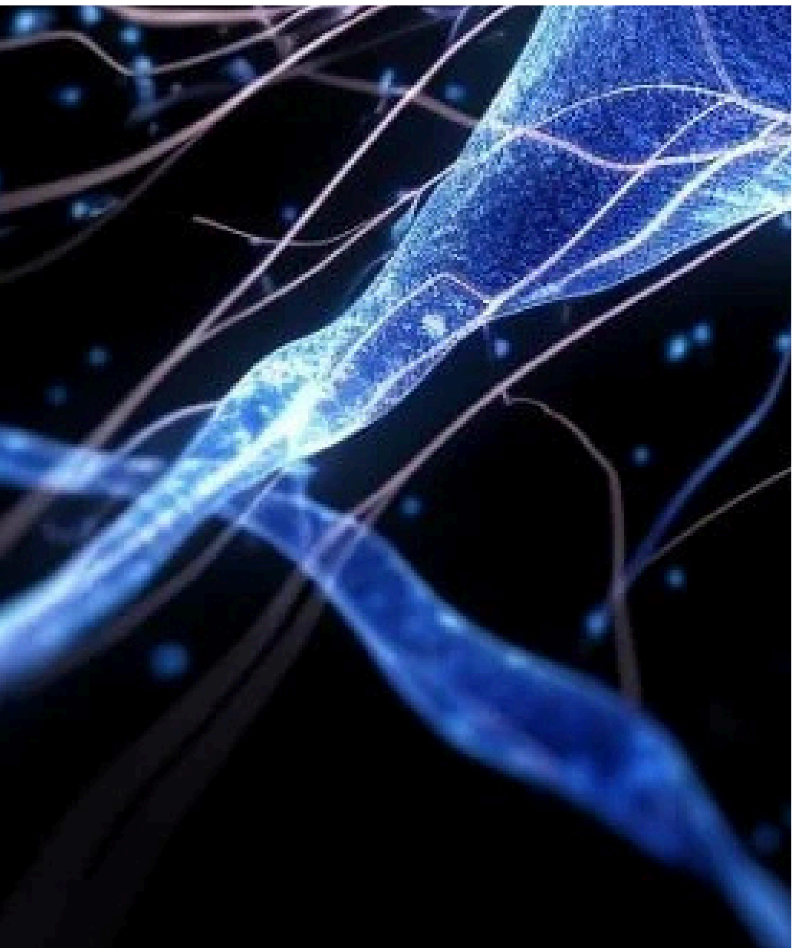
Swinburne University of Technology

FINAL PROJECT

ENHANCING FACE VERIFICATION WITH ATTENTION MECHANISMS, RPPG-BASED LIVENESS, AND EMOTION DETECTION



Lecturer
Mr. Minh Hoang

Name & Student ID
Nguyen Ngoc Lam Dan
104992198

I. Introduction

This project develops a real-time face recognition attendance system that combines face verification, anti-spoofing, and emotion recognition into a unified pipeline. Two embedding approaches, supervised learning and metric learning, are implemented and compared using ROC and AUC. A CNN-based anti-spoofing model enhanced with remote photoplethysmography - CHROM (rPPG-CHROM) [1] improves protection against printed and screen-based attacks, while a pretrained emotion classifier provides real-time expression feedback. All components are integrated into a Tkinter GUI that supports registration, live monitoring, and attendance logging. The system demonstrates how multiple machine learning modules can be combined to create a practical, deployable attendance solution.

