Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan *Vol. 31 No. 2-2026*
ISSN:1813-1786 (Print) 2313-7770 (Online)

Face Spoof Detection in Artificial Neural Networks Using Deep Learning

A. Sattar[1], M. Qadir[2], U. Rashid[3]

[1]*Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.*
[2]*Department of Information Security, The Islamia University of Bahawalpur, Pakistan.*
[3]*Department of Information Technology, University of Layyah, Pakistan.*

[3]umairrashiid@gmail.com

Abstract- Face spoof detection is crucial for biometric security and the reliability of face recognition. Advancements in artificial neural networks and machine learning algorithms have produced promising results in this area. As facial recognition technology is widely used in security, authentication, and access control, there has been a rise in face spoofing incidents. This attention-enhanced face spoof detection framework was developed using a Kaggle dataset and feature extraction techniques. An ensemble of three machine learning models, a Convolutional Neural Network (CNN), an Artificial Neural Network (ANN), and a Long Short-Term Memory (LSTM), was implemented to recognize subtle differences in facial features between real and spoof faces. The deep convolutional neural network achieved a superior accuracy of 98%, while the artificial neural network achieved 63% accuracy, and the long short-term memory achieved 53% accuracy. This study improves the accuracy and reliability of fraud detection systems, benefiting biometric identity, digital criminal investigations, and security measures. The proposed technique can significantly improve the security and reliability of facial recognition systems, raising their perceived reliability and accuracy. The generated system can recognize fake faces in surveillance, access control, and identity authentication systems.

Keywords- Face Spoof Detection; Face Recognition, Fraud Detection Systems, Identity Authentication.

I. INTRODUCTION

Facial biometrics play a central role in face anomaly detection, ensuring enhanced user satisfaction and data accuracy. Facial biometrics are widely deployed for the retention and archiving of user photographs, as historical data requires user identification using private videos and images. This increases the demand for user autonomy and workplace architecture [1]. Human faces are "things" because they can be represented on a computer, but they are also "selves" because they represent the innermost aspect of what a human is. Artificial neurons that draw inspiration from organic brain structures make up AI networks. The "Eigen face algorithm" gives neural networks campaign coefficients that they can use to create database images and lessen face faking [2]. Human faces are referred to as "things" because they can be replicated on a machine, and they are also "selves" because they are capable of expressing deep emotions. Artificial neurons that are modelled after the architecture of the natural brain make up AI networks. The "Eigen face algorithm" gives neural networks access to biometric-based authentication systems, which, in contrast to traditional methods, offer a secure, automated means of identification. These systems are required because of the increasing use of smartphones and cameras. They lessen face faking and make it possible to create database photographs [3]. Popular biometric modalities for security, such as voice and face recognition, are vulnerable to attacks via images and videos. This publication covers methods for identifying spoofing in facial recognition software and the datasets that researchers utilize. To prevent the practice of using attributes from public databases and executing these using a classifier, the anti-spoofing community should focus on the biological component of biometric traits. Future optimization methods may enhance system performance [4]. Artificial neural networks that have been trained to recognize faces in images improve computer learning and decision-making. In contrast, machine learning researchers use computer methods by using sophisticated data and expertise. Predictive maintenance and recommendation engines can both use it to forecast new output values based on historical data. Machine learning algorithms can also be used for facial spoof detection, which identifies, collects, and assesses facial traits to compare with previously recorded images [5].

Researchers have developed a deep CNN architecture that uses the LBP-TOP descriptor to capture spatiotemporal data on fraudulent films and legitimate access, preventing face faking in

biometric applications [6]. The study presents a color texture analysis method for detecting face spoofing, which proves more reliable than gray-scale analogs in unknown settings [7]. A CNN-based method was developed to efficiently identify phony facial expressions using generative adversarial networks, resulting in visually appealing facial photodetection from large data sets [8]. The study developed an anti-spoofing method using the Fisher Score, NUAA Photograph Imposter, and CASIA Face Anti-Spoofing Databases [9].

Research gaps persist in addressing spoofing attacks, despite advances in facial biometrics. Existing methods show promise, but robust solutions are needed for real-world scenarios. Moreover, the state-of-the-art literature highlights facial biometrics' superior accuracy and user satisfaction in face anomaly detection. It uses technologies such as ensemble face algorithms and CNNs to combat spoofing attacks, enhancing the security and reliability of biometric authentication.

This research advances biometric authentication systems utilizing artificial neural networks and machine learning algorithms to identify and prevent spoofing attacks, enhancing security in mobile devices and surveillance systems, and advancing spoof detection technology.

The paper provides a comprehensive exploration of facial biometrics and anti-spoofing measures, covering five main sections: introduction, literature review, methodology, results and findings, and future directions.

