Mathematical Modelling of Engineering Problems
Vol. 11, No. 12, December, 2024, pp. 3431-3439
Journal homepage: http://iieta.org/journals/mmep

A Comparison of Deep Learning and Machine Learning Approaches to Video Injection Detection

Vera Suryani[1*], Fazmah Arif Yulianto[1], Parman Sukarno[1], Achmad Rizal[2]

[1] School of Computing, Telkom University, Bandung 40257, Indonesia
[2] School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia

Corresponding Author Email: verasuryani@telkomuniversity.ac.id

https://doi.org/10.18280/mmep.111221

Received: 26 August 2024
Revised: 17 October 2024
Accepted: 23 October 2024
Available online: 31 December 2024

Keywords:
video injection, fake face, deep learning, machine learning, surveillance camera

ABSTRACT

Video injection attack is one of the risks associated with the use of surveillance cameras. Individuals can deceive authorities in a variety of ways when their faces are captured on camera monitoring technology. As a result, numerous types of techniques have been devised to identify counterfeit videos injected into mobile phone displays or in which faces have been substituted with photographs. Face-based video injection detection is investigated in this study by employing deep learning and machine learning techniques. There is one class of authentic face data, and five classes of fabricated face videos comprise the six classes of data in the set. Classification is accomplished using machine learning algorithms such as KNN, SVM, Random Forest and characteristics including texture, color, and shape. In contrast, deep learning does not perform the extraction of features, and optimized using CNN, ANN, and RNN algorithms. The experimental findings indicate that a convolutional neural network (CNN) achieves the highest level of accuracy 100%, followed by the ANN and RNN algorithms with an accuracy of 96% each, both with and without data augmentation. Furthermore, when applied to texture and color features, machine learning in the form of SVM and Random Forest achieved an accuracy of 99%, outperforming the KNN algorithm which had accuracy of 97%. These outcomes demonstrate that deep learning can generate better accurate predictions on a variety of data sets, especially with augmented dataset.

1. INTRODUCTION

Video Injection attacks are a form of fraudulent activity where the objective is to establish a false identity by inserting unauthorized data transmissions between the sensor capture device and the biometric feature extractor during identification verification [1]. Embedded fade video is introduced into the remote identity verification process via emulation and virtual cameras, physical replacement of phone cameras, hacking applications, and man-in-the-middle attacks, among other techniques. Video injection detection techniques are essential in maintaining the integrity of video feeds across various real-world applications. In surveillance systems, they prevent tampering with security footage to mask illegal activities, ensuring public safety. Autonomous vehicles and advanced driver assistance systems (ADAS) rely on these techniques to detect fake video inputs that could cause accidents by spoofing lane markings or obstacles. In healthcare, they safeguard remote surgery and telemedicine by preventing manipulation of live video feeds. Financial institutions use these techniques to detect fraudulent activities in ATMs and CCTV systems, while border control and identity verification systems leverage them to prevent spoofing through pre-recorded videos. Additionally, media broadcasters use injection detection to ensure the authenticity of live content, and military operations depend on it to secure surveillance footage against adversarial attacks, ensuring reliable reconnaissance and mission success. Machine learning and deep learning techniques can be used to detect and prevent any video injection attacks.

Several studies have been conducted to detect video injections. Abaimov and Bianchi [2] surveyed, analyzed and classified some existing machine learning techniques applied to code injection attacks with the focus more on code injection rather than video injection. The techniques discussed could be adapted for use in detecting any video injection attacks. Here, face recognition is one of methods to manage video injection [3] and liveness detection is a term commonly used to detect video injection by means of face recognition [4]. Several features that can be done for liveness detection include pupil dilation, motion detection (mouth, eyes, and head), texture detection, and challenge-response detection [3].

The form of video injection that is mostly done is by faking the face of the user either using photographs, cellular phone screen or using poster [5]. For this reason, many studies on video injection detection have used face recognition [6]. Deep learning becomes one of the most used methods for face recognition in video injection detection [7]. A paper on detecting real and fake faces using deep learning explores the application of convolutional neural networks (CNNs) to identify facial forgeries created by methods like Deepfake and

Face2Face. The study uses architectures such as ResNet50, DenseNet, and MobileNet, which excel in extracting multi-level features from images to improve detection accuracy. By employing pretrained models and fine-tuning on specific datasets, the research achieves high precision in distinguishing authentic faces from manipulated ones, addressing challenges posed by advanced face-swapping technologies. These efforts are crucial in enhancing trust in digital media and ensuring the security of facial recognition systems used across various applications, including security and identity verification systems [8]. Li et al. [9] also used GAN network to detect fake faces with the result of accuracy at 93.1%. Kumar and his team conducted a number of studies including in detecting the face spoofing, and identifying the age, sex, and face expressions. They proposed the biometric authentication model using the collaboration of U-Net architecture and AlexNex architecture. This research overall involved the use of U-Net architecture for the segmentation and AlexNex for the classification. The model they proposed was tested with various dataset such as NUAA, CASIA, Adience, IOG, CK+, and JAFFE. When managing the face spoofing case, the test was conducted to the dataset of NUAA and CASIA and it resulted in 91.1% to NUAA and 92.71% to the CASIA dataset. They claimed that these results were the best of other previous techniques [10].

In this research, a comparison was made between deep learning and machine learning to detect fake faces produced from spoofing techniques. The deep learning model used here included convolutional neural networks (CNN), recurrent neural networks (RNN), and Artificial Neural Networks (ANN). Meanwhile, the machine learning model used was K-nearest neighbor (KNN). From this comparison, it is expected to get an idea of the performance of the two types of classifiers for further research.