II. Methodology

1. Overall System Architecture

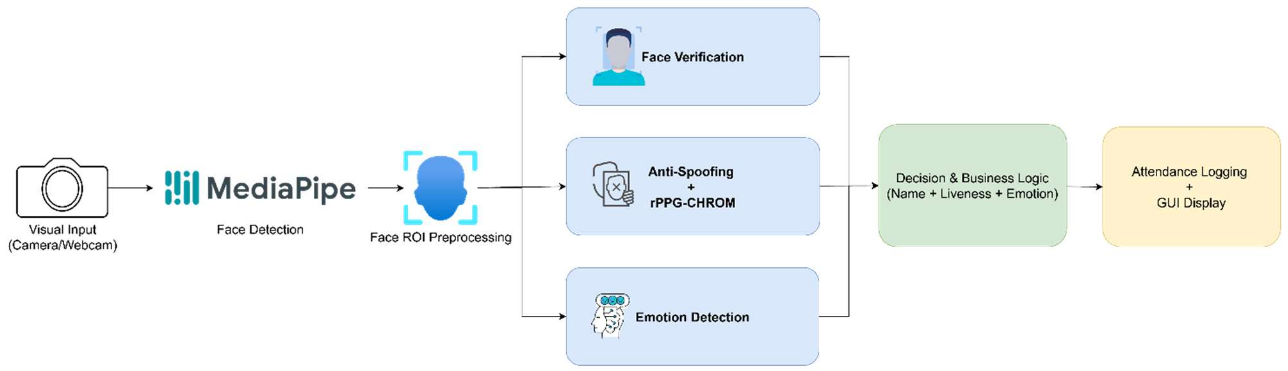


Figure 1. System Architecture

The proposed system follows a multi-stage architecture beginning with a live camera feed that continuously streams raw video frames. These frames are first processed by *MediaPipe* [2] Face Detection to identify facial regions with high precision and low computational overhead. The detected face region of interest (ROI) is then cropped, resized to uniform resolution, and normalized to ensure consistent input quality for all downstream neural networks. Once preprocessing is complete, the frame enters a parallel analysis stage in which three modules operate concurrently: face verification, anti-spoofing, and emotion recognition. The verification module extracts a 512-dimensional embedding vector from a hybrid EfficientNet-B4 backbone [3] enhanced with CBAM [4-6] and a Transformer block [7], and the resulting embedding is compared with stored templates using cosine or Euclidean distance. The anti-spoofing subsystem performs both image-based CNN spoof classification and physiological signal estimation using rPPG-CHROM to detect anomalies associated with printed or digital screen attacks. In parallel, an emotion classifier trained on FER2013 and RAF-DB predicts facial expressions. These outputs are fused by the decision logic module to determine the final identity, liveness status, and emotion label. The results are then displayed on a Tkinter GUI and simultaneously recorded into an attendance CSV file. This architecture enables real-time, robust, and secure attendance processing. The following section evaluates each component using quantitative metrics and visual analysis.

2. Datasets

Multiple datasets were used to train and evaluate the system. The classification dataset contained separate train, validation, and test partitions, each structured into identity-specific folders. This dataset trained the supervised embedding model and provided inputs for metric learning. For verification evaluation, the project used the provided *verification_data* folder along with *verification_pairs_val.txt*, which specifies pairs of images and their ground-truth labels for computing ROC curves. Anti-spoofing was trained using the *CelebA-Spoof* dataset, which includes diverse spoof artifacts such as printed photographs, digital screens, and replay attacks. Emotion recognition used FER2013 as the primary dataset, supplemented by RAF-DB for higher-quality expression samples.

3. Face Detection

Face detection is performed using *MediaPipe* due to its efficiency and robustness in real-time environments. *MediaPipe* returns bounding boxes for each detected face, which are subsequently used to extract the face ROI. This ROI undergoes cropping, resizing to 224×224 or 320×320 pixels, normalization, and optional alignment using facial landmarks. The resulting standardized face image ensures stable model performance regardless of lighting, pose, or image source.

4. Face Embedding Model

The core of the verification system is a hybrid embedding model built upon an EfficientNet-B4 backbone enhanced with Convolutional Block Attention Modules (CBAM) and a lightweight Vision Transformer block. Recent research has shown that hybrid CNN–Transformer architectures can leverage the complementary strengths of convolutional networks and self-attention mechanisms to improve visual representation quality. Works such as CoAtNet [8], Conformer [9], and the ViT–CNN Hybrid model [10] demonstrate that CNNs provide strong local inductive biases while transformers capture long-range global dependencies, resulting in superior generalization in recognition tasks. Motivated by these findings, EfficientNet-B4 is used to extract fine-grained spatial features from facial images, CBAM refines these features through channel and spatial attention, and the Transformer block models global context that CNN layers alone may overlook. After feature extraction, a global average pooling layer and a dense layer produce a 512-dimensional embedding, which is then L2-normalized to ensure a stable magnitude across samples. Two training paradigms were implemented.