II. BACKGROUND AND RELATED WORK

The term fake face detection methods refers to the variety of techniques and algorithms used to analyze media, including digital images and video streams, and to identify artificial faces that are likely to have been altered digitally or physically. There has been a rise in fake faces, including identity theft, online scams, and cyberbullying, among others, which have become common security issues. This has promoted the development of technologies such as advanced photo-editing software and deep-learning algorithms, which, over time, have led to the illusion of seeing real people with unnatural appearances in the mass media [10]. Characteristics such as LBP, GLCM, and facial landmark detection fall within this scope. Facial feature extraction is crucial for automatic visual perception and human face identification systems. However, showing faces on computer devices is challenging due to the limited features. Recent research focuses on shape and feature detection methods that separate facial features from gray and color images. These methods are highly accurate, efficient, and time-efficient [11]. Face landmark detection using the UMB database is a crucial challenge. A PDM model was successfully designed to detect five distinctive areas, even with complex expressions and occlusions. The method's flaws need to be addressed by applying it to multiple databases for improved identification and fitting [12].

The review discusses the performance of deep learning- based facial landmark localization and video frame face correspondences in identifying facial landmark points for various applications [13]. This work enhances facial expression recognition (FER) by using a convolutional neural network classifier and face landmark detection, achieving average recognition rates of 84. 00%, 89. 19%, 85. 42%, and 84. 33% across four benchmark datasets. This study uses a convolutional neural network classifier with face landmark detection to improve facial expression recognition (FER), with average recognition rates of 84. 00%, 89. 19%, 85. 42%, and 84. 33% across four benchmark datasets [14]. In applications such as Virtual Reality, HCI, and Computer Vision, the study offers a precise and reliable machine- learning- based technique for real-time facial landmark detection, deploying CNN [15]. The paper introduces a three- stage cascade convolutional neural network for facial feature learning, achieving high detection accuracy and reliability with an average error rate of less than 1. 1.4% [16]. This work uses local binary patterns to recognize genuine and spoof faces to combat face spoofing attacks. The program uses a regional database of tagged images and deep learning techniques to reduce error rates and improve system efficiency [17]. This paper discusses advancements in face spoof detection and its use in biometric security enhancement, highlighting that techniques such as SVM, LDA, KNN, and ART outperform SVM in detecting spoofed faces [18]. With an emphasis on social networking and financial fraud, the study investigates the risks of face spoof attacks. With accuracy rates ranging from 92% to 97%, it suggests a machine- learning technique that uses convolutional neural networks to identify such attacks precisely [19]. The study enhances deep fake detection by using a hybrid CNN model and the sizable DFDC dataset. Its goals are to aid in the development of evaluation standards to prevent the rapid spread of misleading information, reduce the adverse societal effects of deep fakes on trust, and ensure accurate detection [20]. With good accuracy on benchmark datasets, the paper offers a solid foundation for spoofing attack detection in facial biometric systems employing convolutional autoencoders. Future research aims to improve image reconstruction, include other biometric features, and address spoofing attacks [21]. The literature highlights advancements in face spoofing detection, including a new evaluation process that considers unknown attack types and a one- class anomaly detection method. Examining 20 systems, the research emphasizes the value of feature

selection and the competitiveness of anomaly- based methods [19].

For face spoofing detection, a dual-channel neural network with a deep and a wide channel has been proposed. The architecture performs better than single-channel models and extracts domain-specific features and discriminative patterns. In cross-dataset tests, it performs better and is more generalized [22]. Facial expressions are still the primary choice, and face recognition algorithms make it easier for humans to perform tasks like recognizing faces in pictures. This method also includes identifying face features, including the lips, nose, eyes, and eyebrows. Following that, expression recognition will be used to identify face expressions. The method for identifying a face in an image will be discussed in this article. It uses Sony digital cameras and standard databases like Cohn-Kanade and JAFFE. More proximal facial information about people is stored by the method [14]. Due to security concerns, face recognition systems are being used more and more. But these systems are vulnerable to spoofing assaults. For quality analysis, the approach makes use of fractal dimension descriptors, entropy classification, and quality distortion features. The proposed approach outperforms the Quality Feature model by 5% and the Entropy Fractal Dimension model by 6%, with a maximum accuracy of 98.2%, as demonstrated by experimental data. To evaluate the effectiveness of this strategy, the following study will look at cross-database testing methods [23].

A comparison of face spoofing detection methods based on available literature evaluations is shown in Table 1.

