Video Spoofing Attack Detection Using Convolutional Neural Networks and Ensemble Learning Voting Classifier Methods

1st Siti Vanesa Rahma
School of Computing
Telkom University
Bandung, Indonesia
sitivanesarahma@student.telkomuniversity.ac.id

2nd Vera Suryani
School of Computing
Telkom University
Bandung, Indonesia
verasuryani@telkomuniversity.ac.id

***Abstract*—Facial recognition technology is increasingly being used in various applications, but this has resulted in the emergence of new threats such as spoofing. Existing detection systems still face several shortcomings, such as low accuracy or limited variety of training data, making them vulnerable to spoofing attacks. This research develops a spoofing detection system based on Convolutional Neural Networks (CNN) and ensemble learning methods that combine several models, such as Support Vector Machines (SVM), Random Forests, and Logistic Regression with Voting Classifier techniques tested on the iBeta Level 1 - Liveness Detection Dataset. This approach is done by utilizing a combination of models to improve detection accuracy and reduce the weakness of individual models. The proposed system is tested using validation and test datasets to ensure no overfitting/underfitting occurs. The experimental results in the test data in this study show that the method achieves 97% accuracy on tests and 98% on validation for ensemble learning, as well as 100% on test and validation for CNN-based models. These findings prove the effectiveness of deep learning and ensemble learning approaches in detecting spoofing, thus potentially improving the security of face recognition systems.**

Keywords—spoofing, deep learning, ensemble learning, overfitting, underfitting, accuracy

I. Introduction

Facial recognition technology is currently a very important component for the process of identifying or authenticating identities in life, ranging from public security, finance, military, to everyday life [1]. At the same time, the security threats of these facial recognition systems are growing, especially with Spoofing attacks. Spoofing refers to impersonating a legitimate user through the use of images, videos, or face masks to fool a facial recognition system. Due to the fact that the incidence of this crime continues to grow, it is imperative that crime detection and prevention mechanisms be researched.

Several studies have been conducted in an attempt to find a solution to this problem. However, each of these studies has limitations that need to be addressed. One example is the research conducted by [2], which resulted in accuracy rates between 88 to 90 percent on the validation dataset. On the other hand, the model was unable to account for variations in lighting conditions, which are often experienced in real-world applications. Research conducted by [3] revealed that the Sequential CNN algorithm produced a relatively low accuracy of 87%, while the Naïve Bayes method only managed to get an accuracy of 81.2%. This shows that the architecture and techniques used are not the most effective. When compared to this study, other studies, such as [4] only achieved 77.41% accuracy using the SVM technique because the selected features were limited to RGB average, variance, and luminance. In addition, most of these methods are unable to handle real-world conditions, such as illumination variations, or more complex types of spoofing attacks, such as the use of 3D masks. This emphasizes the need for more sophisticated approaches to detect these attacks. Previous studies using machine learning tend to have performance limitations in lower accuracy than those using deep learning. Therefore, in this study, ensemble learning is used to see if this algorithm can be better when compared to deep learning.

This research aims to overcome the limitations of previous methods by developing a face spoofing detection system that is more accurate and resistant to various attack conditions. This research utilizes Convolutional Neural Networks or commonly called CNN with Xception architecture that excels in detecting complex patterns. In addition, to increase the generalization capacity of the model, a Voting classifier-based ensemble learning strategy that integrates Support Vector Machine or commonly called SVM, Random Forest, and Logistic Regression is used. With the use of these models, this research can handle challenges such as illumination differences, frame variations, as well as detect advanced Spoofing attacks, while improving accuracy over previous methods.

II. Relate Work

Spoofing [5] is the current state of deception of facial recognition technology to disrupt the biometric validation process so that the system cannot correctly recognize real and fake faces. This situation has become a widespread concern in the field of digital media and cybersecurity because there have been various negative impacts from the development of these threats. Research related to Spoofing detection has shown progress to eradicate the negative impact of Spoofing.

