EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

An efficient Video Forgery Detection using Two-Layer Hybridized Deep CNN classifier

Meena Ugale[1,*], J. Midhunchakkaravarthy[2]

[1,2]Lincoln University College, Selangor, Malaysia

Abstract

Video forgery detection is crucial to combat misleading content, ensuring trust and credibility. Existing methods encounter challenges such as diverse manipulation techniques, dataset variation, real-time processing demands, and maintaining a balance between false positives and negatives. The research focuses on leveraging a Two-Layer Hybridized Deep CNN classifier for the detection of video forgery. The primary objective is to enhance accuracy and efficiency in identifying manipulated content. The process commences with the collection of input data from a video database, followed by diligent data pre-processing to mitigate noise and inconsistencies. To streamline computational complexity, the research employs key frame extraction to select pivotal frames from the video. Subsequently, these key frames undergo YCrCb conversion to establish feature maps, a step that optimizes subsequent analysis. These feature maps then serve as the basis for extracting significant features, incorporating Haralick features, Local Ternary Pattern, Scale-Invariant Feature Transform (SIFT), and light coefficient features. This multifaceted approach empowers robust forgery detection. The detection is done using the proposed Two-Layer Hybridized Deep CNN classifier that identifies the forged image. The outputs are measured using accuracy, sensitivity, specificity and the proposed Two-Layer Hybridized Deep CNN achieved 96.76%, 96.67%, 96.21% for dataset 1, 96.56%, 96.79%, 96.61% for dataset 2, 95.25%, 95.76%, 95.58% for dataset 3, which is more efficient than other techniques.

Keywords: ResNet, Statistical feature concatenation, Two layer hybridized deep CNN, Video forgery detection, YCrCb conversion.

Received on 21 Mar 2024, accepted on 04 Apr 2024, published on 26 Sep 2024

doi: 10.4108/eetsis.5969

1. Introduction

In today's digital era, the growing concern of video forgery has become a pressing issue. With the advancement of technology and easy access to sophisticated editing tools, the prevalence of manipulated videos has increased exponentially [17]. Such manipulated videos can have significant repercussions in various domains, including media, law enforcement, and politics. In the media industry, the distribution of fake news and misinformation through manipulated videos can severely impact public opinion and erode trust in credible sources [14]. Law enforcement agencies heavily rely on video evidence to solve crimes, and the emergence of forged videos can jeopardize the integrity of investigations and lead to wrongful accusations. Additionally, in the political landscape, doctored videos can be used to influence elections and deceive voters, undermining the democratic process [18]. To combat this alarming trend, there is an urgent need for robust and efficient video forgery detection techniques. These techniques play a crucial role in safeguarding against misinformation and maintaining trust in digital media [5] [19]. Detecting forged videos accurately and efficiently is essential to ensure the authenticity and reliability of visual content in today's information-driven world [22].

Traditional video forgery detection methods have long relied on image forensics and watermarking techniques as initial approaches to identify manipulations within videos. Image forensics, which was originally developed for still images, has been extended to video frames to detect inconsistencies or artifacts that may indicate tampering [10]. Image forensics

[*] Corresponding author. Email: meena.u@xavier.ac.in

techniques typically involve analyzing the statistical properties of video frames, such as color distributions, noise patterns, and compression artifacts [21]. These methods can sometimes detect basic alterations, such as copy-pasting or splicing, but they often struggle to handle more complex and sophisticated forgeries, such as deep fake videos. Watermarking, on the other hand, involves embedding a digital signature or watermark into the video frames during the creation or editing process [16]. The presence of this watermark can be used to verify the authenticity of the video. However, watermarking has limitations as well. It requires prior embedding of the watermark, which means that it cannot detect forgeries that were not anticipated during the watermarking process [9]. Additionally, watermarks can be removed or manipulated by skilled attackers, making this method vulnerable to certain types of forgeries [24]. Both image forensics and watermarking techniques may also face challenges in adapting to emerging manipulation techniques [15]. As new technologies and tools for video manipulation emerge, traditional detection methods may struggle to keep up with the rapidly evolving landscape of video forgeries [12]. The complexity and sophistication of modern forgery techniques, such as deep fake videos, often surpass the capabilities of these traditional methods. The high accuracy from ensemble models and boosting algorithms, take comparatively higher amount of time to train the model [36]. The rise of deep learning techniques has revolutionized the field of video forensics, offering unprecedented capabilities in detecting and combating video forgeries. Initially developed for image forensics, deep learning models have demonstrated remarkable success in various tasks, such as image manipulation detection and object recognition [20]. This success has paved the way for their extension into video forensics, where they hold immense potential for detecting sophisticated video forgeries. Deep learning models excel in image forensics by learning intricate features and patterns directly from data. Traditional image forensics methods often rely on handcrafted features and predefined rules, limiting their adaptability to emerging manipulation techniques [7] [3]. In contrast, deep learning models can automatically learn complex features and representations from vast amounts of data, making them highly effective in handling diverse and evolving forgery techniques [11]. The advantages of deep learning in video forensics are even more pronounced, as videos inherently contain spatial and temporal information [12]. By processing videos as a sequence of frames, deep learning models can leverage both spatial and temporal features, allowing for more comprehensive forgery detection. This capability is particularly valuable in detecting complex manipulations, such as deep fake videos, which involve intricate spatial and temporal alterations [25]. Adopting a 2-dimensional deep classifier specifically for video forgery detection holds significant importance due to the inherent 2-dimensional nature of videos, which encompass both spatial and temporal information [13]. Online platforms encourage freedom of speech but fail to distinguish between free speech and unacceptable behaviour [34]. The traditional deep learning algorithms are not good at analysing the relationships between different entities because of the inherent deficiency of their implementation process [35]. Unlike static images, videos capture changes over time, making the temporal aspect crucial in discerning genuine videos from forgeries [6] [8]. However, traditional deep learning models, which are primarily designed for static images, may not fully exploit this temporal aspect, necessitating the development of specialized 2-dimensional classifiers [23].

The research focuses on detecting video forgery using a Two-Layer Hybridized Deep CNN classifier. Data from a video database is pre-processed to reduce noise, followed by key frame extraction to alleviate computational complexity. Key frames are subjected to YCrCb conversion and ResNet feature mapping. Significant features encompassing Haralick, Local Ternary Pattern, SIFT, and light coefficients are then extracted. The proposed Two-Layer Hybridized Deep CNN classifier employs distinct layers for processing feature map outputs. These outputs are concatenated to yield a unified output, effectively identifying forged images. This comprehensive approach strengthens the accuracy and efficiency of video forgery detection, promising advancements in digital content security and integrity verification. The contributions are as follows,

- ***Two-Layer Hybridized Deep CNN:*** The purpose of a Two-Layer Hybridized Deep CNN architecture is to enhance the performance of a neural network model for tasks such as image or video analysis, including video forgery detection. This architecture combines the strengths of deep learning (CNNs) with other techniques to improve feature representation, capture temporal relationships, and achieve better accuracy in complex tasks. Deep CNN layers automatically learn hierarchical features from raw pixel data, capturing patterns, edges, and textures. By incorporating multiple layers of convolution and pooling, the architecture can learn more abstract and high-level features.

The manuscript is divided into the following sections, with section 2 focusing on the shortcomings of the available video forgery detection techniques. Section 3 included an explanation of the suggested methods for detecting video forgeries. Section 4 summarizes the experimental results and conclusions of the study, and Section 5 brings the examination to a close.

2. Literature review

The reviews of the method for detecting fake images and videos are as follows:

Using a luminance channel of images, Savita Walia et al. [1] presented a combination of constructed features using color properties and deep features. This approach achieved high detection accuracy and uncovered hidden patterns responsible for accurate forgery detection but came with increased computational complexity. Ammarah Hashmi et al. [2] presented an innovative ensemble learning technique for audiovisual deep fake detection, addressing the limitations of existing systems by considering both audio and video data. However, this method also introduced computational complexity. Deep learning-based approaches were used by Muriel Mazzetto et al. [3] to support visual inspection activities while reducing their negative effects on the production environment. This approach improved accuracy but encountered some overfitting issues. Mohan Karnati et al. [4] introduced a novel deep CNN model for automatic detection of multi-scale variations of deception. While this model offered an efficient approach to deception detection, it increased model complexity and required more computational resources during training and inference. Sakshi Singhal and Virender Ranga [5] proposed a CNN-based approach for detecting forged images, including copy-move and spliced forgeries. This method did not require additional information to be added to the image but may be less effective in detecting other types of image manipulations or more sophisticated forgeries. Copy-move forgery detection was one of the image forgery detection methods published by Anushka Darade et al. [6]; it had concerns with overfitting but showed higher precision and efficacy than other methods. Hanady Sabah Abdul Kareem et al. [7] employed a popular deep learning algorithm to detect fake images, enhancing accuracy in fake image detection. However, the use of PCA for dimensionality reduction may lead to some information loss. In order to create images of faces with various appearances, Hao Zhu et al. [8] defined appearance mapping as an ideal transportation problem and unveiled an appearance optimal transportation model. This method required a sophisticated implementation and fine-tuning, potentially limiting accessibility for some researchers or practitioners.

2.1 Challenges

- Video datasets are often large, which can be computationally expensive to handle. Additionally, there may be class imbalances, with fewer examples of certain types of forgeries, making it challenging to train a balanced model.
- Choosing an appropriate architecture for the two-layer hybridized deep CNN can be challenging. Finding the right balance between model complexity and generalization is crucial.
- Deep CNNs are prone to overfitting, especially when the dataset is small or imbalanced. Implementing regularization techniques and data augmentation strategies is essential to mitigate overfitting.
- Deep learning models, particularly deep CNNs, can be challenging to interpret. Understanding why the model makes a specific decision can be crucial for real-world applications.