Table 1: Comparative Analysis of Face Spoofing Detection Techniques

Reference	**Study Description**	**Methodologies**	**Performance/ Accuracy**
[24]	Reaches an efficiency of 0.89 percent by proposing a face-anti-spoofing neural network model to improve the safety and efficacy of face recognition systems.	Face-anti-spoofing neural network	0.89% efficiency
[25]	The proposed illumination-invariant anti-spoofing technique, combining RGB photos with MSR images and using a convolutional neural network, outperforms current techniques on the CASIA_SURF dataset.	Fusion-based methodology, Multi-Scale Retinex (MSR) images, Convolutional Neural Network	Better outcomes and lower error rates than current techniques
[26]	The study proposes a machine learning-based approach, specifically CNN, for accurate face detection, achieving 92% to 97% accuracy rates across various methods.	Machine Learning (CNN), LTP, DLTP, HOG, CLBP, GLCM	92% to 97%
[27]	Various techniques like PCA, FLD, and FPBM for feature extraction were used, achieving recognition rates of 93.8% to 95.5% for images, including noisy and non-noisy.	PCA, FLD, FPBM	93.8% to 95.5%
[28]	BSF-RCNNVFR, a video-based face recognition system, achieves 94.2% accuracy using bounding box regression, CNN-based feature learning, histogram redistribution, and Background Subtracted Faster RCNN.	BSF-RCNNVFR (Bounding Box Regression, CNN-based feature learning, histogram redistribution)	94.2%
[23]	With a maximum accuracy of 98.2%, the face spoofing detection system outperforms current models by utilizing entropy categorization, fractal dimension descriptor, and quality distortion characteristics.	Entropy categorization, fractal dimension descriptor, quality distortion features	98.2%
[29]	Improves face anti-spoofing using reasoning-based supervision and vision-language models.	Vision-language model with reasoning-based supervision	87%
[30]	Proposes an explainable multimodal framework for face anti-spoofing and spoof-region localization.	Multimodal large language model (MLLM)	Strong performance across FAS tasks

III. MATERIALS AND METHODS

3.1 Experimental Materials

A few crucial elements determine which face-spoofing technique to use next. Facial images from the dataset are first thoroughly pre-processed to ensure that subsequent feature extraction starts with high-quality images. This includes face cropping and frame selection, intended to standardize the

images for the task at hand. The second step is feature extraction, where many techniques are used for both fake and real images, including facial landmark detection, Gray Level Co-occurrence Matrix (GLCM), and Local Binary Pattern (LBP). These features are then concatenated to obtain a complete representation of each face. Other techniques used in the classification of data include Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and other classifiers. Therefore, the classification results are assessed using precision and accuracy metrics, which are useful in selecting the facial spoof detection model. The structure of the methodology is shown in Figure 1.

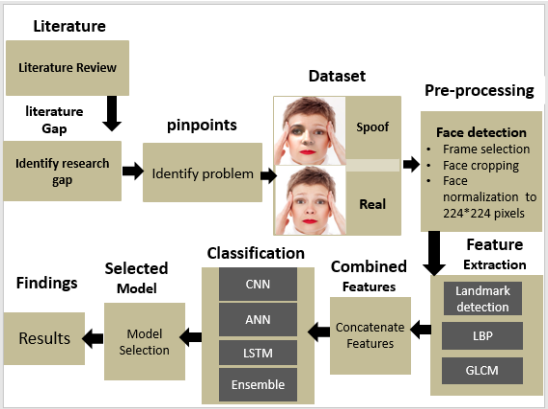

Fig 1. Schematic Diagram

3.2. Dataset

To address the problem of cybersecurity, organizations are equipping themselves with machine learning that enables them to create dependable biometric facial recognition systems, but there is a risk with corrupt photographs. The dataset used in this work, Real and Fake Face Detection, was made available by Kaggle and is held by Yonsei University's Department of Computer Science. The excellent collection of images comprises people's faces touched using the Photoshop software to resemble someone else's face. The validation of professionals ensures that the alteration of the photographs remains valid since the presented photographs can still be recognizable. The collection is divided into two separate directories: the one with 1081 files comprising real faces for training and the other with 960 files of modified images. It can be firmly based on this dataset that faces spoofing detection methods are studied.

3.2.1. Pre-processing of the Dataset

Selected and examined for scientific reasons, the dataset of this study comprises real and fake faces. This provides a wide variety of faces, positions, and lighting conditions, making the analysis very stable. The collection includes 2041 face images in total, with a fair distribution of genuine and fake faces. To address the variability of images within the dataset, all images are resized to the same size. Additionally, preprocessing steps, such as augmentation and normalization, may help achieve a richer, qualitatively sound dataset, thereby providing practical and realistic outcomes.

3.2.2. Feature Extracting

Facial landmark detection is employed for feature extraction in the innovative approach to FER. A convolutional neural network classifier is used to achieve high identification rates [14]. The research eliminates identification errors associated with occlusion, changes in posture, or illumination by learning facial features from a three-stage cascade convolutional neural network. The approach yields a low average error rate of less than 1% in overall system performance, coupled with good detection accuracy and resistance [16]. Facial landmark identification is required for identity recognition and expression analysis, as well as for animation, 3D generation, and beauty care. Face alignment and deep compression models, together with techniques for 2D facial landmark localization, have improved the high performance of deep learning in video frame face correspondence [13]. The feature extraction phase focuses on face landmark detection, which is an essential step in the process. Advanced algorithms are used to identify critical areas on each face, such as the lips, eyes, and nose. Facial landmarks are detected using Python, dlib, and OpenCV. After localizing a face with Haar cascades or HOG, the (x, y)-coordinates of the face areas are found using a facial landmark detector.

Let FF be the set of facial landmarks, each represented as a point (xi, yi)
(xi,yi) in the image.

$F=\{(x1,y1),(x2,y2),...,(xn,yn)\}$F={(x1,y1),(x2,y2),...,(xn,yn)}

While the mathematical formulation for these techniques differs, we can employ techniques like template matching to discover facial landmarks. Here is a general equation for facial landmark recognition using a CNN:

Let II be the input image, and LL be the output landmarks.