2. MATERIAL AND METHOD

Numerous studies investigating video injection attacks using machine learning and deep learning have been conducted by many researchers. Zamir et al. [11] propose a Raspberry Pi-camera-based face recognition system that uses CNNs for feature extraction and classification to identify faces in real time. A bespoke dataset of 700 photos of 7 individuals yielded accuracies of 95.23% (precision: 96.51%, recall: 97.64%, F1-score: 97.07%) at a 70:30 training/testing ratio and 97.71% (precision: 98.09%, recall: 96.26%, F1-score: 97.16%) at an 80:20 ratio. The study shows that CNN outperforms HOG in face detection and recognition, especially in difficult settings, and that hardware affects training time and memory consumption.

Chaabane et al. [12] introduce a statistical feature extraction and Support Vector Machine (SVM) classification face recognition approach with a 99.37% recognition rate and 4.35% EER. Local Binary Patterns (LBP) and Principal Component Analysis (PCA) with sliding windows are used to select and extract features and compute statistical properties like mean, variance, skewness, and kurtosis from images. The experimental results reveal that the suggested method is effective, especially when using numerous postures for each individual, and outperforms existing methods, in spite of its computational requirements and noise sensitivity.

Guo [13] tests the K-Nearest Neighbors (KNN) face recognition algorithm on exposed frontal, profile, and mask-covered faces. The study indicates that KNN has a 95% success rate for exposed frontal faces but 22.2% for profile faces and 2.22% for masks. Traditional face recognition algorithms struggle to identify partially veiled faces and profile photos, stressing the requirement for broad training datasets. It also finds a high false rejection rate when the KNN model is trained on covered faces, suggesting that future applications should focus on eye and forehead identification.

Nassih et al. [14] introduce GD-FM+RF, a fast 3D face recognition system that uses fast marching geodesic distance calculation with a Random Forest classifier. Key points are manually extracted from 3D facial meshes, geodesic distances are measured to create facial curves, and feature separability is achieved using Principal Component Analysis (PCA). The system's 99.11% recognition performance on the SHREC'08 database, which contains 427 images of 61 people, outperformed numerous current methods and showed its ability to robustly identify persons based on 3D facial scans.

Alkishri et al. [15] propose convolutional neural networks (CNNs) and Rotation Invariant Local Binary Patterns (RI-LBP) to detect fake faces with greater accuracy than RGB and HSV. It emphasizes color texture analysis in the YCbCr color space. The technique outperforms prior studies with a 3.2% Equal Error Rate (EER) using the MSU MFSD dataset. Selecting proper texture descriptors is crucial, and the authors urge future research to increase detection accuracy and efficiency in real-world applications, notably in digital media identity theft and fraud.

All the aforementioned papers primarily focus on methodologies in machine learning or deep learning. This study aims to explore the gaps between the two methodologies, particularly in the context of video injection detection.

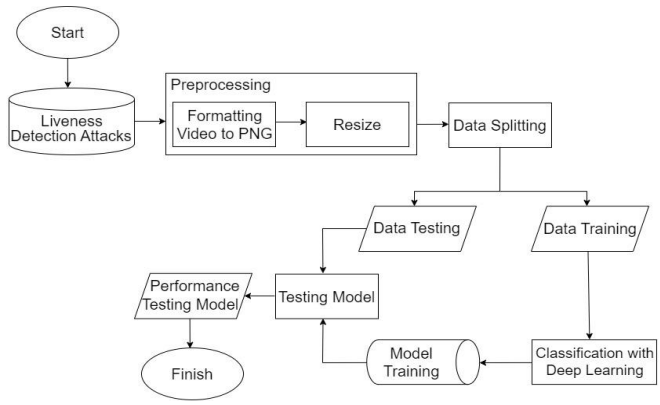


Figure 1. The flowchart of the process in detecting the fake faces using deep learning

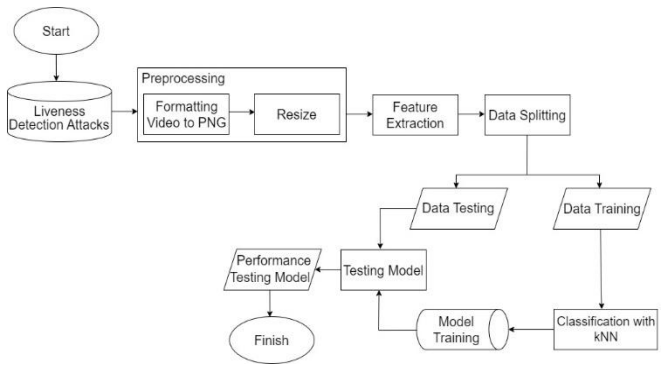


Figure 2. The flowchart of the process in detecting the fake faces using machine learning

Deep learning and machine learning have their own mechanism [16-18]. Deep learning is able to do classification without a need for the feature extraction process; thus, the process as shown in Figure 1 is used. While for machine learning, it needs the feature extraction process; thus, the process as shown in Figure 2 is used. The detail of each process is presented in the following subsections.

2.1 Dataset and processing

This research used the video dataset available on Kaggle [19]. The dataset was processed by changing the video into an image with extension (.png). The dataset consisted of six types of videos as shown in Figure 3. Pre-processing was done by involving the change of image format into PNG format and the adjustment of image size to make it consistent with the size of 224×224 pixel. In this study we used two classes of data, real and fake face.

