16, CNN and LSTM model is designed to be robust to variations in the visual appearance of the video frames. The CNN layers can capture features that are invariant to changes in scale, rotation, and translation, while the LSTM layers can model long-term dependencies across multiple frames. This makes the model less sensitive to variations in the visual appearance of the video frames, such as changes in lighting, camera angles, and object occlusions.

The aforementioned network design combines a long short-term memory (LSTM) model and a convolutional neural network (CNN), designed for image forgery detection. The CNN part of this model makes use of the pre-trained VGG16 model to serve as a feature extractor. The initial image dimensions are 128x128x3, and the VGG16 model is employed to derive features from this input image. Subsequently, the output from the VGG16 model is flattened into a one-dimensional feature vector using a Flatten layer.

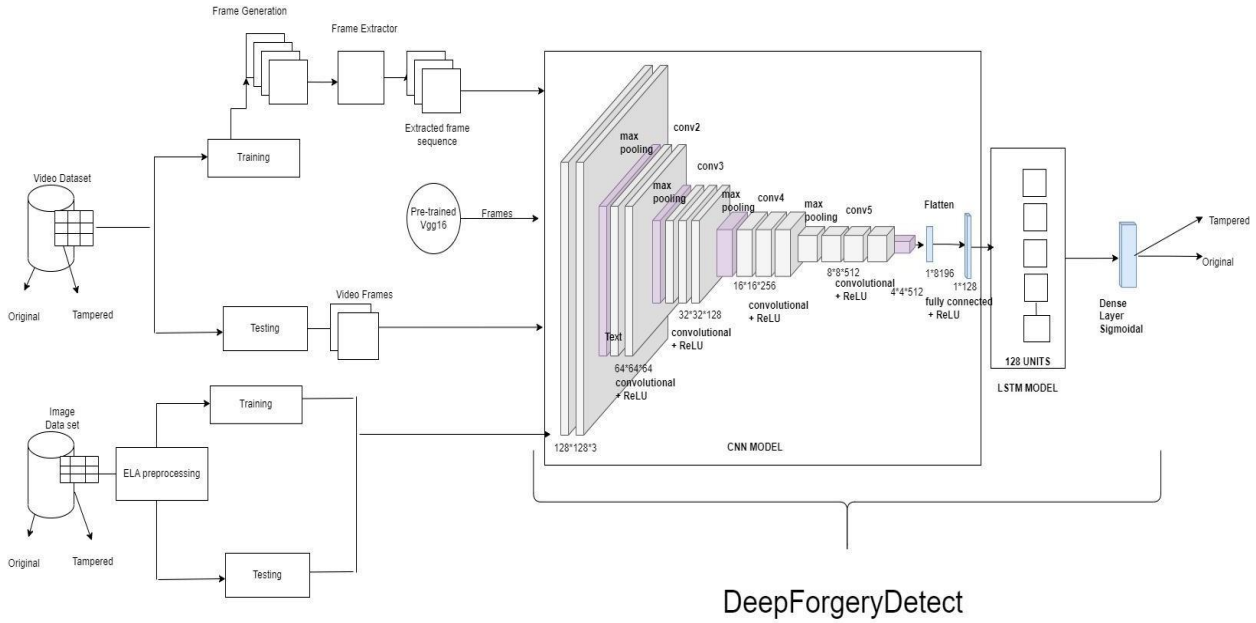


Figure 3. System Architecture of the proposed CNN-LSTM Model

The model is then flattened before an additional dense layer with 128 units and a ReLU activation function is added. The output of the dense layer is then transformed into a tensor of dimensions (1,128) by a Reshape layer, which matches this information.

This tensor serves as input for the LSTM model. The LSTM model comprises a singular LSTM layer containing 128 units, followed by a dense output layer consisting of 2 units and employing a sigmoid activation function. This sigmoid function is chosen for binary classification. The LSTM layer processes the sequence of input features derived from the CNN, generating a single vector representation that encapsulates information from the input sequence. This resultant vector is subsequently channeled through the output layer to yield the ultimate classification outcome. The Adam optimizer and binary cross-entropy loss are used to train the whole structure.

i) Training the network

3.1. VGG16

The VGG16 structure consists of 13 levels of convolution, which are followed by 3 completely linked layers. The following are the forward pass formulae:

A. *Convolutional layer:*

$$a(j) = W(j) * c(j-1) + d(j) \quad (1)$$

$$Z(j) = \mathrm{ReLU}(a(j)) \quad (2)$$

where a (j) is the input to the j-th layer, Z (j) is the outcome of the i-th level, W (j) and d (j) are the dimension and base parameters for the j-th layer, c (j-1) is the output of the Z (i-1) -th levelr, and ReLU is activation function.

A. Max pooling layer:

$$Z(i) = \mathrm{MaxPool}(Z(i-1)) \quad (3)$$

where MaxPool is the max pooling function.

B. Fully connected layer:

$$a(j) = W(j) * c(j-1) + d(j) \quad (4)$$

$$Z(j) = \mathrm{ReLU}(a(j)) \quad (5)$$

where a (i) is the input to the j-th layer, c (j) is the outcome of the j-th layer, W (j) and d (j) are the dimension and base parameters for the i-th layer, Z (i-1) is the outcome of the c (j-1) -th layer, and ReLU is the activation function.

