Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **4**, December **2025**

DeepForgeryDetect: Enhancing social media security through Deep Learning based Forgery Detection

Litty Koshy*

Department of CSE
SCMS School of Engineering and Technology, Ernakulam, India
grace.koshy@gmail.com

Prayla Shyry S.

Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India
praylashyry.cse@sathyabama.ac.in

*Corresponding author: Litty Koshy
Received May 20, 2025, revised September 8, 2025, accepted September 24, 2025.

Abstract. *Nowadays, security and legal applications both heavily rely on surveillance cameras. However, using various video editing software, photos and video recordings can be easily edited. The captured information can be used as evidence of crime scene investigation. The integrity of image and video can have a significant impact on the outcome of a legal case or investigation. Therefore, it's critical to confirm the authenticity of surveillance photos and videos before using them as proof in any legal situation. Existing deep learning-based approaches for picture copy-move forgery detection are ineffective in identifying the boundaries of small, manipulated objects because they do not efficiently leverage high-resolution encoded features. The recovered frame in this proposed article utilizes the spatial information present in the feature maps and increases the precision of identification for small objects. The paper presents a customized CNN -LSTM layer that uses the transfer learning to distinguish between genuine and altered frames. The model evaluation is done using ensemble learning. By incorporating advanced neural network architectures, the model can effectively learn and extract complex features from both videos and images, enhancing its ability to detect copy move forgery. The model produces an accuracy of 95.00% for identifying forged videos of static background, 99.67% for identifying forged videos with moving background and 95.08% for identifying and localizing the forged Image. The model is subjected to testing using a variety of social media images and videos. The experimental data reveals that the suggested model offers a substantial improvement in accuracy compared to cutting-edge methods.*
Keywords: Images; Videos; Social media networks; Custom CNN; LSTM

1. **Introduction.** The distribution of large volumes of multimedia content is made easier by the increasing use of social media platforms. Simultaneously, the development of smartphone apps and digital editing programs like Photoshop, Gimp has made it increasingly effortless for users to modify the images and videos found on these platforms. Consequently, there has been a notable increase in the prevalence of fraudulent images and videos circulating within these networks, enabling their exploitation for fraudulent activities.

Copy-move forgery is a digital image manipulation technique involving the duplication and placement of a section of the image within the same picture, often used to hide

or replicate an object or change its appearance. It involves selecting a specific region or object within the image and replicating it, then placing the duplicate in a distinct position within the same image. This technique is commonly used to tamper or manipulate the image content discreetly 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]. In 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. Figure 1(a) and (b) showcase image sourced from CASIA 2.0 dataset.

(a) (b)

FIGURE 1. Example of copy move forgery technique (a). Real Image (b). Fake image generated from real Image.

Forgery detection of large videos with different frame rates is not possible in real time due to the limited processing power, lack of generalizability, and decreased performance accuracy. This study investigated the problem of detecting forged photos with smooth borders and detecting forged films with static and moving backgrounds at different frame speeds using deep learning. These fake images spread globally after becoming widely popular on social media. While some fake images are innocuous, others could cause fear or damage someone's reputation [2]. As a result, it's essential to create systems that can detect fake multimedia.

2. **Related Work.** Multimedia forgery attacks are divided into: (i) temporal, (ii) spatial, and (iii) spatio-temporal [3]. 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 involving within or across frames through addition, removal or replacement [4]. The spatio-temporal domain is formed by merging the spatial and temporal domains.

This research paper examines the detection of spatial type of forgery, which involves copy-move tampering method. Copy-Move forgery involves replicating image or video segments and embedding them within the same media source, while removal of objects is achieved by filling the region with adjacent pixels sourced from the same frame. An instance of spatial type of forgery is depicted in the figure below. Figure 2(a) displays an authentic video frame sequence and while Figure 2(b) presents a tampered video sequence.

Figure 2(a) shows the actual frame sequences and Figure 2(b) illustrates an example where the frame sequence is duplicated and inserted into another part of the same video.

(A) Actual video frames

(B) Fake video frames

FIGURE 2. Example of spatial forgery. (a) shows the actual frame sequences and (b) illustrates an example where the frame sequence is duplicated.

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, 36], authors used SIFT characteristics, noise residual, and pixel correlation for identifying forgery. In [7], the authors discussed sensor pattern noise to identify forgeries. Authors of [8] also employed noise residuals, quantization attributes and their transformation in order to identify forgery. Most of the image tamper classification methods is based on frequency domain attributes or statistical properties of an image are often explored in [9, 10, 11, 12]. In [13] feature vectors are generated by extracting the mean gradient from each color channel along with additional statistical features of wavelet coefficients. These feature vectors serve as input for a support vector machine (SVM) to differentiate between real objects and counterfeit objects.

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 compressions is used in [17] to detect image tampering specific to 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. In [18], inter-frame copy-move forgery is detected using optical flow (OF). The initial phase of detection involves analyzing the optical flow to 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 anomalies in optical flow and utilize a Dynamic Time Warping (DTW) matching technique to detect copy-move forgeries in media. Both studies highlight the strengths in detecting motion related manipulations.

Convolutional Neural Networks (CNNs) to identify forged images through the detection of abnormal traces such as inconsistencies in illumination direction and noise throughout the entire picture. In [29] uses CNN to assess whether an image is tampered or not. Addressing the issue of low accuracy in the detection results of CNN based copy move

forgery is the key objective of the suggested approach. The primary focus of CNN-based methods 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. In [37], the video clips are divided into static and dynamic segments using global motion estimation then each key frames are selected using extended version of temporally maximum occurrence frame (ETMOF) method for static part and Perceived Motion Energy (PME) for dynamic parts.

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 is a more potent model for both video and picture fraud detection because it makes use of the CNN's capacity to extract spatial data from each frame. 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 various applications.

