ICIC Express Letters
Volume **17**, Number **2**, February **2023**
 ISSN 1881-803X
pp. **235–244**

FACE SPOOFING DETECTION BASED ON DEEP FEATURE EXTRACTION AND INSTANCE-BASED CLASSIFICATION

Niphat Claypo[1,*], Saichon Jaiyen[2] and Anantaporn Hanskunatai[1,*]

[1]Data Science and Computational Intelligence Lab
Department of Computer Science
School of Science
King Mongkut's Institute of Technology Ladkrabang
No. 1, Chalong Krung 1, Chalong Krung Road, Lat Krabang Sub-District, Lat Krabang District
Bangkok 10520, Thailand
[*]Corresponding authors: {58605020; anantaporn.ha}@kmitl.ac.th

[2]School of Information Technology
King Mongkut's University of Technology Thonburi
No. 126, Pracha Uthit Road, Bang Mot Sub-District, Thung Khru District
Bangkok 10140, Thailand
saichon.jai@kmutt.ac.th

Received May 2022; accepted July 2022

Abstract. *Face recognition is an important task in smart home security for detecting a face or monitoring a person in a live video and verifying the identity of an authentic user. However, there have been spoofing face methods that can trick a face recognition algorithm into wrongly verifying the identity of the person. In this paper, we propose a new hybrid framework for spoofing face detection based on Convolutional Neural Network and Long Short-Term Memory (CNNLSTM) and instance-based learning algorithm. In addition, a new dataset called FSA-CCTV is proposed, which contains face images from CCTV video clips with many types of spoofing attacks. The performance of our method was compared to several other anti-spoofing methods: CNN and RI-LBP, SLRNN, HSV+YCbCr, ResNet50, YCbCr+SVM and YCbCr+KNN. The experimental results show that our method yielded 93.2% of Accuracy, 96.8% of Recall, 94% of Precision, 94.8% of F_1-score and 0.93 of AUC on the FSA-CCTV dataset. From the experimental results we can conclude that the proposed algorithm outperforms other approaches and yielded the most stable classification accuracy on the proposed dataset.*
Keywords: Convolutional neural network, Face spoofing attack detection, Feature extraction, Smart security

1. **Introduction.** Smart home technology provides home automation systems for homeowners and facilities that provide security, comfort, convenience, and energy efficiency [1]. In terms of security, residential smart security cameras can monitor, in real time, the happenings around the house. Face recognition is a feature of smart security [2]. It is a necessary method for identifying and verifying faces as well as authenticating users. Face recognition algorithms can automatically detect a face from a real-time video stream and search a face database to find a match [3]. A face recognition system must be able to detect spoofs. A facial liveness detection algorithm, which can decide whether a recognized face is real or spoofed, is an important subroutine in a face recognition system. Its main goals are to stop fraud and ensure an authentic face before the recognized face is matched to a face in the database for identification.

Types of face spoofing attacks are such as printed photo, mask, 3D mask, smartphone, tablet, and ID card. The widespread of Internet and smartphones makes it easy to find

DOI: 10.24507/icicel.17.02.235

images for spoofing faces. A high-performance anti-spoofing method can stop an attack outright and inform the user of the detected anomaly.

Spoofing detection methods have been proposed to solve face spoofing problems by identifying face liveness. The feature extraction method is the main type of data preprocessing for face spoofing detection methods. Texture-based methods are used to analyze color-texture information in face images by extracting low-level feature descriptions from different color spaces. Examples of these methods are Local Binary Patterns (LBP) [4-6], local similar patterns (LSP) [7], and YCbCr color [8]. These methods are traditional face spoofing detection approaches. Recently, Deep Learning (DL) is one of widely used methods for extracting features in an image. Convolutional Neural Network (CNN), a type of DL, is especially good at extracting features in images. Several studies have applied CNNs to detecting face spoofing. Similarly, many researchers have used CNNs to extract features that suit facial liveness detection. The review of these methods will show in Related Works section.

This paper presents a method for eliminating face spoofing problem. The main contributions from our paper are as follows.

- We create a new dataset called FSA-CCTV. This dataset is presented of diverse attacks type for developing the home security system. The FSA-CCTV is a collected image from video frames in long-distance shooting video clips of CCTV cameras. It consists of several different lighting conditions.
- We propose a new hybrid algorithm for classifying spoofed faces. A new structure of the CNN and LSTM is proposed and called CNNLSTM. The proposed method creates a set of CNNLSTM models and uses the models to extract features from images. Finally, instance-based algorithm is used to classify spoofed faces.