Several studies have introduced methods to detect Spoofing using one of the fields of Artificial Intelligent (AI), namely machine learning and deep learning. The machine learning and deep learning approaches used also vary, one of which is in research [6] detecting with CNN, Naive Bayes, KNN, SVM, Random Forest, and Decission Tree algorithms giving an average maximum accuracy of around 87.5%. Another study [7] using the Video Vision Transformer (ViViT) deep learning algorithm found that on the Rose-Youtu Dataset the training set accuracy was obtained at 98.34% but dropped in the validation set and test set by 86.47% and 86.78%. The research [8] shows high accuracy for the use of internal datasets with training accuracy of 95%,

testing accuracy of 93.2%, and validation accuracy of 93% while when implemented into the external dataset HKBU-MARs V1 + database, the accuracy can only reach 86.75% on the test set.

As the use of face recognition technology becomes more widespread in various fields, such as law enforcement, airport security, healthcare, education, marketing, and advertising, counterfeiting or fraud of these systems can occur directly or indirectly, so it is very important to develop detection methods [9]. For this reason, this research introduces detection methods with machine learning and deep learning algorithms.

III. Methodology

In designing the modeling for detecting video face spoofing, several stages are carried out as shown in Fig. 1. The research workflow includes steps starting from searching for datasets to testing and evaluating system performance.

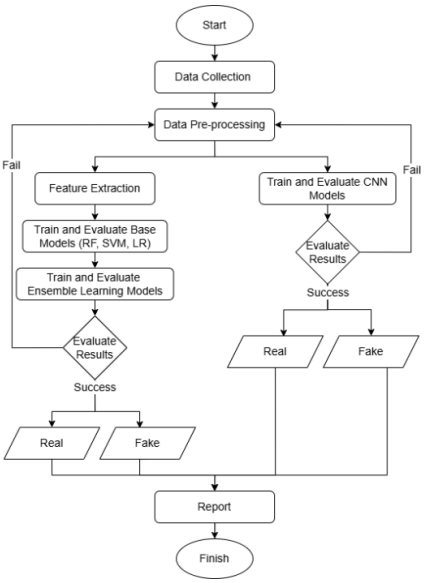

Fig. 1. Workflow Detection System

A. Data Collection and Data Preprocessing

Data Collection

This research utilizes a public dataset from Kaggle called iBeta Level 1 - Liveness Detection Dataset [10], which is specifically designed for the evaluation of liveness detection algorithms. The dataset consists of 567 fakes and 104 reals with the fakes covering 7 different types of spoofing attacks, covering a wide range of scenarios and challenges to ensure a thorough assessment of the algorithm. The attack types in this dataset include:

- Outline: A printed photo of an outline of a person's face.
- Outline3D: A printed photo attached to a cylinder to simulate a 3D face.
- Mask: A printed portrait with hollow eyes for spoofing attacks.
- Mask3D: A 3D mask made of cardboard and connected to resemble a face.
- Phone: Video of a person's face displayed on a smartphone screen.
- Monitor: A video of a person's face displayed on a computer screen.
- Real: A real human face recording as an authentic reference.

This dataset serves as a benchmark in the development and evaluation of the liveness detection system, ensuring the developed algorithm is able to cope with various types of attacks.

Data Preprocessing

The data pre-processing phase is performed to ensure the data is ready to be used in model training and to improve Spoofing detection performance. The data pre-processing steps include:

- Frame Extraction: The video is extracted into frames with an amount of 1 frame per video. Each type of attack is extracted and the results are combined in 1 folder with the name real for real types and fake for Outline, Outline3D, Mask, Mask3D, Phone, and Monitor types.
- Data Augmentation: The main augmentation process in this study utilized the Roboflow tool. Augmentation techniques such as crop (Zoom 0% - 20%) and brightness (±15%) are applied to balance the data which can be seen from Table 1 and improve the generalization of the model to various real-world conditions. In addition, additional augmentation was also applied to the training data at runtime with a combination of flip (horizontal), brightness (±20%), rotation (range [-0.2,0.2] radians), and contrast (0.8 - 1.2). These additional augmentation do not increase the number of images with the aim of memory efficiency, enriching the variety of data during training, and preventing overfitting/underfitting.