C. Softmax layer:

$$a(j) = W(j) * c(j-1) + d\hat{}(j) \quad (6)$$

$$c(i) = \mathrm{Softmax}(a(i)) \quad (7)$$

where a (j) is the input to the i-th layer, c (j) is the output of the j-th layer, W (j) and d (j) are the the dimension and base parameters for the j-th level, c (j-1) is the output of the Z (j-1) -th levelr, and ReLU is the activation function.

Algorithm 1: Learning algorithm for training the VGG16 model:

Input: Input Image Size is 128*128*3

Output: Feature Vector

1. Initialize the VGG16 model with pre-trained weights on ImageNet.

2. Freeze the dimension of the convolutional layers.

3. Add a new fully connected layer to the VGG16 model for classification.

4. Train the fully connected layer by utilizing back propagation with the video forgery detection dataset.

3.2. Convolutional Neural Network

A CNN is a neural network that is good at image processing and object detection. The equations for the forward pass:

3.2.1. Convolutional layer

$$a(j) = W(j) * c(j-1) + d(j) \quad (8)$$

$$c(j) = \mathrm{ReLU}(a(j)) \quad (9)$$

where Z (j) is the input to the j-th layer, a(j) is the result of the i-th layer, W (j) and d (j) are the weight and base parameters for the j-th layer, a (j-1) is the output of the (j-1)-th layer, and ReLU is the activation function.

3.2.2. Max pooling layer

$$Z(i) = \mathrm{MaxPool}(Z(j-1)) \quad (10)$$

where `MaxPool` is the max pooling function.

3.2.3. Fully connected layer

$$a(i) = W(j) * a(j-1) + d(j) \quad (11)$$

$$c(i) = \mathrm{ReLU}(a(j)) \quad (12)$$

where Z (j) is the input to the j-th layer, a(i) is the output of the i-th layer, W (i) and b (i) are the weight and bias parameters for the i-th layer, a (i-1) is the output of the (j-1)-th layer, and ReLU is the activation function.

Algorithm 2: Learning algorithm for Convolutional Neural Network

Input: a batch of images with shape (batch_size, 128, 128, 3), where batch_size is selected according to size of the dataset.

Output: a batch of feature vectors with shape (batch_size, 1, 128)

1. Apply the pre-trained VGG16 model to the batch of images to extract their features.

- Input shape: (batch_size, 128, 128, 3)

- Output shape: (batch_size, 4, 4, 512)

2. Construct convolution Layers

3. Flatten the output from the VGG16 model to obtain a feature vector for each image in the batch.

- Input shape: (batch_size, 4, 4, 512)

- Output shape: (batch_size, 8192)

4. Employ a fully connected layer to the flattened feature vector to reduce its dimensionality to 128, while applying the ReLU activation function to introduce nonlinearity.

- Input shape: (batch_size, 8192)

- Output shape: (batch_size, 128)

- Output equation: x = ReLU(Wx + b) (13)

5. Reshape the output of the fully connected layer to have a shape of (batch _size, 1, 128)

- Input shape: (batch_size, 128)

- Output shape: (batch_size, 1, 128)

Algorithm 3: Learning algorithm for the LSTM model

Inputs:

- Input tensor of shape (batch_size, timesteps, features), where features dimension should match the output shape of the CNN segment of the model.

Outputs:

- Output tensor of shape (batch_size, lstm_units) from the second LSTM layer.

1. Define an LSTM model with two LSTM layers:

- First LSTM layer should have 128 units and return sequences.

- A Dense layer with 1 neuron and sigmoid activation.

Algorithm 4: Learning algorithm for the interconnection of CNN and LSTM model

1. Connect the LSTM model to the CNN model's output:

- Create a Sequential model to merge the CNN and LSTM models.

- Add the CNN as the first layer of the Sequential model.

- Add LSTM model as the second layer of the Sequential model.

2. Compile the model:

- Use binary cross-entropy loss.

- Use the Adam optimizer.

- Use accuracy as a metric.

3. Utilize the input data and labels to train the model.

- Use fit() method with the specified batch size and number of epochs.

- Use the provided validation data for validation during training.

4. Utilize the trained LSTM model for making predictions on new data:

- Use predict () method with the test data to get the predicted output.

In the proposed method, the VGG16 model, which serves as a pre-trained feature extractor, takes an input tensor of dimension (batch_size, image_height, image_width, image_channels), where image_height and image_width is both 128 and image_channels is 3 (corresponding to RGB channels). The output shape of the VGG16 model results in (batch_size, 4, 4, 512), representing the feature maps extracted from the final convolutional layer of the VGG16 model. The CNN component of the model, processes the VGG16 model's output using a TimeDistributed wrapper.

The input shape to the CNN model is (batch size, Num frames, 4, 4, 512), where num_frames signify the count of frames within the input video sequence (which is 10 in this case). The output shape of the CNN model is (batch_size, num_frames, 8192), which corresponds to the flattened feature maps obtained from the CNN layers. Finally, the LSTM portion of the model accept the output of the CNN model as its input.

The initial layer consists of an LSTM architecture featuring 128 units. The quantity of units dictates the intricacy and capacity of the LSTM's capability to discern patterns within the sequential data. Subsequently, the output generated by the LSTM layer is directed to the subsequent layer. The purpose of this Dense layer is to transform the output of the LSTM layer into a probability distribution that includes the two classes. By employing the sigmoid activation function, the resulting values are constrained to a range between 0 and 1, denoting the probabilities of belonging to each respective class.

