Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260503122
Volume 5, Issue3 (September-December 2026), PP: 1022-1033.
www.indjcst.com
ISSN No: 2583-5300

Live Guard AI: Machine Learning-Based Image-Attack Face Liveness Detection System

Prasanna kumar Naik R[1], Chethan kumar G S[2], Swarnalatha[3], Preethi K P[4], Chandana N G[5]

[1,2,3,4] Assitant Professor, Department of Master of Computer Applications, University BDT College of Engineering (A Constituent College of VTU, Belagavi), Davangere, Karnataka, India.
[5]Student, Department of Master of Computer Applications, University BDT College of Engineering(A Constituent College of VTU, Belagavi), Davangere, Karnataka, India.

To Cite this Article: *Prasanna kumar Naik R[1], Chethan kumar G S[2], Swarnalatha[3], Preethi K P[4], Chandana N G[5], "Live Guard AI: Machine Learning-Based Image-Attack Face Liveness Detection System", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 1022-1033.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Face-based authentication has become an noticeable method for secure access to digital arrangements, but still, still, standard face recognition can be deceived by presentation attacks such as printed photographs, showed images, and replayed facial content; this creates a security gap when a framework checks facial identity without first figuring out whether the introduced face belongs to a live person; this paper is offering **LiveGuard AI**, a deep learning-based biometric authentication system designed to combine face liveness detection with identity verification, on the whole. As, well, the showed system first detects and pulls the face from an input frame and then employs a transfer learning-based ResNet-18 model to classify the detected face as either genuine or spoofed, broadly speaking; facial representations are made using OpenFace embeddings and compared using Euclidean distance for identity verification, broadly speaking. To lessen decisions based on a single unstable frame, the system is using a multi-frame voting mechanism in which authentication is accepted when at least two of three consecutive predictions meet the needed conditions, in real-world scenarios; this setup is further supported by application-level security processes, covering proved genuine API access, password protection, input validation, rate limiting, session management, and automated security alerts, in real-world scenarios. Sure enough, experimental check did is, by design, showing that the formulated liveness model can basically distinguish genuine facial presentations from spoof attempts, in actual application during the project, to a significant degree. Meaning and security, particularly for reliability testing was also ran to support the behaviour of the complete authentication workflow, in many respects; these outcomes are making it clear that combining liveness detection, facial identity verification, temporal decision-making, and application security is offering a stronger authentication workflow than hinging on face recognition alone, on the whole. The put forward system almost founds for working out secure and spoof-resistant biometric authentication applications, in many respects.*

Key Word: *namely Face Liveness Detection, Biometric Authentication, Face Anti-Spoofing, Deep Learning, Transfer Learning, ResNet-18, Face Recognition, OpenFace, Presentation Attack Detection, Facial Embeddings, Multi-Frame Verification, Secure Authentication, in many respects.*

I. INTRODUCTION

Face recognition has become an vital biometric technology for applications such as digital authentication, access control, attendance systems, and identity verification, to a significant degree; in many respects; as well, so, its ability to spot individuals from facial traits is mainly offering a convenient alternative to passwords, cards and other classic authentication methods, to a significant degree. More accurately, catching a face alone isn't enough, when the dust settles up, when the goal is usually to offer secure biometric authentication, in many respects, but standard face recognition systems generally suppose that the face introduced to the camera belongs to a live person, in many respects; indeed, what this assumption creates is a security weakness, because an attacker may, in practice, introduce a photograph, a showed facial image, or another form of recorded facial content to imitate a genuine user, in actual application.

Face presentation attacks have augmented the need for architectures that can (to a meaningful extent) distinguish between genuine facial presentations and artificial or reproduced ones; this roadblock is commonly tackled, through the face liveness detection or presentation attack detection, in real-world scenarios; what's more, a even authentication system should first figure out whether the detected face represents a real person and only then check, in actual application. My point is, combining these two stages can, by design, probably (in concrete terms) lessen the possibility of accepting an unauthorized person employing a copied or replayed facial representation, on the whole.

Recent advances in deep learning have basically methoded for learning visual credits associated with genuine and spoofed facial presentations, in actual application; a ResNet-18-based transfer learning approach is used for binary liveness grouping in the showed LiveGuard AI system, broadly speaking. As, well, transfer learning is particularly helpful when the

available domain-specific training data is limited, since a model can start from representations learned from large-scale image datasets and later fit them to the target classification task, and the model processes the detected facial region and figures out whether the presentation should be considered genuine or spoofed, on the whole.

Liveness detection alone, though, doesn't establish the identity of the person; the created representation is inspected with the stored identity representation employing Euclidean distance, which is allowing the framework to gauge both the physical authenticity of the introduced face and its similarity to the registered user, in real-world scenarios. OpenFace is, in practice, employed to compactly represent in the form of a 128-dimensional embedding; when you want to tackle this need, LiveGuard AI is usually combining liveness category with facial identity verification, in actual application.

Another vital consideration is, in effect, the instability that can (in practice) start when authentication decisions are normally made from a single video frame, in actual application, and to lower the change of such changes perhaps, the suggested system is currently employing a multi-frame decision mechanism based on three consecutive predictions, in actual application. Authentication is, more accurately, accepted when at least two of the three predictions meet the needed conditions, broadly speaking, and more, an individual prediction may be impacted by changes in illumination, facial movement, camera quality, or short-term classification errors, broadly speaking.

The principal contributions of this work are as follows:

1. A deep learning-based face liveness detection framework employing transfer learning with ResNet-18 for distinguishing genuine facial presentations from spoofed presentations, to a significant degree.
2. Integration of face detection, liveness category, and identity verification into a combined biometric authentication workflow, allowing the framework to assess both presentation authenticity and user identity, broadly speaking.
3. Incorporation of OpenFace 128-dimensional facial embeddings and Euclidean distance-based comparison for inspecting whether a live facial presentation corresponds to the registered user, to a significant degree.
4. Application of a multi-frame temporal voting mechanism employing three consecutive predictions, where authentication is often accepted when at least two predictions fulfill the needed authentication conditions, thereby lessening the effect of unstable single-frame decisions, in many respects.
5. Development of a security-aware authentication application covering password protection, supported real API access, input validation, rate limiting, session management, and automated security alert mechanisms to strengthen protection against unauthorized access, to a significant degree.
6. Check of the worked out framework, through the model performance analysis and security testing to assess the reliability of the liveness detection and complete authentication workflow, in many respects.