The remaining portion 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 efficient network architecture for video and image forgery detection which consists of two primary elements: DeepForgeryDetect employs a sophisticated system that combines both Long Short-Term Memory (LSTM) and a Convolutional Neural Network (CNN) technology to accurately identify forged images and videos. The CNN component additionally makes use of the VGG16 model, a deep neural network architecture that has been pre-trained on a large image dataset for image categorization.

The VGG16 network comprises of thirteen convolution layers along with three fully connected layers and is known for its capacity to capture high-level semantic features from individual frames of the input video. These features go beyond basic pixel information and capture more abstract concepts like texture, shapes and objects. After the CNNs have extracted features, a sequence of LSTM layers is used to model the temporal dependencies between successive video frames. As these layers process the current frame, they preserve the context of earlier frames. By adapting the VGG16 network's parameters to the objective of video forgery detection, a transfer learning technique is used to improve and optimize the model's performance. The pre-trained VGG16's weight parameters are locked, and CNN and LSTM layers are trained on a sizable dataset of images and videos, both with and without forgeries.

The output of the model is a binary classification that classifies every frame as either authentic or fake. The efficacy of integrating CNN and LSTM networks for forgery detection is demonstrated by the model's excellent accuracy on a range of video and image forgery datasets.

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 VGG16 features can further refine this representation and model the temporal dependencies in the video, leading to higher accuracy in detecting video forgeries. Figure 3 visualizes the key components and their relationships within the proposed system.

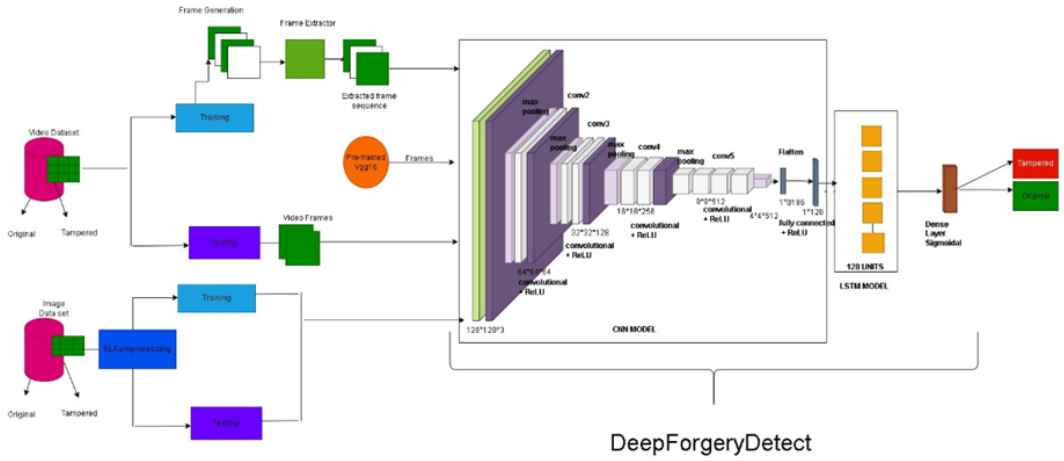


FIGURE 3. System Architecture of the proposed CNN-LSTM Model

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 utilize spatial filters to extract features that remain consistent variations 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.

In order to detect visual forgeries, the network architecture combines the spatial features prowess of CNNs with the temporal dependency modeling capabilities of long short-term memory (LSTM) model. In this model, the CNN component utilizes the trained VGG16 model to function as a feature extractor. The initial image dimensions are $128 \times 128 \times 3$; undergo feature extraction using the VGG16 architecture. Following this, the VGG16 model's output is transformed into a one-dimensional feature vector using Flatten layer. Subsequently, the model undergoes flattening and an extra dense layer with 128 units and a ReLU activation function is incorporated. The output of the dense layer is then transformed into a tensor of dimensions $(1, 128)$ by a Reshape layer.

An input for the LSTM model is provided by this tensor. The 128-unit single LSTM layer of the LSTM model is followed by a 2-unit dense output layer that uses a sigmoid activation function, that is very useful for binary classification. After processing the input feature sequence that was received from the CNN, the LSTM layer creates a single vector representation that contains all the information from the input sequence. The final classification result is then obtained by feeding this final vector through the output layer. The entire architecture is trained using binary cross-entropy loss and the Adam optimizer.

3.1. **Training the network.**

3.1.1. *VGG16.* The VGG16 architecture comprises of thirteen convolution layers, followed by three fully connected layers. The equations for the forward pass:

1. Convolution layer:
$$p(i) = m(i) * q(i-1) + b(i) \tag{1}$$
$$q(i) = \text{ReLU}(p(i)) \tag{2}$$
where $m(i)$ and $b(i)$ are the weight and bias parameters for the i-th layer, $p(i)$ is the input to the i-th layer, $q(i)$ is the output of the i-th layer, $q(i-1)$ is the output of the $(i-1)$-th layer, and ReLU is the activation function.
2. Max pooling layer:
$$q(i) = \text{MaxPool}(q(i-1)) \tag{3}$$
where MaxPool is the max pooling function.
3. Fully connected layer:
$$p(i) = m(i) * p(i-1) + b(i) \tag{4}$$
$$q(i) = \text{ReLU}(p(i)) \tag{5}$$
where $q(i)$ is the output of the i-th layer, $q(i-1)$ is the output of the $(i-1)$-th layer, $m(i)$ and $b(i)$ are the weight and bias parameters for the i-th layer, $p(i)$ is the input to the i-th layer, and ReLU is the activation function.
4. Softmax layer:
$$p(i) = m(i) * q(i-1) + b(i) \tag{6}$$
$$q(i) = \text{Softmax}(p(i)) \tag{7}$$
where $q(i)$ is the output of the i-th layer, $q(i-1)$ is the output of the $(i-1)$-th layer, $m(i)$ and $b(i)$ are the weight and bias parameters for the i-th layer, $p(i)$ is the input to the i-th layer, and ReLU is the activation function.