The rest of this paper is organized as follows: Section 2 reviews anti-spoofing face detection articles in the literature; Section 3 suggests the proposed dataset; Section 4 describes the proposed method; Section 5 explains the evaluation methods, the experimental setup, and the experimental outcomes; and finally, Section 6 suggests directions for future works and concludes the paper.

2. **Related Works.**

In this section, we review existing face spoofing detection approaches mainly focused on features extraction with the texture-based and CNN methods. The related methods are divided into five groups as follows.

2.1. **Texture based methods.**

In [9], Yang et al. proposed a person-specific face anti-spoofing method based on HOG and MsLBP feature extraction methods. Li et al. [10] proposed a method that converted images into YCbCr color space and then extracted features by LBP method. In the same vein, Fourati et al. [11] used Image Quality Assessment (IQA) and motion cues of face image to identify face spoofing.

2.2. **CNN based methods.**

CNN is a specialization method for detected patterns in image. Chen et al. [12] proposed a Two-Stream Convolutional Neural Network (TSCNN). The TSCNN uses a Multi-Scale Retinex (MSR) space to solve illumination problem. Muhammad and Melo [13] proposed an SLRNN for spoofed face classification. The SLRNN method combined CNN and adding Long Short-Term Memory (LSTM) together. Rehman et al. [14] presented a face liveness detection method that incorporated a disparity layer in the CNN to learn dynamic disparity maps. Wirianto and Mauritsius [15] presented an Indonesia Labelled Face in the Wild (ILFW) dataset. The researchers suggested a DCNN network architecture called ResNet100 for face recognition.

2.3. **Texture-based and CNN based methods.** The texture-based feature and CNN based features were proposed by Chen et al. [16]. The research introduced Face Anti-spoofing Region-based Convolutional Neural Network (FARCNN), an instance of FAR-CNN that is based on improved faster Region-based Convolutional Neural Network (R-CNN) framework for face extraction and extraction features. They extended the faster R-CNN method and Retinex-based LBP to cover face anti-spoofing tasks with diverse illumination conditions. It classifies spoofing faces by using an SVM classifier.

2.4. **CNN based and machine learning methods.** Shao et al. [17] proposed a feature learning model for 3D-mask face anti-spoofing. A VGG was used to extract features and deep dynamic textures. Li et al. [18] presented a technique for extracting features by CNN, and detected spoofed faces by SVM. George and Marcel [19] presented a Multi-Channel Convolutional Neural Network (MCCNN). The new loss function was presented in that study.

2.5. **Texture based and CNN based and machine learning methods.** Khammari [20] presented a new method that extracted features by using LBP and WLD. The output features were encoded by CNN. The output from CNN was inputted to SVM to identify live or spoofed face. Chen et al. [21] extended the face algorithm from Li et al. [10]. That extension was based on RI-LBP and CNN and used SVM to identify face spoofing.

Texture-based methods extract features from low-level feature descriptions from different color spaces, and these lose some features, which makes the accuracy disappear. CNN-based methods can extract deep challenging features and extract from multi-view of an image by convolution layers, providing high performance for spoofing face detection.

3. **The FSA-CCTV Dataset.** The new dataset called FSA-CCTV is proposed. We fetched video streams from the IP cameras, following the Real Time Streaming Protocol (RTSP). Faces are detected by using a Haar Cascades technique [22]. Haar Cascades method is fast to detect and is an effective detection method. Then, we resized images to 224×224, and used data augmentation to increase the number of images. The types of face spoofing attacks of this dataset are fake images and images of a person wearing a mask from smartphone, iPad, office card, and ID card under multiple lighting conditions. Sample images from our proposed dataset are shown in Figure 1. The first row in the figure shows spoofed faces. The first two images in the second row are images of spoofed faces taken in IR mode, and the last two pictures are genuine faces. The dataset consists of 2,045 spoofed face images and 1,015 genuine face images. This dataset uses the data augmentation technique for increasing number of data, reduces overfitting and decreases variance of model. The spoofed faces are expanding to 3,724 images and the genuine faces increase to 2,396 images.

FIGURE 1. Sample images from the FSA-CCTV dataset

4. **Face Spoofing Detection Method.** This section explains the proposed anti-spoofing approach.