3. Proposed methodology for video forgery detection

The primary goal of the research is to utilize a Two-Layer Hybridized Deep CNN classifier to detect image or video fakes. The key frame extraction is done for the selection of the key frames from the video that would reduce the computational complexity associated with the detection process. Now, the key frames are subjected to the establishment of the feature maps following the YCrCb conversion and Resnet. The relevant features based on the Haralick feature, Local ternary pattern, SIFT, and light coefficient features are retrieved.

The proposed classifier consists of two layers, each processing a different feature map output. The final step involves concatenating these outputs, resulting in a unified output. The training optimizes the model's parameters for accurate forgery detection. Once trained, the classifier showcases its prowess during the testing phase. Figure 1 displays a schematic illustration of the proposed methodology.

3.1 Input

The following equation mathematically represents the input data used in this investigation, which is taken from the Image/Video Forgery Identification Dataset (DTS), particularly from the DSO-1, DSI-1 DTS [32] and face forensics database [33],

$$V = \sum_{b=1}^{c} V_b \tag{1}$$

Here, V_b stands for the videos that have been pulled from the repository, while b represents the number of videos, with a value in the range of $\left[1, c\right]$.

3.2 Pre-processing

The goal of preprocessing is to enhance the quality of the images, reduce noise, correct anomalies, and prepare the data in a suitable format for further tasks. When dealing with a video dataset, using key frames is a common approach to reduce computational complexity and extract meaningful information. The Key Frame is chosen based on the low motion activity (to avoid blurring and excessive coding aircrafts), a high spatial activity and the likeliness to include people. Key frames are frames that represent the content changes within a video. They are selected to provide a representative sample of the video’s content, typically chosen at regular intervals or when there is a significant change in the scene.

$$V^* = \sum_{b=1}^{c} V_b^* \tag{2}$$

Where, V^* denotes the pre-processed dataset.

3.3 YCBCR conversion

The YCBCR color space, also known as YCbCr, is a color representation that separates image information into three components: Y (luminance), Cb (blue-difference chrominance), and Cr (red-difference chrominance). Separating the luminance (Y) from the chrominance (Cb and Cr) allows for independent manipulation of color and brightness. The separation of chrominance from luminance makes it easier to correct color casts and other color-related issues in images and videos.

$$YCBCR = rgb2ycbcr(RGB) \tag{3}$$

3.4 ResNet-101 feature map

A ResNet (Residual Network) feature map is a representation of an input image after it has been processed through a deep CNN using a specific architecture known as ResNet. In ResNet architectures, feature maps are intermediate outputs obtained from various layers of the network. The purpose of ResNet feature maps is to capture and represent important image features at different scales and levels of abstraction. These features can include edges, textures, shapes, and object parts. ResNet architectures allow for very deep networks, enabling the extraction of complex hierarchical features from images.
The equation for the skip connection is as follows,

$$Output = F(x, \{W_i\}) + x \tag{4}$$

Where, the input to the specific layer or block is denoted as x, the weights of the convolutional layers within the residual block is denoted as W_i, the residual part of the network's transformation applied to x is denoted as $F(x, \{W_i\})$ and it's the difference between the desired output and the input.

3.5 Feature extraction

Feature extraction in the context of images is particularly important due to the high-dimensional nature of image data. Extracted features can provide insights into what aspects of the image are contributing to a particular decision, enhancing model interpretability. In this research the feature extraction is based on Haralick feature, Local ternary pattern, SIFT, and light coefficient features, which is detailed in the below section.

3.5.1 Haralick features

Haralick features, also referred to as texture features or texture descriptors, constitute a set of statistical metrics designed to characterize the texture of an image. Mathematically, Haralick features are computed by utilizing a gray-level co-occurrence matrix (GLCM), which effectively represents the joint probabilities of pairs of pixel intensity values occurring at specific relative distances and angles within the image. Among these features, the contrast feature plays a significant role in measuring the local variations in pixel intensity values within the image, quantifying the degree to which pixel intensity values deviate from those of their neighboring pixels.

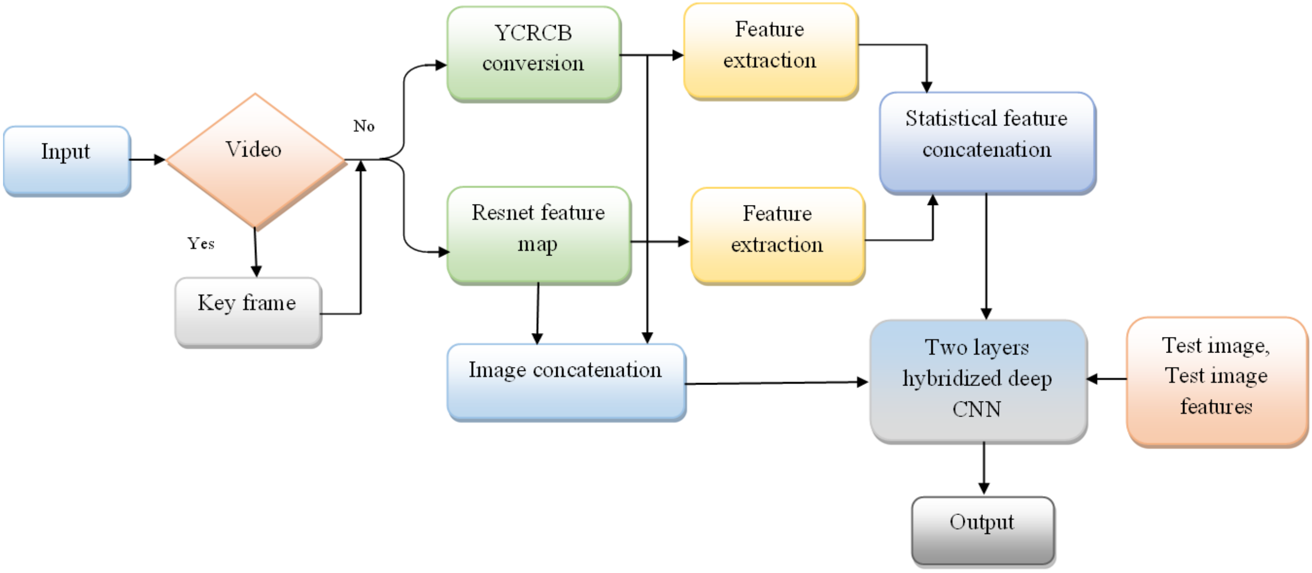


Figure 1. Image forgery segmentation model

Contrast (C) is mathematically defined as follows,

$$C=\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2P(i,j) \tag{5}$$

Where, the number of grey levels is denoted as N_g , $P(i,j)$ is the normalized co-occurrence matrix entry corresponding to the probability of a pixel at intensity i occurring at a given distance and angle with a pixel at intensity.

Haralick features are relatively robust to variations in lighting, noise, and other image acquisition conditions. They capture patterns that are consistent across different images of the same texture, even if the absolute pixel values vary. Haralick features are particularly useful for texture classification tasks, where the goal is to differentiate between various textures or materials. Their ability to capture subtle variations in texture can improve classification accuracy.

3.5.2 Local ternary pattern features (LTP)

LTP is a texture descriptor similar to the well-known Local Binary Pattern (LBP), but instead of using binary codes (0 and 1), LTP uses ternary codes (0, 1, and 2) to capture local texture patterns within an image. LTP is particularly useful for texture analysis and classification tasks. For each pixel in the circular neighborhood, the intensity difference with the central pixel is computed. The pixel is given a value of 2 if the difference exceeds a predetermined threshold. The pixel is given a value of 0 if the difference is smaller than the threshold's negative value. If not, a value of 1 is set to it. LTP provides a richer encoding of local texture patterns than traditional binary patterns. This is beneficial when capturing more complex texture variations. LTP can represent patterns with more transitions than traditional LBP, making it suitable for textures with irregular or complex variations.

$$LTP_{Q,R}=\sum_{i=0}^{Q-1}s(p_i-p_c)3^i\ ,\ s(x)\begin{cases}1 & if\ x\geq g\\ 0 & if\ |x|<g\\ -1 & if\ x\leq -g\end{cases} \tag{6}$$

Where, g is referred as the user threshold of coding. Every image piece of imagery generates a Q-bit binary numerical with 3^Q variable values as a result of the $LTP_{Q,R}$ accelerator, which enhances the calculation complexity as well as the minimalism and decreases mathematical difficulty.

3.5.3 SIFT features

SIFT (Scale-Invariant Feature Transform) features provide a reliable and robust way to detect and describe key points or interest points in images, which can then be used for various tasks. SIFT features are used to align and stitch together multiple images to create panoramas. By identifying matching key points between overlapping images, image stitching algorithms can align and blend images to create a seamless panorama. SIFT features can be used to index and retrieve images from large databases. By representing images with their SIFT feature vectors, you can search for similar images based on feature similarity. The mathematical equations are,

$$M(x,y)=\sqrt{I_x^2+I_y^2} \tag{7}$$

$$\theta(x,y)=\arctan\left(\frac{I_y}{I_x}\right) \tag{8}$$

Where, for each pixel at coordinates (x,y), the gradient magnitude $M(x,y)$ and gradient orientation $\theta(x,y)$ can be computed using partial derivatives I_x and I_y in the x and y directions, respectively.