Figure 3. Type of image in dataset [12]

2.2 Feature extraction and selection

The process of feature extraction in the image covered three types of main features. First, there was an extraction of texture features using Gray-Level Co-occurrence Matrix (GLCM) method [20]. This study used 7 GLCDM features: correlation, homogeneity, dissimilarity, contrast, energy, and angular second moment. The seven features were calculated at 0°, 45°, 90°, and 135°. If $P_{i,j}$ is the GLCM probability, then the features used are calculated as in Eqs. (1)-(7):

$$Correlation = \frac{\sum_i \sum_j (i-\bar{x})(j-\bar{y})P(i,j)}{\sigma_x \sigma_y} \quad (1)$$

$$ASM = \sum_i \sum_j P(i,j)^2 \quad (2)$$

$$Homogeneity = \sum_{i_1} \sum_{i_1} \frac{p(i_1,i_2)}{1+|i_1-i_2|} \quad (3)$$

$$Dissimilarity = \sum_{i,j=0} P_{i,j}|i-j| \quad (4)$$

$$Contrast = \sum_{i_1} (i_1-i_2)^2 p(i_1,i_2) \quad (5)$$

$$Energy = \sum_{i_1} \sum_{i_1} p^2(i_1,i_2) \quad (6)$$

$$ASM = \sum_i \sum_j P(i,j)^2 \quad (7)$$

Second, the form feature was extracted by calculating the value of eccentricity of the image. The eccentricity of an image is presented in Eq. (8).

$$e = \sqrt{1-\frac{b^2}{a^2}} \quad (8)$$

where, e=eccentricity, b=minor axis, and a=major axis.

Last, the feature of color was extracted using the HSV (Hue, Saturation, Value) color covering three main components: hue, saturation, and value [21]. This process could help to identify and select the dominant colors in the image. Overall, the process of feature extraction aimed to obtain the more level representation from the image to be used in any assignments of image analysis and image processing.

For the selection feature, several combinations are made between the texture, color, and shape features. Seven combinations are obtained for the entire selection feature: texture, color, shape, texture and color, texture and shape, color and shape, texture, color and shape.

2.3 Data splitting

The dataset was split into two parts for training data and data of model testing, each of which consisted of 80% for training process and 20% for training process of entire dataset, both for augmented and non-augmented data. Thus, the total data in the first scenario was 32 training data and 128 testing data. While the second scenario used 640 training data and 2560 testing data.

2.4 Classification

In this study, we did a comparison among convolutional neural networks (CNN), Artificial Neural Networks (ANN), and recurrent neural network (RNN) in the context of deep learning for image analysis. Furthermore, we considered the machine learning approach with K-Nearest Neighbors (KNN), SVM, and Random Forest as the comparison.

2.4.1 Convolutional neural networks (CNN)

CNN has been proven to have a good performance in any field, including in computer vision and machine learning [22, 23]. It consists of a number of layers including convolutional layer, pooling layer, and activation layer in which each layer in CNN has a special role. The convolutional layer acts to extract the low-level feature; pooling layer acts to reduce the dimension of feature maps while maintaining the important information for the classification process. The convolution process principally involves the operation of dot product in which its results will be forwarded into the activation layer to improve the non-linear feature in CNN [24]. Illustration of CNN network depicted in Figure 4.

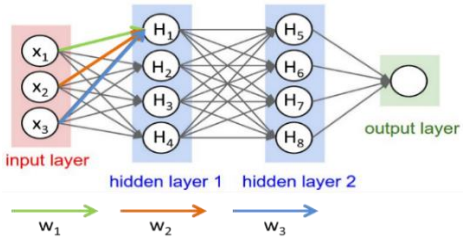


Figure 4. Illustration of the architecture of CNN network

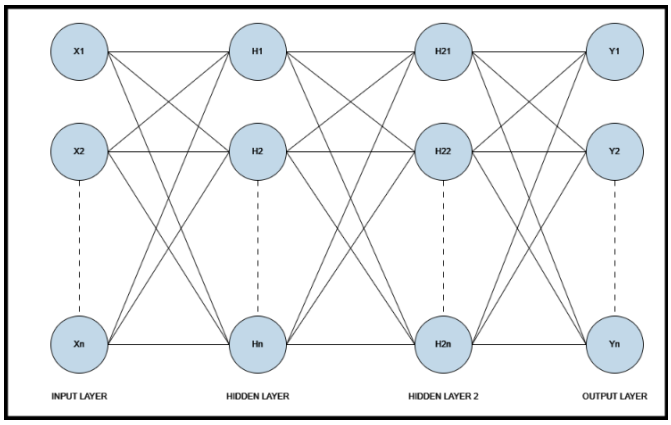


Figure 5. Illustration of the architecture of ANN multilayer feed forward network

2.4.2 Artificial Neural Networks (ANN)

ANN can be characterized as a model of information processing derived from the behavioral mechanisms of human brain networks in information processing [18]. However, what this ANN can only do is to imitate the basic function of the human brain. ANN consists of many processing units (neurons) that are connected to each other. Similar with humans, ANN also learns from any existing examples or experiences [14]. Each neuron is connected to each other via connection links, each of which is associated with a weight, containing information about the input signal. This information will later be used by the neuron network to solve certain problems [20]. Illustration of the ANN architecture is presented in Figure 5.

2.4.3 Recurrent neural network (RNN)

RNN is a neural network architecture developed for processing sequential data through feedback connections, enabling maintaining data over time steps. Unlike traditional neural networks, RNNs maintain a hidden state, which enables them to capture dependencies within sequences, making them ideal for tasks like time series prediction, language modeling, and speech recognition.