$L=f(I)$ L=f (I)

Where ff represents the CNN model that takes the input image II and predicts the facial landmarks LL. The detailed mathematical expression of the CNN model is contingent upon its architecture, encompassing the number of layers, varieties of activation functions, and parameters for training. Many computer vision tasks, including face alignment, blink detection, head pose estimation, face swapping, and face part extraction, can be performed with the help of these landmarks. For additional investigation, the coordinates of these points are stored in a CSV file format. By utilizing deep learning approaches, the algorithm lowers error

rates and boosts system efficiency by detecting genuine faces using local binary patterns [17].

One statistical technique for capturing the spatial connections between pixels in an image is called GLCM. Denote the co-occurrence matrix P(i,j|θ,d)P(i,j|\theta, d)P(i,j|θ,d) where iii and j are the gray levels of two pixels, θ\thetaθ is the direction, and it is the distance between the two pixels. The co-occurrence matrix's elements are determined by calculating the frequency at which pixel pairings with gray levels iii and j occur at the designated offset (d,θ)(d, \theta)(d,θ). The grey level co-occurrence matrix is then normalized frequently by dividing each element of the matrix by the total of all the elements.

In terms of math, the GLCM is expressed as:

P(i,j|θ,d)=1Nd∑x=1N∑y=1Mδ(f(x,y)=i,f(x+dcos⁡(θ),y+dsin⁡(θ))=j)P(i,j|\theta, d) = \frac{1}{N_d} \sum_{x=1}^{N} \sum_{y=1}^{M} \delta(f(x,y) = i, f(x+d\cos(\theta), y+d\sin(\theta)) = j)P(i,j|θ,d)=Nd1∑x=1N∑y=1M δ(f(x,y)=i,f(x+dcos(θ),y+dsin(θ))=j)

Where NdN_dNd is the number of valid pixel pairs at distance D, N, and MMM are the dimensions of the image, f(x,y)f(x,y)f(x,y) is the intensity at pixel (x,y)(x,y)(x,y), and δ(⋅)\delta(\cdot)δ(⋅) is the Kroenke delta function which returns 1 if the condition is true and 0 otherwise. Techniques LBP descriptors are widely used, but their efficiency decreases when similar textual characteristics are observed. More effective methods for distinguishing between spoofed and non-spoofed faces include LDA, KNN, and ART [18]. LBP is a texture descriptor that can be used to identify and classify different textures. By comparing a core pixel's intensity with that of its surrounding pixels, it encodes the local structure of an image.

LBP codes of the central pixel PcP_cPc and its neighbors are calculated in a circularly symmetric neighborhood of radius R as follows: The binary pattern is obtained using the following conditions: 0 for the case where the intensity of the neighbor is not equal to or greater than the central pixel, and 1 otherwise. After so doing, a decimal value that corresponds to the obtained LBP code of the center pixel is then obtained through the application of the binary pattern produced.

The LBP code of a pixel PcP_cPc is calculated mathematically as follows:

LBP P,c(P)=∑p=0P−1s(gp−gc)·2pLBP_{P,c}^{(P)} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^pLBPP,c(P)=∑p=0P−1s(gp−gc)·2p

Whereby the following definition applies to the sign function, s(⋅)s(\cdot)s(⋅): The number of neighbors is denoted by PPP, while the intensities of the central pixel and the pthp^{th}pth neighbor are represented by gpg_cgc and gpg_pgp, respectively.

3.3. Concatenation and Data Integration

All of these, including GLCM features, LBP histograms, and face landmark coordinates, must be compiled into one data set for the various methods. This composition also enhances the predictive strength, as well as the categorization precision. To address this issue, the researchers employed a gray-level co-occurrence matrix, local binary patterns, and facial landmarks for detecting the required patterns for differentiation between the original and modified faces.

IV. CLASSIFICATION

Labelling the incoming data based on the patterns that are learned from the training data set is known as a fundamental step in learning and artificial intelligence. It can be used in text, image, audio, and other data processing, among others.

4.1. CNN

Or CNNs, which are artificial neural networks, are capable of emulating the human hierarchy through which spatial constituents can be identified from photographs. Its convolutional model is constituted by convolutional and pooling layers as well as a fully connected layer. CNNs employ pixels to analyze input images and perform image categorization and identification. There are several components in this to specify the architecture, and these are: convolution, striding, padding, pooling layers, fully connected layers, and non-linear activation (ReLU). Convolutional neural networks, more commonly abbreviated as CNNs, are favourite for featuring information from images and categorizing the images as real or fake. CNNs process picture data with the help of many mathematical operations. The main equations and mathematical parts of a CNN are as follows:

The main equations and mathematical parts of a CNN are as follows:

A) Convolutional Layer: Feature maps are created by the convolutional layer by applying a collection of filters, or kernels, to the input image. The following equation can designate the operation:

FeatureMapij(k)=∑m=0M−1∑n=0N−1Input(i+m)(j+n)·Kernelmn(k)+b(k)\text{FeatureMap}_{ij}^{(k)} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \text{Input}_{(i+m)(j+n)} \cdot \text{Kernel}_{mn}^{(k)} + b^{(k)}FeatureMapij(k)=∑m=0M−1∑n=0N−1 Input(i+m)(j+n)·Kernelmn(k)+b(k)

- FeatureMapij(k)\text{FeatureMap}_{ij}^{(k)}FeatureMapij(k): The value at position (i,j)(i,j)(i,j) in the kkk-th feature map.
- Input(i+m)(j+n)\text{Input}_{(i+m)(j+n)}Input(i+m)(j+n): The value at position (i+m,j+n)(i+m, j+n)(i+m,j+n) in the input image.