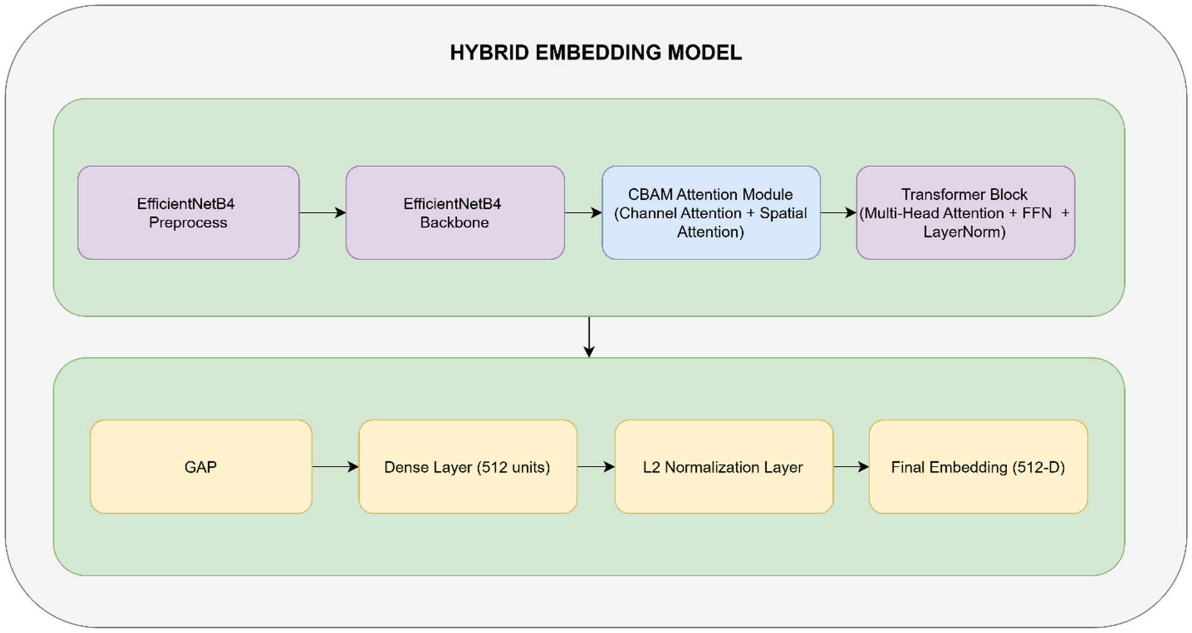

HYBRID EMBEDDING MODEL
EfficientNetB4 Preprocess
EfficientNetB4 Backbone
CBAM Attention Module (Channel Attention + Spatial Attention)
Transformer Block (Multi-Head Attention + FFN + LayerNorm)
GAP
Dense Layer (512 units)
L2 Normalization Layer
Final Embedding (512-D)

Figure 2. Hybrid Embedding Model Architecture

4.1. Supervised Classification Approach

In the supervised classification approach, the hybrid EfficientNet-B4 + CBAM + Tiny ViT backbone was trained as a multi-class classifier using categorical cross-entropy with label smoothing (ε = 0.1) and the Adam optimizer. Mixed precision (mixed_float16) was enabled to exploit GPU throughput while keeping the logits and loss in float32 for numerical stability. Training followed a three-phase curriculum that gradually increased input resolution and the number of trainable layers.

In Phase 1 (head warmup), the backbone was frozen and only the classification head was trained on 224×224 face crops with a batch size of 64. Data augmentation included random horizontal flips, small rotations (±5°), zooming (±10%), and contrast jitter, implemented through a Keras Sequential augmentation layer. The model was optimized with Adam at a learning rate of 1×10^{-3} for 10 epochs, with approximately 200 steps per epoch and 60 validation steps.

In Phase 2 (partial fine-tuning), the input resolution was increased to 256×256 and the top third of backbone layers were unfrozen based on their depth index. The batch size was reduced to 48 to fit the larger inputs, and the learning rate was lowered to 5×10^{-4} to avoid destabilizing previously learned weights. Training again ran for 10 epochs with around 220 training steps and 60 validation steps per epoch.

In Phase 3 (full fine-tuning), all backbone layers were unfrozen and the input resolution was further increased to 320×320 while reducing the batch size to 32. A smaller learning rate of 1×10^{-4} was used to refine the entire network without catastrophic forgetting. This phase was also trained for 10 epochs (≈240 steps per epoch, 60 validation steps), after which the classifier was saved and the 512-dimensional L2-normalized embedding layer was exported as the final supervised face embedding model.