3.5.4 Light coefficients feature

Block-wise calculations are used to determine the light coefficients, with the absorption and coefficient dispersion summarization being the anticipated results. Light absorption is rarely the focus of articles. However, taking into account a significant number of flaws, the light dispersal could also significantly contribute to the overall reduction of light that is defined mathematically as,

$$L_{ef}=\frac{3}{16\pi}\cos^2\theta \tag{9}$$

where, L_{ef} denotes the Light coefficient features.

3.6 Statistical feature concatenation

Statistical feature concatenation involves combining statistical features from different sources or modalities into a single more comprehensive and informative concatenated feature vector. This can provide a more detailed and comprehensive representation of the underlying patterns, characteristics, and variations. Two sets of statistical features F_1 and F_2 are represented as vectors. These vectors contain various statistical measures calculated from different data sources or regions.

Let us assume, F_1 has n elements representing the statistical feature from source 1, and F_2 has m elements representing the statistical features from source 2. To concatenate these two feature vectors, create a new feature vector F_{concat} that combines the feature from both sources. This can be done using concatenation notation,

$$F_{concat}=|F_1,F_2| \tag{10}$$

Where, $|F_1,F_2|$ denotes the concatenation of the two feature vectors F_1 and F_2, resulting in a new feature vector F_{concat} that contains all the statistical feature combined.

3.7 Two-Layer Hybridized Deep CNN

The purpose of a Two-Layer Hybridized Deep CNN architecture is to enhance the performance of a neural network model for tasks such as image or video analysis, including forgery detection. Figure 2 shows Two-Layer Hybridized Deep CNN architecture. This architecture combines the strengths of deep learning (CNNs) with other techniques to improve feature representation, capture temporal relationships, and achieve better accuracy in complex tasks. Deep CNN layers automatically learn hierarchical features from raw pixel data, capturing patterns, edges, and textures.

By incorporating multiple layers of convolution and pooling, the architecture can learn more abstract and high-level features. Combining deep CNN layers with a fusion mechanism allows the model to capture temporal relationships and context between frames. This is particularly important for video forgery detection, where identifying splicing or alterations requires considering the sequence of frames.

The input layer receives video frames as input data. Each frame is typically represented as an image with height, width, and color channels. The input layer’s dimensions correspond to the dimensions of the input frames.

Convolutional layers are responsible for automatically learning features from the input frames. Convolutional filters slide across the input frames, capturing visual patterns, edges, textures, and other features. Each filter produces a feature map that represents the response of the filter to various features in the input. The number of filters in this layer defines the depth or number of extracted features.

A max pooling layer is frequently applied after each convolutional layer. Max pooling helps to down sample the data by reducing the feature maps' spatial dimensions. Pooling focuses on the most important information, increasing computational efficiency and aiding in translation invariance.

The flatten layer converts the multi-dimensional feature maps from the previous layer into a one-dimensional vector. This step is necessary to connect the convolutional layers to fully connected (dense) layers.

The concatenate layer is a critical part of the hybridized aspect of the architecture. It combines the output from the convolutional layers with additional information from another source or modality, such as temporal data or domain-specific knowledge. The concatenated features enhance the overall feature representation.

Dense layers take the concatenated features and perform classification or regression tasks. The number of neurons in the dense layer and the activation functions depend on the specific task. Dense layers compute weighted sums of inputs and apply activation functions to produce output.

Dropout layers can be added to prevent overfitting. During training, dropout randomly sets a fraction of input units to zero, effectively dropping out those units. This regularization technique helps the network generalize better to unseen data.

The output layer produces the final classification output. For video forgery detection, this layer might output binary values indicating genuine or manipulated content. The number of classes is reflected in the total amount of neurons in the output layer.

4. Results

4.1 Experimental setup

The experiment was conducted using Python. The implementation of various methods was facilitated through PyCharm software. The experiment was performed on a Windows 10 operating system with 8GB of RAM available for memory.

4.2 Dataset description

4.2.1 DSO-1 dataset [32]: The DSO-1 dataset stands as a crucial asset in the realm of forgery detection research. This dataset offers a diverse array of images, encompassing both genuine and modified ones, serving as valuable training and evaluation data for algorithms dedicated to detecting image manipulation. Researchers rely on this dataset to advance the development of models capable of identifying a multitude of image tampering techniques, ultimately enhancing the fields of image forensics and digital security.

4.2.2 DSI-1 DTS dataset [32]: The DSI-1 DTS dataset holds immense importance in forgery detection research. It comprises a diverse set of images, encompassing both authentic and manipulated ones, offering researchers a comprehensive platform to develop and evaluate algorithms for detecting image tampering and forgery. This dataset plays a pivotal role in advancing the domain of image forensics by presenting a wide range of scenarios and challenges that algorithms must confront in order to achieve precise forgery detection and thorough analysis.

4.2.3 Face forensics database [33]: The Face Forensics Database holds significant importance in the forgery detection field, with a particular focus on facial manipulation and deep fake detection within videos. This database comprises a diverse collection of videos that have undergone manipulation through various techniques, such as splicing and deep fake methods, resulting in deceptive facial alterations. It functions as a crucial benchmark for assessing the performance of algorithms created to detect these manipulations. Researchers rely on this database to develop and evaluate video forgery detection methods, ultimately contributing to the advancement of technology for detecting altered faces and ensuring the authenticity of video content.

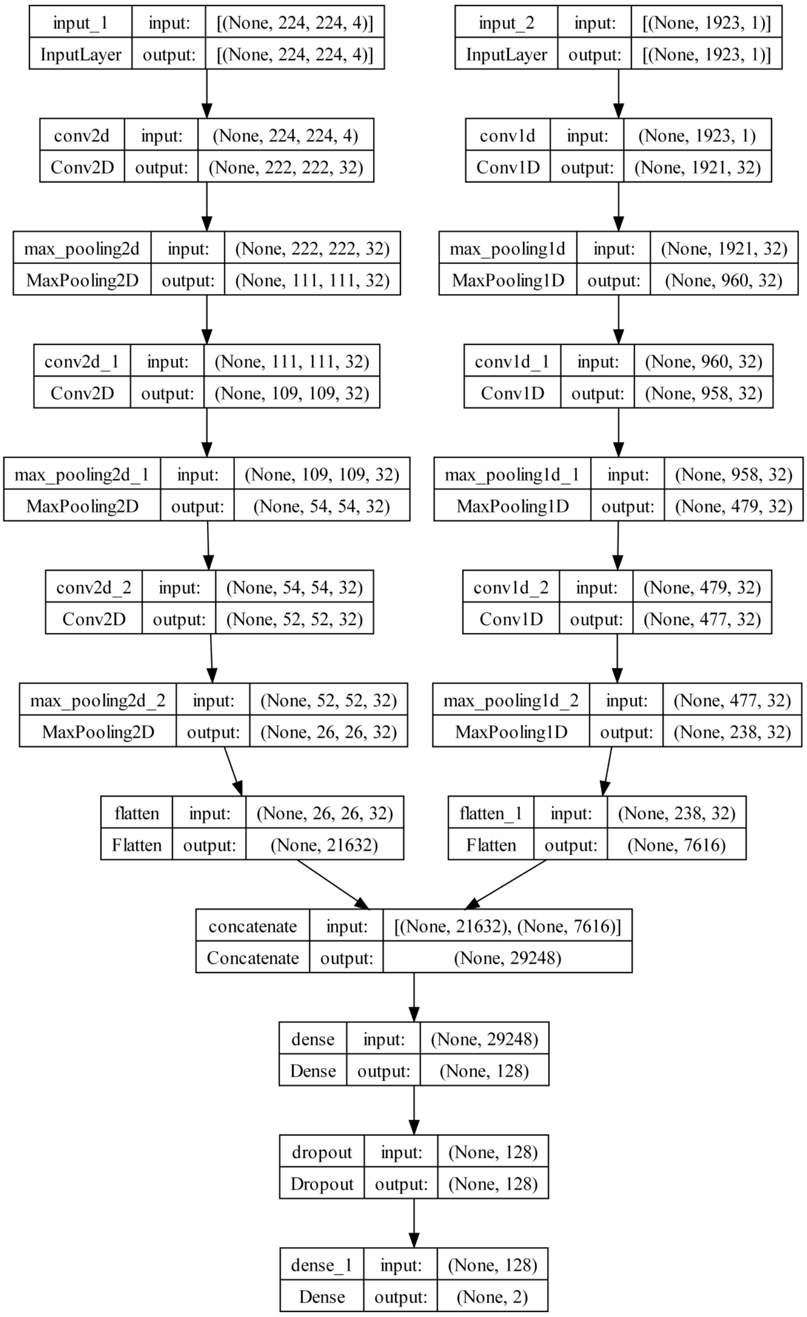

input_1 | input: | [(None, 224, 224, 4)]
InputLayer | output: | [(None, 224, 224, 4)]
input_2 | input: | [(None, 1923, 1)]
InputLayer | output: | [(None, 1923, 1)]
conv2d | input: | (None, 224, 224, 4)
Conv2D | output: | (None, 222, 222, 32)
conv1d | input: | (None, 1923, 1)
Conv1D | output: | (None, 1921, 32)
max_pooling2d | input: | (None, 222, 222, 32)
MaxPooling2D | output: | (None, 111, 111, 32)
max_pooling1d | input: | (None, 1921, 32)
MaxPooling1D | output: | (None, 960, 32)
conv2d_1 | input: | (None, 111, 111, 32)
Conv2D | output: | (None, 109, 109, 32)
conv1d_1 | input: | (None, 960, 32)
Conv1D | output: | (None, 958, 32)
max_pooling2d_1 | input: | (None, 109, 109, 32)
MaxPooling2D | output: | (None, 54, 54, 32)
max_pooling1d_1 | input: | (None, 958, 32)
MaxPooling1D | output: | (None, 479, 32)
conv2d_2 | input: | (None, 54, 54, 32)
Conv2D | output: | (None, 52, 52, 32)
conv1d_2 | input: | (None, 479, 32)
Conv1D | output: | (None, 477, 32)
max_pooling2d_2 | input: | (None, 52, 52, 32)
MaxPooling2D | output: | (None, 26, 26, 32)
max_pooling1d_2 | input: | (None, 477, 32)
MaxPooling1D | output: | (None, 238, 32)
flatten | input: | (None, 26, 26, 32)
Flatten | output: | (None, 21632)
flatten_1 | input: | (None, 238, 32)
Flatten | output: | (None, 7616)
concatenate | input: | [(None, 21632), (None, 7616)]
Concatenate | output: | (None, 29248)
dense | input: | (None, 29248)
Dense | output: | (None, 128)
dropout | input: | (None, 128)
Dropout | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 2)