The number of Dense layer units matches to the number of classes in the classification task. In this instance, it is set to 2 ie,the video or image is original or tampered.

4. EXPERIMENTAL SETUP

The implementation of the model took place within Google Colab, utilizing the Keras library and making use of 100 GPU units. The images underwent resizing, being adjusted to dimensions of 128 by 128. To determine the optimal combination of hyper parameters, numerous iterations were conducted, varying batch sizes and dropout probabilities. The proposed model was evaluated on three publicly available datasets: VIFFD [30], ViFoDAC [31], REWIND_3D [32], CASIA 2.0 [33].

a) CASIA 2.0 dataset consists of a total of 12,616 images, encompassing 7,492 genuine images and 5,124 altered images. The manipulation images underwent alterations through methods such as like copy-move and image splicing, along with subsequent actions of cropping and resizing.

b) VIFFD is a dataset designed to detect video inter-frame forgeries, also known as copy-move forgeries, where entire frames are copied from one part of a video and pasted into another. The dataset includes 136 training and 136 testing videos, totaling 272 videos.

c) ViFoDAC is a dataset comprises 16 authentic videos alongside 16 manipulated videos, with durations spanning approximately 10 to 62 seconds each. The genuine videos were captured utilizing a camera, while the manipulated videos underwent editing through various tools to introduce objects within the video sequence. This dataset proves valuable for identifying forgeries that involve the external addition of objects to videos, as well as any instances of forgeries where alterations occur in the background.

d) REWIND_3D is a dataset serves the purpose of detecting forgery involving small objects within videos. It encompasses 10 original videos and 10 tampered videos, each lasting around 6 to 18 seconds. The videos were recorded using a stationary camera, maintaining a consistent background throughout. This dataset is particularly effective in identifying forgeries where small objects are incorporated into pre-existing videos without causing changes to the background.

e) MICC-F220: This dataset comprises a collection of 220 images, equally divided into 110 manipulated images and 110 authentic originals.

Rather than employing conventional images for image-based learning, the model makes use of error level analysis (ELA) images, resulting in enhanced convergence speed and accuracy within deep learning models.

5. EXPERIMENT RESULTS

The performance evaluation of the VGG16-CNN-LSTM architecture involved conducting experiments on both images and videos with static and moving back grounds. The training process for the static camera scenario involved using a dataset where the cameras were stationary, while the moving camera scenario involved training the model using a dataset where the cameras were in motion. By testing the model on different scenarios, we were able to assess its robustness and generalization capabilities. The experimental outcomes show the proposed VGG16-CNN-LSTM framework performs well in different scenarios. In the case of the static camera scenario, the achieved results show that, for a dataset split of 75-25, the model attained a training accuracy of 95.00% and a validation accuracy of 90%. This signifies that the model has effectively captured the underlying data features and can generalize to novel data with a commendable level of accuracy. In the case of the moving camera scenario, the model achieved a higher training accuracy of 98.67% and a validation accuracy of 95%. This indicates that the model is more robust in handling the variations caused by the movement of the camera. The model is also trained using different image dataset with produces an accuracy of 95.08%. The VGG16-CNN-LSTM architecture seems to be effective in capturing the spatiotemporal features of the data and has shown promising results in both static and moving camera scenarios. A confusion matrix provides insight into the classification model's performance on a specific test data by comparing the model's predictions to the actual labels.

The confusion matrix comprises four key components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy and confusion matrix for static background video dataset and moving background video dataset given by the model is depicted below. Table 1. show the performance analysis of the developed model using different dataset and different test split.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

5.1. Performance Analysis of DeepForgeryDetect Network against different dataset and different test split

Fig 4. shows the graph of performance analysis of Deep Forgery Detect Network with other methods. These outcomes indicate the model's performance in detecting Image and video forgery based on the given background conditions.

Table 1. Performance Analysis of the model for different dataset and different test split

Test Split of Video Dataset	Accuracy of the proposed Model		
	Dataset	**80-20**	**72-25**
Static background	VIFFD	.9427	.9500
	REWIND_3D	.9347	.9458
Moving background	ViFoDAC	.9860	.9867
Image dataset	CASIA_2.0	.9408	.9508

5.2. Performance Analysis of DeepForgeryDetect Network with state-of-the-art methods in terms of accuracy

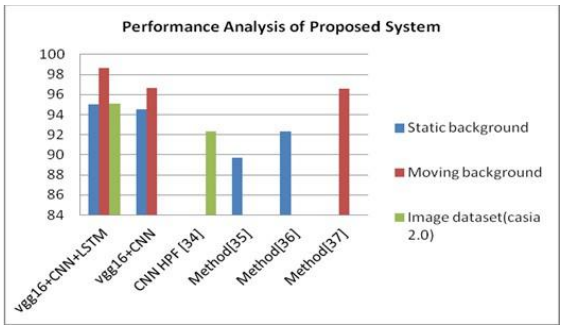


Figure 4. Comparison of Deep Forgery Detect Network with state-of-the-art methods

The model achieved a higher accuracy of 98.67% when the background was moving and 95.00% accuracy in the static background scenario with a test split of 72 and 25 and 95.08% accuracy for the image dataset. These results suggest that the model performs better in identifying and localizing video forgery compared to the existing detection mechanism. The moving background might provide additional cues or features that help the model differentiate between genuine and tampered video frames.