Algorithm 1 Learning algorithm for training the VGG16 model

1: **Input:** Input Image Size is $128 \times 128 \times 3$
2: **Output:** Feature Vector
3: Initialize the VGG16 model with pre-trained weights obtained from ImageNet.
4: Freeze the weights of the convolution layers.
5: Add a new fully connected layer to the VGG16 model for classification.
6: Train the fully connected layer by utilizing back propagation with the video forgery detection dataset.

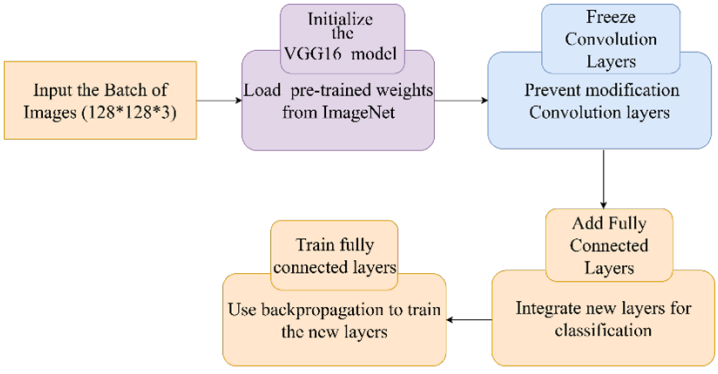


FIGURE 4. Training the VGG16 model

Figure 4 depicts the flow of VGG16 learning algorithm.

3.1.2. *Convolutional Neural Network.* A CNN is adept at image processing and object detection within the realm of neural networks. The equations for the forward pass:

1. Convolution layer:
$$p(i) = m(i) * q(i-1) + b(i) \tag{8}$$
$$q(i) = \text{ReLU}(p(i)) \tag{9}$$
where $m(i)$ and $b(i)$ are the weight and bias parameters for the i-th layer, $p(i)$ is the input to the i-th layer, $q(i)$ is the output of the i-th layer, $q(i-1)$ is the output of the $(i-1)$-th layer, and ReLU is the activation function.
2. Max pooling layer:
$$q(i) = \text{MaxPool}(q(i-1)) \tag{10}$$
where MaxPool is the max pooling function.
3. Fully connected layer:
$$p(i) = m(i) * p(i-1) + b(i) \tag{11}$$
$$q(i) = \text{ReLU}(p(i)) \tag{12}$$
where $m(i)$ and $b(i)$ are the weight and bias parameters for the i-th layer, $p(i)$ is the input to the i-th layer, $q(i)$ is the output of the i-th layer, $q(i-1)$ is the output of the $(i-1)$-th layer, and ReLU is the activation function.

Algorithm 2 Learning algorithm for Convolution Neural Network

1: **Input:** a batch of images with shape $(\text{batch\_size}, 128, 128, 3)$, where batch_size is selected according to size of the dataset.
2: **Output:** a batch of feature vectors with shape $(\text{batch\_size}, 1, 128)$
3: Apply the pre-trained VGG16 model to the batch of images to extract their features.
Input shape: $(\text{batch\_size}, 128, 128, 3)$
Output shape: $(\text{batch\_size}, 4, 4, 512)$
4: Construct convolution Layers.
5: Flatten the output from the VGG16 model to obtain a feature vector for each image in the batch.
Input shape: $(\text{batch\_size}, 4, 4, 512)$
Output shape: $(\text{batch\_size}, 8192)$
6: Apply 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: $(\text{batch\_size}, 8192)$
Output shape: $(\text{batch\_size}, 128)$
Output equation:
$$x = \text{ReLU}(Wx + b) \tag{13}$$
7: Reshape the output of the fully connected layer to have a shape of $(\text{batch\_size}, 1, 128)$
Input shape: $(\text{batch\_size}, 128)$
Output shape: $(\text{batch\_size}, 1, 128)$

The flow of the learning algorithm of VGG 16 and CNN is depicted in Figure 4 and Figure 5.

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 are both 128 and image_channels is 3 (corresponding to RGB channels). The output shape of the VGG16 model results in $(\text{batch\_size}, 4, 4, 512)$,

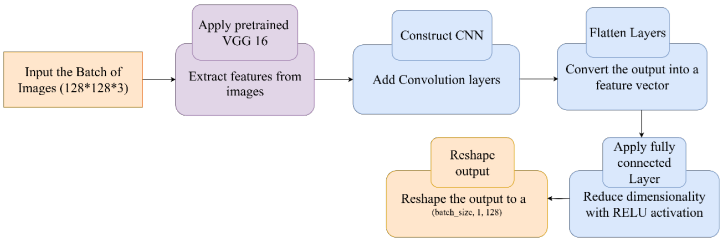


FIGURE 5. Flow diagram of CNN Learning

Algorithm 3 Learning algorithm for the LSTM model

1: **Inputs:** Input tensor of shape (batch_size, timesteps, features), where features dimension should match the output shape of the CNN segment of the model.
2: **Outputs:** Output tensor of shape (batch_size, lstm_units) from the second LSTM layer.
3: Define an LSTM model with two LSTM layers:
First LSTM layer should have 128 units and return sequences.
A Dense layer containing a single neuron with 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 with following configurations:
Employ binary cross-entropy loss.
Utilize the Adam optimizer.
Measure 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.