4.2. Metric Learning Approach

For metric learning, the same hybrid backbone was reused as an embedding extractor and trained using a triplet loss objective. To avoid the numerical instabilities (NaN gradients) previously encountered under mixed precision, the global policy was set to full float32 during metric training. The training dataset was built from the classification identities using image_dataset_from_directory at a resolution of 256×256 and batch size 64, with integer labels indicating identity.

A custom batching pipeline constructed mini-batches containing multiple identities so that anchor, positive, and negative examples could be drawn within a single batch. The loss function was a batch-all triplet loss with a margin of 0.3 and semi-hard negative mining: all valid triplets were formed where negatives were harder than the positive but still violated the margin. The TripletModel wrapper subclassed keras.Model and overrode train_step to compute the embeddings, sanitize non-finite values with tf.where, and backpropagate the triplet loss through only the backbone parameters.

Optimization was performed using Adam with a learning rate of 3×10^{-5} and gradient clipping (clipnorm=1.0) to further guard against exploding gradients. The metric model was trained for 3 epochs with 150 steps per epoch, which was sufficient to reshape the embedding space while avoiding overfitting on the relatively small identity set. After training, the backbone was saved as a standalone metric embedding model and evaluated on the verification pairs using cosine similarity and negated Euclidean distance.

4.3. Similarity Metric

To verify whether two face embeddings represent the same person, cosine similarity and Euclidean distance were both evaluated. ROC curves were generated using the verification pairs, and optimal thresholds were selected based on the highest AUC and minimal false acceptance rate. Cosine similarity exhibited stable performance, with thresholds typically around 0.55, although the exact value depends on dataset variation and embedding distribution.

5. Anti-Spoofing Module

The CNN-based anti-spoof classifier was implemented by attaching a lightweight binary head on top of the same hybrid backbone used for face embeddings. The CelebA-Spoof dataset was preprocessed by parsing the official JSON label files, where attribute index 43 indicates live (0) versus spoof (1). To keep training feasible on a single T4 GPU, a maximum of 60,000 samples was drawn using random subsampling, and then split into training and validation sets with a 90/10 stratified split to preserve the live/spoof ratio.

Each image was loaded from disk, decoded with TensorFlow I/O, resized to 224×224, and normalized using the EfficientNet preprocessing function. The dataset was converted into a tf.data pipeline with shuffling, batched at 32 samples per step, and prefetched with AUTOTUNE for efficient streaming. The anti-spoof head consisted of a dense layer with 256 ReLU units, batch normalization, dropout (0.3), and a final sigmoid neuron producing a spoof probability.

Training followed a two-stage fine-tuning schedule in full float32 precision. In **Stage 1**, the backbone was frozen and only the anti-spoof head was trained for 6 epochs using Adam with a learning rate of 1×10^{-3}, binary cross-entropy loss, and metrics including accuracy and AUC. The model saw approximately 800 training steps and 150 validation steps per epoch.

In **Stage 2**, deeper layers starting from blocks 6 and 7 of the backbone were unfrozen using a name-based rule, enabling the network to adapt higher-level features to spoof cues. The model was then recompiled with a reduced learning rate of 1×10^{-5} and trained for an additional 6 epochs with the same steps-per-epoch configuration. ReduceLROnPlateau, EarlyStopping (monitoring validation AUC), and ModelCheckpoint callbacks were used in both stages to stabilize optimization and retain the best-performing weights.

6. Emotion Detection Module

Emotion recognition is handled by a dedicated classifier that reuses the hybrid EfficientNet-B4 + CBAM + Tiny ViT backbone as a feature extractor and adds a task-specific head. On top of the 512-dimensional embedding, the head applies dropout (0.4), a dense layer with 512 ReLU units, batch normalization, another dropout layer, and a final dense layer with 7-way softmax output corresponding to the standard emotion set {angry, disgust, fear, happy, neutral, sad, surprise}. The output layer is forced to float32 to maintain numerical stability under mixed-precision inference.