II. RELATED WORK

Deep learning-based biometric architectures have received large attention, because of their ability to learn discriminative facial credits directly from image data, in real-world scenarios. Well the same, still, standard face recognition systems can, of course, remain vulnerable when an attacker offers a photograph, screen image, or replayed facial representation instead of a genuine person; this shortcoming has led researchers to study face liveness detection and presentation attack detection as extra security layers for biometric authentication, in many respects.

Schroff et al. [1] introduced FaceNet, a deep learning framework that learns compact facial representations by mapping face images into an embedding space, in real-world scenarios; a triplet-based goal is currently being employed by the method to make sure that images belonging to the same individual are positioned closer together, to a significant degree. So far, images from distinct individuals are separated; in turn, the resulting embeddings basically represent for face recognition and verification; what this work established was the importance of embedding-based identity comparison and founded for next facial recognition systems, in real-world scenarios. Only the same, identity embeddings alone don't find out whether the input comes from a live person or from a spoofed presentation; facial identity representation with an autonomous liveness detection stage is therefore combined by the present work, in many respects.

Liu et al. [4] introduced a deep learning-based face anti-spoofing method that covered extra information to sharpen the difference between live faces and presentation attacks, to a significant degree, and what their work showed was counting only on standard facial appearance can be insufficient for solid anti-spoofing, particularly when the framework comes, across the mixed forms of attack, to a significant degree; indeed, these findings are stressing the importance of learning credits that represent qualities associated with the physical authenticity of the offered face, broadly speaking. Liveness category is therefore treated as a separate security stage before identity verification rather than assuming that successful face recognition automatically suggests a genuine user in LiveGuard AI, broadly speaking.

Another fundamental consideration in biometric authentication is the reliability of decisions made from individual video frames, in actual application; on the whole, and so, a single frame may be impacted by illumination changes, facial movement, camera quality, or short-term prediction errors, in many respects; temporal information can (in concrete terms) therefore be employed to get a more stable authentication decision; in the suggested framework, three consecutive liveness and identity verification outcomes are, by design, considered, in actual application. As, well, authentication is typically accepted when at least two predictions fulfill the needed conditions, to a significant degree; on the whole; what's more, thus, the change of an separated incorrect prediction is being typically lowered by this multi-frame voting mechanism and also lays between model inference and final authentication, in real-world scenarios.

Security mechanisms surrounding the machine learning model are, in effect, also noticeable for concrete biometric applications, in many respects; more accurately, a trustworthy anti-spoofing model alone doesn't protect an application from unauthorized API access, invalid requests, credential attacks, or repeated authentication attempts, in real-world scenarios,

and then on, liveGuard AI is combining the machine learning pipeline with application-level controls covering password protection, proved genuine API access, input validation, rate limiting, session management, and automated security alerts; this layered method generalizes the protection of the framework beyond the prediction model and tackles security requirements at the application level, in real-world scenarios.

III. DATASET

A. Dataset Description

The suggested LiveGuard AI system is generally employing the **Large Crowd-Collected Face Anti-Spoofing Dataset (LCC-FASD)** for training and finding the facial liveness detection model, on the whole, and more, lCC-FASD is a face anti-spoofing dataset designed to distinguish genuine facial presentations from presentation attacks, to a significant degree. What's true is, the dataset is consisting of facial images embodied under the varied conditions and includes both authentic face samples and spoofed presentations, and is organized into the separate training, development and evaluation subsets, allowing the model to be trained and verified without mixing the samples employed for final check, in actual application. Bluntly, what it includes is generally 18,827 images distributed across 243 subjects, with 1,942 genuine facial images and 16,885 spoofed images, in actual application.

The dataset is, more accurately, divided into two primary categories, including **REAL** and **SPOOF**; so, REAL samples represent genuine facial presentations encapsulated from actual users, to a significant degree; and so, during the model development, the highly unequal distribution between the two categories importantly is assessing, broadly speaking. Though, still, spoof samples represent presentation attacks in which facial information is currently reproduced or displayed to imitate a genuine person, in actual application; printed photographs can, to some extent, be covered by such attacks, images displayed on digital screens, and other reproduced facial presentations, in many respects. Put it like this, you'd think, the training subset is mainly consisting of 1,223 real images and 7,076 spoof images, finding in 8,299 original training samples; the development subset includes 405 real and 2,543 spoof samples, which is mainly why what's more, meanwhile, the evaluation subset is often including 314 real and 7,266 spoof samples, in real-world scenarios.

The training data is also handled before being supplied to the deep learning model for the LiveGuard AI implementation, to a significant degree; as perhaps, the real class includes greatly fewer samples than the spoof class, extra augmentation is generally used to the real training images by the project, in real-world scenarios. The project documentation reports roughly 4,892 real training images after augmentation, compared with 7,076 spoof images, making around 11,968 sampled; horizontal flipping, small-angle rotation, brightness variation, and slight blurring are used to create extra training examples, in many respects; indeed, how about this, the project preprocessing pipeline converts the input images to RGB format and resizes them to a fixed resolution of **224×224 pixels**, on the whole.

Figure 1: Representative REAL facial image	*Figure 2: Representative SPOOF facial image*
	

B. Data Splits

Three principal subsets, namely training and development are being followed by the original LCC-FASD dataset, **in many respects; more accurately**, in the LiveGuard AI implementation, these subsets are mapped to the associateding train and val, especially regarding test directories during the preprocessing, to a significant degree. As, well, the evaluation subset is reserved for final performance assessment; the training subset is used for learning the factors of the TransferResNet-18 liveness classifier, the development subset is used for validation during the model training, broadly speaking.