- Kernelmn(k)\text{Kernel}_{mn}^{(k)}Kernel mn(k): The value at position (m,n)(m,n)(m,n) in the kkk-th kernel (filter).
- b(k)b^{(k)}b(k): The bias term for the kkk-th kernel.
- MMM and NNN: The dimensions of the kernel.

B) Activation Function**:** After convolution, an activation function is applied element by element. The most common activation function used by CNNs is the Rectified Linear Unit (ReLU):
ReLU(x) = max_0,x= \max(0, x) \text{ReLU}(x)ReLU(x) equals max(0,x)
C) Pooling Layer: The pooling layer, which often uses max pooling, reduces the spatial dimensions of the feature maps. The following is how max pooling works:
\text{MaxPool}_{ij} = \max_{(m,n) \in \text{PoolRegion}} \text{Input}_{(i+m)(j+n)} MaxiPoolij=max⁡(m,n)∈PoolRegionInput(i+m)(j+n)MaxPoolij is equal to max(m,n) ∈ PoolRegionInput(i+m)(j+n).
MaximumPolij\text{MaxPool}_{ij}MaxPoolij: The pooled feature map's value at location (i,j)(i,j)(i,j).
PoolRegion\text{PoolRegion} (1)PoolRegion: The area that receives the greatest operation (such as a 2x2 region).
D) Fully Connected Layer: The fully connected (dense) layer flattens the feature maps and performs a weighted sum, followed by an activation function (usually softmax for classification):
Outputk=σ(∑iWki·FlattenedInputi+bk)\text{Output}_k = \sigma \left(\sum_{i} W_{ki} \cdot \text{FlattenedInput}_i + b_k \right)Outputk=σ(∑iWki·FlattenedInputi+bk)

- Outputk\text{Output}_kOutputk: The output value for the kkk-th class.
- WkiW_{ki}Wki: The weight connecting the iii-th input to the kkk-th output.
- FlattenedInputi\text{FlattenedInput}_iFlattenedInputi: The iii-th element of the flattened input.
- bkb_kbk: The bias term for the kkk-th output.
- σ\sigmaσ: The activation function, typically softmax for the output layer:

σ(z)k=ezk∑jezj\sigma(z)_k = \frac{e^{z_k}}{\sum_{j} e^{z_j}}σ(z)k=∑jezjezk
The convolution layer generates the output image, while the strides determine the step size used to move the filter/kernel over the input volume or image. Padding addresses dimensionality reduction caused by convolutional operations on input data, with two main types: zero-padding and valid-padding. ReLU adds non-linearity to the network so that it can interpret and represent the relationships found in the data. Pooling layers reduce the spatial dimensions of feature maps while also conserving important information, cutting down on processing costs, and avoiding overfitting. The picture input matrix is converted into a vector representation and made into a model by the fully connected layer. Python modules such as Matplotlib, sklearn, TensorFlow, and NumPy are used in the development, assembly, and fit-training of the model. After that, the model is ready for testing, and it achieves a 98% accuracy rate. The CNN procedure includes the following steps: importing the dataset, splitting it up into feature and target variables, and feature scaling.

4.2. LSTM

In cases of videos and their frames, a specific type of recurrent artificial neural network called long short-term memory networks, or LSTM networks for short, is often used. It does this by collecting sequential patterns that can be used to make comparisons with instances in the data set, as well as temporal correlations, which afford distinction of a fake face from a real one. Featuring the sequences of feature vectors, LSTM models provide classifications of temporal behavior to help detect abnormal motion and appearance. Since LSTM utilizes previous frames in making a decision on the authenticity of a given frame, it is well-suited for modeling sequences of data. Because of its memory cells' long-term retrieval and storing capabilities, the network can identify deep false artifacts by identifying long-range dependencies in visual footage. From video frames, LSTM collects high-level traits that are then utilized to forecast the video's authenticity.

A) Cell State (ct): This represents the long-term data stored in the LSTM cell. It is updated at each time step using both previous data and the most recent input.