5.2.1. Scene is moving but camera is static

Fig 5 (a) shows the graph of Total loss and Validation loss and Fig. 5(b) shows the graph of Total Accuracy Vs Total Validation accuracy by considering the number of epochs and Fig 6 shows the confusion matrix for a scene, which has static background.

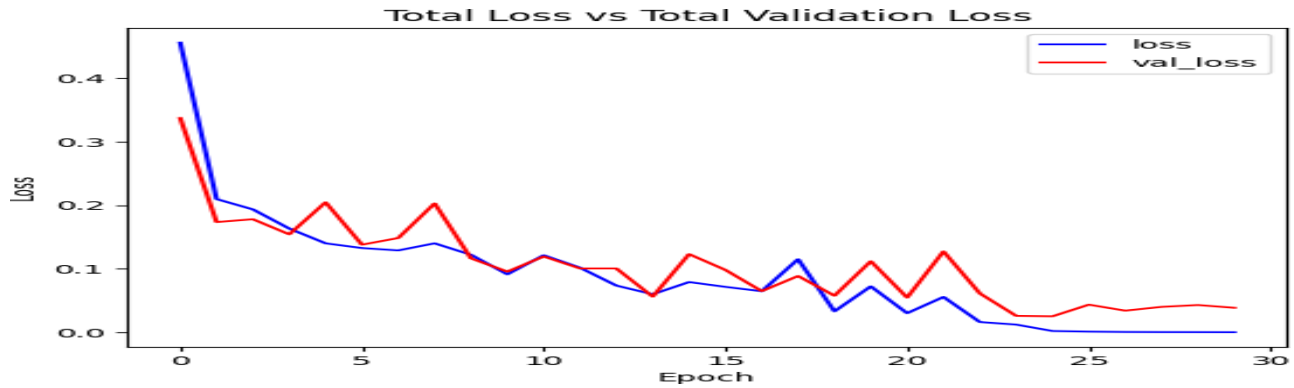


Figure 5 (a). Graph showing Epoch Vs Loss in Static Scene.

5.2.2. Scene is moving and camera is moving

Fig 7 (a) shows the graph of Total loss and Validation loss and Fig. 7(b) shows the graph of Total Accuracy Vs Total Validation accuracy by considering the number of epochs and Fig 8. Shows the confusion matrix for a scene, which has moving background.

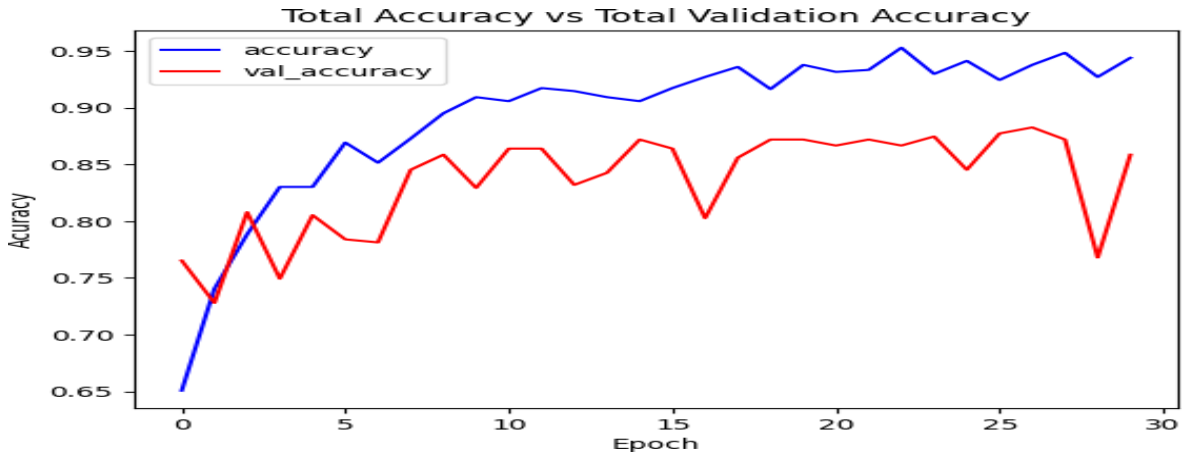


Figure 5 (b). Graph showing Epoch Vs Accuracy in Static Scene.