Table 1. Total Data

Type	Original	Augmentation	Total
Real	**104**	**412**	**516**
Fake	**567**	**0**	**567**

- Normalization: Pixels are normalized to the value range [0, 1] to speed up model convergence during training.
- Labeling: Frames are labeled as real/1 or fake/0 according to the categories in the dataset as shown in Fig. 2.

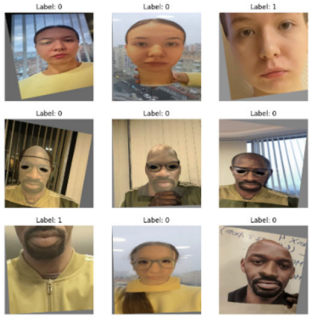

Fig. 2. Sample Data

- Data Split: Based on [11], the data will be divided into training (60%), validation (20%), and test (20%) sets to ensure the trained model has good generalization.

This process ensures the data has sufficient quality and diversity to support the training of a robust model to detect different types of Spoofing attacks.

B. Feature Extraction

Features are extracted based on texture, shape, and color attributes using OpenCV. Texture extraction is performed using Local Binary Pattern (LBP) encoding, followed by histogram calculation [12]. For shape, the Canny edge

detection algorithm is used as a pre-processing step to extract edge features from grayscale images with low noise level and high accuracy, which is effective in feature segmentation for advanced analysis such as face recognition [13]. Meanwhile, color attributes are extracted using HSV color space, resulting in a histogram of Hue, Saturation, and Value distributions [14]. These feature combinations are evaluated with base models to measure relevance, evaluate feature extraction capabilities, and find the combination of models and features that provide the best accuracy of model performance.

C. Train and Evaluate Base Model

The baseline model in this project consists of SVM, Random Forest, and Logistic Regression, which are used for classification of video frames based on the extracted features. SVM separates data into classes using a hyperplane to detect patterns in the video. Random Forest utilizes a collection of decision trees for classification, while Logistic Regression serves as a simple baseline that predicts classes based on the linear relationship of features to probabilities. All three models were imported from the Python scikit-learn library and underwent hyperparameter tuning using RandomizedSearchCV to find the best parameters for training.

From the results of the feature extraction that has been done, the basic model is built with parameters as in Table 2.

TABLE 2. BASE MODEL PARAMETERS

Model	Parameter	Value	Best Param
Support Vector Machine (SVM)	Kernel	['linear', 'rbf', 'poly']	rbf
	C (Regularization Strength)	uniform(0.1, 1.0, 10.0)	np.float64 (7.896910002727692)
	Gamma	['scale', 'auto']	Scale
	degree	randint(2, 5)	2
	Probability	True	True
	Random_state	42	42
Logistic Regression	Penalty	['l1', 'l2']	l1
	Solver	['liblinear', 'saga']	liblinear
	C (Regularization Strength)	uniform(0.1, 1.0, 10.0)	np.float64 (1.6601864044243653)
	Maximum Iterations	1000	1000
	Random_state	42	42
Random Forest	Number of Trees (n_estimators)	randint(50, 100, 300)	253
	max_depth	randint(10, 50)	39
	min_samples_split	randint(2, 20)	2
	min_samples_leaf	randint(1, 10)	2
	max_features	['sqrt', 'log2']	Log2
	Bootstrap	[True, False]	False
	Random_state	42	42
	Class_weight	balanced	balance

D. Train and Evaluate the Ensemble Learning Model

The base model is trained individually and then optimized using a RandomizedSearchCV approach to find the best parameters. In this case, the SVM, random forest, and logistic regression base models are trained first. After the basic model is trained, classifiers are obtained with the best prediction results and put together using ensemble learning by classifying the three models using voting. The voting used is hard voting, which is the way the most votes are then the final prediction results of several models produced. If most models predict a class, then the class becomes the prediction result. By combining the results of multiple algorithms through this technique, the voting classifier can improve the accuracy and reliability of pattern detection on videos, as each model contributes to strengthening the final decision. The parameter used to build the ensemble learning model is the estimator with the value ('Random Forest', best_rf_model), ('Logistic Regression', best_lr_model), dan ('SVM', best_svm_model).