The following is the cell state update equation:

- Previous Cell State (ct-1): carries data from previous time steps.
- Forget Gate (ft): determines how much information from the previous cell state is lost; values range from 0 to 1. To determine it, a sigmoid activation function is employed:
  *ft = σ(Wi * [ht-1, xt] + bi)*
- Input Gate (it): The current input amount (xt) that is incorporated into the cell state is calculated. Its calculation also employs a sigmoid function:
  *it = σ(Wi * [ht-1, xt] + bi)*
- Candidate Cell State (Ct'): indicates that new information has to be added to the cell condition. Using the hyperbolic tangent (tanh) activation function, this is calculated:
  *Ct' = tanh(Wi * [ht-1, xt] + bi)*
- Updated Cell State (ct): combines the prospective cell state with the previous cell state by means of the forget gate.
  *ct = ft * ct-1 + it * Ct'*

B) Hidden State (ht): At a particular time step, this is the LSTM cell's output. The updated cell state and the output gate are used to calculate it:
- Output Gate (ot): regulates the portion of the cell state that is represented in the output. It follows the same forget and input gate protocol, using a sigmoid function:
  $ot = \sigma(Wi * [ht-1, xt] + bi)$
- Hidden State (ht): brings together the result of the tanh function and the modified cell state:
  $ht = ot * tanh(ct)$

C) Activation Functions: To add non-linearity, LSTMs use activation functions such as tanh and sigmoid (σ). Before being fed into the following layer, these functions change the weighted sums of the inputs.

D) Weight Matrices (Wi) and Bias Vectors (bi): During the training phase, these parameters are obtained. They ascertain the degree to which each component affects the computations.

The fundamental mathematical ideas behind LSTMs for spoof detection are as follows. Complicated matrix operations and backpropagation are used in the actual implementation to get the ideal weights and biases. Since LSTM networks capture information, which is learned in short training sessions and apply it in long training, then LSTM networks yield better results than simple RNN's. These consist of forget, input, and output gates, which allow information flow or inhibit such information flow. LSTMs are designed explicitly as recurrent neural networks, which work with time-series inputs and which have an O (1) complexity per time step and weight. The goal of this work is to show that LSTM can be used to provide standard/pathological classifications of ECG signals, to show that classification accuracy can be constant for heterogeneity-mitigated data across heterogeneous data over robotic systems.

4.3. ANN

To enhance the classification operation, this work used a multi-layer ANN trained via backpropagation for replication of the interaction of distinctive features with discrete target labels. The topography of the ANN model comprised of one output layer, which employed the softmax function, one or more hidden layers that used the ReLU activation function, and the last input layer. ANNs are discrete models of machine learning that address, send, and process information. They derive this from the several layers of interlinked neuron circuits in the human brain. Artificial neural networks (ANNs) operate through 'training'; they are designed for specific applications such as identifying patterns or categorising data. The synaptic connections between neurons may be changed as a result of learning, and thus ANNs could adapt to a generic range of tasks. An artificial neural network (ANN) is composed of an input layer that processes input data in several formats, including text, numbers, audio files, and picture pixels. The center part of the ANN model contains hidden layers that process the input data mathematically to find patterns. To get the intended outcome, the output layer, which is the middle layer, carefully calculates. Utilizing activation function F(X), the input signal is turned on. Among them are the activation functions tanh, ReLu, sigmoid, and others. The following example uses the sigmoid activation function in real life at nodes in both levels.

$$F(X)=\frac{1}{1+e^{-x}}$$

The value at hidden layer 1 is equal to (1 * 0.1) + (1 * 0.1) + (1 * 0.1) = 0.3 when we solely look at computations at hidden node 1, H1, in the ANN figure above.

$$Y_{in}=\sum X_i * W1_{ij}$$

Sigmoid (0.3) = 1/ (exp (0.3) +1) = 0.57 is the result of using the sigmoid activation function.

$$Y_{out}=F\ (Y_{in})$$

Every concealed node performs comparable computations, after which the results are sent to the output layer. Node value in the output layer.

The value at the output node is equal to (0.57 * 0.1) + (0.57 * 0.1) + (0.57 * 0.1) = 0.228

$$Z_{in}=\sum Y_{out} * W2_{jk}$$

Sigmoid (2.28) = 1/ (exp (2.28) +1) = 0.56 is the result of using the sigmoid activation function.

$$Output=F\ (Z_{in})$$

The smallest processing component in ANNs is the neuron, which uses a sigmoid activation function for inputs and weights. The weights between connective nodes are referred to as biases and are derived through training data for several iterations. The ANN consists of two hidden layers, each with four neurons, a single output layer with two output units, and a single input layer with four input units.

4.4. Ensemble Classification

Combining CNN with LSTM and CNN with ANN produced greater classification performance when the results of multiple models were cascaded. Thus, the investigation was dedicated to the set of possibilities of amplifying the advantages of each paradigm separately, such as majority voting and weighted averaging. The employed mean of all the models' forecasts and the process of majority voting made it possible to select the path that would correspond to the majority. Whereas common averaging assigned equal weights to each of the models, weighted averaging offered more weight to the models' predictions that performed better. The idea of the ensemble of models was to minimize the weakness of each model and improve the error tolerance capacity of the classification system by boosting the strong facets of all the models.