4.1. **Network architecture of CNNLSTM.** CNNLSTM is used for feature extraction in this study. The outputs of CNNSLTM are the new instances for instance-based learning to recognize spoofing faces. The architecture of CNNLSTM for extracting features from images is inspired by AlexNet [23]. We improve the neural network model of AlexNet by adding LSTM algorithm to its convolutional layer. The LSTM is designed to avoid a long-term dependency problem [13]. It consists of several memory cells. Each memory cell has three elements: write, read, and forget (delete). The convolutional layers in the CNNLSTM are extended from those of AlexNet as explained as follows. In the first and second convolutional layers, we set the number of convolutional kernels to 256 kernels, and in the fifth convolutional layer was set to 512×5 kernels. In the final representation layer, LSTM, a Recurrent Neural Network (RNN) was used to receive inputs from prior convolutional layers. A Batch-Normalization (BN) method was applied to the outputs of convolutional layers before they were inputted into the LSTM. The output of BN is tuple of integer fed into the LSTM layer. A set of 512 outputs from LSTM output units was fed into the next layer. The fully connected layers contain 1,000 hidden neurons. The activation function of all the convolutional layers is a Rectified Linear Unit (ReLU). The output layer contains one output neuron with a sigmoid activation function. The CNNLSTM model in this study was trained with a Stochastic Gradient Descent (SGD) method.

4.2. **The proposed feature learning network.** The overall process of creating feature learning networks of the proposed algorithm is illustrated in Algorithm 1.

Algorithm 1: Pseudo code of the feature learning network method.

Input: $\mathbf{S}$: A training set, K: The number of the subdatasets, θ: A performance threshold, m_{k-1}: The pre-trained weights

Output: E: A set of CNNLSTM models, C: A set of representation datasets

1: Randomly split the training dataset into K sub datasets: $\mathbf{S} \rightarrow \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots, \mathbf{s}_K\}$.
2: **For** $k = 1, 2, \ldots, K$ **do:**
3: $AUC = 0$
4: **While** $AUC < 0$ **do:**
5: Train CNNLSTM from m_{k-1} with $\mathbf{s}_k$ to obtain the m_k model.
6: Extract deep features $\mathbf{F}_k$ from $\mathbf{s}_k$ by m_k.
7: Normalize every feature vectors $\mathbf{f}_{ik}$ in $\mathbf{F}_k$ with an $L2$-norm technique.
8: **For** i **in** $y \in \{0, 1\}$ **do:**
9: Compute the centroid of a class $\mathbf{c}_{ik}$ from $\mathbf{F}_{k \in Nik}$.
10: **End For**
11: Calculate the Euclidean distances from centroid $\mathbf{c}_{ik}$ to every feature of all $\mathbf{F}_k$ instances.
12: Predicted label h_j of $\mathbf{s}_k$ by using $\mathbf{c}_{ik}$ and $\mathbf{F}_k$.
13: Compute AUC value from predicted labels h_j with true label y.
14: **End While**
15: $m_k \in E$ and $\mathbf{c}_{ik} \in C$.
16: **End For**

To construct CNNLSTM models, our method randomly selects instances of data in the $\mathbf{S}$ and put them into several subdatasets. The subdataset $\mathbf{s}_k$ consists of N input data vectors $\mathbf{x}$ and label y: $\mathbf{s}_k = \{(\mathbf{x}_{1k}, y_{1k}), \ldots, (\mathbf{x}_{jk}, y_{jk}), \ldots, (\mathbf{x}_{Nk}, y_{Nk})\}$ (Line1). These subdatasets aim to create diversity CNNLSTM models and representation of class data.

The subdataset $\mathbf{s}_k$ is the training set for a CNNLSTM, m_k (Line5). The current model gets the weights from previous model m_{k-1}.

A set of extracted feature vectors $\mathbf{F}_k = \{\mathbf{f}_{1k}, \mathbf{f}_{2k}, \ldots, \mathbf{f}_{Nk}\}$ is extracted from model m_k (Line6). The features $\mathbf{f}_{jk}$ are extracted from a data $\mathbf{x}_{jk}$: $\mathbf{f}_{jk} = m_k(\mathbf{x}_{jk})$.