E. Train and Evaluate CNN Model

CNNs are used particularly for image recognition tasks such as Spoofing identification in videos as they could learn relevant and complex visual features from image or video data by performing feature extraction from each video frame using convolution layers and pooling layers. In this research, a CNN model is built using Xception as a base that has been trained on ImageNet. This base model extracts low to medium features from the input image, such as edges, patterns, or textures. The model accepts images of 224x24 pixels with 3 channels (RGB).

After the output of the base model, some additional convolution layers are added to deepen the feature extraction, the layers include:

1. Convolution Block 1

   The first convolution block uses 32 filters 3x3 with ReLU to capture simple features. Followed by 2x2 pooling to reduce the size while retaining information. Batch normalization is used to normalize the output of the convolution layer. Dropout of 30% was randomly disabled to reduce overfitting.
2. Convolution Block 2

   The second convolution layer uses 64 filters, followed by pooling, batch normalization, and dropout with the same parameters as the first block.
3. Convolution Block 3

   The last convolution layer uses 128 filters to extract high-level features. In this part, the dropout is increased to 0.5 (50%) to prevent overfitting in more complex layers.
4. Flatten

   After feature extraction, the output of the last convolutional layer is converted into a one-dimensional vector using flatten.
5. Dense (256 neurons)

   A fully connected layer with 256 neurons is used to process the extracted features. ReLU activation function is used in this layer to add non-linearity.
6. Dropout (0.5)

   Randomly disables 50% of the neurons to prevent overfitting.
7. Output Layer (Softmax)

   The output layer has the number of neurons corresponding to the number of classes (num_classes) with a softmax activation function to generate probabilities for each class.

Table 3 shows the parameters of the layers that can minimize the overfitting/underfitting state of the CNN model training model.

TABLE 3. CNN MODEL PARAMETERS

Layer	Parameter	Value
Input dan Base Model	include_top	False
	weights	'imagenet'
	input_shape	(224, 224, 3)
Convolutional 1	filters	32
	kernel_size	(3,3)
	strides	1
	padding	'same'
	activation	relu
Pooling 1	pool_size	(2,2)
Dropout 1	dropout_rate	0,3
Convolutional 2	filters	64
	kernel_size	(3,3)
	strides	1
	padding	'same'
	activation	relu
Pooling 2	pool_size	(2,2)
Dropout 2	dropout_rate	0,3
Convolutional 3	filters	128
	kernel_size	(3,3)
	strides	1
	padding	'same'
	activation	relu
Pooling 3	pool_size	(2,2)
Dropout 3	dropout_rate	0,5
Fully Connected	dense_units	256
	activation	relu
Dropout 4	dropout_rate	0,5
Output Layer	dense_units	num_classes
	activation	Softmax
Optimization	optimizer	Adam
	loss_function	Categorical Crossentropy
	metrics	Accuracy

F. Evaluation

The CNN, base model, and ensemble systems were tested using separate testing and validation data to ensure the accuracy and reliability of the predictions in distinguishing fake from real videos. Evaluation metrics such as accuracy, precision, recall, and f1-score are used to evaluate the performance of the system [15].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision+\ Recall} \quad (4)$$

IV. RESULT AND DISCUSSION

A. Result

In the evaluation stage, system testing is carried out on test and validation data to see the performance results of the model built. Test data is a new dataset that is not used during training or validation. Validation data is used to measure the model's performance on new data, ensuring the model does not only work on training data.

The following is the feature description:
T : Texture
C : Color
S : Shape
T-C: Texture-Color
S-C: Shape-Color
T-S: Texture-Shape
T-C-S: Texture-Color-Shape.