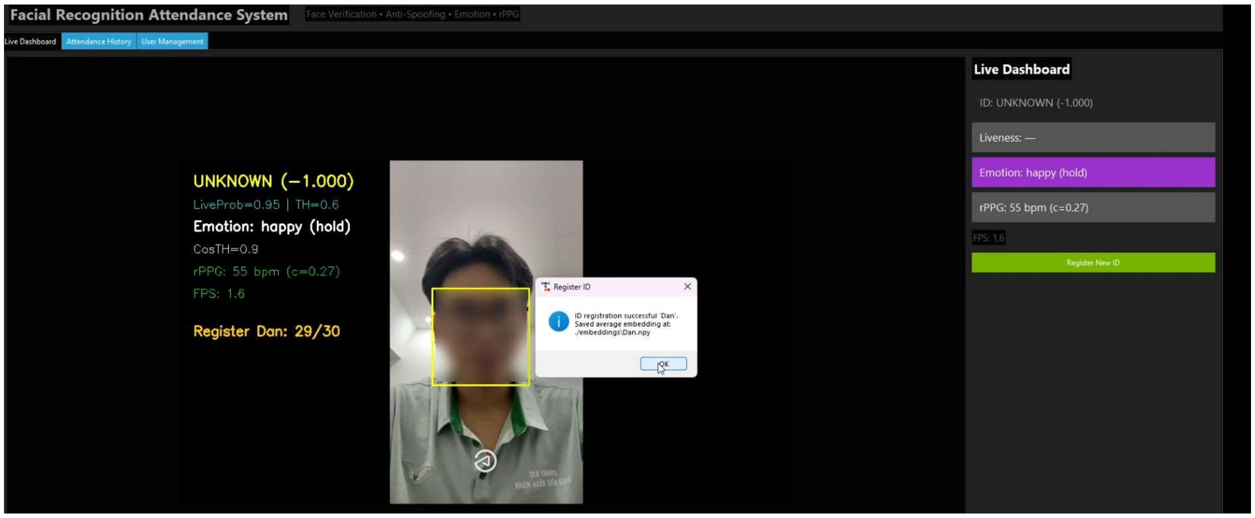
Training followed a two-stage transfer learning pipeline. In Stage 1 (FER2013 pre-training), the backbone weights were frozen and only the emotion head was optimized on the FER2013 training split, using 224×224 crops and a batch size of 32. The dataset was loaded via image_dataset_from_directory and augmented with random horizontal flips, brightness jitter, and contrast perturbations to increase robustness to pose and lighting variations. The model was trained for 10 epochs with approximately 300 steps per epoch and 80 validation steps, using Adam with a learning rate of 1×10^{-3} and categorical cross-entropy loss.

In Stage 2 (RAF-DB fine-tuning), higher-quality RAF-DB samples were used to refine the model. CSV annotation files were parsed to map RAF labels into the same 7-class scheme, and file paths were joined with the corresponding train/test image directories. Custom TensorFlow datasets were built via tf.data.Dataset.from_tensor_slices, shuffling the training set and resizing images to 224×224. The last blocks (block5, block6, block7) of the backbone were unfrozen to allow moderate adaptation to RAF-DB's distribution while keeping earlier layers stable. The model was then trained for another 10 epochs on the RAF train split with roughly 500 training steps and 100 validation steps per epoch, using Adam at a reduced learning rate of 1×10^{-5}.

After training, the best RAF-DB checkpoint (selected by highest validation accuracy) was reloaded and recompiled with Adam (1×10^{-4}) for evaluation. This two-stage training strategy explains the observed performance gap between FER2013 (46.63% accuracy) and RAF-DB (75.78% accuracy), as the model first learns coarse, noisy patterns from FER and then specializes on the cleaner RAF-DB distribution before being deployed in the real-time GUI.

7. Real-Time GUI (Tkinter)

A Tkinter-based graphical interface integrates all system modules into a functional application. The GUI includes multiple tabs for live monitoring, attendance history, and user management. Users can register new identities, capture embeddings, and manage existing templates. The live dashboard displays the camera feed with face bounding boxes, identity predictions, liveness results, and emotion labels. Attendance events, including timestamp, user ID, liveness status, and emotion are automatically appended to a CSV file. Additional optimizations such as FPS smoothing and frame skipping ensure that the system maintains real-time performance despite the computational load.