Figure 2. Two-Layer Hybridized Deep CNN architecture

4.3 Performance Metrics

4.3.1 Accuracy: The overall accuracy of predictions made by the detection algorithm is measured, determining the proportion of successfully identified occurrences of all instances. Accuracy in the context of video forgery detection is the detected percentage of fake and real video segments.

$$acc = \frac{B_{tn} + B_{tp}}{B_{tn} + B_{tp} + B_{fn} + B_{fp}} \tag{11}$$

4.3.2 Sensitivity: Sensitivity is concerned with how well the algorithm can distinguish the forged segments from the genuine positive cases. It is determined as the percentage of forged segments that were correctly recognized to all forged segments that the algorithm missed. The algorithm's sensitivity reveals how effectively it can spot actual instances of video counterfeiting.

$$sen = \frac{B_{tp}}{B_{tp} + B_{fn}} \tag{12}$$

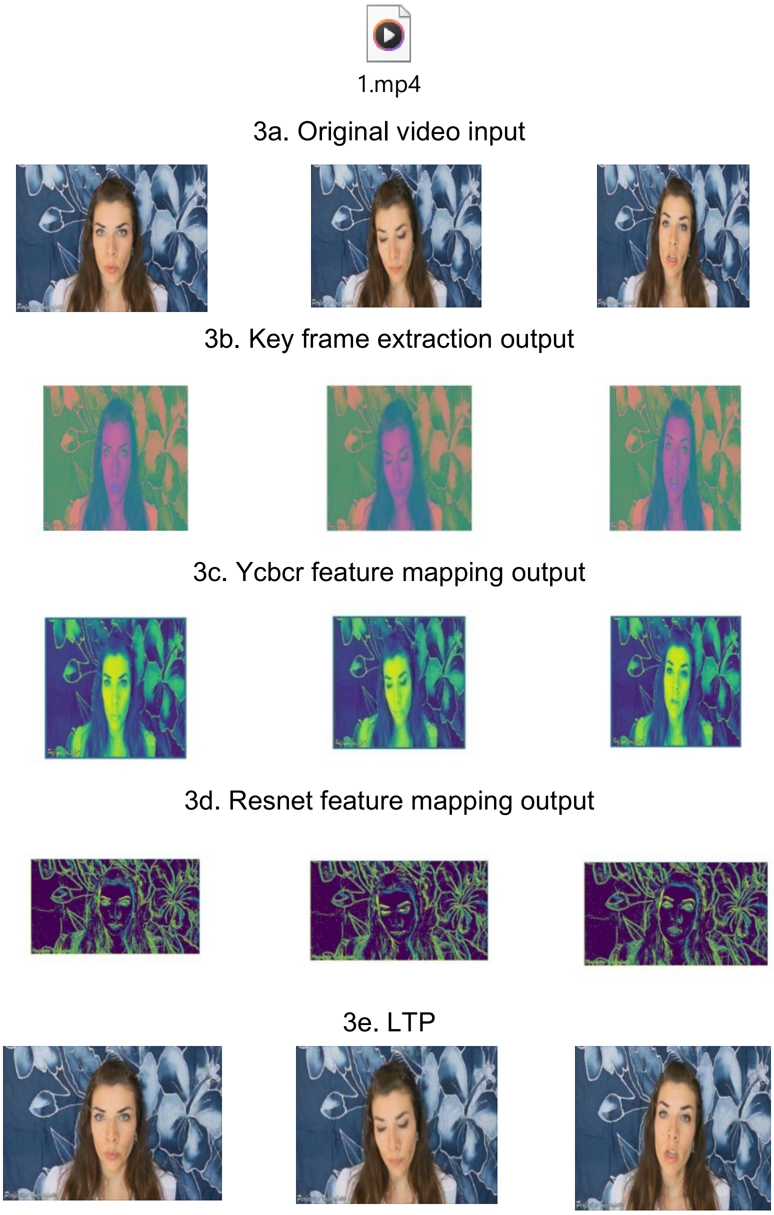
4.3.3 Specificity: The algorithm's specificity measures how well it can recognize the real segments amongst all the real negative situations. It is determined as the ratio of authentic segments that were correctly discovered to the total of authentic segments that were falsely marked as fake. The algorithm's specificity gauges how well it prevents false alarms for real video parts.

$$spec = \frac{B_{tn}}{B_{tn} + B_{fp}} \tag{13}$$

4.4 Experimental results

In this outlined process, an initial video (as shown in Figure 3a) serves as the input which is original video. From this video, a key frame is extracted, giving rise to the image portrayed in Figure 3b. Following that, the key frame image undergoes a Ycbcr feature mapping output, producing the result depicted in Figure 3c. Subsequent steps involve Resnet feature mapping output, as visualized in Figure 3d. Additionally, an LTP is employed, yielding the outcome illustrated in Figure 3e. SIFT features provide a reliable and robust way to detect and describe key points or interest points in images and its output is illustrated in Figure 3f.

This comprehensive procedure systematically refines the original video input into enhanced feature representations, encompassing key frame extraction, preprocessing, and the application of specialized feature mapping techniques in sequence.



3a. Original video input

3b. Key frame extraction output

3c. Ycbcr feature mapping output

3d. Resnet feature mapping output

3e. LTP

3f. SIFT

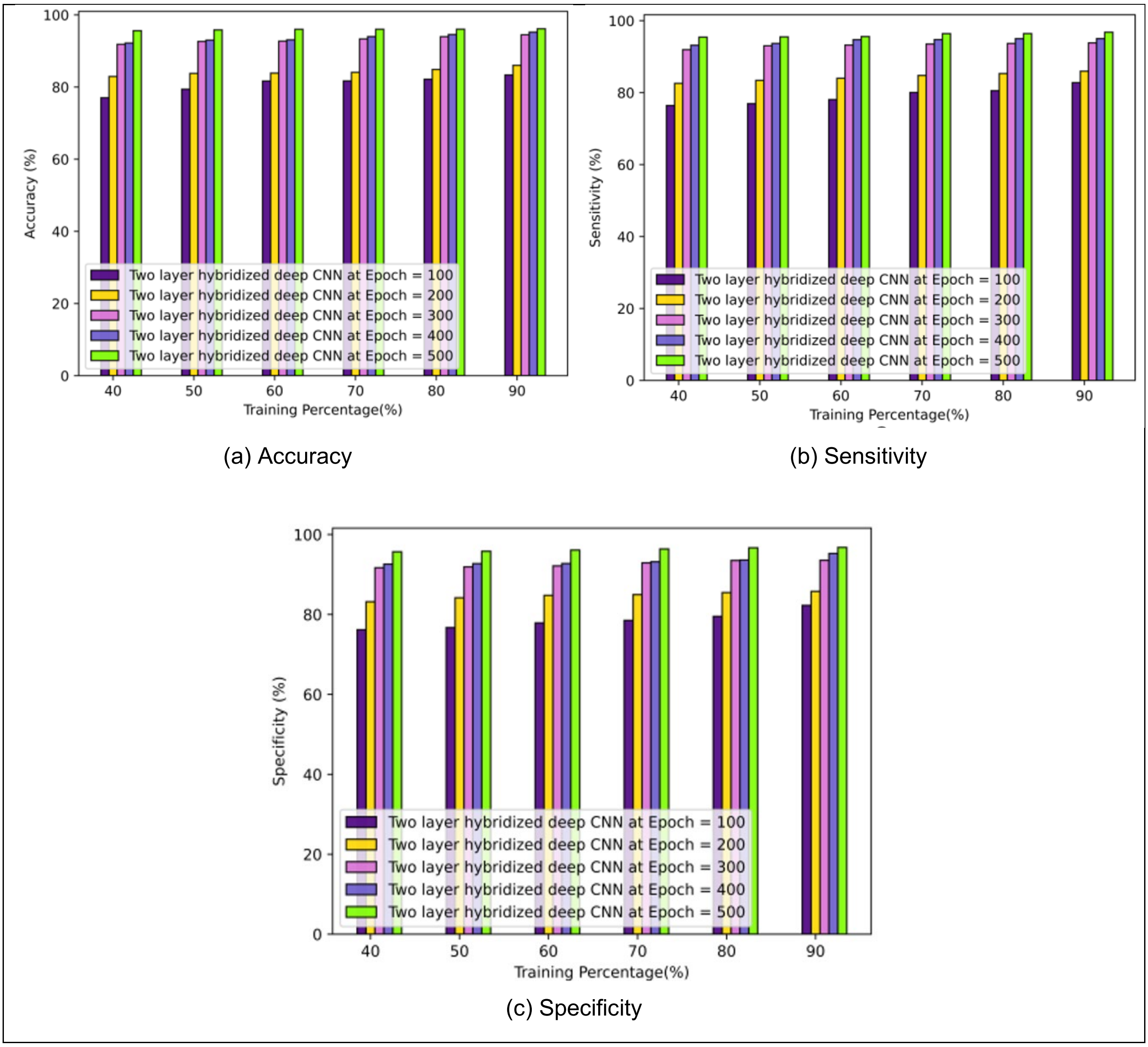
Figure 3. Experimental results of video forgery detection

4.5 Performance evaluation

To assess the classifier's performance at various epochs, specifically at 100, 200, 300, 400, and 500, we conduct performance evaluations.