All feature vectors in $\mathbf{F}_k$ are normalized by using an $L2$-norm technique (Line7), and let $\mathbf{f}_{jk} = L2(\mathbf{f}_{jk})$. The norm of $\mathbf{f}_{jk}$ is calculated as follows.

$$|\mathbf{f}_{jk}| = \sqrt{\sum_{l=1}^{n^k} |v_{lk}|^2}, \tag{1}$$

where v_{lk} is an extracted value in $\mathbf{f}_{jk}$, n^k is size of $\mathbf{f}_{jk}$. For the proposed method to be able to recognize patterns of genuine and spoofing faces, a centroid of a class, $\mathbf{c}_{ik}$ is computed in this step. The centroid $\mathbf{c}_{ik}$ is an average of every feature vector in $\mathbf{F}_k$ from instances of class ith. The centroids $\mathbf{c}_{ik}$ is the representation of training data. The centroid $\mathbf{c}_{ik}$ is calculated as follows:

$$\mathbf{c}_{ik} = \frac{1}{Nik} \sum_{\mathbf{f}_{jk} \in Nik} \mathbf{f}_{jk}, \tag{2}$$

where Nik is the number of all data of class i in $\mathbf{s}_k$ (Line9). The similarity distances of every class between $\mathbf{f}_{jk}$ and $\mathbf{c}_{ik}$ were computed in Line11. The distance is a Euclidean distance. The prediction class h_j is selected by the minimum distance of a class (Line12). Compute the similarity value and predicted label h_j is defined as

$$h_j = \arg\min_i \|\mathbf{f}_{jk} - \mathbf{c}_{ik}\|_2, \text{ for } j = 1 : Nk, \tag{3}$$

where Nk is the size of $\mathbf{s}_k$. The performance of m_k measures in every five training epochs, in terms of the Area Under the Curve (AUC). If AUC value is less than a threshold θ, the m_k must be retrained for another 5 epochs (go to Line5), $\mathbf{F}_k$ and $\mathbf{c}_{ik}$ are recomputed, otherwise the training would be stopped and the model m_k is a member of E; $m_k \in E$; and $\mathbf{c}_{ik} \in C$ (Line15). Our training process minimizes the distance between the extracted features and a centroid of a class which have the same class label and maximizes the distance between the centroid of genuine and the centroid of spoof faces.

For the next subdataset $\mathbf{s}_{k+1}$, we use pre-trained weights from the previous iteration k to learn the current data $m_{k+1} \rightarrow m_k$. To reduce training time and improve classification accuracy, it was convenient to use a pre-trained weight from a previous model. The models in E are different from a decision boundary model of the same architecture. The centroids

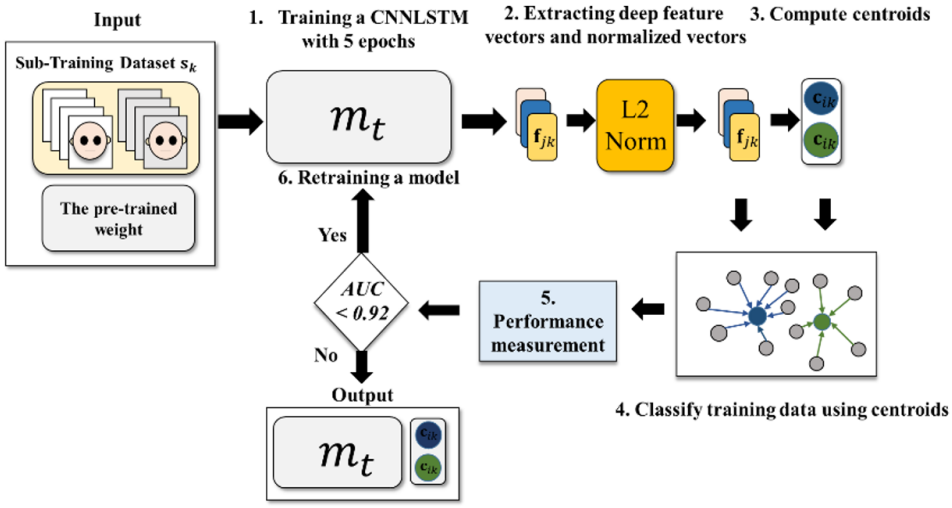

FIGURE 2. Overall process of feature learning networks for face spoofing detection

in C are the representation of the patterns of spoofing attack. The overview of feature learning method is illustrated in Figure 2.

4.3. **Classification of face spoofing.** The proposed instance-based classification method is discussed in this section. This method is a technique for preventing bias in class prediction. The distance weights are computed for every class. The pseudo code of the classification steps is shown in Algorithm 2.