V. Discussion and Findings

In the study, concatenated feature vectors were utilized to train CNN, LSTM, and ANN models by supervised learning methodology. To avoid overfitting, a training and validation group for the dataset was established. Generalization was enhanced by data augmentation approaches, and the loss function was decreased through the use of optimization algorithms such as Adam. Grid search and cross-validation were used to get the best results while adjusting hyperparameters like learning rate, batch size, and epoch number. Both precision and accuracy measures were used to assess the trained models; precision calculated the percentage of true positive classifications, while accuracy looked at the overall accuracy of the model's predictions. Reliability and robustness were verified using cross-validation.

Effective learning of complex patterns and features was demonstrated by a CNN model that was trained across 10 epochs and showed training accuracy of 99.76% and validation accuracy of 55.98%. However, decreased validation accuracy suggests overfitting. Convolutional neural networks (CNNs), widely used in facial recognition, perform better with an increasing number of convolutional layers. The model's design comprised two dense layers, three activations and pooling layers, and three convolutional layers, all of which had 99% accuracy. Applications including speech recognition, natural language processing, and time series forecasting benefit from the usage of recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM). However, considering their poor training and validation accuracy, it seems that they would not be suitable for classification tasks. For optimal performance, a higher number of convolutional layers is needed, yielding a 54% accuracy rate. The fact that this model requires less precision than the CNN design means that more intricate structures and optimizations might be required. Artificial Neural Networks (ANNs), which are made up of layers of connected neurons, are used in deep learning models. With final training scores of 63.01% and validation scores of 54.07%, the ANN model performed at a reasonable level. But beyond training data, it struggled with generalization. The best results were obtained with higher convolutional layer counts, which required less time to complete and resulted in 63.01%accuracy rates.

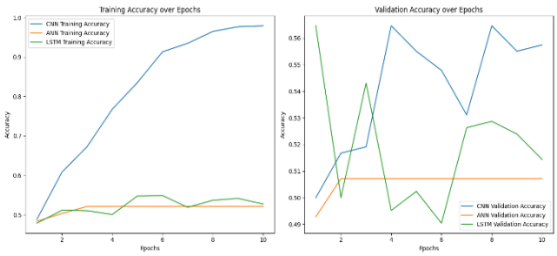

Fig 2. Analyzing the Accuracy Over Epochs Graph

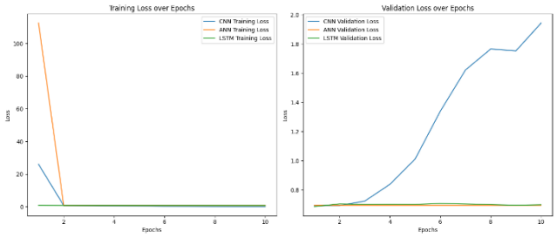

Fig 3. Analyzing the Loss Over Epochs Graph

The CNN shows the best training accuracy as the 10th epoch gets closer to 1.0, demonstrating its strong performance throughout training. Additionally, as compared to the ANN and LSTM models, the CNN achieves the highest validation accuracy, however it varies. Despite their lower accuracy, the ANN and LSTM models show more consistent validation loss values, which could suggest better generalization. The application of ensemble techniques is recommended to balance these benefits and drawbacks among multiple models.

VI. Conclusion

The paper uses a Kaggle dataset as proof of an effective method for face spoofing detection. Three feature extraction methods, the GLCM approach, LBP, and facial landmark detection, were used to distinguish real faces from fake ones. An ensemble approach using CNN, ANN, and LSTM models was used. While the ANN and LSTM models had lower accuracy of 63% and 53%, respectively, the CNN model reached a high accuracy of 98%. The research contributes to the development of face spoofing detection algorithms. It can be applied in real-life scenarios, particularly in access control systems, video surveillance systems, and identity verification processes.

VII. Acknowledgement

This work was completed as part of my research thesis under the supervision of Dr. Shahid Farid for the MSCS program in the Computer Science department of Bahauddin Zakariya University, Multan.

REFERENCES

[1] A. A. Una, "Classification Technique for Face-Spoof Detection in Artificial Neural Networks using Concepts of Machine Learning," 2021.

[2] V. Panwar, D. Kumar Sharma, K. V. Pradeep Kumar, A. Jain, and C. Thakar, "Experimental investigations and optimization of surface roughness in turning of en 36 alloy steel using response surface methodology and genetic algorithm," in *Materials Today: Proceedings*, Elsevier Ltd, 2020, pp. 6474–6481. doi: 10.1016/j.matpr.2021.03.642.

[3] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: 10.1109/TCSVT.2003.818349.

[4] P. Kavitha and K. Vijaya, "A Study on Spoofing Face Detection System." http://www.ijpam.eu

[5] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," May 01, 2021, *Springer*. doi: 10.1007/s42979-021-00592-x.

[6] Institute of Electrical and Electronics Engineers, *2017 2nd International Conference on Image, Vision and Computing (ICIVC 2017): August June 2-4, 2017, Chengdu, China.*

[7] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face Spoofing Detection Using Colour Texture Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, Aug. 2016, doi: 10.1109/TIFS.2016.2555286.

[8] H. Mo, B. Chen, and W. Luo, "Fake Faces Identification via Convolutional Neural Network," 2018, doi: 10.1145/3206004.