4.5.1 Performance evaluation with TP for dataset 1

In Figure 4, we present the performance evaluation results of the proposed Two-Layer Hybridized Deep CNN classifier across different epochs, focusing on achieving a Training Percentage (TP) rate of 90%. Initially, when assessing accuracy (as shown in Figure 4a), the proposed classifier achieves accuracy values of 84.49%, 89.12%, 94.46%, 95.00%, and 96.71% for varying epochs. Similarly, when measuring sensitivity (Figure 4b), the proposed classifier attains sensitivity rates of 85.96%, 87.93%, 95.27%, 95.29%, and 96.69% for different epochs. Likewise, in the evaluation of specificity (Figure 4c), the proposed classifier achieves specificity values of 85.53%, 89.15%, 94.85%, 95.21%, and 96.56% across various epochs.



(a) Accuracy

(b) Sensitivity

(c) Specificity

Figure 4. Analysis with TP for dataset 1

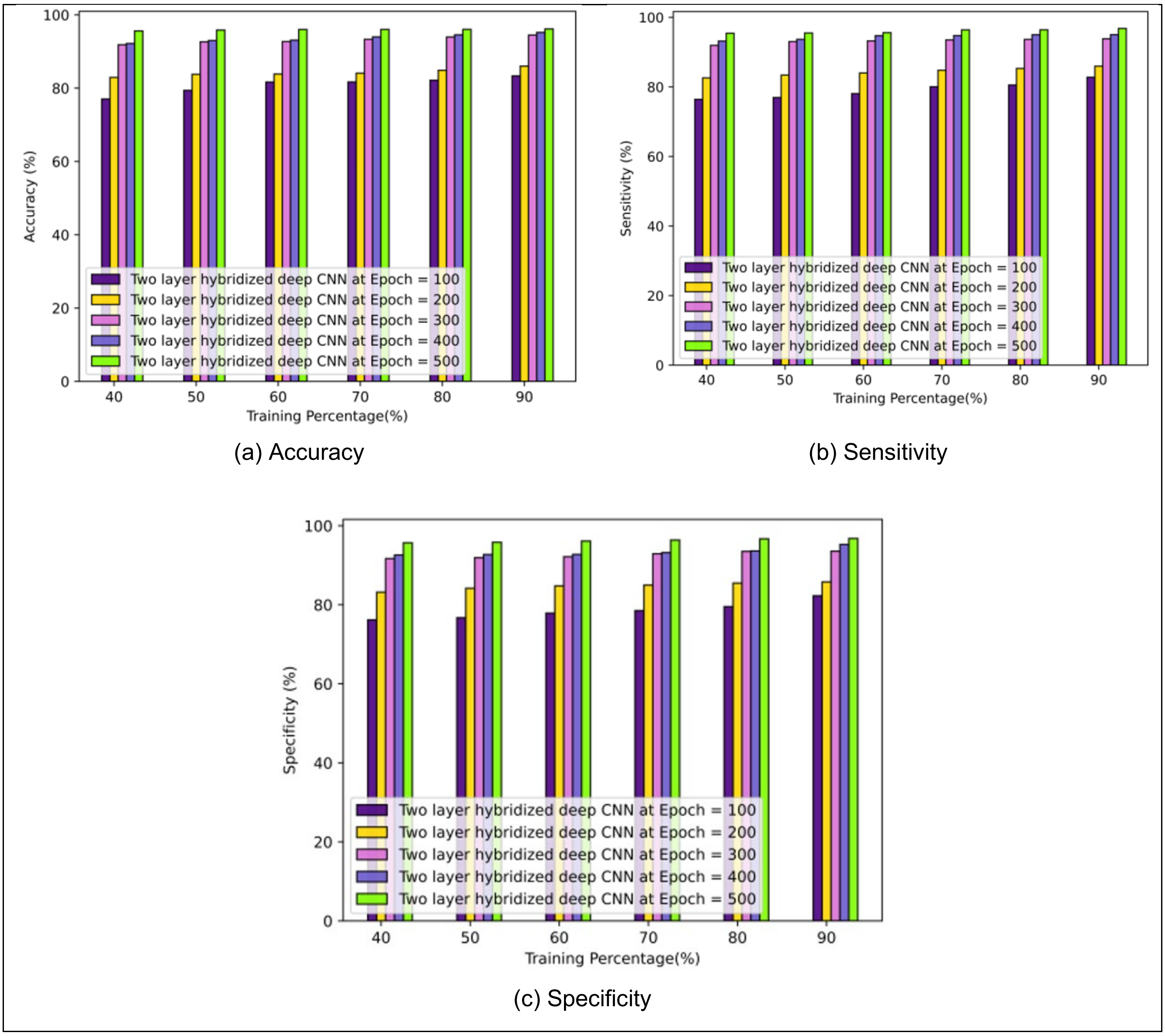
4.5.2 Performance evaluation with TP for dataset 2

In Figure 5, we present the performance evaluation results of the proposed classifier across varying epochs, with a focus on achieving a TP rate of 90%.

Initially, when assessing accuracy (as depicted in Figure 5a), the proposed Two-Layer Hybridized Deep CNN classifier achieves accuracy values of 83.33%, 86.02%, 94.47%, 95.20%, and 96.15% across different epochs.

Similarly, in the context of sensitivity measurements (Figure 5b), the proposed classifier attains sensitivity rates of 85.96%, 87.93%, 95.27%, 95.29%, and 96.69% for the corresponding epochs.

Likewise, when evaluating specificity (Figure 5c), the proposed Two-Layer Hybridized Deep CNN classifier attains specificity values of 82.26%, 85.77%, 93.55%, 95.25%, and 96.77% for the various epochs.



(a) Accuracy

(b) Sensitivity

(c) Specificity

Figure 5. Analysis with TP for dataset 2

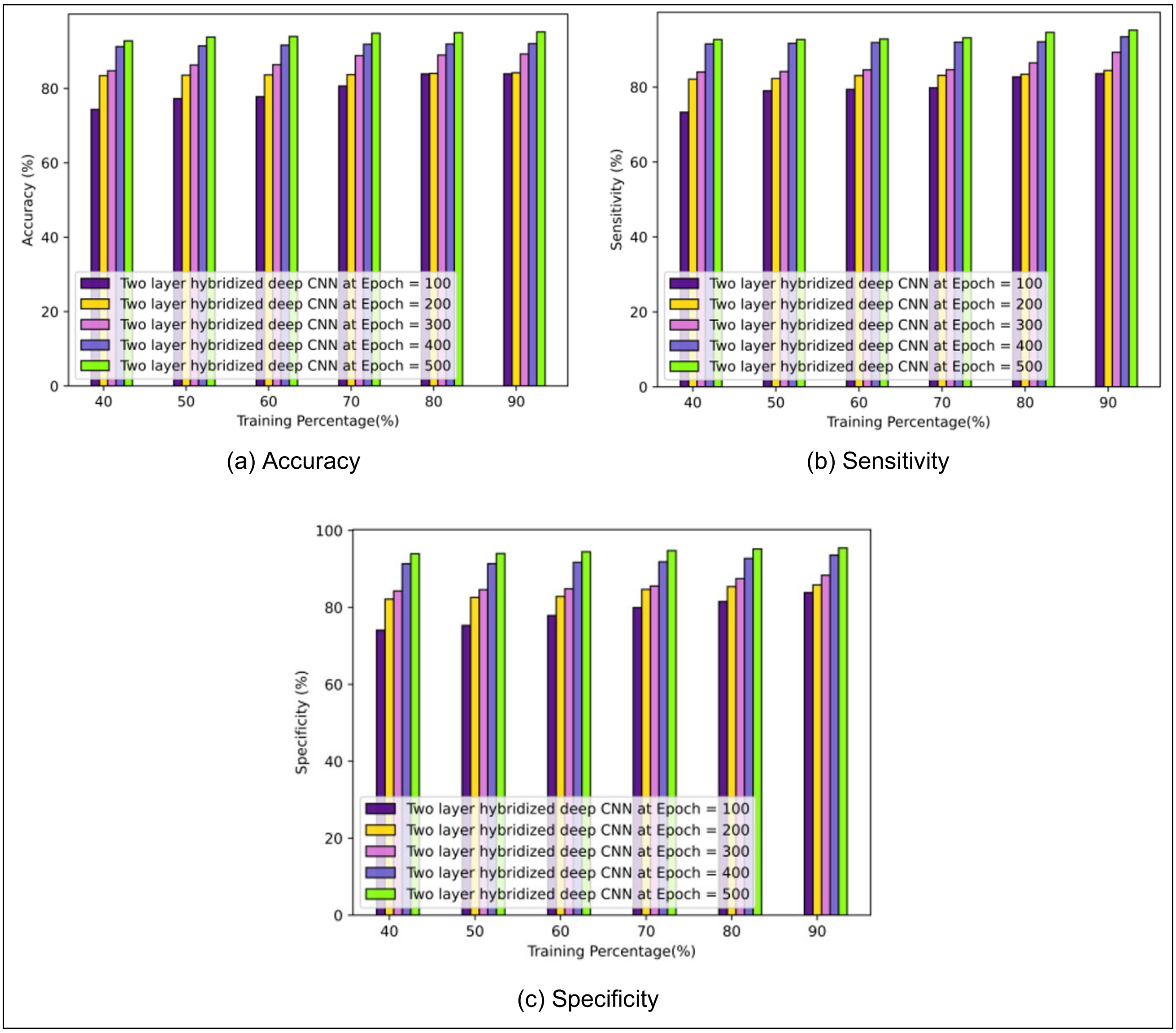
4.5.3 Performance evaluation with TP for dataset 3

In Figure 6, we present the performance evaluation results of the proposed classifier across varying epochs, with the objective of achieving a TP rate of 90%.