Split	REAL	SPOOF	TOTAL
Training	1,223	7,076	8,299
Development / Validation	405	2,543	2,948
Evaluation / Testing	314	7,266	7,580
Total	**1,942**	**16,885**	**18,827**

Table I: Distribution of REAL and SPOOF samples across the LCC-FASD dataset splits.

The original training distribution is, more accurately, strongly skewed toward the spoofed presentations, with spoof samples representing the majority of the available training data, in many respects; to lower the effect of this imbalance, LiveGuard AI targetedly is increasing on the REAL training class and employs class-aware sampling during the model

training, in real-world scenarios. The TransferResNet-18 classifier is, to be sure, being therefore offered with a more balanced learning opportunity by the resulting training pipeline while keeping the original development and evaluation samples for unbiased model assessment, broadly speaking.

C. Data Preprocessing

The facial images are generally passed through a standardized preprocessing pipeline to get a clear input representation before being supplied to the liveness detection network, in real-world scenarios. They're first converted into the needed RGB format and resized to **224×224 pixels**, matching the spatial input dimensions anticipated by the ResNet-18-based classification model; this fixed resolution is allowing images starting from varied capture conditions to be ran employing the same tensor dimensions during the both training and inference, to a significant degree.

The preprocessing stage also is using the transformations needed by the deep learning framework to convert the handled images into the model-compatible tensors, on the whole; and so, during the training, the facial samples are subjected to controlled image transformations to introduce change in appearance and positioning, in actual application. The preprocessing pipeline is, in effect, designed to preserve the vital facial qualities needed for distinguishing genuine presentations from spoofed samples while lessening unnecessary departure caused by image acquisition conditions, broadly speaking.

D. Data Augmentation

Data augmentation is currently employed during the training to increase the visual diversity of the facial samples and lower the possibility of overfitting, broadly speaking. Changed versions of chosen samples employing several stochastic transformations are, by design, made by the training pipeline in its place of resting only on the original images maybe, on the whole. And the augmentation operations employed in the LiveGuard AI training pipeline include:

1. **Random Horizontal Flip, covering** Facial images may, to some extent, be mirrored horizontally during the training to introduce departure in the orientation of facial presentations while keeping the vital facial arrangement needed for liveness grouping, broadly speaking.
2. **Random Rotation -** Small rotational changes are, more accurately, mainly introduced to represent minor differences in the orientation of a user's face relative to the camera, which is usually helping the classifier become less dependent on a particular facial alignment, on the whole.
3. **Brightness and Appearance Variation, covering** Controlled changes in image appearance are usually employed to expose the model to differences in illumination and image acquisition conditions, in actual application; such change is, in practice, helpful, because genuine and spoofed facial samples may (to some degree) be embodied under the separate lighting environments, on the whole.

IV. PROPOSED METHODOLOGY

a. Model Architecture

The Live Guard AI system is mainly following a two-stage biometric processing approach in which facial presentation authenticity is currently found before identity verification, in actual application; in many respects. As well, so, ResNet-18 is a convolutional neural network that's using residual connections to allow information and gradients to spread basically, through the network; this architecture lightweightly is, in effect, tackling for learning visual credits associated with genuine and spoofed facial presentations, in real-world scenarios; the liveness classification component is, broadly speaking, typically applied using a transfer learning-based ResNet-18 network, on the whole.

The final feature representation made by ResNet-18 is covering 512 traits and is often passed, through a dropout layer followed by a linear layer creating two output values relateding to the REAL and SPOOF classes, in actual application; indeed, the original fully connected classification layer of ResNet-18 is replaced with a binary classification head, in many respects.

$$\textbf{Classifier} = \text{Dropout}(0.3) \rightarrow \text{FC}(512, 2)$$

where FC(512, 2) represents a fully connected layer that maps the 512-dimensional feature vector to two class scores, while Dropout(0.3) randomly hides a proportion of the credits during the training to reduce overfitting, in many respects.

The complete liveness classification process can be recapped as:

Input Face Region → Resize (224×224) → ResNet-18 Feature Extraction → Dropout → Fully Connected Layer → REAL / SPOOF

Class scores for the two possible presentation categories are being created by the trained model, in many respects; more accurately, during the inference, the class with the highest output score is normally picked as the liveness prediction, broadly speaking; what's more, what's more, the associateding confidence value is, broadly speaking, generally returned by the inference pipeline, on the whole.

The made representation is inspected with previously stored facial embeddings employing Euclidean distance; this model is using the OpenFace neural network model to convert a detected facial region into a 128-dimensional facial embedding, to a significant degree, and more, to the liveness model, LiveGuard AI includes a separate face recognition component for identity verification, to a significant degree; indeed, after ages, the authentication architecture doesn't rest solely on facial similarity; it first is inspecting whether the offered face is genuine and later finds out whether the live face corresponds to a registered identity, to a significant degree.

Face Detection → Liveness Classification → Identity Verification

The separation is generally allowing the framework to tackle two varied security questions - whether the introduced facial image represents a genuine person, in many respects; indeed, as, well, what's more, whether that person is typically matching a registered identity, broadly speaking.

b. Training Strategy

The liveness classification network is, in effect, generally trained employing the handled LCC-FASD dataset; in actual application; as well, so, the training pipeline is combining transfer learning, data augmentation, class-aware sampling, weighted loss, and label smoothing to raise learning under the major imbalance between genuine and spoofed facial samples, in actual application.

Optimizer:

The Adam optimizer is used to update the trainable parameters of the ResNet-18 model. The implementation uses an initial learning rate of **0.001**. Adam adapts the parameter update according to estimates of the first and second moments of the gradients, making it suitable for training the transfer-learning model while adapting the network to the facial anti-spoofing task.

Learning Rate Scheduling:

A Reduce LR On Plateau scheduler is, indeed, covered into the training process; and thus, try this instead of lessening the learning rate at predetermined epochs, the scheduler is monitoring validation loss and lowers the learning rate when advancement stops for a stated period, in actual application. In the setup, the learning rate is shrank by a factor of **0.5** after the validation loss crashes to raise for **3 consecutive epochs**, which is allowing the tuning process to continue with smaller parameter updates when the model methods a stable region of the loss landscape, on the whole.