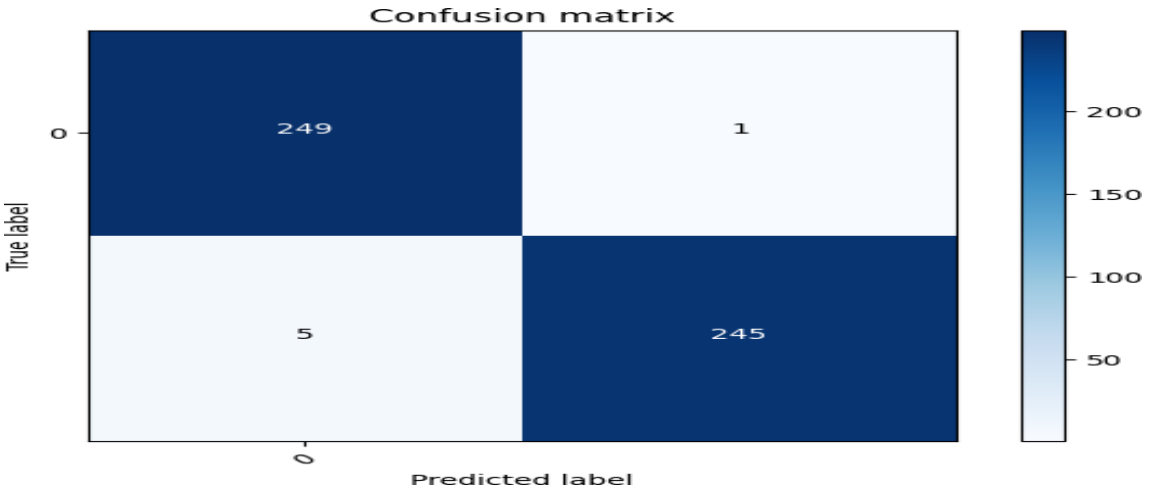


Figure 6. Confusion matrix for a static background



Figure 7 (a). Graph showing Epoch Vs Loss in moving Scene.

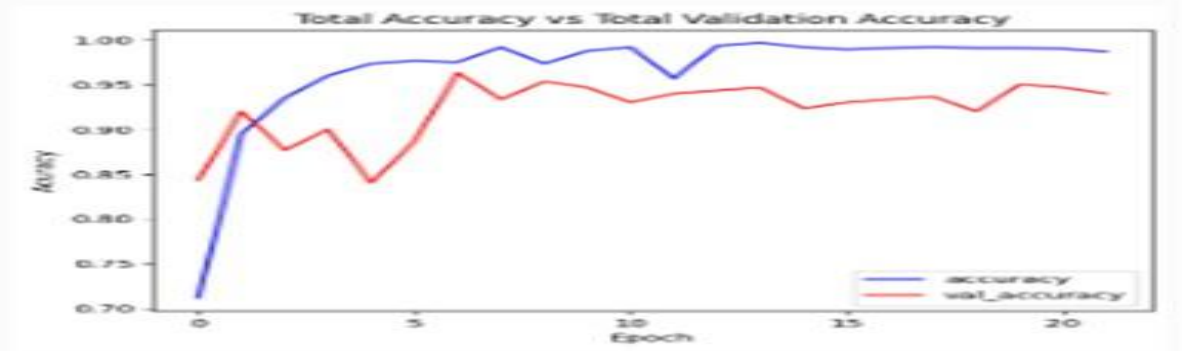


Figure 7 (b). Graph showing Epoch Vs Accuracy in moving Scene.

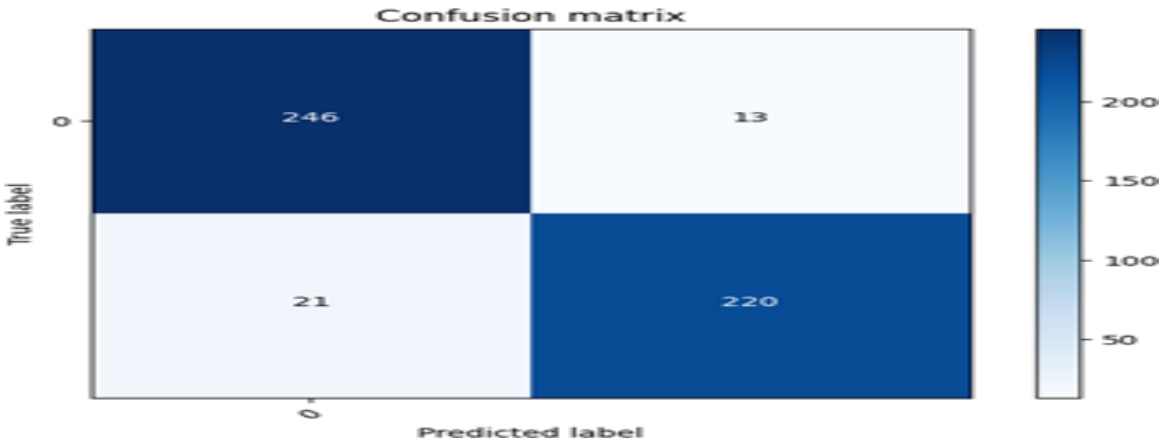


Figure 8. Confusion matrix for a Moving Scene

5.2.3. Image Tampering Detection using ELA

Training a single model for both image and video forgery detection can promote generalization. The model learns to extract relevant features and identify common forgery patterns from both images and videos. This broader understanding helps the model perform better on unseen data, making it more robust and versatile in detecting various types of forgeries. Image is preprocessed using Error Level Analysis (ELA) and trained using the VGG16 + CNN + LSTM model. By leveraging the advantages of ELA as a preprocessing step, the VGG16 + CNN + LSTM model can benefit from enhanced forgery detection, improved tamper localization, noise reduction, and detection of compression artifacts. This can lead to more accurate and reliable forgery detection results.