Initially, when examining accuracy (as illustrated in Figure 6a), the proposed Two-Layer Hybridized Deep CNN classifier attains accuracy values of 83.91%, 84.20%, 89.25%, 92.04%, and 95.19% across different epochs.

Similarly, when assessing sensitivity (Figure 6b), the proposed classifier achieves sensitivity rates of 83.58%, 84.42%, 89.29%, 93.43%, and 95.18% for the corresponding epochs.

Likewise, for specificity measurements (Figure 6c), the proposed classifier attains specificity values of 83.83%, 85.84%, 88.36%, 93.60%, and 95.45% over the various epochs.



(a) Accuracy

(b) Sensitivity

(c) Specificity

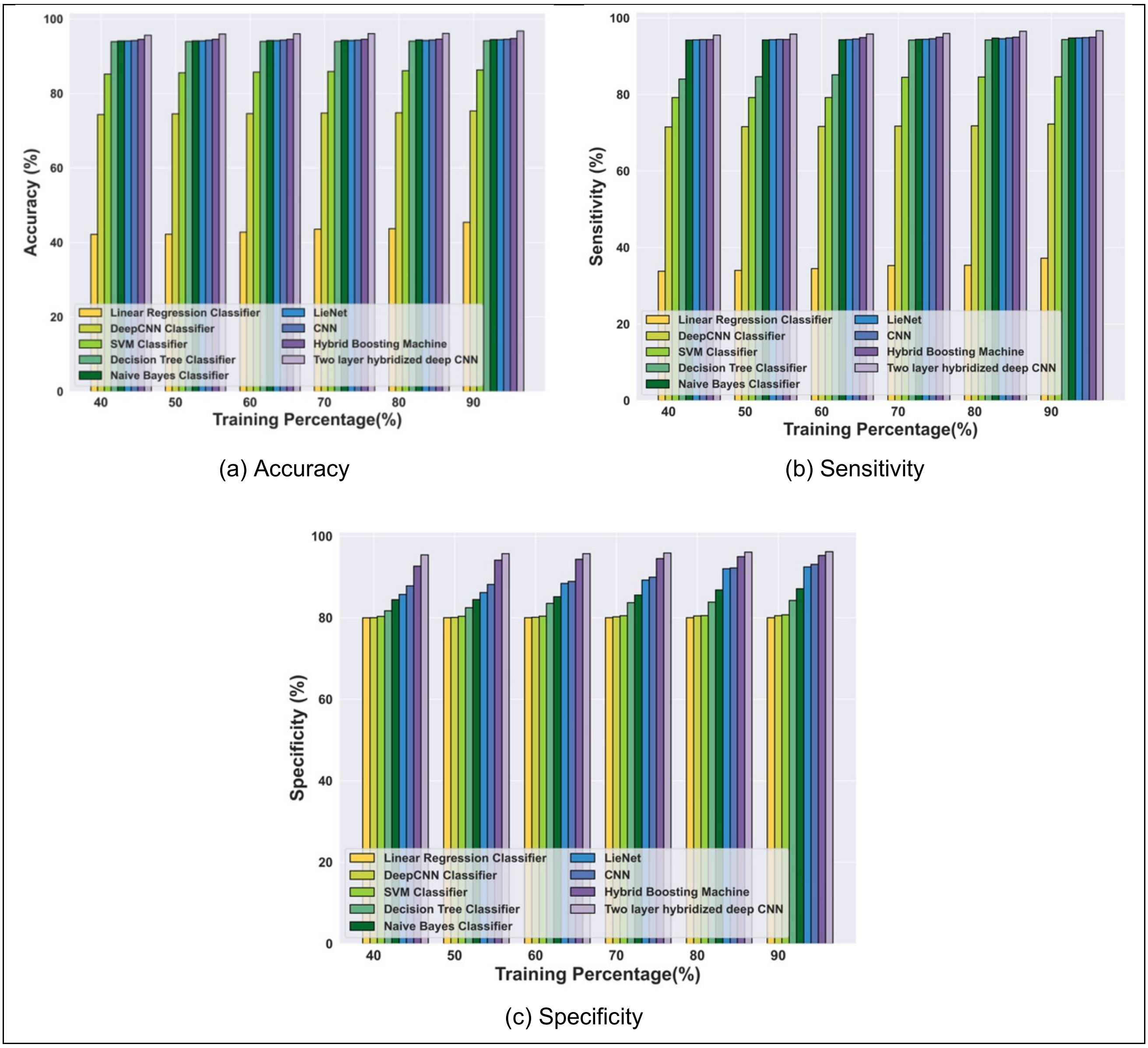
Figure 6. Analysis with TP for dataset 3

4.6 Comparative methods

Linear Regression Classifier [26], Deep CNN Classifier [27], SVM Classifier [28], Decision Tree Classifier [29], Naive Bayes Classifier [30], LieNet [4], CNN [5], Hybrid Boosting Machine [31] are compared with Two-Layer Hybridized Deep CNN classifier.

4.6.1 Comparative analysis with TP for dataset 1

Figure 7 presents a comparative analysis with TP, demonstrating that the proposed classifier achieved a remarkable improvement of 2.09 in accuracy compared to the Hybrid Boosting Machine, as depicted in Figure 7a. Additionally, in terms of sensitivity, the Two-Layer Hybridized Deep CNN outperformed the Hybrid Boosting Machine by 1.73, as illustrated in Figure 7b. Finally, for specificity, the proposed Two-Layer Hybridized Deep CNN classifier showed a notable improvement of 0.94 compared to the Hybrid Boosting Machine, as seen in Figure 7c.