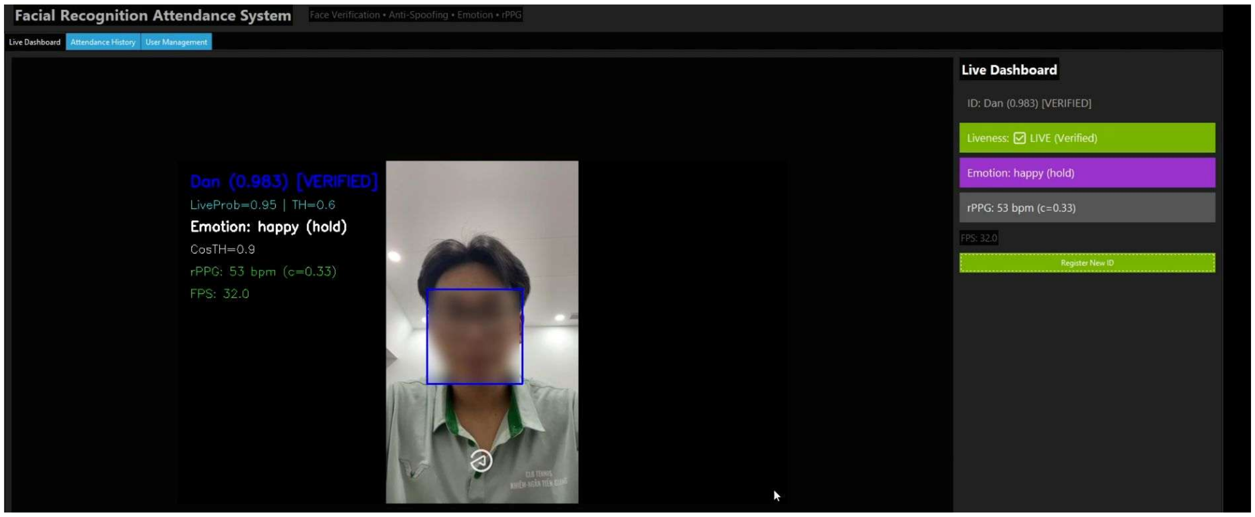
UNKNOWN (−1.000)
LiveProb=0.95 | TH=0.6
Emotion: happy (hold)
CosTH=0.9
rPPG: 55 bpm (c=0.27)
FPS: 1.6

Register Dan: 29/30

ID registration successful 'Dan'.
Saved average embedding at:
./embeddings\Dan.npy

ID: UNKNOWN (-1.000)
Liveness: —
Emotion: happy (hold)
rPPG: 55 bpm (c=0.27)
FPS: 1.6

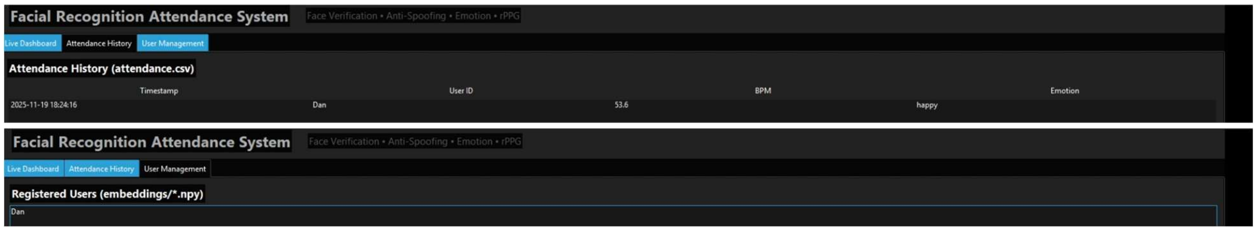
Figure 3. Successful face registration with embedding stored in the database


Facial Recognition Attendance System
Face Verification • Anti-Spoofing • Emotion • rPPG
Live Dashboard
Attendance History
User Management
Live Dashboard
Register New ID

Dan (0.983) [VERIFIED]
LiveProb=0.95 | TH=0.6
Emotion: happy (hold)
CosTH=0.9
rPPG: 53 bpm (c=0.33)
FPS: 32.0

ID: Dan (0.983) [VERIFIED]
Liveness: ☑ LIVE (Verified)
Emotion: happy (hold)
rPPG: 53 bpm (c=0.33)
FPS: 32.0

Figure 4. Successful real-time verification showing identity, liveness, and emotion


Facial Recognition Attendance System
Face Verification • Anti-Spoofing • Emotion • rPPG
Live Dashboard
Attendance History
User Management