Algorithm 2: Pseudo code of spoof face classification.

Input: E: CNNLSTM models, C: A set of centroid class data, $\mathbf{x}$: An unseen data

Output: $h_{final} = \arg\max_i \sum_i \sum_k \log\left(\frac{1}{ds_{ik}}\right)$

1: **For** $k = 1, 2, \ldots, K$ **do:**
2: Extract features $\mathbf{f}_k = m_k(\mathbf{x})$.
3: Normalize the feature vectors $\mathbf{f}_k$ by $L2$-norm.
4: **For** $i = 1 : y$ **do:**
5: Calculate the distance d_{ik} between centroid $\mathbf{c}_{ik}$ to features $\mathbf{f}_k$.
6: Compute the distance weight ds_{ik}.
7: **End For**
8: **End For**

The sets of models E and centroids C are the inputs of Algorithm 2. The final output is calculated as follows:

$$h_{final} = \arg\max_i \sum_i \sum_k \log\left(\frac{1}{ds_{ik}}\right), \tag{4}$$

where h_{final} is the predicted class, i is class index, k is the number of models and ds_{ik} is the distance weights calculated by Equation (6). According to this equation, a high distance weight ds_{ik} would produce a low voting value, and vice versa. The predicted class h_{final} is selected with a maximum voting value of the individual class. The proposed classification method is illustrated in Figure 3.

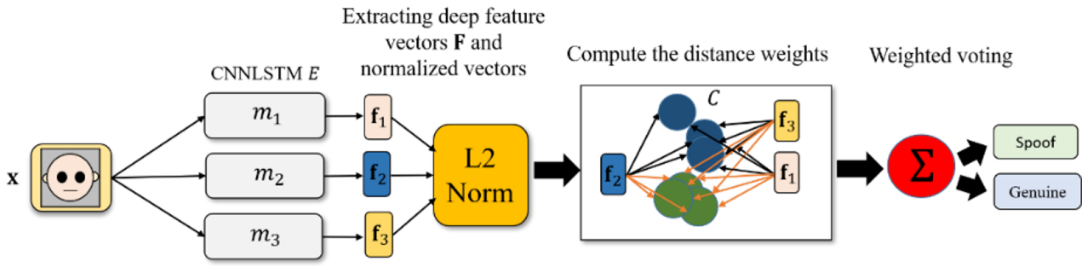

FIGURE 3. The overall process of face spoofing classification

The first step extracts feature from unseen data: $\mathbf{f}_k = m_k(\mathbf{x})$ (Line2). The similarity distance α_{ik} between $\mathbf{f}_k$ and $\mathbf{c}_{ik}$ is calculated by Equation (5). Before α_{ik} is computed, the $\mathbf{f}_k$ is normalized by $L2$-norm technique: $\mathbf{f}_k = L2(\mathbf{f}_k)$ (Line3). The $\mathbf{f}_k$ is normalized into the same vector space of $\mathbf{c}_{ik}$. The distances values are expressed as follows:

$$\alpha_{ik} = \|\mathbf{f}_k - \mathbf{c}_{ik}\|_2, \text{ for } k = 1 : K, \tag{5}$$

where α_{ik} is a distance value of the ith class (Line5). Each feature vector is computed distance of every centroid of a class data in C. The voting weights are computed as follows:

$$ds_{ik} = \frac{\alpha_{ik}}{\sum_i \alpha_{ik}}. \tag{6}$$

The weights are the average distance between the feature vectors and centroids of spoofed and authentic faces (Line6). The main concept of distance measure is inspired by a face

identification method [24], applied to spoofing detection. If an unseen data $\mathbf{x}$ is an authentic face, then the features extracted from CNNLSTM would be similar to the centroids of an authentic face, and the distance weight computed by Equation (6) would be high. Otherwise, the distance weight would be low.

5. **Experiment and Results.** This section describes and discusses the experiment and experimental results on our proposed face-spoofing dataset. The results of each method on our dataset were shown of the performance of spoofed face detection for smart home security.

5.1. **Experimental setup.** The FSA-CCTV dataset has 6,120 images, including 3,724 positive class images, and 2,396 negative class images. Let the genuine faces be negative class and spoofed faces be positive class. The experiment was done with five-fold cross-validation technique. The results were averaged over five testing folds. The input training threshold, θ, was set to 0.92 and K, was set to 3, 5, and 10. To evaluate our method, the results were compared to those achieved by several widely known face anti-spoofing algorithms.