(a) Accuracy

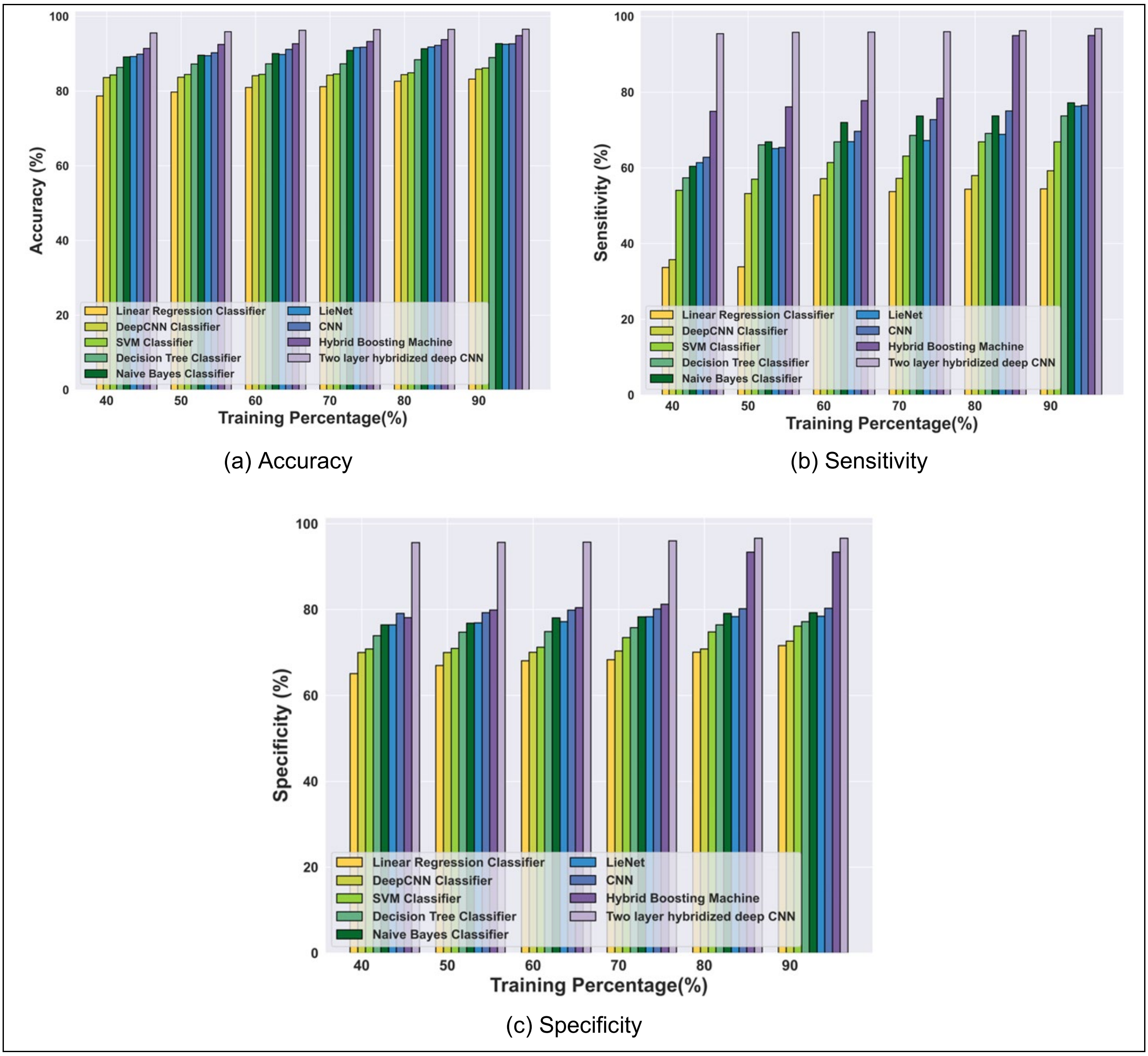
(b) Sensitivity

(c) Specificity

Figure 7. Analysis with TP for dataset 1

4.6.2 Comparative analysis with TP for dataset 2

Figure 8 presents a comparative analysis involving TP, revealing that the proposed classifier achieved a notable improvement of 1.76 in accuracy when compared to the Hybrid Boosting Machine, as depicted in Figure 8a. Similarly, in the assessment of sensitivity, the proposed classifier displayed a substantial improvement of 1.88 compared to the Hybrid Boosting Machine, as observed in Figure 8b. Finally, when evaluating specificity, the Two-Layer Hybridized Deep CNN classifier demonstrated a remarkable improvement of 3.35 compared to the Hybrid Boosting Machine, as illustrated in Figure 8c.



(a) Accuracy

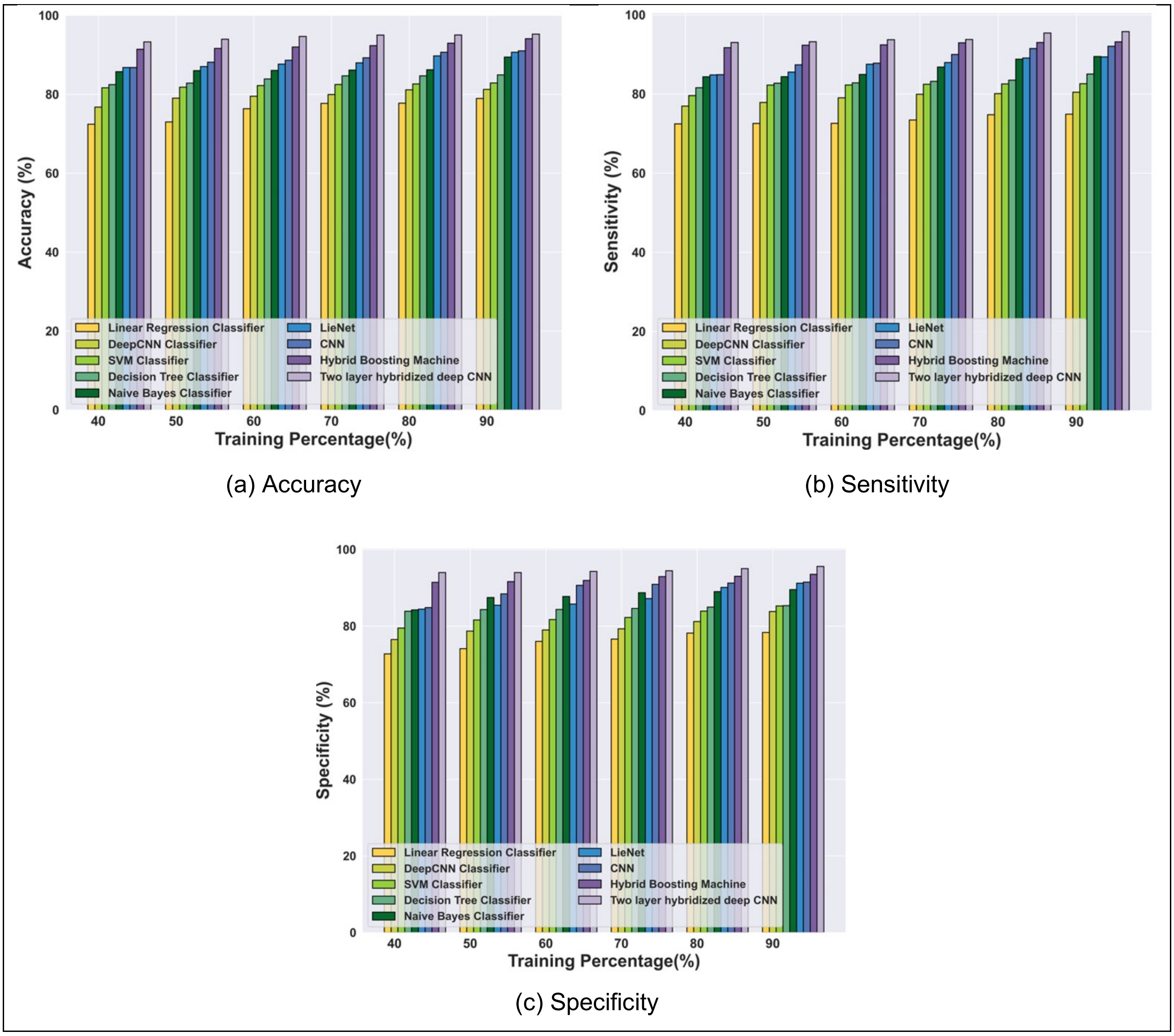
(b) Sensitivity

(c) Specificity

Figure 8. Analysis with TP for dataset 2

4.6.3 Comparative analysis with TP for dataset 3

Figure 9 presents the comparative analysis involving TP, highlighting that the proposed classifier achieved a significant improvement of 1.23 in accuracy when compared to the Hybrid Boosting Machine, as shown in Figure 9a. Likewise, in terms of sensitivity measurement, the proposed classifier demonstrated a substantial improvement of 2.71 compared to the Hybrid Boosting Machine, as illustrated in Figure 9b. Finally, when assessing specificity, the proposed Two-layer Hybridized Deep CNN classifier exhibited a noteworthy improvement of 2.15 in comparison to the Hybrid Boosting Machine, as indicated in Figure 9c.



(a) Accuracy

(b) Sensitivity

(c) Specificity

Figure 9. Analysis with TP for dataset 3

Table 1. Comparative discussion of the proposed Two-Layer Hybridized Deep CNN classifier

Sr. No.	Methods	DSO-1 Dataset			DSI-1 Dataset		
			TP (90)			TP (90)	
		Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)
1	Linear Regression Classifier	45.40	37.22	80.00	83.20	54.45	71.61
2	Deep CNN Classifier	75.32	72.27	80.50	85.82	59.22	72.65
3	SVM Classifier	86.32	84.62	80.71	86.16	66.89	76.13
4	Decision Tree Classifier	94.14	94.37	84.23	88.96	73.72	77.18
5	Navie Bayes Classifier	94.47	94.76	87.11	92.69	77.18	79.27
6	LieNet	94.47	94.72	93.80	92.14	73.30	79.13
7	CNN	94.48	94.85	93.97	92.49	76.91	80.32
8	Hybrid Boosting Machine	94.74	95.00	95.27	94.86	94.97	93.38
9	**Proposed Classifier**	**96.76**	**96.67**	**96.21**	**96.56**	**96.79**	**96.61**

Sr. No.	Methods	Face Forensics Dataset		
		TP (90)		
		Accuracy (%)	Sensitivity (%)	Specificity (%)
1	Linear Regression Classifier	78.92	74.90	78.33
2	Deep CNN Classifier	81.25	80.45	83.78
3	SVM Classifier	82.85	82.59	85.25
4	Decision Tree Classifier	84.92	85.05	85.35
5	Navie Bayes Classifier	89.43	89.47	89.52
6	LieNet	90.88	90.47	89.16
7	CNN	90.89	92.03	91.62
8	Hybrid Boosting Machine	94.09	93.17	93.52
9	**Proposed Classifier**	**95.25**	**95.76**	**95.58**

4.7 Comparative discussion

The effectiveness of the proposed Two-Layer Hybridized Deep CNN classifier has been substantiated through comprehensive comparisons with various existing approaches. For dataset 1, the Two-Layer Hybridized Deep CNN classifier achieved outstanding outcomes of 96.76%, 96.67%, and 96.21%, respectively. Similarly, for dataset 2, the results were 96.56%, 96.79%, and 96.61%, while for dataset 3, they reached 95.25%, 95.76%, and 95.58%, all with a TP (training percentage) value of 90.

5.Conclusion

In this research the Two-Layer Hybridized Deep CNN method is developed for the detection of video forgery. The primary objective is to enhance accuracy and efficiency in identifying manipulated content. The process commences with the collection of input data from a video database, followed by diligent data pre-processing to mitigate noise and inconsistencies. To streamline computational complexity, the research employs key frame extraction to select pivotal frames from the video. Subsequently, these key frames undergo YCbCr conversion to establish feature maps, a step that optimizes subsequent analysis. These feature maps then serve as the basis for extracting significant features, incorporating Haralick features, Local Ternary Pattern, Scale-Invariant Feature Transform (SIFT), and light coefficient features. This multifaceted approach empowers robust forgery detection. The detection is done using the proposed classifier that identifies the forged image. The outputs are measured using accuracy, sensitivity, specificity and the proposed Two-Layer Hybridized Deep CNN classifier achieved 96.76%, 96.67%, 96.21% for dataset 1, 96.56%, 96.79%, 96.61% for dataset 2, 95.25%, 95.76%, 95.58% for dataset 3, which is more efficient than other techniques. In the future, the utilization of hybrid optimization techniques will be integrated into the classifier training process, further enhancing its detection performance.