[9] A. Benlamoudi, D. Samai, A. Ouafi, S. E. Bekhouche, A. Taleb-Ahmed, and A. Hadid, "Face spoofing detection using local binary patterns and Fisher Score," in *3rd International Conference on Control, Engineering and Information Technology, CEIT 2015*, Institute of Electrical and Electronics Engineers Inc., Aug. 2015. doi: 10.1109/CEIT.2015.7233145.

[10] W. Alkishri, J. H. Yousif, A.-B. Mahmood, and M. Al-Bahri, "Deepfake Image Detection Methods Using Discrete Fourier Transform Analysis and Convolutional Neural Network," 2023, doi: 10.17605/OSF.IO/BNPC4.

[11] Y.-M. Wu, H.-W. Wang, Y.-L. Lu, S. Yen, and Y.-T. Hsiao, "Facial Feature Extraction and Applications: A Review," 2012.

[12] A. Abdu Yusuf, Z. Sufyanu, F. Susilawati Mohamad, K. Nanaa, and C. Author, "Facial Landmark Detection and Estimation under Various Expressions and Occlusions," *International Journal of Sciences: Basic and Applied Research (IJSBAR) International Journal of Sciences: Basic and Applied Research*, vol. 18, no. 2, pp. 40–50, 2014, http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

[13] Y. Yan, X. Naturel, T. Chateau, S. Duffner, C. Garcia, and C. Blanc, "A survey of deep facial landmark detection." https://hal.science/hal-02892002

[14] S. Khandait, "An Efficient Approach to Facial Feature Detection for Expression Recognition," 2009. https://www.researchgate.net/publication/228710078

[15] R. Vasavi, M. Keerthi, M. Thaslima, P. N. Sree, and S. Afreen, "Facial Landmark Detection Using Machine Learning." www.ijfmr.com

[16] X. Chen, L. Huang, and Y. Chen, "Facial Landmark Detection based on Cascade Neural Network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2019. doi: 10.1088/1742-6596/1237/3/032048.

[17] A. Shukla, V. Dave, A. Mishra, and R. Scholar, "Spoof Detection Using Local Binary Pattern in Face".

[18] P. Sharma, N. Chauhan, and A. Professor, "Issue 6 www.jetir.org (ISSN-2349-5162)," 2019. www.jetir.org

[19] Institute of Electrical and Electronics Engineers and Institute of Electrical and Electronics Engineers. Delhi Section, *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom): 23rd-25th March, 2022, New Delhi, India.*

[20] S. Thaseen Ikram, S. Chambial, and D. Sood, "A Performance Enhancement of Deepfake Video Detection through the use of a Hybrid CNN Deep Learning Model 169 Original Scientific Paper."

[21] S. Arora, M. P. S. Bhatia, and V. Mittal, "A robust framework for spoofing detection in faces using deep learning," *Visual Computer*, vol. 38, no. 7, pp. 2461–2472, Jul. 2022, doi: 10.1007/s00371-021-02123-4.

[22] S. Hashemifard and M. Akbari, "A Compact Deep Learning Model for Face Spoofing Detection," Jan. 2021, http://arxiv.org/abs/2101.04756

[23] N. Daniel and A. Anitha, "Texture and quality analysis for face spoofing detection," *Computers and Electrical Engineering*, vol.

94, Sep. 2021, doi: 10.1016/j.compeleceng.2021.107293.

[24] "A Novel Face-Anti spoofing Neural Network Model for Face Recognition and Detection."

[25] K. Gopalakrishnan, T. Soundarya, and S. Santhiya, "Face Anti-Spoofing using Deep Learning." https://ssrn.com/abstract=4089543

[26] Institute of Electrical and Electronics Engineers and Delhi Section, Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom): 23rd-25th March, 2022, New Delhi, India, 2022.

[27] R. H. Abiyev, "Facial feature extraction techniques for face recognition," *Journal of Computer Science*, vol. 10, no. 12, pp. 2360–2365, 2014, doi: 10.3844/jcssp.2014.2360.2365.

[28] B. Ravi Teja, D. Mythili, L. Duvva, S. Bethu, and Y. Garapati, "Deep Learning Feature Extraction Architectures for Real-Time Face Detection," *SN Comput Sci*, vol. 4, no. 5, Sep. 2023, doi: 10.1007/s42979-023-02023-5.

[29] J. Min, K. Lim, M. Kim, D. Kim, S. Oh, E. Kim, and H. Jang, "Analyzing the effect of reasoning-based supervision on face anti-spoofing," *Scientific Reports*, vol. 16, Art. no. 13360, 2026, doi: 10.1038/s41598-026-43800-5.

[30] H. Wang, Y. Shi, Z. Tao, Y. Gao, L. Zhang, X. Lin, J. Feng, X. Yuan, Z. Yu, and X. Cao, "FaceShield: Explainable Face Anti-Spoofing with Multimodal Large Language Models," *Proc. AAAI Conf. Artif. Intell.*, vol. 40, no. 12, pp. 9811–9819, 2026, doi: 10.1609/aaai.v40i12.37945.