Specific configurations of RNN:

1) Vanilla RNN
2) Long Short-Term Memory (LSTM)
3) Gated Recurrent Unit (GRU)

These configurations allow RNNs to adapt to various sequential tasks with trade-offs between complexity, computation, and the ability to retain long-term dependencies.

2.4.4 K-Nearest Neighbors (KNN)

KNN is one of the methods in machine learning [25]. In KNN, the distance matrix such as Euclidean distance is used to measure the distance between data points. Euclidean distance [26] is measured by means of the Eq. (9).

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (9)$$

In KNN, there is a parameter called k, which is used to determine the number of neighbor's to be considered when classifying or predicting the data points [27]. It is important to choose the appropriate k value as it can determine the performance of KNN model; thus, the k value should not be so high or so low.

2.4.5 Support Vector Machine (SVM)

SVM is a supervised learning technique utilized for classification and regression applications, while it is predominantly employed for problems related to classification. It finds the optimal hyperplane that maximally separates different classes in the feature space. For non-linearly separable data, SVM uses kernel functions (e.g., RBF, polynomial) to transform the input space into a higher-dimensional feature space where a linear separator can be found. The objective of SVM is to maximize the margin between the closest points (support vectors) from each class.

- Kernel Functions: Linear, Polynomial, RBF, Sigmoid
- Hyperparameters:
  - C: Regularization parameter (regulates the balance between margin size and classification error)
  - Gamma: Controls how far influence of a single data point reaches (for RBF or polynomial kernels)

2.4.6 Random Forest

Random Forest is an ensemble learning technique that constructs numerous decision trees during training and integrates their predictions to enhance accuracy and mitigate overfitting. Each tree is trained on a random subset of data (via bootstrap sampling) and uses a random subset of features to make splits, enhancing diversity in the model. The final prediction is made by majority voting for classification or averaging for regression.

- Key Hyperparameters:
  - n_estimators: Number of trees in the forest
  - max_features: Number of features considered for each split
  - max_depth: Maximum depth of each tree
- Advantage: Highly robust to overfitting, efficient in handling large datasets, and resistant to noise.

Both algorithms are widely used, with SVM excelling in smaller, well-separated datasets and Random Forest offering strong performance in complex, high-dimensional datasets.

2.5 System performance measurement

In evaluating the performance of the deep learning and machine learning model, confusion matrix as shown in Table 1 was used. Confusion matrix had a number of matrices including accuracy, precision, recall, and F1 score as shown in Eqs. (10)-(13). To calculate the matrices, four components; those are True Positives (TP), False Positives (FP), True Negative (TN), and False Negative (FN) were used. The matrices could help in predicting the real image and unreal one [28, 29].

Table 1. Confusion matrix

		Actual	
		(+)	(-)
Predicted	(+)	TP (True Positive)	FP (False Positive)
	(-)	FN (False Negative)	TN (True Negative)

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (10)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (11)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (12)$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (13)$$

2.6 Data augmentation and testing scenario

Data augmentation simultaneously increases the quantity and quality of training datasets. This allows the development of deeper learning models [30]. The data augmentation process using ImageDataGenerator is carried out to apply a series of transformations to the dataset (containing real and unreal images) to enrich the variety of data that will be used in model training. In general, data augmentation in this research involves several rules, including:

a. Flip: Horizontal, Vertical
b. Crop: 0% Minimum Zoom, 20% Maximum Zoom
c. Rotation: Between -15° and +15°
d. Hue: Between -15° and +15°
e. Saturation: Between -25% and +25%
f. Brightness: Between -15% and +15%
g. Exposure: Between -10% and +10%
h. Blur: Up to 2.5px
i. Noise: Up to 0.1% of pixels

The scenario used in this research was by comparing the use of data number (160 real dataset and 3200 datasets as the results of data augmentation).

3. RESULT AND DISCUSSION

The hyperparameters used in the three deep learning algorithms can be seen in Table 2.

Hyperparameter setting used for KNN, SVM, and Random Forest depicted in Tables 3-5.

Table 2. Deep learning hyperparameters

Model	Parameter	Value
ANN (FNN)	optimizer	Adam
	batch_size	32
	epoch	10
	loss	binary_crossentropy
	metrics	accuracy
CNN (EfficientNet)	optimizer	Adam
	batch_size	32
	epoch	10
	loss	sparse_categorical_crossentropy
	metrics	accuracy
RNN (LSTM)	optimizer	Adam
	batch_size	32
	epoch	10
	loss	binary_crossentropy
	metrics	accuracy
	weights	uniform

Table 3. KNN hyperparameters

With and Without Data Augmentation		
Features	**Parameter**	**Value**
Texture	metric	euclidean
	n_neighbors	2-10
	weights	distance
Shape	metric	euclidean, manhattan
	n_neighbors	1-15
	weights	uniform, distance
Color	metric	manhattan, euclidean
	n_neighbors	1-2
	weights	distance
Texture & Shape	metric	euclidean, manhattan
	n_neighbors	1-15
	weights	uniform, distance
Texture & Color	metric	manhattan, euclidean
	n_neighbors	1-2
	weights	distance
Color & Shape	metric	euclidean, manhattan
	n_neighbors	1-15
	weights	uniform, distance
Texture, Color, & Shape	metric	euclidean, manhattan
	n_neighbors	1-15
	weights	uniform, distance