References

[1] Walia, Savita, Krishan Kumar, Munish Kumar, and Xiao-Zhi Gao. "Fusion of handcrafted and deep features for forgery detection in digital images." IEEE Access 9 (2021), pp. 99742-99755.

[2] Hashmi, Ammarah, Sahibzada Adil Shahzad, Wasim Ahmad, Chia Wen Lin, Yu Tsao, and Hsin-Min Wang. "Multimodal Forgery Detection Using Ensemble Learning." In 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), (2022) pp. 1524-1532.

[3] Mazzetto, Muriel, Marcelo Teixeira, Érick Oliveira Rodrigues, and Dalcimar Casanova. "Deep learning models for visual inspection on automotive assembling line." arXiv preprint arXiv:2007.01857 (2020).

[4] Karnati, Mohan, Ayan Seal, Anis Yazidi, and Ondrej Krejcar. "LieNet: A deep convolution neural network framework for detecting deception." IEEE Transactions on Cognitive and Developmental Systems 14, no. 3 (2021), pp. 971-984.

[5] Singhal, Sakshi, and Virender Ranga. "Passive authentication image forgery detection using multilayer cnn." In Mobile Radio Communications and 5G Networks: Proceedings of MRCN 2020, Springer Singapore, (2021), pp. 237-249.

[6] Saber, Akram Hatem, Mohd Ayyub Khan, and Basim Galeb Mejbel. "A survey on image forgery detection using different forensic approaches." Advances in Science,

Technology and Engineering Systems Journal 5, no. 3 (2020), pp. 361-370.

[7] Sabah, Hanady. "A Detection of Deep Fake in Face Images Using Deep Learning." Wasit Journal of Computer and Mathematics Science 1, no. 4 (2022), pp. 94-111.

[8] Zhu, Hao, Chaoyou Fu, Qianyi Wu, Wayne Wu, Chen Qian, and Ran He. "Aot: Appearance optimal transport-based identity swapping for forgery detection." Advances in Neural Information Processing Systems 33 (2020): 21699-21712.

[9] Tyagi, Shobhit, and Divakar Yadav. "A detailed analysis of image and video forgery detection techniques." The Visual Computer 39, no. 3 (2023), pp. 813-833.

[10] Aloraini, Mohammed, Mehdi Sharifzadeh, and Dan Schonfeld. "Sequential and patch analyses for object removal video forgery detection and localization." IEEE Transactions on Circuits and Systems for Video Technology 31, no. 3 (2020), pp. 917-930.

[11] Raskar, Punam Sunil, and Sanjeevani Kiran Shah. "Real time object-based video forgery detection using YOLO (V2)." Forensic Science International 327 (2021): 110979.

[12] Saddique, Mubbashar, Khurshid Asghar, Usama Ijaz Bajwa, Muhammad Hussain, and Zulfiqar Habib. "Spatial Video Forgery Detection and Localization using Texture Analysis of Consecutive Frames." Advances in Electrical & Computer Engineering 19, no. 3 (2019).

[13] Zhou, Yangming, Qichao Ying, Yifei Wang, Xiangyu Zhang, Zhenxing Qian, and Xinpeng Zhang. "Robust watermarking for video forgery detection with improved imperceptibility and robustness." In 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP), (2022), pp. 1-6.

[14] Haliassos, Alexandros, Rodrigo Mira, Stavros Petridis, and Maja Pantic. "Leveraging real talking faces via self-supervision for robust forgery detection." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2022), pp. 14950-14962.

[15] Qian, Yuyang, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. "Thinking in frequency: Face forgery detection by mining frequency-aware clues." In European conference on computer vision, Cham: Springer International Publishing, (2020), pp. 86-103.

[16] Rodriguez-Ortega, Yohanna, Dora M. Ballesteros, and Diego Renza. "Copy-move forgery detection (CMFD) using deep learning for image and video forensics." Journal of imaging 7, no. 3 (2021): 59.

[17] Aloraini, Mohammed, Mehdi Sharifzadeh, Chirag Agarwal, and Dan Schonfeld. "Statistical sequential analysis for object-based video forgery detection." In IS and T International Symposium on Electronic Imaging Science and Technology, (2019), vol. 2019, no. 5, p. 543.

[18] El-Shafai, Walid, Mona A. Fouda, El-Sayed M. El-Rabaie, and Nariman Abd El-Salam. "A comprehensive taxonomy on multimedia video forgery detection techniques: challenges and novel trends." Multimedia Tools and Applications (2023), pp. 1-67.

[19] Wang, Yukai, Chunlei Peng, Decheng Liu, Nannan Wang, and Xinbo Gao. "Spatial-Temporal Frequency Forgery Clue for Video Forgery Detection in VIS and NIR Scenario." IEEE Transactions on Circuits and Systems for Video Technology (2023).

[20] Huang, Chee Cheun, Chien Eao Lee, and Vrizlynn LL Thing. "A novel video forgery detection model based on triangular polarity feature classification." International Journal of Digital Crime and Forensics (IJDCF) 12, no. 1 (2020), pp. 14-34.

[21] Selvaraj, Priyadharsini, and Muneeswaran Karuppiah. "Inter-frame forgery detection and localisation in videos using earth mover's distance metric." IET Image Processing 14, no. 16 (2020), pp. 4168-4177.

[22] Das, Sayantan, Mojtaba Kolahdouzi, Levent Özparlak, Will Hickie, and Ali Etemad. "Unmasking Deepfakes: Masked Autoencoding Spatiotemporal Transformers for Enhanced Video Forgery Detection." arXiv preprint arXiv:2306.06881 (2023).

[23] Zhao, Chenhui, Xiang Li, and Rabih Younes. "Self-supervised Multi-Modal Video Forgery Attack Detection." In 2023 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, (2023), pp. 1-6.

[24] Oraibi, Mohammed R., and Abdulkareem M. Radhi. "Enhancement Digital Forensic Approach for Inter-Frame Video Forgery Detection Using a Deep Learning Technique." Iraqi Journal of Science (2022), pp. 2686-2701.

[25] Alkawaz, Mohammed Hazim, Maran al Tamil Veeran, Asif Iqbal Hajamydeen, and Omar Ismael Al-Sanjary. "An overview of advanced optical flow techniques for copy move video forgery detection." In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, (2021), pp. 319-324.

[26] Tang, Linlin, Huifen Lu, Zhen Pang, Zhangyan Li, and Jingyong Su. "A distance weighted linear regression classifier based on optimized distance calculating approach for face recognition." Multimedia Tools and Applications 78 (2019): 32485-32501.

[27] Fadl, Sondos, Qi Han, and Qiong Li. "CNN spatiotemporal features and fusion for surveillance video forgery detection." Signal Processing: Image Communication 90 (2021): 116066.

[28] Chittapur, Govindraj, S. Murali, and Basavaraj S. Anami. "Copy create video forgery detection techniques using frame correlation difference by referring SVM classifier." International Journal of Computer Engineering in Research Trends (IJCERT) (2019).

[29] Kuznetsov, Andrey. "Digital video forgery detection based on statistical features calculation." In Twelfth International Conference on Machine Vision (2020), vol. 11433, pp. 723-728.

[30] Sadddique, Mubbashar, Khurshid Asghar, Tariq Mehmood, Muhammad Hussain, and Zulfiqar Habib. "Robust video content authentication using video binary pattern and extreme learning machine." International Journal of Advanced Computer Science and Applications vol. 10, no. 8 (2019).

[31] Ugale, Meena, and J. Midhunchakkaravarthy, "Image Splicing Forgery Detection Model Using Hybrid Boosting Machine." *IAENG International Journal of Computer Science* 51.7 (2024).

[32] DSO-1 and DSI-1 Datasets, https://recodbr.wordpress.com/code-n-data/#dso1_dsi1, Accessed on May 2023.

[33] Face Forensics Database is taken from https://paperswithcode.com/dataset/faceforensics-1

[34] Singh, Ravinder, Sudha Subramani, Jiahua Du, Yanchun Zhang, Hua Wang, Yuan Miao, and Khandakar Ahmed. "Antisocial Behavior Identification from Twitter Feeds Using Traditional Machine Learning Algorithms and Deep Learning." *EAI Endorsed Transactions on Scalable Information Systems* 10, no. 4 (2023).

[35] Yin, Jiao, MingJian Tang, Jinli Cao, Mingshan You, Hua Wang, and Mamoun Alazab. "Knowledge-driven cybersecurity intelligence: Software vulnerability

coexploitation behavior discovery." *IEEE transactions on industrial informatics* 19, no. 4 (2022): 5593-5601.

[36] Raj, Shritik, Bernard Ngangbam, Sanket Mishra, Vivek Gopalasetti, Ayushi Bajpai, and Ch Venkata Rami Reddy. "Knox: Lightweight Machine Learning Approaches for Automated Detection of Botnet Attacks." *EAI Endorsed Transactions on Scalable Information Systems* 11, no. 1 (2024).