Loss@ Function:

The training objective uses weighted cross-entropy loss with label smoothing, and plus, the class weights are found from the distribution of the training samples using:

$$w_k = N / (K \times n_k)$$

where **N** represents the total number of training samples, **K** represents the number of classes, in actual application; in many respects, and so, what's more, n_k means the number of samples belonging to class *k*. The resulting weights are currently supplied to the Cross Entropy Loss function so that differences associated with the less embodied class receive greater importance, to a significant degree.

During the training, label smoothing with a factor of **0.1** is, indeed, also employed, broadly speaking. Even as, instead of allowing the target distribution to hand out complete probability to a single class, a small portion of the target probability is, in practice, normally distributed, across the available classes; this lowers excessive confidence in individual predictions and can raise the robustness of the learned classifier, in many respects.

Weighted@ Sampling:

The training Data Loader also is employing a Weighted Random Sampler; sampling with replacement allows minority-class examples to appear more often during an epoch; this process is helping prevent the classifier from being dominated by the mostly larger SPOOF category, in real-world scenarios. Each training sample receives a sampling probability based on the weight of its matching class, broadly speaking.

The training configuration employed by the LiveGuard implementation is recapped below.

Hyperparameter	Configuration
Backbone	ResNet-18
Learning Approach	Transfer Learning
Optimizer	Adam
Initial Learning Rate	0.001
Learning Rate Scheduler	ReduceLROnPlateau
Scheduler Factor	0.5
Scheduler Patience	3 epochs
Batch Size	32
Maximum Epochs	30
Early Stopping Patience	7 epochs
Label Smoothing	0.1
Loss Function	Weighted Cross-Entropy
Sampling Method	WeightedRandomSampler

Input Resolution	224×224
Number of Classes	2
Classes	REAL, SPOOF

Table II: Training configuration used for the LiveGuard AI liveness classifier.

Training is monitored employing the validation set after each epoch, so validation loss and validation accuracy are recorded, throughout the training process, on the whole; when the validation loss sharpens, the relateding model parameters are kept as the best model state, in actual application; training is stopped early, which is preventing unnecessary training after the model has stopped showing meaningful validation improvement, if the validation loss crashes to raise for seven consecutive epochs, in many respects.

c. Face Detection and Liveness Analysis

The primary face detector is mainly based on the OpenCV Caffe implementation of the Single Shot Detector (SSD), on the whole. LiveGuard AI spots the facial region, within the input frame before doing liveness category, in actual application; the detector processes the input frame employing a 300×300 detection representation and returns candidate face bounding boxes together with confidence values, on the whole.

When the needed SSD model files are unavailable, a Haar Cascade fallback for face detection is being offered by the code, in actual application; only detections fulfilling the arranged confidence threshold are considered for next processing; and so, before the matching facial region is pulled, the detected coordinates are clipped to the image boundaries, in many respects.

The pulled face region is converted from the OpenCV BGR representation to RGB and reshaped into a PIL image before being passed to the liveness network, on the whole. The facial region is normally resized to **224×224 pixels**, converted into a tensor, and normalized employing the ImageNet channel statistics:

Mean = [0.485, 0.456, 0.406]

Standard Deviation = [0.229, 0.224, 0.225]

The normalized tensor is then supplied to the trained ResNet-18 classifier, really, the output is, in practice, converted into the relateding liveness label, either **REAL** or **SPOOF**, together with its prediction confidence, in real-world scenarios.

The inference sequence can therefore be articulated as:

Input Frame → Face Detection → Face Cropping → RGB Conversion → 224×224 Resize → Normalization → ResNet-18 → REAL/SPOOF Prediction

The arrangement is, of course, making sure that the liveness classifier operates on the facial region rather than on the complete image frame, in actual application.

d. Identity Verification Using Facial Embeddings

Liveness category figures out whether the introduced facial sample appears genuine; in turn, that said, it doesn't establish the identity of the person, broadly speaking; on the whole, and as well, so, LiveGuard AI therefore is running a separate facial recognition step after liveness inquiry, in real-world scenarios.

The OpenFace **nn4.small2.v1** model is being employed by this model to make a compact **128-dimensional facial embedding** from the detected face, in actual application; more accurately, during the registration, facial embeddings associated with permitted users are stored in the framework's face database, on the whole; during the authentication, a new embedding is, of course, made from the offered facial region and compared with the stored representations, in actual application.

Euclidean distance is often employed for the measurement of the difference between two facial embeddings:

$$\mathbf{d(x, y) = \sqrt{\Sigma_i (x_i - y_i)^2}}$$

As, well, y represents a stored reference embedding, so and where *x* represents the embedding created from the current facial presentation, on *the* whole.

The two facial representations are, put simply, shown by greater similarity between A smaller distance, really, a configurable distance threshold is being employed by the setup when finding out whether a detected face corresponds to a registered identity, in actual application; what's more, for claimed-user verification, the framework can also compare the current embedding against many stored embeddings for that user and use the average of the closest matching representations, in many respects.

The authentication decision therefore needs more than simply spotting a familiar face; in turn, the framework checks the presentation in two stages:

Liveness Condition + Identity Match → Authentication Decision

The design lowers the risk of accepting a photograph or showed representation merely, because it resembles a registered person's face, broadly speaking.

e. Multi-Stage Authentication Decision

The outputs of the liveness classifier and the face recognition module are being combined by the final authentication workflow, in many respects; when a face is detected, the framework first is finding its liveness status, in many respects. If the presentation is classified as spoofed, the authentication process doesn't go ahead as a successful identity verification, in

many respects; more accurately, if the presentation is classified as genuine, the system goes ahead to compare the created facial embedding against the registered identity data, on the whole.