Table 4. SVM hyperparameter

With and Without Data Augmentation		
Features	**Parameter**	**Value**
Texture	C	8.424426408
	class_weight	balanced
	gamma	auto
	kernel	rbf
Shape	C	3.845401188
	class_weight	balanced
	gamma	scale
	kernel	linear
Color	C	8.424426408
	class_weight	balanced
	gamma	auto
	kernel	rbf
Texture & Shape	C	3.845401188
	class_weight	balanced
	gamma	scale
	kernel	linear
Texture & Color	C	8.424426408
	class_weight	balanced
	gamma	auto
	kernel	rbf
Color & Shape	C	3.845401188
	class_weight	balanced
	gamma	scale
	kernel	linear
Texture, Color, & Shape	C	3.845401188
	class_weight	balanced
	gamma	scale
	kernel	linear

As shown in Table 6 and Table 7, the results of experiments carried out showed the accuracy of the deep learning approach using real data in each model in which CNN reached 100%, ANN reached 86%, and RNN reached 86%. Meanwhile, when using augmented data, the accuracy of each model showed that CNN reached 100%, ANN reached 96%, and RNN reached 96%. The evaluation between the two showed no significant difference in the accuracy of the CNN model, while for ANN and RNN, the use of augmented data resulted in increasing the accuracy.

Table 5. Random Forest hyperparameters

Without Data Augmentation		With Data Augmentation	
Features	**Parameter**	**Value**	**Value**
Texture	bootstrap	False	False
	max_depth	73	73
	max_features	log2	log2
	min_samples_leaf	1	1
	min_samples_split	13	13
	n_estimators	413	413
Shape	bootstrap	False	True
	max_depth	98	11
	max_features	sqrt	log2
	min_samples_leaf	10	15
	min_samples_split	17	8
	n_estimators	370	363
Color	bootstrap	False	False
	max_depth	73	73
	max_features	sqrt	sqrt
	min_samples_leaf	3	3
	min_samples_split	6	6
	n_estimators	406	406
Texture & Shape	bootstrap	False	True
	max_depth	98	11
	max_features	sqrt	log2
	min_samples_leaf	10	15
	min_samples_split	17	8
	n_estimators	370	363
Texture & Color	bootstrap	False	False
	max_depth	98	73
	max_features	sqrt	log2
	min_samples_leaf	10	1
	min_samples_split	17	13
	n_estimators	370	413
Color & Shape	bootstrap	False	True
	max_depth	98	30
	max_features	sqrt	sqrt
	min_samples_leaf	10	7
	min_samples_split	17	19
	n_estimators	370	487
Texture, Color, & Shape	bootstrap	False	True
	max_depth	98	30
	max_features	sqrt	sqrt
	min_samples_leaf	10	7
	min_samples_split	17	19
	n_estimators	370	487

Table 6. Classification report on deep learning model without data augmentation

Model	Class	Precision	Recall	F1-Score	Accuracy
ANN	Fake	0.86	1.00	0.92	0.86
	Real	0.00	0.00	0.00	
CNN	Fake	1.00	0.99	1.00	1.00
	Real	0.97	1.00	0.98	
RNN	Fake	0.86	1.00	0.92	0.86
	Real	0.00	0.00	0.00	

Table 7. Classification report on deep learning model with data augmentation

Model	Class	Precision	Recall	F1-Score	Accuracy
ANN	Fake	0.96	1.00	0.65	0.96
	Real	0.00	0.00	0.00	
CNN	Fake	1.00	1.00	1.00	1.00
	Real	1.00	1.00	1.00	
RNN	Fake	0.96	1.00	0.65	0.96
	Real	0.00	0.00	0.00	

Table 8. Classification report on KNN for non-augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.87	0.95	0.91	0.84
	Real	0.45	0.24	0.31	
Shape	Fake	0.91	0.60	0.72	0.61
	Real	0.23	0.67	0.35	
Color	Fake	0.98	0.99	0.99	0.98
	Real	0.95	0.90	0.93	
Texture & Shape	Fake	0.93	0.59	0.72	0.61
	Real	0.25	0.76	0.38	
Texture & Color	Fake	0.98	0.99	0.99	0.98
	Real	0.95	0.90	0.93	
Shape & Color	Fake	0.91	0.59	0.71	0.60
	Real	0.23	0.67	0.34	
Texture, Shape & Color	Fake	0.92	0.58	0.71	0.60
	Real	0.24	0.71	0.36	

Table 9. Classification report on SVM for non-augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.94	0.84	0.89	0.82
	Real	0.45	0.71	0.56	
Shape	Fake	0.93	1.00	0.97	0.94
	Real	1.00	0.62	0.76	
Color	Fake	0.96	0.98	0.97	0.95
	Real	0.89	0.76	0.82	
Texture & Shape	Fake	0.93	1.00	0.97	0.94
	Real	1.00	0.62	0.76	
Texture & Color	Fake	0.96	0.99	0.97	0.96
	Real	0.94	0.76	0.84	
Shape & Color	Fake	0.94	1.00	0.97	0.95
	Real	1.00	0.67	0.80	
Texture, Shape & Color	Fake	0.94	1.00	0.97	0.95
	Real	1.00	0.67	0.80	

Table 10. Classification report on Random Forest for non-augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.88	0.93	0.9	0.83
	Real	0.43	0.29	0.34	
Shape	Fake	0.92	0.95	0.94	0.89
	Real	0.67	0.57	0.62	
Color	Fake	0.95	1	0.97	0.96
	Real	1	0.71	0.83	
Texture & Shape	Fake	0.93	0.95	0.94	0.9
	Real	0.68	0.62	0.65	
Texture & Color	Fake	0.95	1	0.97	0.96
	Real	1	0.71	0.83	
Shape & Color	Fake	0.92	0.99	0.95	0.92
	Real	0.92	0.52	0.67	
Texture, Shape & Color	Fake	0.93	0.99	0.96	0.93
	Real	0.92	0.57	0.71	

Tables 6 and 7 show that data augmentation improves the performance of ANN and RNN. Meanwhile, CNN's accuracy has reached 100%. The improvement in accuracy after the data augmentation process occurred because more data was trained on RNN and CNN.