representing the feature maps extracted from the final convolution 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 signifies the count of frames within the input video sequence. 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 LSTM's capability to discern patterns within the sequential data. Subsequently, the output generated by the LSTM layer is directed to the subsequent layer. After the LSTM layer, a Dense layer is added with 2 units and a sigmoid activation function Following the LSTM layer, a Dense layer with 2 units and a sigmoid activation function is incorporated. This Dense layer serves to reshape the LSTM layer's output into a probability distribution encompassing the two classes. By employing

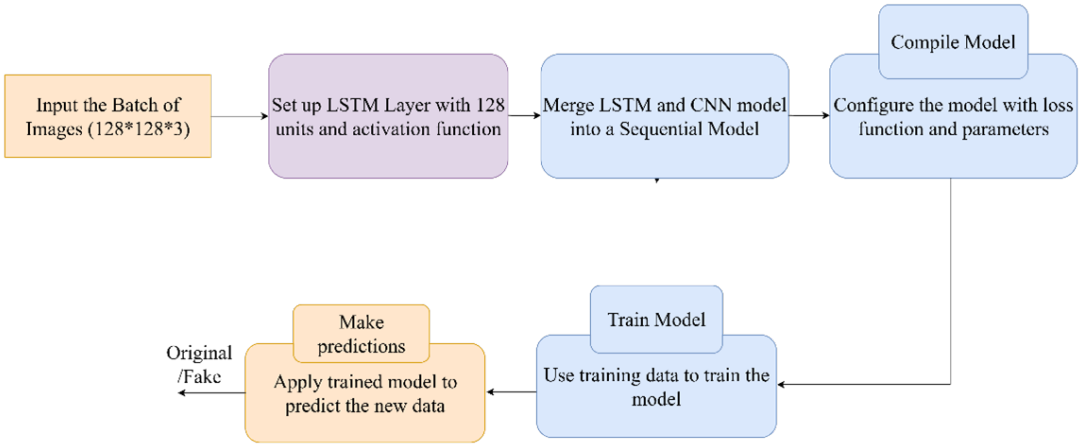


FIGURE 6. Learning of VGG16-CNN-LSTM Model

the sigmoid activation function, the resulting values are constrained to a range between 0 and 1, denoting the probability of belonging to each respective class.

The number of Dense layer units corresponds to the number of classes in the classification task. In this instance, it is set to 2 i.e., the video or image is original or tampered. The flow of Learning algorithm is depicted in Figure 6.

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 128×128. To determine the optimal combination of hyper parameters, numerous iterations were conducted, varying batch sizes and dropout probabilities. The proposed system was evaluated on publicly available datasets such as VIFFD [30], ViFoDAC [31], REWIND_3D [32], CASIA 2.0 [33] and one custom social-media dataset for robustness checks under real-world compression.

- **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.
- **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.
- **ViFoDAC:** is a dataset comprises 16 authentic videos alongside 16 manipulated videos, with durations spanning between 10 and 62 seconds each. The authentic content was recorded using 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.
- **REWIND_3D:** is a dataset that serves the purpose of detecting forgery involving small objects within videos. It encompasses 10 original videos and 10 tampered videos, each lasting between 6 and 18 seconds. A stationary camera was used to record the videos, ensuring a constant background throughout. This dataset is particularly effective in identifying fakes where small objects are incorporated into pre-existing videos without causing changes to the background.

- **MICC-F220:** This dataset comprises a collection of 220 images, equally divided into 110 manipulated images and 110 authentic originals.
- **Custom set:** compressed and down-scale images and videos gathered from social platforms for qualitative assessment.

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 backgrounds. 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 findings demonstrate the functionality of the suggested VGG16-CNN-LSTM architecture 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 achieved 90% validation accuracy and 95% training accuracy.

This means that the model is able to accurately generalize to new data and has successfully captured the fundamental properties of the data. The model obtained a higher training accuracy of 99.67% and a validation accuracy of 95% in the moving camera scenario. This suggests that the model can withstand fluctuations brought on by the camera's movement more robustly. Additionally, the model is trained on several image datasets, yielding 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. By contrasting the model's predications with the actual labels, a confusion matrix offers information about how well the classification model performed on a particular test set of data.

True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the four main components of the confusion matrix. Table 1 shows the performance analysis of the developed model using different datasets and different test split.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

5.1. **Evaluation of the DeepForgeryDetect Network across diverse dataset and varying test split ratios.** Table 1 presents the performance evaluation of the proposed model across different datasets with varying test split ratios.

TABLE 1. Model performance across different dataset and different test split

Test Split of Video Dataset	**Accuracy of the proposed Model (%)**		
	Dataset	**80-20**	**75-25**
Static background	VIFFD	94.27	95.00
	REWIND_3D	93.47	94.58
Moving background	ViFoDAC	98.60	99.67
Image dataset	CASIA_2.0	94.08	95.08

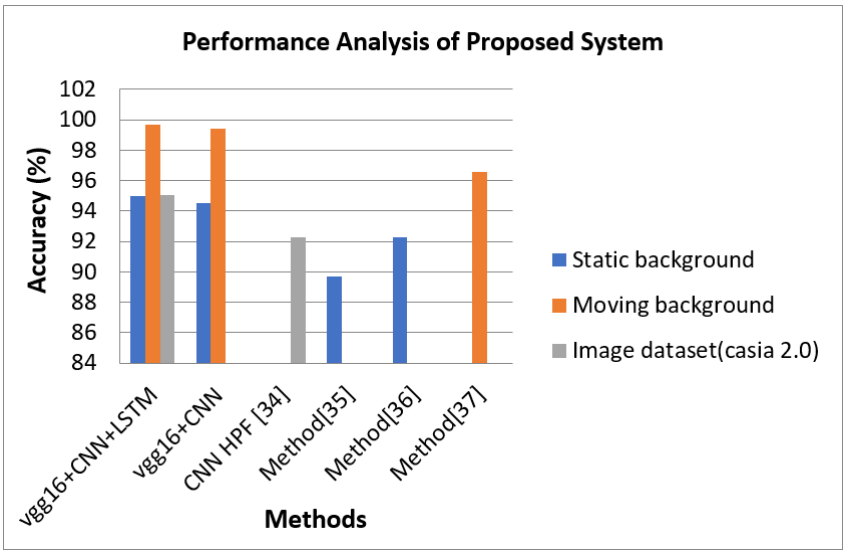


FIGURE 7. Comparison of DeepForgeryDetect Network with state-of-the-art Approaches