Figure 9 (a). Actual Image

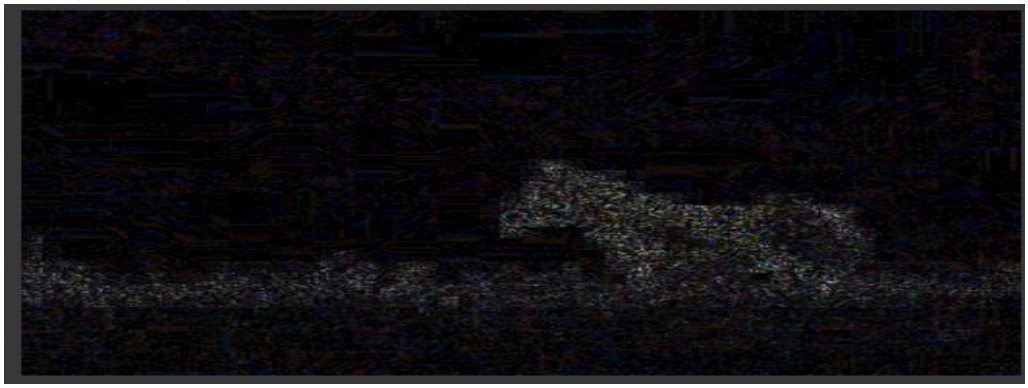


Figure 9 (b). ELA of the Real Image



Figure 10 (a). Fake Image



Figure 10 (b). ELA of the Fake Image

Fig .9 (a) and Fig.9 (b) show the Actual image and ELA of the Actual image. Fig.10 (a) and Fig.10 (b) show the Fake image and ELA of the Fake image. Fig 11 (a) shows the graph of Total loss and Validation loss and Fig. 11(b) shows the graph of Total Accuracy Vs Total Validation accuracy by considering the number of epochs and Fig 12. Shows the confusion matrix for image dataset.

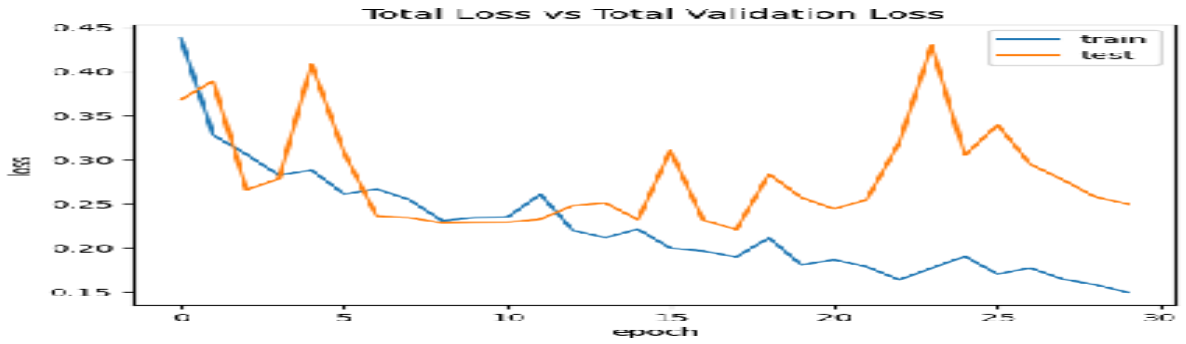


Figure 11 (a). Epoch Vs Loss for image dataset.

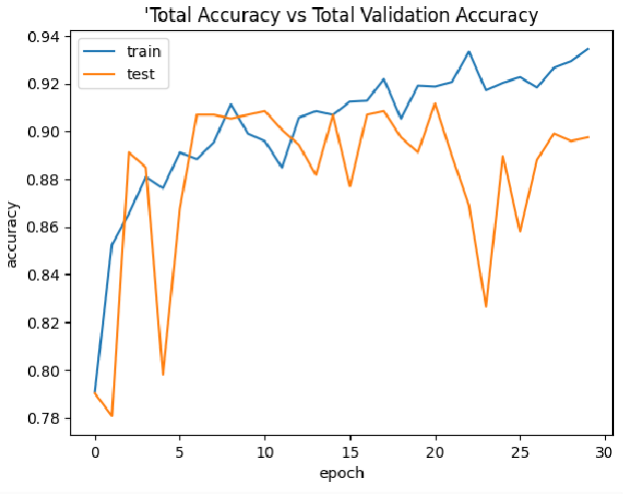


Figure 11(b). Epoch Vs Accuracy for image dataset

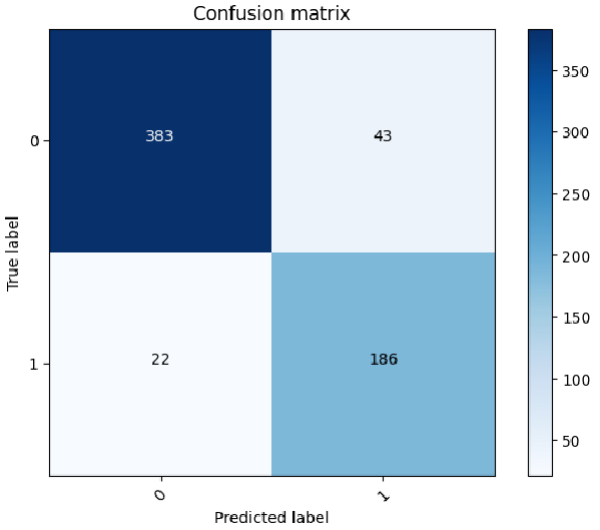


Figure 12. Confusion matrix for a Image dataset

iii) ROC Curve:

The effectiveness of a binary classification model can be shown visually in the ROC (Receiver Operating Characteristic) curve. It shows how true positive rate (TPR) and false positive rate (FPR) balance out across a range of classification levels.