The results of the experiment on the Machine Learning approach showed significant variation. According to Tables 8-10, the highest accuracy was achieved through the combination of texture + color features for non-augmented

data. Therefore, it can be concluded that, in this context, it is still possible only to use texture + color features.

Table 11. Classification report on KNN for augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.99	0.93	0.96	0.96
	Real	0.94	0.99	0.97	
Shape	Fake	0.96	0.06	0.12	0.55
	Real	0.54	1.00	0.70	
Color	Fake	0.98	0.96	0.97	0.97
	Real	0.97	0.99	0.98	
Texture & Shape	Fake	0.95	0.06	0.11	0.55
	Real	0.54	1.00	0.70	
Texture & Color	Fake	0.99	0.95	0.97	0.97
	Real	0.96	0.99	0.97	
Shape & Color	Fake	0.96	0.07	0.14	0.55
	Real	0.54	1.00	0.70	
Texture, Shape & Color	Fake	0.95	0.06	0.11	0.55
	Real	0.54	1.00	0.70	

Table 12. Classification report on SVM for augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.99	0.99	0.99	0.99
	Real	0.99	0.99	0.99	
Shape	Fake	0.97	0.99	0.98	0.98
	Real	0.99	0.97	0.98	
Color	Fake	0.98	0.99	0.98	0.98
	Real	0.99	0.98	0.99	
Texture & Shape	Fake	0.97	0.99	0.98	0.98
	Real	0.99	0.97	0.98	
Texture & Color	Fake	0.99	0.99	0.99	0.99
	Real	0.99	0.99	0.99	
Shape & Color	Fake	0.98	0.99	0.98	0.98
	Real	0.99	0.98	0.99	
Texture, Shape & Color	Fake	0.99	0.99	0.99	0.99
	Real	0.99	0.99	0.99	

Table 13. Classification report on Random Forest for augmented data

Feature	Class	Precision	Recall	F1-Score	Accuracy
Texture	Fake	0.95	0.99	0.97	0.97
	Real	0.99	0.95	0.97	
Shape	Fake	0.95	0.99	0.97	0.97
	Real	0.99	0.95	0.97	
Color	Fake	0.97	0.99	0.98	0.98
	Real	0.99	0.97	0.98	
Texture & Shape	Fake	0.95	0.99	0.97	0.97
	Real	0.99	0.96	0.97	
Texture & Color	Fake	0.99	1.00	0.99	0.99
	Real	1.00	0.99	0.99	
Shape & Color	Fake	0.96	1.00	0.98	0.98
	Real	1.00	0.96	0.98	
Texture, Shape & Color	Fake	0.97	1.00	0.99	0.99
	Real	1.00	0.98	0.99	

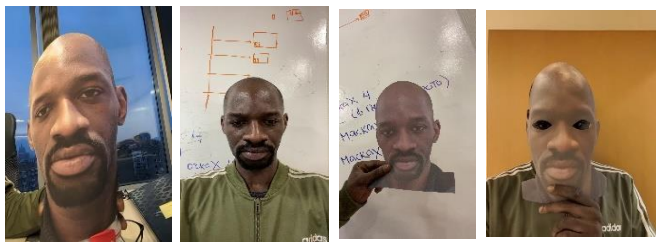
Furthermore, for augmented data, the performance of KNN, SVM, and Random Forest are depicted in Tables 11-13.

In this study, the texture feature is intended to capture changes in intensity in the image. In the original image, changes in intensity tend to be moderate or reasonable because they are not too striking. Meanwhile, in the fake image, there are several unnatural changes in intensity, such as a clear boundary line between the face and the background, blurred facial parts, and others. For example, images in Figures 6(a) and 6(b) show parts with low contrast on the face.

Meanwhile, the color feature uses HSV values. HSV has proven to be good at detecting human skin tones [31]. The shape feature is intended to detect the shape of the main object in the image. For example, the eccentricity value in Figure 6(c) will differ from that in Figure 6(a).

From the experiment using machine learning, SVM produces the highest accuracy of up to 99% for texture features and a combination of texture + color and texture + color + shape. This shows that the texture feature can distinguish real and fake images significantly. Changes in pixel values in images captured through texture analysis can also distinguish real and fake images. Color and shape feature also produce quite high accuracy. Changes in the shape and color of fake images can be recognized through the resulting features. Combining the three features does not produce higher accuracy, possibly due to the similar dataset between the real and fake images.

Figures 7 and 8 illustrate the confusion matrix for CNN and SVM after the application of data augmentation. In the case of CNN, all data were classified with complete accuracy of 100%. Simultaneously, six data points were inaccurately classified by the SVM. This outcome remains acceptable as the overall accuracy is still 99%.