5.2. **Performance Analysis of DeepForgeryDetect Network benchmarked against state-of-the-art methods, in terms of accuracy.** Figure 7 shows that the model achieved a higher accuracy of 99.67% on videos with moving background and 95.00% accuracy on static background videos with a test split of 75-25 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.* Figure 8(a) shows the graph of Total loss and Validation loss and Figure 8(b) shows the graph of Total Accuracy Vs Total Validation accuracy by considering the number of epochs and Figure 9 shows the confusion matrix for a scene, which has static background.

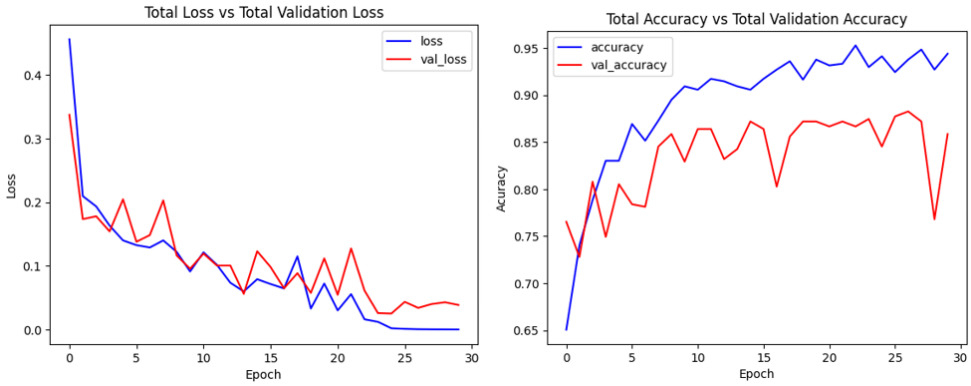


FIGURE 8. (a) Graph showing Epoch Vs Loss in Static Scene. (b) Graph showing Epoch Vs Accuracy in Static Scene.

5.2.2. *Scene is moving and camera is moving.* Figure 10(a) and Figure 10(b) depict the graphs of total loss Vs validation loss and total accuracy Vs total validation accuracy,

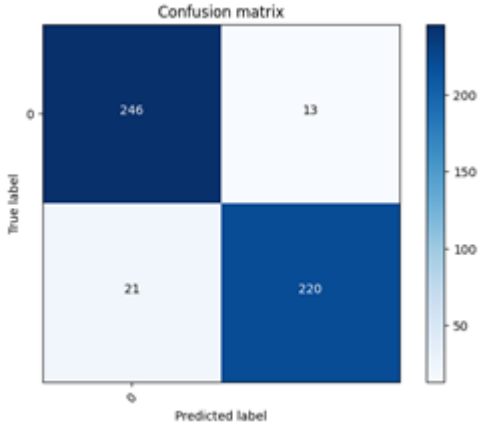


FIGURE 9. Confusion matrix for a static background

respectively, taking into account the number of epochs. and Figure 11 shows the confusion matrix for a scene, which has moving background.

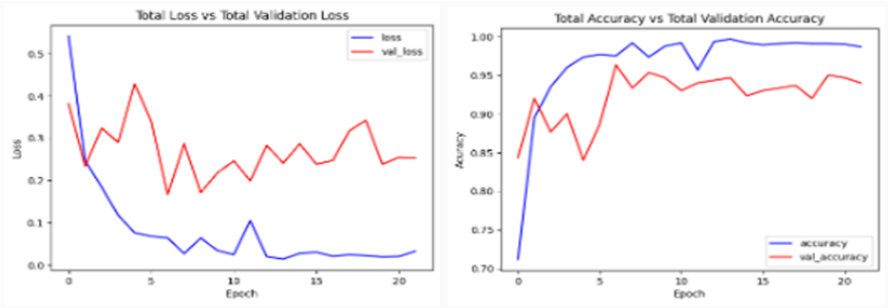


FIGURE 10. (a) Graph showing Epoch Vs Loss in moving Scene. (b) Graph showing Epoch Vs Accuracy in moving Scene.

5.2.3. *Hyperparameter Selection.* The number of LSTM units was determined through a limited grid search across 64, 128 and 256 units. Model fails to capture temporal dependencies effectively with 64 units, while with 256 units increased the training time and causes overfitting with consistent gains in performance. So, a configuration with 128 units is selected with best balance with expressiveness and generalization.

Generalization can be encouraged by training a single model for both image and video forgery detection. This helps to identify common counterfeit patterns and extract relevant information from both images and movies. This deeper comprehension improves the model's performance on invisible data, increasing its resilience and adaptability in identifying different kinds of forgeries. Error Level Analysis (ELA) is used for image preprocessing, while the VGG16 + CNN + LSTM model is used for training. 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. As a result, results for forgery detection may be more precise and trustworthy.

Both the actual image and its ELA are shown in Figure 12(a) and Figure 12(b). Figure 13(a) and Figure 13(b) display the ELA and the fake picture. Figure 14(a) and Figure 14(b) show the graphical representations of total loss vs validation loss and total accuracy versus total validation accuracy, respectively, taking into account the number of epochs. The picture dataset's confusion matrix is displayed in Figure 15.