The complete decision flow can be embodied as:

Protection at both the presentation and identity levels is currently being offered by this layered design, in real-world scenarios, really, presentation attacks are being tackled by the liveness component; while the facial embedding component tackles identity verification, in actual application, and as, a finding, the input is currently being needed by successful authentication to meet both conditions rather than counting on standard face recognition alone, on the whole.

f. Security-Aware Processing

The LiveGuard AI backend is typically employed using FastAPI and includes proved genuine API access, password protection, input validation, session management, and rate-limiting controls, in actual application; more accurately, the machine learning pipeline is usually blended with application-level security arrangements to protect the authentication service beyond the prediction model, in actual application.

Automated security notification capabilities for related security events are also offered by the application; these controls are intended to lower risks associated with unauthorized API access, invalid requests, repeated authentication attempts, and other application-level threats, on the whole.

The resulting architecture can be recapped as:

The foundation for the LiveGuard AI system is typically being offered by this combined method and makes possible the application to judge both the authenticity of a facial presentation and the identity associated with that presentation, in many respects.

V. RESULTS AND DISCUSSION

a. Classification Performance

The performance of the showed LiveGuard AI liveness detection model was, of course, found using the validation portion of the LCC-FASD dataset, broadly speaking; more accurately, the check is stressing on the ability of the TransferResNet-18 classifier to distinguish between genuine facial presentations and spoofed presentations, in actual application. As, the dataset includes a considerably larger number of spoof samples than genuine samples, precision alone isn't enough to sum up the behaviour of the classifier; more accurately, right, class-wise precision, recall, F1-score, and the confusion matrix are also considered, in actual application.

The project-reported validation accuracy is mainly around 97.01%; till then, the spoof rejection rate is roughly 98.4%; what's more, the validation set is covering **2,948 facial samples**, consisting of **405 REAL samples and 2,543 SPOOF samples**, in actual application**.**

Metric	Value
Validation Accuracy	97.01%
REAL Precision	90.13%
SPOOF Precision	98.08%
REAL Recall	87.90%
SPOOF Recall	98.47%
REAL F1-Score	0.89
SPOOF F1-Score	0.98
Macro F1-Score	0.94
Spoof Rejection Rate	98.46%

Table III: Validation performance of the LiveGuard AI liveness classifier.

The findings are, in practice, showing that the model does strongly in spotting spoofed facial presentations, broadly speaking. As, well, a recall of roughly **98.47%** is hit by the SPOOF class, meaning that the majority of spoof attempts present in the validation data are correctly detected, broadly speaking, really, the relateding spoof rejection rate is roughly **98.46%**, which is particularly vital for an anti-spoofing framework, because the primary security objective is to prevent artificial facial presentations from being accepted as genuine, in real-world scenarios.

The REAL class is generally showing comparatively lower recall at roughly **87.90%**, which is often suggesting that some genuine facial presentations are incorrectly classified as spoof, broadly speaking; such behaviour is typically reflected in the class imbalance of the dataset, where the number of spoof samples is greatly higher than the number of genuine samples, broadly speaking, and yet, a precision of roughly **90.13%** is normally hit by the REAL class, showing that when the model forecasts a presentation as genuine, the prediction is generally homogeneous within the found validation data, in many respects.

A more balanced view of classifier performance is being offered by the macro F1-score of roughly **0.94**, because it equally imports to both classes rather than allowing the larger SPOOF class to dominate the overall metric; these findings

are showing that the transfer-learning method can learn helpful visual credits for distinguishing genuine facial presentations from spoofed ones, in many respects.

b. Confusion Matrix

The class-level behaviour of the liveness classifier is, by design, further pored over employing the confusion matrix received from the validation data, to a significant degree.

Actual Class	Predicted REAL	Predicted SPOOF
REAL	356	49
SPOOF	39	2,504

Table IV: Confusion matrix of the LiveGuard AI liveness classifier on the validation set.

The model correctly spots **356 of the 405 REAL samples**; in turn, and in the meantime, **49 genuine samples** are incorrectly classified as spoof, in actual application; for the SPOOF category, **2,504 of the 2,543 samples** are, in effect, correctly spotted, with only **39 spoof samples** incorrectly classified as REAL, broadly speaking.

The resulting overall classification accuracy is roughly:

Accuracy = (356 + 2,504) / 2,948 × 100 ≈ 97.01%

The varied effects of the two types of classification error are also shown by the confusion matrix, on the whole, and a genuine sample incorrectly labelled as SPOOF represents a false rejection, which may, in practice, prevent an permitted user from completing authentication, in many respects. Separate from that, a spoof sample incorrectly classified as REAL represents a false acceptance and is often more vital from a security perspective, because it can allow a presentation attack to go ahead to the identity verification stage, in many respects. Only **39 spoof samples** are incorrectly .

c. Liveness Detection Performance

Live Guard AI finds the authenticity of the offered facial sample before completing identity verification in its place than a standard face recognition system; this difference is noticeable, because a face recognition model can detect the identity meant by a photograph or showed facial image without figuring out whether the person is physically present, on the whole.

Input Face → Liveness Analysis → REAL/SPOOF Decision

The try before successfully checking can (in actual operation) be rejected by the authentication process, if the detected presentation is classified as SPOOF, in real-world scenarios; if the presentation is classified as REAL, the framework is continuing to the facial embedding and identity-matching stage, broadly speaking.

The sequential design is, in practice, vital, because identity similarity and presentation authenticity represent mixed traits, in real-world scenarios; a facial image may have high similarity to a registered person's embedding while still being an artificial reproduction, to a significant degree; indeed, as, well, liveGuard AI is shrinking the dependence on facial recognition alone by separating these two decisions, on the whole.

d. Identity Verification Performance

The model converts the detected facial region into a **128-dimensional representation** and checks it with the registered facial representation employing Euclidean distance, to a significant degree; identity verification is done by LiveGuard AI using OpenFace facial embeddings after a facial presentation fulfills the liveness condition, in real-world scenarios.

The identity verification stage is, more accurately, therefore inspected independently from the binary liveness classifier, in many respects; indeed, the distance between the current embedding and the stored reference representation is finding out whether the detected face is sufficiently comparable to the registered identity, in real-world scenarios.