5.2. **Experimental results.** This research conducted the experiment only on the FSA-CCTV dataset because of limitations in the personal data. The performances of the proposed method and several state-of-the-art methods were evaluated. Performance values are reported as an average of five replicates from five runs.

Table 1 shows the classification accuracy on each testing fold, the average accuracy of five folds, and the Standard Deviation (SD) value on the FSA-CCTV dataset. The result shows that the average classification accuracies of the proposed methods with all K sizes are better than those of the other compared methods in identifying spoofed faces. The accuracy results of our methods are still high, with a low variance in several testing folds (shown by SD value behind $\pm$ in the average column). For the accuracy of the first and third testing folds, the accuracies of the proposed methods are higher than those of the other compared methods. The fifth testing fold of our methods with $K = 5$ and $K = 10$ is equal to CNN and RI-LBP, and HSV+YCbCr with 93% of accuracy. Considering the accuracy of the second and the fourth folds, SLRNN method has the highest accuracy in this testing fold. YCbCr+SVM method has the lowest classification performance in spoofed faces detection than those of other methods. On the other hand, the accuracy of CNN and RI-LBP, SLRNN, HSV+YCbCr, and ResNet50 are dropped on some testing folds. These methods have high variance of accuracy, especially SLRNN is very sensitive to a spoofing attack in this dataset. We used a t-test with alpha equal to 0.01 to evaluate the differences between the accuracy of the proposed methods and other methods in this

Table 1. The performance on the FSA-CCTV dataset in terms of *Accuracy*

Methods	**Fold (%)**					**Average (%)**
	1	**2**	**3**	**4**	**5**	
CNN and RI-LBP	93	92	89	89	**93**	$91.2 \pm 2* \circ\star$
SLRNN	91	**96**	84	**95**	84	$90 \pm 5.7* \circ\star$
HSV+YCbCr	91	88	88	89	**93**	$89.8 \pm 2.1* \circ\star$
ResNet50	93	91	87	91	84	$89.2 \pm 3.6* \circ\star$
YCbCr+SVM	85	87	88	88	88	$87.2 \pm 1.3* \circ\star$
CNNLSTM	89	93	89	91	91	$90.6 \pm 1.6* \circ\star$
YCbCr+KNN	92	92	91	91	92	$91.6 \pm 0.5* \circ\star$
The proposed method $K = 3$	93	93	**95**	93	91	93 ± 1.4
The proposed method $K = 5$	93	93	94	93	**93**	**93.2** $\pm$ 0.4
The proposed method $K = 10$	**94**	92	94	93	**93**	**93.2** $\pm$ 0.8

experiment ($*$, $\circ$, $\star$, indicate the proposed method with $K = 3$, $K = 5$, $K = 10$ is significantly better than the corresponding algorithm, respectively). The results show that our method with every K size is significantly better than the other methods by comparing with the average accuracy.

Table 2 shows the average values of *Recall*, *Precision*, and F_1-score of the proposed method and the other compared methods on the FSA-CCTV dataset. The proposed method of all K sizes has higher average *Recall* value than other methods. The proposed method with $K = 3$ has the highest value of *Recall*. While YCbCr+SVM provides the lowest *Recall* with 88%. In addition, our method of all K sizes is significantly better than the other algorithms. The third column contains *Precision* values of spoofed faces. The obtained *Precision* values indicate an acceptable classification accuracy for the model. It can be seen in the column that the proposed method is inferior to SLRNN, and CNN and IR-LBP. HSV+YCbCr is the worst of *Precision* value. The F_1-score of all methods is shown in the fourth column. Our models with $K = 3$ and $K = 5$ are significantly better than all other methods except CNN and RI-LBP in terms of F_1-score. While the proposed method with $K = 10$ is significantly better than YCbCr+SVM and equal to CNN and RI-LBP method. YCbCr+SVM provides the lowest value of F_1-score.