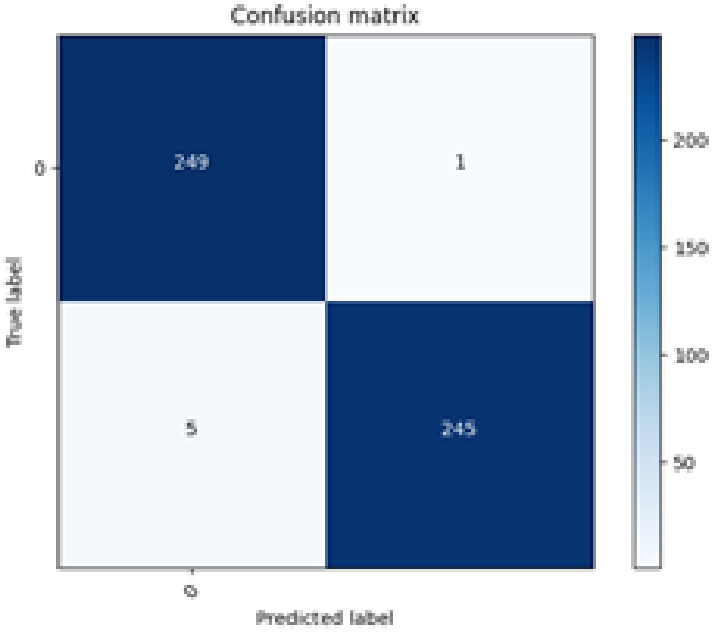
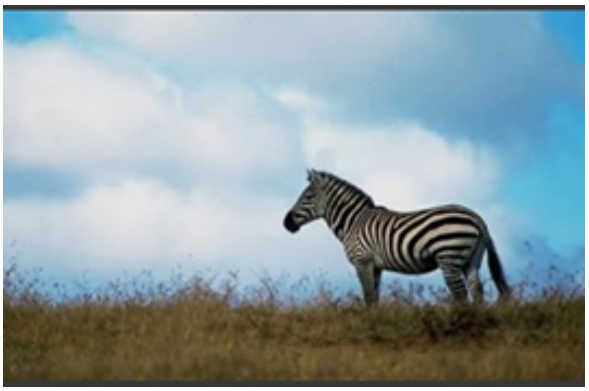


FIGURE 11. Confusion matrix for a Moving Scene



(A) Actual Image

(B) ELA of the Real Image

FIGURE 12. (a) Actual Image. (b) ELA of the Real Image.

(A) Fake Image

(B) ELA of the Fake Image

FIGURE 13. (a) Fake Image. (b) ELA of the Fake Image.

5.2.4. *ROC Curve.* By showing the equilibrium between the true positive rate (TPR) and false positive rate (FPR) at different classification thresholds, the Receiver Operating

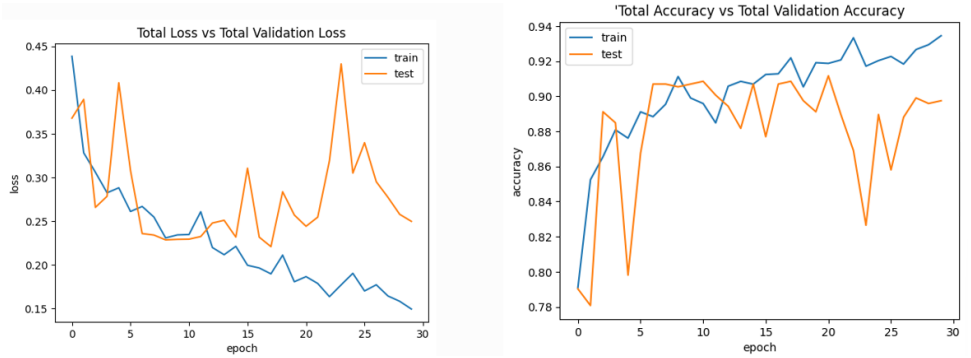


FIGURE 14. (a) Epoch Vs Loss for image dataset. (b) Epoch Vs Accuracy for image dataset.

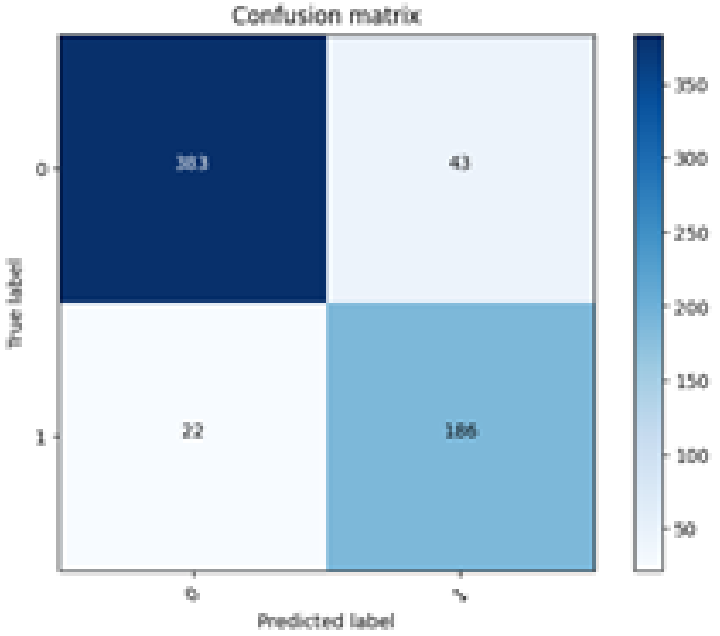


FIGURE 15. Confusion matrix for a Image dataset

Characteristic (ROC) curve graphically represents the performance of a binary classification model. Plotting TPR vs FPR while varying the categorization threshold produces this curve. TPR is the ratio of accurately predicted positive instances (also known as true positives) to the total number of actual positive occurrences. It is also known as sensitivity, recall, or hit rate. On the other hand, the ratio of all true negative cases to falsely anticipated positive cases (false positives) is known as the false positive rate (FPR). The AUC (Area Under the Curve) graph for the suggested system is shown in Figure 16.

$$TPR = \frac{TP}{TP + FN} \tag{15}$$

$$FPR = \frac{FP}{TN + FP} \tag{16}$$

Table 2 presents the performance measures in terms of Precision, Recall, F1-Score Accuracy and AUC on Test dataset.