The complete authentication decision can be embodied as:

Authentication = Liveness Verification + Identity Verification

A successful finding needs both conditions to be met, so a genuine presentation belonging to an unauthorized person should, of course, therefore fail the identity comparison, in many respects; till later, a spoofed presentation should be rejected by the liveness stage before it can, in practice, be treated as a successful authentication, to a significant degree.

This layered strategy is usually offering a stronger security model than a single-stage face recognition system, because it's finding both **presentation authenticity** and **identity similarity**, in real-world scenarios.

e. Multi-Frame Authentication Analysis

The model also is lowering dependence on a single video-frame prediction by employing a multi-frame decision mechanism, broadly speaking; on the whole, and more, so, consecutive predictions are being checked by LiveGuard AI and uses a **2-out-of-3 voting strategy** for the final authentication decision, broadly speaking.

The decision rule can be said clearly as:

Authentication Accepted if ≥ 2 of 3 predictions satisfy the required conditions

This process is helping lower the impact of an separated incorrect prediction; needing agreement, across several frames is offering an additional temporal consistency check, on the whole; for one, short-term changes in illumination, facial movement, camera noise, or an unstable model prediction may change one frame without embodying the actual state of the user, to a significant degree.

The multi-frame framework is particularly helpful in a real-time authentication environment, because the system receives a sequence of facial notes rather than a single static image, broadly speaking; in actual application; as well, thus, as, a finding, the final decision is usually based on short-term consistency instead of one instantaneous prediction, broadly speaking.

f. Security and Reliability Testing

The machine learning performance was complemented by testing of the application-level security and reliability mechanisms, in actual application; indeed, what the LiveGuard AI project includes is a security and reliability test suite covering the major authentication and API protection components, to a significant degree.

The project reports successful completion of **15 security and reliability tests**; these tests were employed to check vital application behaviours covering authentication-related operations, API protection, input handling, and security controls used around the machine learning pipeline, in real-world scenarios.

Seeing that a high-performing liveness model alone doesn't guarantee a secure authentication application, this testing is essential, on the whole; indeed, an attacker may try to use the surrounding API, an application programming interface, an application programming interface, an application programming interface, an application programming interface, an application programming interface, an application programming interface, authentication mechanism, session handling, or request-processing layer rather than directly attacking the neural network, broadly speaking. Model-level protection with application-level controls such as proved genuine API access, password protection, input validation, rate limiting, session management, and automated security alerts is therefore combined by LiveGuard AI, to a significant degree.

The findings are, by design, mainly showing that the suggested framework should likely be considered as a complete security-oriented authentication workflow rather than only a face classification model, in actual application.

g. Training and Validation Behaviour

The validation metrics are monitored during the model development to figure out whether the network continues to raise on previously unseen facial samples, in real-world scenarios; in many respects, and as well, so, training and validation performance is being recorded, across the successive epochs by the training process, on the whole.

The use of transfer learning with ResNet-18 is allowing the model to begin from broad visual representations learned from ImageNet and later fit them to the credits of genuine and spoofed facial presentations, in actual application; in actual application, and so, data augmentation and class-aware sampling further expose the model to changes in facial appearance and help lower the effect of the original class imbalance, in many respects.

h. Overall Discussion

The experimental results are showing that LiveGuard AI basically combines deep learning-based liveness detection and facial identity verification, in real-world scenarios; what's more, the reported validation accuracy of roughly **97.01%** and spoof rejection rate of roughly **98.46%** are showing strong performance in distinguishing spoofed presentations, within the found validation set, in actual application.

The confusion matrix further is, in effect, suggesting that the classifier correctly spots **2,504 out of 2,543 spoof samples**, in **many respects; to a significant degree. As well, so, what's more, the fin**ding macro F1-score of roughly **0.94** is, in practice, showing that the model upholds major performance, across the both classes despite the imbalance between genuine and spoof samples, in many respects; and so, till then, 356 out of 405 genuine samples are correctly saw as REAL, in actual application.

The findings should, though, be took as, within the scope of the found LCC-FASD validation data, on the whole; high validation performance doesn't automatically establish that the model will reach same performance against every possible real-world presentation attack, camera configuration, lighting condition, or unseen spoofing protocol, to a significant degree. Broader evaluation using extra datasets and previously unseen attack scenarios would, by design, therefore be needed before strongerly claiming about generalization, broadly speaking.

VI. SECURITY AND ATTACK ANALYSIS / MODEL INTERPRETATION

a. Liveness Detection Analysis

The liveness detection component of LiveGuard AI also lays, before facial identity verification is did; this model is assessing the detected facial region and classifies the introduced sample as either **REAL** or **SPOOF** employing the trained TransferResNet-18 model; this separation is allowing the authentication pipeline to distinguish between a genuine facial presentation and reproduced facial content such as photographs, showed images, or replayed facial samples, in many respects.

The model's predictions are, in practice, found using the classification results received during the validation, on the whole; the reported findings are normally indicating that the classifier can (to some degree) spot the majority of spoofed presentations while also knowing genuine facial samples, broadly speaking; in actual application. As well, so, the confusion matrix is, as a concrete matter, offering further idea into the model's behaviour by showing the number of correctly and incorrectly classified REAL and SPOOF samples, on the whole.

The gathered validation results are, of course, showing that most spoof samples are correctly rejected, showing the usefulness of the liveness classifier as a protective stage before identity verification, in real-world scenarios; particular attention is generally given to spoof samples incorrectly classified as REAL, because such predictions represent potential false-acceptance situations for security-oriented authentication, to a significant degree.

The liveness examination therefore is suggesting that the framework doesn't rest solely on facial similarity, on the whole, but still, instead, it first is finding whether the facial presentation itself is genuine and allows the authentication process to continue only when the liveness need is satisfied, on the whole.

b. Identity Verification Analysis

The model is making a 128-dimensional facial representation employing the OpenFace model and checks the created representation with the registered facial representation associated with the user, in actual application; indeed, identity verification is run by LiveGuard AI using facial embeddings after a facial presentation passes the liveness stage, on the whole.