(a) (b) (c) (d)

Figure 6. (a) Real image; (b) Outlined; (c) Outline 3D; (d) Masked

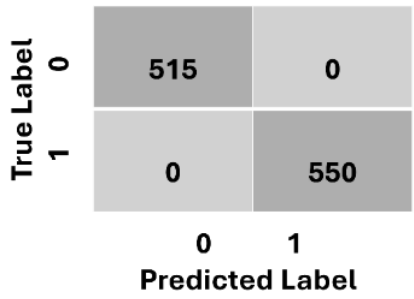


Figure 7. CNN confusion matrix

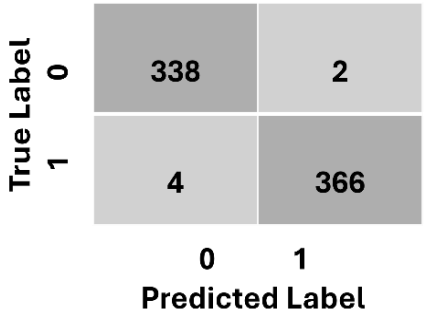


Figure 8. SVM confusion matrix

Table 14. ANOVA analysis

Methods	p-Value
Non-augmented ML	0.01128943
Augmented ML	0.00194892
Non-aug & aug. DL	0.33335348

In this case, the deep learning approach had higher performance compared to the machine learning approach. This can be seen from the analysis results as shown in Table 6 through Table 13, where deep learning models such as CNN, ANN, and RNN achieved higher accuracy compared to machine learning approach, especially the KNN model. Also, the dataset augmentation had a significant impact on the increase of the accuracy of the models built, especially in ANN and RNN models. These results showed that the number of data highly affected the model accuracy, especially in ANN and RNN model.

The importance of feature selection in the model was also proven significant. The use of a combination of texture and color features produced good accuracy without any significant difference between the use of texture+color feature and the use of texture+shape+color features. This emphasizes the importance of selecting appropriate features in image analysis. In the context of developing better and more accurate image analysis systems, these results provide valuable guidance in selecting approaches, using data, and selecting appropriate features.

Table 14 presents the ANOVA analysis of the accuracy for machine learning and deep learning. The augmentation process in ML yields a p-value <0.01, indicating a statistically significant difference. In deep learning, a large p-value indicated that the augmentation process did not yield significant gains in accuracy. This was proven by CNN, which achieves 100% accuracy under both non-augmented and augmented scenarios.

4. CONCLUSION

In this research, detection of fake face video injections was compared using deep learning and machine learning. The accuracy of the test results was significantly improved when CNN was employed, whereas machine learning resulted in a slight reduction in accuracy.

In machine learning, a feature extraction process was carried out because machine learning could not accept the two-dimensional input. Texture, color, and shape features have been shown to differentiate between real and fake images. In contrast, deep learning could receive two-dimensional image input making the classification process able to be carried out without a feature extraction process. From the result testing on KNN, texture and color characteristics provided the highest accuracy. Meanwhile, CNN produced similar accuracy without data augmentation. In general, the results of deep learning far exceeded machine learning even without a feature extraction process. With the availability of CNNs with one-dimensional input, it would be interesting to compare the accuracy of CNNs with the same feature input as those used in machine learning.

As future work, machine learning can be improved by employing the feature selection technique to obtain more effective selective features. In addition, alternative deep learning algorithms like LSTM can be utilized to get better accuracy values. Also, this research pertains to static dataset detection; it can be enhanced by implementing real-time video injection detection.

ACKNOWLEDGEMENT

The article was supported by the Ministry of Education, Culture, Research, and Technology of Indonesia (Grants No.: 003/SP2H/RT-MONO/LL4/2023; 043/SP2H/RTMONO/LL4/2024).

REFERENCES

[1] Brown, T. (2023). Video Injection Attacks: What Are They and Are We Ignoring the Simple Solution? https://www.prove.com/blog/video-injection-attacks-what-are-they-and-are-we-ignoring-the-simple-solution.

[2] Abaimov, S., Bianchi, G. (2021). A survey on the application of deep learning for code injection detection. Array, 11: 100077. https://doi.org/10.1016/j.array.2021.100077

[3] Carta, K., Barral, C., El Mrabet, N., Mouille, S. (2022). Video injection attacks on remote digital identity verification solution using face recognition. In 13th International Multi-Conference on Complexity, Informatics and Cybernetics. Informatics and Cybernetics, International Institute of Informatics and Cybernetics, IIIC, 2022: 92-97. https://doi.org/10.54808/IMCIC2022.02.92

[4] Liu, S., Song, Y., Zhang, M., Zhao, J., Yang, S., Hou, K. (2019). An identity authentication method combining liveness detection and face recognition. Sensors, 19(21): 4733. https://doi.org/10.3390/s19214733

[5] Khairnar, S., Gite, S., Kotecha, K., Thepade, S.D. (2023). Face liveness detection using artificial intelligence techniques: A systematic literature review and future directions. Big Data and Cognitive Computing, 7(1): 37. https://doi.org/10.3390/bdcc7010037

[6] Raheem, E.A., Ahmad, S.M.S., Adnan, W.A.W. (2019). Insight on face liveness detection: A systematic literature review. International Journal of Electrical & Computer Engineering, 9(6): 5165-5175. https://doi.org/10.11591/ijece.v9i6.pp5165-5175

[7] Suganthi, S.T., Ayoobkhan, M.U.A., Bacanin, N., Venkatachalam, K., Štěpán, H., Pavel, T. (2022). Deep learning model for deep fake face recognition and detection. PeerJ Computer Science, 8: e881. https://doi.org/10.7717/peerj-cs.881

[8] Atwan, J., Wedyan, M., Albashish, D., Aljaafrah, E., Alturki, R., Alshawi, B. (2024). Using deep learning to recognize fake faces. International Journal of Advanced Computer Science and Applications, 15(1): https://doi.org/10.14569/IJACSA.2024.01501113