The proposed model shows very high detection performance on a wide range of datasets. It has the best performance on moving background videos (Precision = 98.6%, Recall =

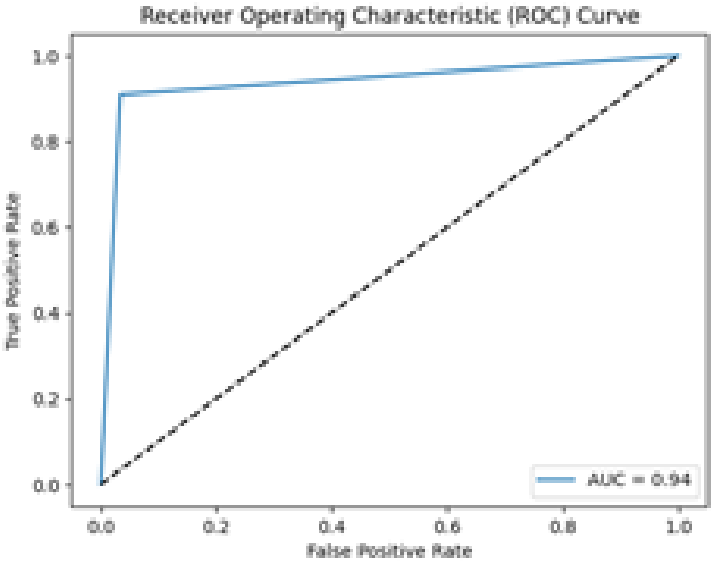


FIGURE 16. Indicating the model performance using ROC Curve

TABLE 2. Performance measures in terms of Precision, Recall, F1-Score Accuracy and AUC on Test dataset

Case / Dataset	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)	AUC
Case 1 – Static background	92.13	94.98	93.53	93.2	0.96
Case 2 – Moving background	98.6	98.03	98.81	99.67	0.99
Case 3 – Image dataset	94.57	89.91	92.18	89.78	0.96

98.03%, F1-score = 98.81%, AUC $\approx$ 0.99), showing great accuracy on difficult dynamic situations. The performance holds well on static backgrounds and image sets, with all AUC scores $\geq$ 0.96, verifying the model's robust generalization on changing content types and background complexities.

6. **Conclusion.** In today's world, surveillance cameras are essential for both security and legal applications. However, the widespread availability of advanced video editing software has simplified the process of altering photos and video recordings. This raises concerns about the authenticity and integrity of captured visual evidence, which can have a significant impact on legal cases and investigations. Thus, it is essential to verify the originality of surveillance photos and videos before utilizing them as evidence. Traditional methods for detecting picture copy forgery have limitations in accurately identifying boundaries of small manipulated objects. They fail to effectively utilize high-resolution encoded features and spatial information, which are crucial for precise edge detection of small objects. This proposed article leverage spatial information present in encoded features; leading to improved edge identification for small objects. The paper introduces a customized CNN-LSTM architecture that utilizes transfer learning to differentiate between genuine and altered frames in videos. Additionally, ensemble learning techniques are employed for model evaluation. The experimental results showcase the effectiveness of the proposed model, outperforming current state-of-the-art methods.

The model achieves an accuracy of 95.00% in identifying forged videos with a static background and 99.67% in identifying forged videos with a moving background. Extensive testing against various social media images and videos further validates the model's performance. These results highlight the superiority of the proposed approach in video

forgery detection, providing a reliable solution for verifying the authenticity of visual evidence in security and legal applications.

Ethical Approval. No human or animal research conducted by the authors is included in this article.

Funding. The authors declare that no money, grants, or other kinds of support were used to create this paper.

Conflict of Interest. The authors have no justifiable financial or non-financial motivations to make this content public.

Informed Consent. No human or animal research conducted by the authors is included in this article.

Data Availability statement. The data used in this study is available upon request from the corresponding author. Researcher's interested in accessing the data should contact `grace.koshy@gmail.com` to initiate the data access process.

Authorship Contributions. Conceptualization: Litty Koshy; **Methodology:** Litty Koshy; **Formal analysis and investigation:** Litty Koshy, Dr. S. Prayla Shyry; **Writing - original draft preparation:** Litty Koshy; **Writing - review and editing:** Litty Koshy; **Resources:** Litty Koshy, Dr. S. Prayla Shyry; **Supervisions:** Dr. S. Prayla Shyry.

REFERENCES

[1] L. Fazio, "Curbing fake news: Here's why visuals are the most potent form of misinformation," 2020. [Online]. Available: `https://scroll.in/article/953395/curbing-fake-news-here's-why-visuals-are-the-most-potent-form-of-misinformation`. Accessed on: Jan. 2, 2021.

[2] R. Eveleth, "Hurricane Sandy: Five ways to spot a fake photograph," BBC Future, 2014. [Online]. Available: `https://www.bbc.com/future/article/20121031-how-to-spot-a-fake-sandy-photo`.

[3] J. Tao, L. Jia, and Y. You, "Review of passive-blind detection in digital video forgery based on sensing and imaging techniques," in *Proc. Int. Conf. Optoelectron. Microelectron. Technol. Appl.*, Shanghai, China, Jan. 2017, Art. no. 102441.

[4] D. D'Avino, D. Cozzolino, G. Poggi, and L. Verdoliva, "Autoencoder with recurrent neural networks for video forgery detection," University Federico II of Naples, Naples, Italy, 2017.