Ensemble Learning

Based on the ensemble learning evaluation in Tables 4 and Tables 5, the combination of texture and color features shows the best performance. This is reflected in accuracy, precision, recall and F1-score metrics defined in formulas (1)-(4), with values reaching 97% on the test data and 98% on the validation data with T-C feature. These results confirm the superiority of the approach in detecting spoofing accurately and consistently.

TABLE 4. ENSEMBLE LEARNING TEST EVALUATION

Aspects	Feature	Performance			
		Accuracy	Precission	Recall	F1-score
Voting Classifier	**T**	86%	86%	86%	86%
	C	95%	95%	95%	95%
	S	94%	94%	94%	94%
	T-C	**97%**	**97%**	**97%**	**97%**
	S-C	94%	94%	94%	94%
	T-S	93%	93%	93%	93%
	T-C-S	94%	94%	94%	94%

TABLE 5. ENSEMBLE LEARNING VALIDATION EVALUATION

Aspects	Feature	Performance			
		Accuracy	Precission	Recall	*F1-score*
Voting Classifier	**T**	90%	90%	90%	90%
	C	94%	94%	94%	94%
	S	90%	90%	90%	90%
	T-C	**98%**	**98%**	**98%**	**98%**
	S-C	91%	91%	91%	91%

	T-S	90%	90%	90%	90%
	T-C-S	95%	95%	95%	95%

Convolutional Neural Networks (CNN)

From the evaluation of the CNN data in Tables 6 the results of the evaluation metrics are the same between the test and validation data, which reached a perfect result of 100%. This indicates that the model learned very well on the training data and can implement it very well on completely new data. To avoid any indication of overfitting/underfitting, Fig. 3 and Fig. 4 show the training process. From these figure, the validation accuracy tends to be greater than training and the validation loss is smaller than training. This indicates that the model is indeed very suitable to handle the dataset in this study and there is no overfitting/underfitting.

TABLE 6. CNN DATA EVALUATION

Aspect	Performance			
	Accuracy	Precission	Recall	F1-score
CNN (Test)	**100%**	**100%**	**100%**	**100%**
CNN (Validation)	**100%**	**100%**	**100%**	**100%**

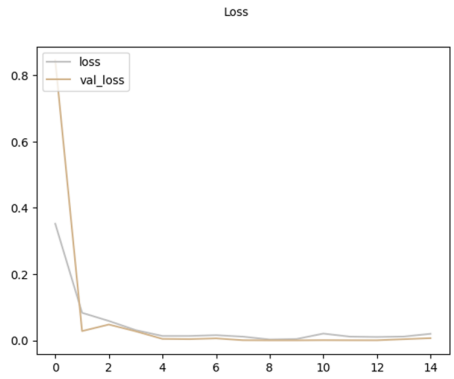

Fig. 3. Training and Validation Loss Curve for CNN Mode

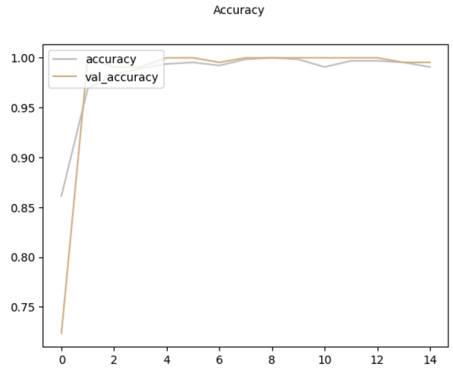

Fig. 4. Training and Validation Accuracy Curve for CNN Model

Baseline Model

From the evaluation of the base model data in Table 7, the results of the evaluation metrics achieve the best results on the S-C feature in SVM. Random Forest achieves the best evaluation results in T-C features with 96% in all evaluation metrics. In Logistic Regression, the best results are found in the combination of S, S-C, T-S, and T-C-S features with results reaching 94% in each evaluation metric.

While evaluating the basic model data in Table 8, the results of the evaluation metrics above achieved the best results in the C and texture-color-shape T-C-S features in SVM which reached 93% in each evaluation metric. In Random Forest, the performance results of the evaluation metrics achieved the best results in the C and T-C features with 95% results in each evaluation metric. Meanwhile, in Logistic Regression the best results are in the combination of T-C-S features with results reaching 94% in each evaluation metric.