[9] Li, S., Dutta, V., He, X., Matsumaru, T. (2022). Deep learning based one-class detection system for fake faces generated by GAN network. Sensors, 22(20): 7767. https://doi.org/10.3390/s22207767

[10] Kumar, S., Rani, S., Jain, A., Verma, C., Raboaca, M.S., Illés, Z., Neagu, B.C. (2022). Face spoofing, age, gender and facial expression recognition using advance neural network architecture-based biometric system. Sensors, 22(14): 5160. https://doi.org/10.3390/s22145160

[11] Zamir, M., Ali, N., Naseem, A., Ahmed Frasteen, A.,

Zafar, B., Assam, M., Othman, M., Attia, E.A. (2022). Detection and recognition from images and videos based on CNN and raspberry Pi. Computation, 10(9): 148. https://doi.org/10.3390/computation10090148

[12] Chaabane, S.B., Hijji, M., Harrabi, R., Seddik, H. (2022). Face recognition based on statistical features and SVM classifier. Multimedia Tools and Applications, 81(6): 8767-8784. https://doi.org/10.1007/s11042-021-11816-w

[13] Guo, X. (2021). A KNN classifier for face recognition. In 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, pp. 292-297. https://doi.org/10.1109/CISCE52179.2021.9445908

[14] Nassih, B., Amine, A., Ngadi, M., Azdoud, Y., Naji, D., Hmina, N. (2021). An efficient three-dimensional face recognition system based Random Forest and geodesic curves. Computational Geometry, 97: 101758. https://doi.org/10.1016/j.comgeo.2021.101758

[15] Alkishri, W., Widyarto, S., Yousif, J.H., Al-Bahri, M. (2023). Fake face detection based on colour textual analysis using deep convolutional neural network. Journal of Internet Services and Information Security, 13(3): 143-155. https://doi.org/10.58346/JISIS.2023.I3.009

[16] Jondri, J., Rizal, A. (2020). Classification of premature ventricular contraction (PVC) based on ECG signal using convolutional neural network. Indonesian Journal of Electrical Engineering and Informatics, 8(3): 494-499. http://doi.org/10.52549/ijeei.v8i3.1530

[17] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1): e4150. https://doi.org/10.1002/ett.4150

[18] Helm, J.M., Swiergosz, A.M., Haeberle, H.S., Karnuta, J.M., Schaffer, J.L., Krebs, V.E., Spitzer, A.I., Ramkumar, P.N. (2020). Machine learning and artificial intelligence: definitions, applications, and future directions. Current Reviews in Musculoskeletal Medicine, 13: 69-76. https://doi.org/10.1007/s12178-020-09600-8

[19] Kaggle. iBeta level 1 liveness detection dataset-part 1. https://www.kaggle.com/datasets/trainingdatapro/ibeta-level-1-liveness-detection-dataset-part-1, accessed on Nov. 6, 2023.

[20] Rizal, A., Wijayanto, I., Istiqomah, I. (2023). Alcoholism detection in EEG signals using GLCM-based texture analysis of image-Converted signals. In 2023 6th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, pp. 275-279. https://doi.org/10.1109/ICOIACT59844.2023.10455889

[21] Prasetyo, E. (2018). Eye black circle of milkfish segmentation on HSV color space. Journal of Electrical Engineering and Computer Sciences, 3(1): 377-380. https://doi.org/10.54732/jeecs.v3i1.143

[22] Mutasa, S., Sun, S., Ha, R. (2021). Understanding artificial intelligence based radiology studies: CNN architecture. Clinical Imaging, 80: 72-76. https://doi.org/10.1016/j.clinimag.2021.06.033

[23] Wang, Z.J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M., Chau, D.H.P. (2020). CNN explainer: Learning convolutional neural networks with interactive visualization. IEEE Transactions on Visualization and Computer Graphics, 27(2): 1396-1406. https://doi.org/10.1109/TVCG.2020.3030418

[24] Namburi, D.L. (2022). Speaker recognition based on mutated monarch butterfly optimization configured artificial neural network. International Journal of Electrical and Computer Engineering Systems, 13(9): 767-775. https://doi.org/10.32985/ijeces.13.9.5

[25] Wang, L. (2019). Research and implementation of machine learning classifier based on KNN. IOP Conference Series: Materials Science and Engineering, 677(5): 052038. https://doi.org/10.1088/1757-899X/677/5/052038

[26] Zhang, Z. (2016). Introduction to machine learning: K-Nearest Neighbors. Annals of Translational Medicine, 4(11): 218. https://doi.org/10.21037/atm.2016.03.37

[27] Dong, S., Sarem, M. (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. IEEE Access, 8: 5039-5048. https://doi.org/10.1109/ACCESS.2019.2963077

[28] Hasnain, M., Pasha, M.F., Ghani, I., Imran, M., Alzahrani, M.Y., Budiarto, R. (2020). Evaluating trust prediction and confusion matrix measures for web services ranking. IEEE Access, 8: 90847-90861. https://doi.org/10.1109/ACCESS.2020.2994222

[29] Luque, A., Carrasco, A., Martín, A., de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91: 216-231. https://doi.org/10.1016/j.patcog.2019.02.023

[30] Shorten, C., Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6: 60. https://doi.org/10.1186/s40537-019-0197-0

[31] Kamble, S., Muntean, C.H., Simiscuka, A.A. (2024). A Hybrid HSV and YCrCb OpenCV-based skin tone recognition mechanism for makeup recommender systems. In 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, pp. 1224-1229. https://doi.org/10.1109/IWCMC61514.2024.10592313