[5] S. Jia, Z. Xu, H. Wang, C. Feng, and T. Wang, "Coarse-to-fine copy-move forgery detection for video forensics," *IEEE Access*, vol. 6, pp. 25323–25335, 2018.

[6] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting video forgeries based on noise characteristics," in *Advances in Image and Video Technology*, vol. 5414. Tokyo, Japan: Springer, 2009, pp. 306–317.

[7] R. C. Pandey, S. K. Singh, and K. K. Shukla, "Passive copy-move forgery detection in videos," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, Allahabad, India, Sep. 2014, pp. 301–306.

[8] D. K. Hyun, S.-J. Ryu, H. Y. Lee, and H. K. Lee, "Detection of upscale crop and partial manipulation in surveillance video based on sensor pattern noise," *Sensors*, vol. 13, no. 9, pp. 12605–12631, 2013.

[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.

[10] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *IEEE International Conference on Multimedia and Expo*, 2007.

[11] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *18th International Conference on Pattern Recognition*, 2006.

[12] B. Mahdian and S. Saic, "Detection of copy–move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2, pp. 180–189, 2007.

[13] W. Wang, J. Dong, and T. Tan, "Exploring DCT coefficient quantization effects for local tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1653–1666, 2014.
[14] Y. Richao, Y. Gaobo, and Z. Ningbo, "Detection of object-based manipulation by the statistical features of object contour," *Forensic Sci. Int.*, vol. 236, pp. 164–169, 2014.
[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, 2018.
[16] L. Su, T. Huang, and J. Yang, "A video forgery detection algorithm based on compressive sensing," *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 6641–6656, 2015.
[17] L. Li, X. Wang, W. Zhang, G. Yang, and G. Hu, "Detecting removed object from video with stationary background," in *Proc. Int. Workshop Digit. Forensics Watermarking*, Taipei, Taiwan, 2013, pp. 242–252.
[18] O. I. Al-Sanjary, A. A. Ahmed, A. A. B. Jaharadak, M. A. M. Ali, and H. M. Zangana, "Detection clone an object movement using an optical flow approach," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang, Malaysia, 2018, pp. 388–394.
[19] V. Subramanyam and S. Emmanuel, "Video forgery detection using HOG features and compression properties," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Banff, AB, Canada, Sep. 2012, pp. 89–94.
[20] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP for object localization," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1393–1400.
[21] 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, 2018.
[22] V. Conotter, J. F. O'Brien, and H. Farid, "Exposing digital forgeries in ballistic motion," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 283–296, 2012.
[23] Y. Yao, Y. Shi, S. Weng, and B. Guan, "Deep learning for detection of object-based forgery in advanced video," *Symmetry*, vol. 10, no. 1, p. 3, 2017.
[24] D. Xu, X. Shen, Y. Lyu, X. Du, and F. Feng, "MC-Net: Learning mutually-complementary features for image manipulation localization," *Int. J. Intell. Syst.*, 2022. doi: 10.1002/int.22826.
[25] Y. Wu, W. Abdalmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9535–9544.
[26] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder-decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, pp. 3286–3300, 2019.
[27] M. A. Elaskily, M. H. Alkinani, A. Sedik, and M. M. Dessouky, "Deep learning based algorithm (ConvLSTM) for copy move forgery detection," *J. Intell. Fuzzy Syst.*, vol. 40, pp. 4385–4405, 2021.
[28] X. H. Nguyen and Y. Hu, "VIFFD - A dataset for detecting video inter-frame forgeries," Mendeley Data, V6, 2020. doi: 10.17632/r3ss3v53sj.6.
[29] S. Pawar, G. Pradhan, B. Goswami, and S. Bhutad, "ViFoDAC- Video Forgery Detection And Classification," IEEE Dataport, 2022. doi: 10.21227/63t2-ea77.
[30] X. H. Nguyen, Y. Hu, G. Khan, V. Le, and T. Truong, "Three-dimensional Region Forgery Detection and Localization in Videos," *International Journal of Image, Graphics and Signal Processing*, vol. 11, pp. 1–13, 2019.
[31] K. M. Hosny, A. M. Mortda, N. A. Lashin, and M. M. Fouda, "A New Method to Detect Splicing Image Forgery Using Convolutional Neural Network," *Applied Sciences*, vol. 13, no. 3, p. 1272, 2023.
[32] B. Singh and D. K. Sharma, "Image Forgery over Social Media Platforms - A Deep Learning Approach for its Detection and Localization," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 705–709.
[33] J. Bakas, R. Naskar, and M. Nappi, "Object-based forgery detection in surveillance video using capsule network," *J Ambient Intell Human Comput*, vol. 14, pp. 3781–3791, 2023.
[34] G. Qadir, S. Yahaya, and A. T. S. Ho, "Surrey university library for forensic analysis (SULFA) of video content," in *Proc. IET Conf. Image Process. (IPR)*, London, U.K., 2012, pp. 1–6.
[35] P. Ghadekar, P. Maheshwari, R. Shah, A. Shaha, V. Sonawane, and V. Shetty, "Video Forgery Dataset," Sep. 2022. [Online]. Available: `https://www.kaggle.com/datasets/rajshah1/video-forgery-dataset`.

[36] S.-Q. Zhang, T. Wu, X.-H. Xu, Z.-M. Cheng, S.-L. Pu, and C.-C. Chang, "No-reference image blur assessment based on SIFT and DCT," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 1, pp. 219–231, 2018.

[37] S.-Q. Zhang, C.-J. Hsu, Y.-X. Zheng, and H.-J. Ding, "A novel key-frame extraction approach for semantic video processing," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 14, no. 3, pp. 136–147, 2023.