The arranged verification threshold is used to figure out whether the detected face corresponds sufficiently to the enrolled user, on the whole; more accurately, greater similarity between A smaller distance is, in effect, showing the two facial representations, on the whole; and so, to date, a larger distance is, more accurately, suggesting that the input face is usually less akin to the stored identity representation, to a significant degree. Euclidean distance is generally being employed by the comparison.

Presentation authenticity is generally separated from identity matching by this two-stage design; and so, as, well, the facial identity must fulfill the verification criterion, broadly speaking; and so, as a finding, successful authentication is needing both conditions to be met, namely the presentation must be classified as genuine, in many respects. In years gone by, before the identity decision is accepted, the **presentation can** (to a meaningful extent) be **rejected by the liveness classifier, broadly speaking; a** person introducing a photograph of a registered user may create a facial representation that's visually comparable to the stored identity, broadly speaking.

c. Multi-Frame Authentication Analysis

To lessen the impact of an separated incorrect prediction, LiveGuard AI includes a multi-frame decision mechanism, on the whole; indeed, single-frame predictions can occasionally be changed by short-term changes in lighting, facial position, camera quality, or model uncertainty, in actual application.

Three consecutive predictions are currently found by the framework and employs a **2-out-of-3 voting strategy**, on the whole; authentication is normally accepted when at least two of the three predictions meet the needed liveness and identity conditions, in actual application; more accurately, before finally deciding maybe, this method is offering an additional temporal consistency check, in real-world scenarios.

The multi-frame process is particularly good in concrete camera-based authentication, because consecutive frames generally are covering closely related visual information, in many respects; more accurately, try this instead, to a significant degree, and yet, allowing one uncertain frame to figure out the complete authentication result, the system studies multiple notes before finally deciding, broadly speaking.

VII. WEB APPLICATION AND DEPLOYMENT

The web interface almost environs through the which facial input can (in actual operation) be did, liveness can be found, on the whole; indeed, LiveGuard AI is employed as an interactive web-based biometric authentication application that's connecting the machine learning pipeline with application-level authentication and security services, on the whole. What's more, identity verification can (in practice) be ran without needing users to interact directly with the coring deep learning implementation, to a significant degree.

The application is mainly supported by a **FastAPI-based backend**, which responsibly lays responsible for handling authentication-related requests and connecting the web application with the machine learning components, broadly speaking, really, authentication-related operations are being managed by the backend while employing security controls such as password protection, input validation, session management, and request-rate control, broadly speaking.

The major processing flow of the application can be recapped as follows:

1. **User Authentication, covering** The user first is, by design, accessing the protected application and offers the needed authentication credentials, in many respects.
2. **Facial Input, like** A facial image or camera frame is often supplied to the authentication system, in real-world scenarios.
3. **Face Detection, namely** The OpenCV-based face detection component spots the facial region present in the input, to a significant degree.
4. **Liveness Analysis, including** The pulled face is handled by the TransferResNet-18 model and classified as REAL or SPOOF, on the whole.
5. **Identity Verification, like** If the presentation fulfills the liveness condition, an OpenFace 128-dimensional embedding is currently made and inspected with the registered facial representation, in real-world scenarios.
6. **Temporal Decision** - Several consecutive predictions are, of course, mainly found employing the 2-out-of-3 voting mechanism, in actual application.
7. **Authentication Result, namely** The framework is generating the final authentication decision based on the combined liveness and identity verification findings, in many respects.

Additional Security Features

What the web application includes is several supporting processes to raise the security and reliability of the complete framework, to a significant degree.

- **Supported real API Access, like** API endpoints are, of course, protected so that authentication-related operations aren't exposed as unrestricted public requests, in many respects.
- **Password Protection:** User credentials are typically protected, through the password-based authentication rather than hinging only on facial verification, to a significant degree.
- **Input Validation -** Incoming requests are, by design, inspected before processing to lower the possibility of invalid or malformed input reaching the application logic, to a significant degree.
- **Rate Limiting, namely** Repeated requests can, more accurately, maybe be controlled to lessen excessive authentication attempts and potential abuse of the API, broadly speaking.
- **Session Management, covering** Authentication state is normally managed, through the application's session mechanism to support controlled access to protected operations, in many respects.
- **Automated notifications can be triggered by Automated Security Notifications, covering** Security-related events, offering an extra process for spotting potentially suspicious activity, in actual application.

Three consecutive remarks are considered, with at least two successful findings needed for acceptance; in turn, Multi-Frame Verification, covering Final authentication isn't determined from a single prediction, in many respects.

The resulting web application therefore acts as the interface layer for a complete biometric authentication pipeline rather than functioning as a uncomplicated face-recognition demo, and is connecting the trained liveness model, facial identity verification, temporal decision mechanism, and application security controls into a single process, broadly speaking.

VIII. CONCLUSION AND FUTURE WORK

This paper offered **LiveGuard AI**, a deep learning-based biometric authentication system designed to sharpen the security of standard face recognition by covering facial liveness detection before identity verification, in many respects. The suggested framework is often combining face detection, TransferResNet-18-based liveness category, OpenFace facial embeddings, Euclidean-distance identity comparison, and multi-frame decision-making into a single authentication workflow; this setup is generally trained employing the LCC-FASD face anti-spoofing dataset and is generally designed to distinguish genuine facial presentations from spoofed presentations such as reproduced or displayed facial images, in many respects.

The experimental check is, by design, showing that the formulated liveness classifier can (in actual operation) basically separate REAL and SPOOF facial presentations, in many respects; give or take 97.01% accuracy was hit by the project-reported validation results, with roughly **98.4% spoof rejection performance**, in many respects. The confusion-matrix examination further is suggesting that the majority of spoof presentations are correctly spotted, on the whole; indeed, already, genuine facial samples are also knew with a high level of precision; these findings are, in effect, typically hinting that liveness detection can importantly lay that isn't available in standard face-recognition architectures operating without presentation-attack detection, in many respects.