Figure 4 shows the *ROC* curves of all compared algorithms on the FSA-CCTV dataset. The highest point of the three *ROC* curves of the proposed method with different K is at

TABLE 2. The performance on the FSA-CCTV dataset in term of *Recall*, *Precision*, and F_1-score

Methods	**Recall (%)**	**Precision (%)**	**F_1-score (%)**
CNN and RI-LBP	93.8$*$ $\circ\star$	95.2	94.6
SLRNN	90.4$*$ $\circ\star$	**96**	92.8$*\circ$
HSV+YCbCr	92.6$*$ $\circ\star$	91.6$\circ$	92.2$*\circ$
ResNet50	91.2$*$ $\circ\star$	93.6	92.2$*\circ$
YCbCr+SVM	88$*$ $\circ\star$	93	90.4$*$ $\circ\star$
CNNLSTM	92$*$ $\circ\star$	92$\circ$	92$*\circ$
YCbCr+KNN	91.6$*$ $\circ\star$	94	93$*\circ$
Proposed method $K = 3$	**96.8**	92	**94.8**
Proposed method $K = 5$	96	94	**94.8**
Proposed method $K = 10$	95.8	94	94.6

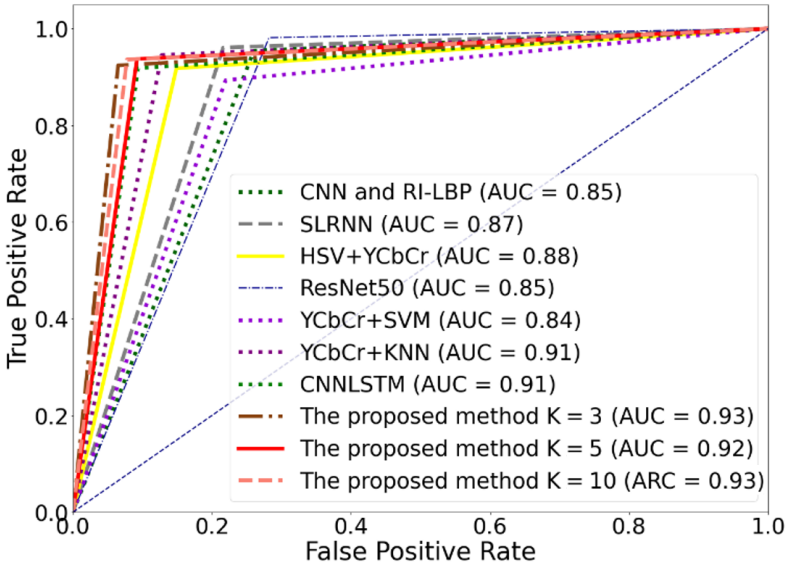

FIGURE 4. ROC curve on the FSA-CCTV dataset

the upper left corner. The proposed method of all K sizes outperforms all other methods – providing the higher value of AUC at 0.93, 0.92, and 0.93, respectively. YCbCr+SVM has the lowest AUC than other methods. The classification results from the proposed method are indicated that our method is efficient in spoofed face detection and provides high accuracy in identifying authentic faces. While the other comparison methods are high variance and have low ROC value meaning that those approaches are unbalanced in terms of classification performance.

The experimental results indicate that the CNN-based methods have higher precision than the color texture-based methods. The merging of the features extracted from CNN with the instance-based classification method of our algorithm can enhance the performance of the detection. However, when classifying with high lighting cases, the proposed method and those of other methods misclassified these images because these images have few gradients and depth shapes. Conversely, these have high performance on IR images.

6. **Conclusions.** This paper proposes the FSA-CCTV dataset and a new hybrid approach for face spoofing attack problem based on deep neural network and instance-based algorithms. The classification results of the proposed method on the FSA-CCTV dataset are compared with other widely known anti-spoofing methods. In the proposed method, a new architecture of deep neural network, called CNNLSTM, is used to extract features from facial images. CNNLSTM is trained on the proposed face-spoofing dataset. A new distance weight voting procedure is used to predict the final class. The performance of the proposed method is evaluated with 5-fold cross-validation technique. Experimental results show that the proposed method outperforms in classification performance than the other anti-spoofing classification methods. The proposed method is suitable for smart security on CCTV.

For future work, we plan to apply the proposed algorithm to different types of anti-spoofing tasks. The number of models generated by CNNLSTM can be varied, and the structure of CNNLSTM can be designed suitable for a particular dataset. The number of extracted features can be reduced, and other normalization methods can be applied to relieving span resources. This method can also be extended to an incremental learning that learns incremental data.

REFERENCES

[1] M. R. Alam, M. B. I. Reaz and M. A. M. Ali, A review of smart homes-past, present, and future, *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol.42, no.6, pp.1190-1203, 2012.