TABLE 7. BASELINE MODEL TEST EVALUATION

Aspects	Feature	Performance			
		Accuracy	Presisi	Recall	F1-Score
SVM	**T**	86%	88%	86%	85%
	C	92%	92%	92%	92%
	S	94%	94%	94%	94%
	T-C	91%	92%	91%	91%
	S-C	**96%**	**96%**	**96%**	**96%**
	T-S	93%	93%	93%	93%
	T-C-S	94%	95%	94%	94%
Random Forest	**T**	80%	81%	80%	79%
	C	98%	98%	98%	98%
	S	88%	88%	88%	88%
	T-C	**96%**	**96%**	**96%**	**96%**
	S-C	94%	94%	94%	94%
	T-S	76%	79%	76%	74%
	T-C-S	92%	92%	92%	92%
Logistic Regression	**T**	82%	86%	82%	82%
	C	89%	90%	89%	89%
	S	**94%**	**94%**	**94%**	**94%**
	T-C	87%	88%	87%	86%
	S-C	**94%**	**94%**	**94%**	**94%**
	T-S	**94%**	**94%**	**94%**	**94%**
	T-C-S	**94%**	**94%**	**94%**	**94%**

TABLE 8. BASELINE MODEL VALIDATION EVALUATION

Aspects	Feature	Performance			
		Accuracy	Presisi	Recall	Accuracy
SVM	**T**	88%	88%	88%	88%
	C	**93%**	**93%**	**93%**	**93%**
	S	89%	89%	89%	89%
	T-C	93%	94%	93%	93%
	S-C	92%	92%	92%	92%
	T-S	92%	92%	92%	92%
	T-C-S	**93%**	**93%**	**93%**	**93%**
Random Forest	**T**	83%	84%	83%	83%
	C	**95%**	**95%**	**95%**	**95%**
	S	85%	85%	85%	85%
	T-C	**95%**	**95%**	**95%**	**95%**
	S-C	89%	90%	89%	89%
	T-S	76%	80%	76%	75%

	T-C-S	92%	92%	92%	92%
Logistic Regression	**T**	85%	85%	85%	85%
	C	91%	92%	91%	91%
	S	91%	91%	91%	91%
	T-C	89%	90%	89%	89%
	S-C	92%	92%	92%	92%
	T-S	92%	92%	92%	92%
	T-C-S	**94%**	**94%**	**94%**	**94%**

B. Discussion

Based on the analysis of results of each model in the Tables 4 – Tables 8, it can be seen that CNN shows outstanding performance, reaching 100% in accuracy, precision, recall, and f1-score on both test and validation datasets. This indicates that the CNN in the model in this study excels in feature extraction and has a remarkable capacity for generalization to new data. With no signs of overfitting or underfitting, the model is believed to be adept at learning features and generalizing to unprecedented datasets. CNN is the optimal model for this study due to its outstanding ability to identify patterns in images. In addition, CNN is clearly capable of outperforming several machine learning models, specifically SVM, Random Forest, and Logistic Regression. Despite the good results on testing machine learning models, the findings from CNN were superior.

The second method used, Ensemble Learning with Voting Classifier, significantly improved the model, especially the T-C feature combination, achieving performance results of 97% on test data and 98% on validation data in accuracy, precision, recall, and f1-score. Although there is a slight difference in the test and validation results of the ensemble model, the performance of this model is quite stable with high accuracy, so it can be said that this model also works well in this study even though its performance cannot equal the CNN results.