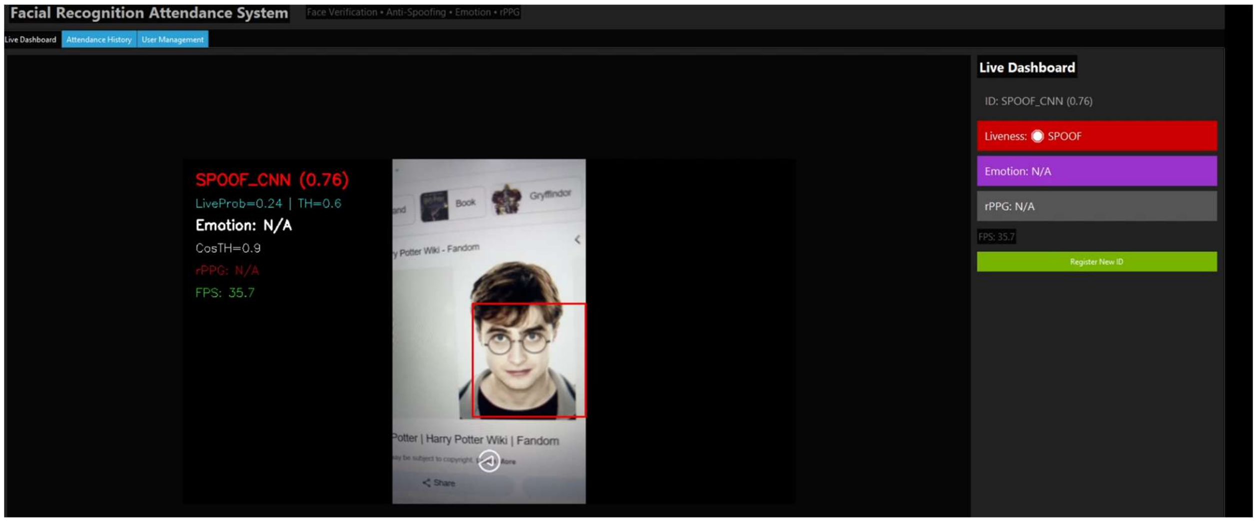
Attendance History (attendance.csv)

Timestamp	User ID	BPM	Emotion
2025-11-19 18:24:16	Dan	53.6	happy

Registered Users (embeddings/*.npy)

Dan

Figure 5. Attendance History and User Management after Registered and Verified


Facial Recognition Attendance System
Face Verification • Anti-Spoofing • Emotion • rPPG
Live Dashboard
Attendance History
User Management
Live Dashboard
Register New ID

SPOOF_CNN (0.76)
LiveProb=0.24 | TH=0.6
Emotion: N/A
CosTH=0.9
rPPG: N/A
FPS: 35.7

ID: SPOOF_CNN (0.76)
Liveness: ● SPOOF
Emotion: N/A
rPPG: N/A
FPS: 35.7

Figure 6. Spoof detected.

III. Results & Discussion

1. Face Verification Results

Figure 7 illustrates the ROC curves for both the supervised and metric-learning embedding models. The supervised hybrid model, which combines EfficientNet-B4, CBAM attention, and a Transformer block, achieved an AUC of 0.8811 using the Euclidean distance metric, outperforming its cosine similarity score of 0.8532. This indicates that Euclidean distance provides slightly better separability for the supervised embedding space. The metric-learning model showed competitive performance, reaching an AUC of 0.8759 for both cosine and Euclidean metrics, demonstrating that triplet loss is effective at structuring the embedding space even without identity-specific supervision.