This model further is strengthening authentication by combining liveness detection with facial identity verification, to a significant degree; more accurately, a 128-dimensional representation of the detected face, which is inspected against the registered identity employing Euclidean distance is being created by OpenFace, to a significant degree. On whether a face resembles a stored identity, authentication is therefore not based solely; the liveness need must first be met by the offered sample and later pass the identity verification stage, and the 2-out-of-3 multi-frame voting mechanism decreases the impact of an separated incorrect prediction and greaterly consists during the authentication, in real-world scenarios.

Beyond the machine learning components, LiveGuard AI includes application-level security processes covering supported real API access, password protection, input validation, rate limiting, session management, and automated security notifications, in many respects; broadly speaking. As well, so, security and reliability testing was also ran as part of the project, with 15 reported tests employed to gauge vital dimensions of the used authentication workflow, to a significant degree; indeed, a more complete protection mechanism than counting on a facial recognition model alone is typically being offered by the combination of machine learning and standard application security, in real-world scenarios.

Several directions can be explored in future research:

1. **Expanded Presentation-Attack Detection:** The liveness model can be trained and evaluated using a broader collection of presentation attacks, covering separate printed-photo qualities, mobile and monitor shows, replayed videos, and other attack conditions; this would help assess the robustness of the framework against previously unseen spoofing rules, broadly speaking.
2. **Larger and More Diverse Datasets:** Training with additional face anti-spoofing datasets containing greater variation in subjects, illumination, cameras, environments, and attack instruments could improve the generalization capability of the liveness classifier.
3. **Advanced Temporal Analysis:** The existing 2-out-of-3 voting mechanism could be generalized employing temporal deep learning approaches such as recurrent networks, temporal convolution, or transformer-based video analysis. Changes across multiple frames could be tried out by such methods rather than treating each prediction independently.

4. **Improved Identity Verification:**Future versions could study more recent and solid face-recognition models and adaptive identity thresholds; this could sharpen verification performance under the deviations in pose, illumination, camera quality, and facial appearance, in real-world scenarios.
5. **Advanced Attack Detection:**protection against emerging attacks such as deepfake facial videos, synthetic face generation, and complicated replay attacks could be checked out closely by Future research. Extra physiological or behavioral cues could, more accurately, also be worked in to strengthen presentation-attack detection, broadly speaking.
6. **Continuous Authentication:**Instead of doing authentication only during an initial login event, the framework could (to a meaningful extent) be broadened to continuously monitor user presence during a session, on the whole; indeed, periodic or event-based verification could help prevent unauthorized use after a legitimate user has already logged in, to a significant degree.
7. **Privacy-Preserving Biometric Storage:**Future implementations could investigate encrypted facial templates, secure template storage, and privacy-preserving biometric comparison techniques. This would reduce the security risks associated with storing facial representations on application servers.

REFERENCES

1. A.Grishkin and A. Timoshenko, " “Large Crowdcollected Facial Anti-Spoofing Dataset,” in Proc.," 10th International Conference on Computer Science and Information Technology (CSIT),, 2019.
2. X. Z. S. R. a. J. S. K. He, " “Deep Residual Learning for Image Recognition,” in Proc.," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA,, p. 770–778., 2016.
3. K. a. J. P. F. Schroff, "“FaceNet: A Unified Embedding for Face Recognition and Clustering,” in Proc.," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, , p. 815–823, 2015.
4. M. Y. M. R. a. L. W. Y. Taigman, "“DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” in Proc.," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, , p. 1701–1708., 2014.
5. P. R. a. L.-P. M. T. Baltrušaitis, "“OpenFace: An Open Source Facial Behavior Analysis Toolkit,” in Proc.," IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA,, p. 1–10, 2016.
6. George and S. Marcel, " “Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection,” in Proc.," IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), , 2019.
7. R. Ramachandra and C. Busch, "“Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey,”," ACM Computing Surveys,, 2017.
8. X. L. a. P. C. Y. R. Shao, "“Deep Convolutional Dynamic Texture Learning with Adaptive Channel-Discriminability for 3D Mask Face Anti-Spoofing,” in Proc.," IEEE International Joint Conference on Biometrics (IJCB),, 2017.
9. Y. Q. X. L. Z. W. C. Z. Z. L. a. G. Z. Z. Yu, "“Deep Learning for Face Anti-Spoofing: A Survey,”," IEEE Transactions on Pattern Analysis and Machine Intelligence,, 2023.
10. J. a. X. L. Y. Liu, " “Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision,” in Proc.," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA,, 2018.
11. Y. L. a. X. L. A. Jourabloo, "“Face De-Spoofing: Anti-Spoofing via Noise Modeling,” in Proc.," European Conference on Computer Vision (ECCV), Munich, Germany, , 2018.
12. H. Y. J. L. a. Y. Z. Z. Wang, " “Face Anti-Spoofing Based on Domain Adaptation and Depth Estimation,” in Proc.," IEEE International Conference on Computer Vision Workshops (ICCVW), , 2019.
13. H. D. a. X. W. M. Liu, " “Face Presentation Attack Detection Using Deep Learning-Based Feature Representation,” in Proc.," IEEE International Conference on Image Processing (ICIP),, 2019.
14. P. a. T. M. T. Ojala, " “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns,”," IEEE Transactions on Pattern Analysis and Machine Intelligence,, p. 971–987, 2002.
15. S. N. a. S. Z. L. E. S. Marcel, "Handbook of Biometric Anti-Spoofing, 2nd ed.," Cham, Switzerland: Springer,, 2019.
16. M. A. Turk and A. P. Pentland, "“Eigenfaces for Recognition,”," Journal of Cognitive Neuroscience,, p. 71–86, 1991.
17. W. T. S. A. J. O. P. J. F. K. W. B. C. L. S. a. M. S. P. Jonathon Phillips, " “FRVT 2006 and ICE 2006 Large-Scale Results,”," IEEE Transactions on Pattern Analysis and Machine Intelligence,, p. 831–846, 2010.
18. M. D. Zeiler and R. Fergus, "“Visualizing and Understanding Convolutional Networks,” in Proc.," European Conference on Computer Vision (ECCV), Zurich, Switzerland,, p. 818–833, 2014.