[2] M. R. Chandra, B. V. Kumar and B. S. Babu, IoT enabled home with smart security, *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, 2017.

[3] B. Ríos-Sánchez, D. Costa-da-Silva, N. Martín-Yuste and C. Sánchez-Avila, Deep learning for facial recognition on single sample per person scenarios with varied capturing conditions, *Applied Sciences*, vol.9, no.24, 5474, 2019.

[4] D. He and L. Wang, Texture unit, texture spectrum, and texture analysis, *IEEE Trans. Geoscience and Remote Sensing*, vol.28, pp.509-512, 1990.

[5] G. Zhao and M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.29, no.6, pp.915-928, 2007.

[6] M. Heikkilä and M. Pietikäinen, A texture-based method for modeling the background and detecting moving objects, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.28, pp.657-662, 2006.

[7] H. R. Pourreza, M. Masoudifar and M. ManafZade, LSP: Local similarity pattern, a new approach for rotation invariant noisy texture analysis, *2011 18th IEEE International Conference on Image Processing*, Belgium, 2011.

[8] C. Poynton, *Digital Video and HDTV*, Morgan Kaufmann, 2003.

[9] J. Yang, Z. Lei, D. Yi and S. Z. Li, Person-specific face antispoofing with subject domain adaptation, *IEEE Trans. Information Forensics and Security*, vol.10, no.4, pp.797-809, 2015.

[10] L. Li, P. L. Correia and A. Hadid, Face recognition under spoofing attacks: Countermeasures and research directions, *IET Biometrics*, vol.7, no.1, pp.3-14, 2018.
[11] E. Fourati, W. Elloumi and A. Chetouani, Anti-spoofing in face recognition-based biometric authentication using image quality assessment, *Multimedia Tools and Applications*, pp.865-889, 2019.
[12] H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson and S. Z. Li, Attention-based two-stream convolutional networks for face spoofing detection, *IEEE Trans. Information Forensics and Security*, vol.15, pp.578-593, 2020.
[13] U. Muhammad and W. Melo, Face anti-spoofing via sample learning based Recurrent Neural Network (RNN), *The British Machine Vision Conference (BMVC)*, Cardiff, UK, 2019.
[14] Y. A. U. Rehman, L.-M. Po and M. Liu, SLNet: Stereo face liveness detection via dynamic disparity maps and convolutional neural network, *Expert Systems with Applications*, vol.142, 2020.
[15] Wirianto and T. Mauritsius, The development of face recognition model in Indonesia pandemic context based on DCNN and Arcface loss function, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1513-1530, 2021.
[16] H. Chen, Y. Chen, X. Tian and R. Jiang, A cascade face spoofing detector based on face, *IEEE Access*, vol.7, pp.170116-170133, 2019.
[17] R. Shao, X. Lan and P. C. Yuen, Joint discriminative learning of deep dynamic textures for 3D mask face anti-spoofing, *IEEE Trans. Information Forensics and Security*, vol.14, no.4, pp.923-938, 2018.
[18] H. Li, P. He, S. Wang, A. Rocha, X. Jiang and A. C. Kot, Learning generalized deep feature representation for face anti-spoofing, *IEEE Trans. Information Forensics and Security*, vol.13, no.10, pp.2639-2652, 2018.
[19] A. George and S. Marcel, Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks, *IEEE Trans. Information Forensics and Security*, vol.16, pp.361-375, 2021.
[20] M. Khammari, Robust face anti-spoofing using CNN with LBP and WLD, *IET Image Processing*, vol.13, no.11, pp.1880-1884, 2019.
[21] F. M. Chen, C. Wen, K. Xie, F. Q. Wen, G. Q. Sheng and X. G. Tang, Face liveness detection: Fusing colour texture feature and deep feature, *IET Biometrics*, vol.8, no.6, pp.369-377, 2019.
[22] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro and Ramyashree, Facial recognition using Haar cascade and LBP classifiers, *Global Transitions Proceedings*, vol.2, pp.330-335, 2021.
[23] A. Krizhevsky, I. Sutskever and G. E. Hinton, Advances in neural information processing systems, *Advances in Neural Information Processing Systems (NIPS2012)*, vol.25, 2012.
[24] F. Schroff, D. Kalenichenko and J. Philbin, FaceNet: A unified embedding for face recognition and clustering, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.815-823, 2015.