The ROC curve is constructed by plotting TPR against FPR while adjusting the classification threshold.TPR, also known as sensitivity, recall, or hit rate, represents the ratio of accurately predicted positive instances (true positives) to the total count of actual positive instances. On the other hand, FPR denotes the number of false positives, which are instances of negative data that were predicted incorrectly, to all occurrences of negative data.

$$TPR = \frac{TP}{TP+FN} \quad (15)$$

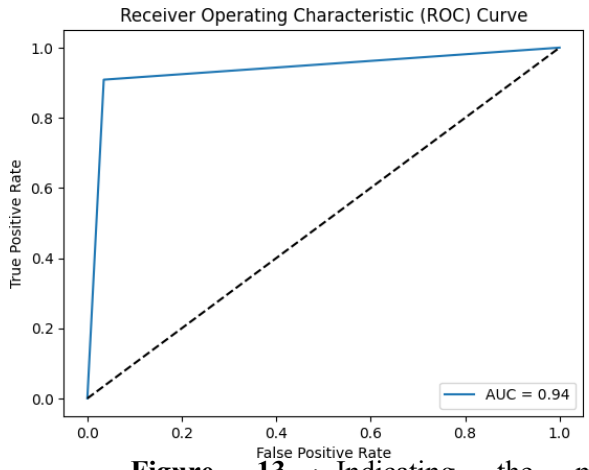


Figure 13. Indicating the model performance using ROC Curve

6. CONCLUSION

Surveillance cameras are essential in today's environment for both legal and security purposes. However, modifying images and videos has become easier because to the widespread availability of sophisticated video editing software. This calls into question the veracity and integrity of recorded visual evidence, which may have a big influence on court proceedings and investigations. Therefore, before using surveillance images and videos as evidence, it is crucial to confirm their authenticity. The limits of tiny modified objects cannot be effectively identified by traditional deep learning methods for detecting photo copy-move forgeries. They don't make good use of spatial information and high-resolution encoded features, which are essential for accurate edge identification of tiny objects. In order to enhance edge detection for tiny objects, this suggested article makes use of spatial information included in encoded characteristics. In order to distinguish between real and modified video frames, the research presents a customized CNN-LSTM architecture that makes use of transfer learning. Additionally, model assessment is done using ensemble learning approaches. The results of the experiments demonstrate the efficacy of the suggested paradigm, surpassing previous techniques. while recognizing forged films with a static backdrop, the model achieves an accuracy of 95.00%, and while identifying forged videos with a moving background, it achieves an accuracy of 98.67%. The model's performance is further validated by extensive testing against a variety of social media photos and videos. These findings demonstrate the effectiveness of the suggested method in detecting video forgeries and provide a dependable way to confirm the legitimacy of visual evidence in security and legal applications.

$$FPR = \frac{FP}{FP+TN}$$

REFERENCES

[1] L. Fazio, "Curbing fake news: Here's why visuals are the most potent form of misinformation" from ttps://scroll.in/article/953395/curbing-fake-news-here's-why-visuals-are-the most-potent-form-of-misinformation last viewed on January 2, 2020. [Google Scholar] [Cross Ref] [Publisher Link] [2] J. Tao, L. Jia, and Y. You, Review of sensing and imaging-based passive-blind detection in digital video forgeries. Vol. 10244, pp. 320-325, International Conference on Optoelectronics and Microelectronics Technology and Application, 2017. [Cross Ref] [Google Scholar] [Publisher Link] SPIE [3] R. Eveleth, Hurricane Sandy: Five indicators of a phony image. 2020. [Google Scholar] [Cross Ref] [Publisher Link] [4] Autoencoder with recurrent neural networks for video forgery detection, D. D'avino, D. Cozzolino, G. Poggi, and L. Verdoliva. 2017. arXiv preprint arXiv:1708.08754. [Google Scholar] [Cross Ref] [Publisher Link] [5] Coarse-to-fine copy-move forgery detection for video forensics, S. Jia, Z. Xu, H. Wang, C. Feng, and T. Wang. IEEE Access, 6 (2018), pp. 25323-25335. [Google Scholar] [Cross Ref] [Publisher Link] [6] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting video forgeries based on noise characteristics," Advances Image Video Technology, vol. 5414, pp. 306–317, 2009. [Google Scholar] [Cross Ref] [Publisher Link]

[7] Passive copy-move forgery detection in videos, R. C. Pandey, S. K. Singh, and K. K. Shukla. IEEE International Conference on Computer and Communication Technology (ICCCT), 2014, pp. 301-306. [Google Scholar] [Cross Ref] [Publisher Link]

[8]D.-K. Hyun, S.-J. Ryu, H.-Y. Lee, and H.-K. Lee, "Upscale crop and partial manipulation detection in surveillance video based on sensor pattern noise," Sensors, vol. 13, no. 9, pp. 12605–12631, 2013. [Google Scholar] [Cross Ref] [Publisher Link]