CONCLUSSION

Based on the evaluation of model performance on validation and test data, the results obtained show excellent performance to address the limitations of previous research, with an accuracy range of 97% for test and 98% for validation for the ensemble learning voting classifier model and 100% for the CNN model in both test and validation. The small difference between the validation and test results indicates that the model can generalize the new data very well, without showing signs of overfitting/underfitting. The ensemble learning performance itself has not been able to outperform CNN in this study. But in general, the CNN and ensemble learning models are quite stable and able to provide accurate predictions to detect the spoofing videos. For future research, it is recommended to explore training with the latest models that are more accurate in making predictions, and consider using more alternative feature extraction methods .This can help improve the performance of the model in the face of more diverse data variations.

REFERENCES

[1] M. Wang and W. Deng, "Deep Face Recognition: A Survey," *Neurocomputing,* vol. 429, no. 14, pp. 215-244, 2020.

[2] R. B. Hadiprakoso, H. Setiawan and Girinoto, "Face Anti-Spoofing Using CNN Classifier & Face liveness Detection," in *2020 3rd International Conference on Information and Communications Technology* , Yogyakarta, 2020.

[3] A. A. Mohamed, M. M. Nagah, M. G. Abdelmonem, M. Y. Ahmed, M. El-Sahhar and F. H. Ismail, "Face Liveness Detection Using a sequential CNN," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, Nevada, 2021.

[4] S. Ganorkar, S. Rajankar and G. Rajpurohit, "Face Liveness Detection Using Machine Learning," *International Journal Of Scientific & Technology Research,* vol. 8, no. 09, pp. 337-340, 2019.

[5] A. O'Brien, "Spoof: A barrier to the acceptance of Facial Recognition Systems," Innovative Technology Ltd, 11 November 2020. [Online]. Available: https://www.intelligent-identification.com/spoof-a-barrier-to-the-acceptance-of-facial-recognition-systems. [Accessed 26 November 2024].

[6] M. A. Malik, T. Mazhar, I. Haq, T. Shahzad, Y. Y. Ghadi, F. Maleek and H. Hamam, "A Novel Deep Learning-Based Method for Real-Time Face Spoof Detection," *Research Square,* 29 September 2023.

[7] M. Marais, J. Connan, D. L. Brown and A. Z. Boby, "Facial Liveness and Anti-Spoofing Detection using Vision Transformers," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2023*, Champagne Sports Resort, 2023.

[8] S. Mondal, "Implementation of Human Face and Spoofing Detection Using Deep Learning on Embedded Hardware," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020.

[9] Kapersky Company, "What is Spoofing – Definition and Explanation," Kapersky, 19 February 2020. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/spoofing. [Accessed 26 November 2024].

[10] trainingdata.pro, "iBeta 1 - 42,280 Liveness Detection Dataset," Training Data, 2023. [Online]. Available: https://www.kaggle.com/datasets/trainingdatapro/ibeta-level-1-liveness-detection-dataset-part-1. [Accessed October 2024].

[11] B. Purnama, Pengantar Machine Learning Konsep dan Praktikum dengan Contoh Latihan Berbasis R dan Python, Bandung, Jawa Barat: Informatika Bandung, 2019.

[12] S. Khairnar, S. Gite, K. Kotecha and S. D. Thepade, "Face Liveness Detection Using Artificial IntelligenceTechniques: A Systematic Literature Review and Future Directions," *Big Data and Cognitive Computing,* vol. 7, no. 1, pp. 1-35, 2023.

[13] L. P. R. Noviana, I. P. E. Indrawan and G. I. Setiawan, "Analysis of Canny Edge Detection Method for Facial Recognition in Digital Image Processing," *Jurnal Manajemen dan Teknologi Informasi (JMTI),* vol. 15, no. 2, pp. 15-22, 2024.

[14] A. Octaviani, D. S. Prasvita, K. R. T. Zulkarnain and S. Hinggit, "Klasifikasi Tingkat Kematangan pada Buah Rambutan Berdasarkan Fitur Warna Menggunakan KNN dan Ekstraksi Warna HSV," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, Jakarta, 2021.

[15] A. Yilmaz, A. A. Demircali, S. Kocaman and H. Uvet, *Comparison of Deep Learning and Traditional Machine Learning Techniques for Classification of Pap Smear Images,* Istanbul: arXiv, 2020.