[9] J. Goodwin and G. Chetty, "Blind video tamper detection based on fusion of source features," in Proc. Int. Conf. Digit. Image Comput., Techn. Appl. Noosa, QLD, Australia, Dec. 2011, pp. 608–613. [Google Scholar] [Cross Ref] [Publisher Link]

[10] G. Li, S. Sun, D. Tu, and Q. Wu. Using dwt and svd, a sorted neighborhood method may identify duplicate areas in picture forgeries. In the 2007 IEEE International Conference on Multimedia and Expo. [Google Scholar] [Cross Ref] [Publisher Link]

[11] W. Luo, G. Qiu, and J. Huang. reliable identification of digital picture region-duplication forgeries. 2006 saw the 18th International Conference on Pattern Recognition. [Cross Ref] [Google Scholar] [Publisher Link] [12] B. Mahdian and S. Saic. The use of blur moment invariants to detect copy-move forgeries. International Journal of Forensic Science, Volume 171, Issue 2, pages 180–189, 2007. [Google Scholar] [Cross Ref] [Publisher Link]

[13] W. Wang, T. Tan, and J. Dong. investigating the impacts of dct coefficient quantization for the detection of local tampering. IEEE Transactions on Information Forensics and Security, volume 9, issue 10, pages 1653–1666, 2014. [Cross Ref] [Google Scholar] [Publisher Link] [14] Y. Richao, Gaobo, and Z. Ningbo, "Detection of object-based manipulation by the statistical features of object contour," Forensic Science International, vol. 236, pp. 164–169, March 2014. [Cross Ref] [Google Scholar] [Publisher Link] [15] L. Su and C. Li, "A novel passive forgery detection algorithm for video region duplication," Multidimensional Syst. Signal Process. vol. 29, no. 3, pp. 1173–1190, July 2018. [Google Scholar] [Cross Ref] [Publisher Link]

[16] L. Su, T. Huang, and J. Yang, "A compressive sensing-based video forgery detection algorithm," Multimedia Tools Appl., vol. 74, no. 17, pp. 6641–6656, Sep. 2015. [Cross Ref] [Google Scholar] [Publisher Link] [17] V. Conotter, H. Farid, and J. F. O'Brien, "Exposing digital forgeries in ballistic motion," IEEE Trans. Inf. Forensics Security, vol. 7, no. 1, February 2012, pages. 283–296. [Google Scholar] [Cross Ref] [Publisher Link]

[18] L. Li, X. Wang, W. Zhang, G. Yang, and G. Hu, "Detecting removed object from video with stationary background," Proceedings of the International Workshop Digit. Forensics Watermarking, Taipei, Taiwan, pp. 242–252, 2013. [Cross Ref] [Google Scholar] [Publisher Link] [19] Shan Jia, Zhengquan Xu, Hao Wang, Chunhui Feng, and Tao Wang, "Coarse-to-Fine Copy-Move Forgery Detection for Video Forensics," IEEE Access, vol. 6, pp. 25323-25335, 2018. [Google Scholar] [Cross Ref] [Publisher Link]

[20] O. I. Al-Sanjary, A. A. Ahmed, A. A. B. Jaharadak, M. A. M. Ali, and H.M. Zangana, "Detection clones an object movement using an optical flow approach," in IEEE Symp. Comput. Appl. Ind. Electron. (ISCAIE) Penang, Malaysia, April, pp. 388–394, 2018. [Google Scholar] [Cross Ref] [Publisher Link]

[21] V. Subramanyam and S. Emmanuel, "Video forgery detection using HOG features and compression properties," in IEEE 14th International Workshop Multimedia Signal Process (MMSP), Banff, AB, Canada, September, pp. 89–94, 2012. [Cross Ref] [Google Scholar] [Publisher Link] [22] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP for object localization," in Proc. CVPR, El Paso, CA, USA, pp. 1393–1400, 2011. [Cross Ref] [Google Scholar] [Publisher Link]

[23] L. Su and C. Li, "A novel passive forgery detection algorithm for video region duplication," Multidimensional Syst. Signal Process. vol. 29, no. 3, pp. 1173–1190, July 2018. [Google Scholar] [Cross Ref] [Publisher Link]

[24] L. Su, T. Huang, and J. Yang, "A compressive sensing-based video forgery detection algorithm," Multimedia Tools Appl., vol. 74, no. 17, pp. 6641–6656, Sep. 2015. [Cross Ref] [Google Scholar] [Publisher Link] [25] V. Conotter, H. Farid, and J. F. O'Brien, "Exposing digital forgeries in ballistic motion," IEEE Trans. Inf. Forensics Security, vol. 7, no. 1, February 2012, pages. 283–296. [Google Scholar] [Cross Ref] [Publisher Link]

[26] Y. Yao, Y. Shi, S. Weng, and B. Guan, "Deep learning for object-based forgery detection in advanced video," Symmetry, vol. 10, no. 1, p. 3, 2017. [Cross Ref] [Google Scholar] [Publisher Link]

[27] D. Xu, X. Shen, Y. Lyu, X. Du, and F. Feng, MC-Net: Acquiring complementary characteristics for localization of image alteration. Int. J. Intell. Syst. 2022. [Google Scholar] [Cross Ref] [Publisher Link]

[28] Y. Wu, W. Abdalmageed, and P. Natarajan, Mantra-net: Manipulation tracing network for anomalous feature identification and location of picture forgeries. Within IEEE/CVF