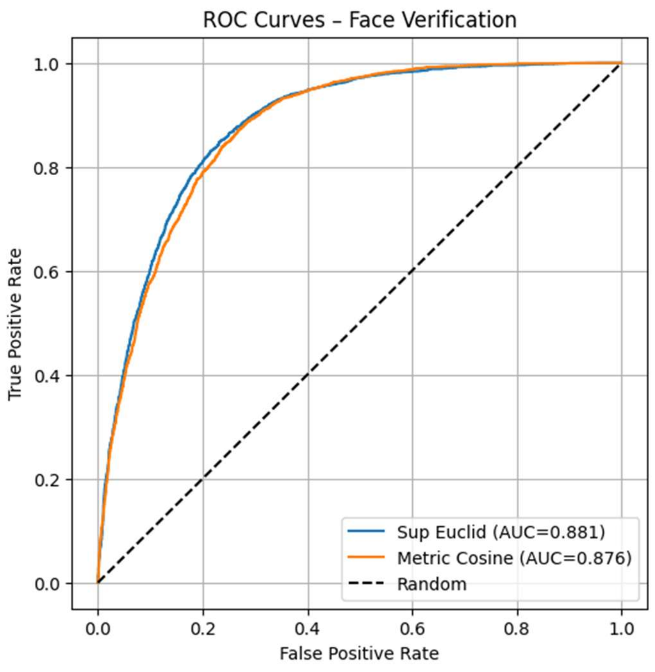


Figure 7. Verification ROC

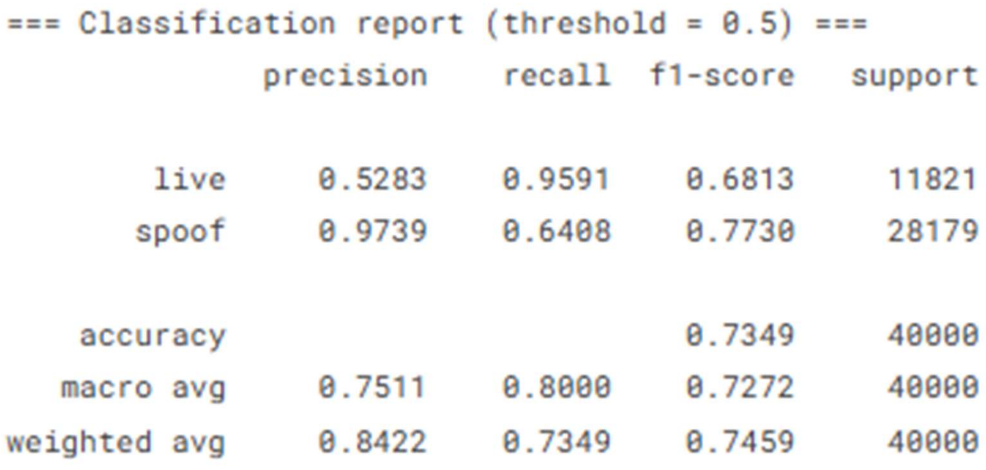
Threshold analysis (Table 1) shows that the optimal cosine threshold for the supervised model **(**0.5803**)** is higher than that of the metric model (0.3845), reflecting differences in embedding scale and distribution between the two training paradigms. Although the metric model achieves marginally higher cosine AUC, the supervised model remains more stable during training and less sensitive to numerical precision. Earlier experiments confirmed that metric-learning models tend to produce NaN gradients under mixed-precision (float16) training, requiring full FP32 for stable optimization.

Method	AUC (Cosine)	Best T (Cosine)	AUC (Euclid)	Best Threshold
Supervised	0.8532	0.5803	0.8811	–610.1769
Metric	0.8759	0.3845	0.8759	–1.1095

Table 1. Supervised and Metric Accuracy Comparison Table

Overall, the supervised hybrid approach offers the best balance between accuracy, stability, and consistency for deployment in a real-time attendance system. The ROC analysis confirms that the learned embeddings effectively separate positive and negative identity pairs, demonstrating strong verification performance suitable for real-world use.

2. Anti-Spoofing Results



```
=== Classification report (threshold = 0.5) ===
              precision    recall  f1-score   support

        live     0.5283    0.9591    0.6813     11821
       spoof     0.9739    0.6408    0.7730     28179

    accuracy                         0.7349     40000
   macro avg     0.7511    0.8000    0.7272     40000
weighted avg     0.8422    0.7349    0.7459     40000
```

Figure 8. Anti-spoof Classification Report

The anti-spoofing module was evaluated on the CelebA-Spoof test set, which includes a diverse range of attack types such as printed photos, screen displays, and replay videos. As shown in Figure 8, the model achieved an overall accuracy of 73.49%, with very strong performance on spoof detection but weaker reliability when predicting live samples. The precision for the *spoof* class reached 0.9739, indicating that the model is highly confident and accurate when identifying fake faces. However, the *live* precision of 0.5283 suggests that real users are sometimes misclassified as spoof, a common issue in spoof datasets where fake cues, flat textures, screen reflections, or edge artifacts, are often easier to learn than subtle indicators of live skin.

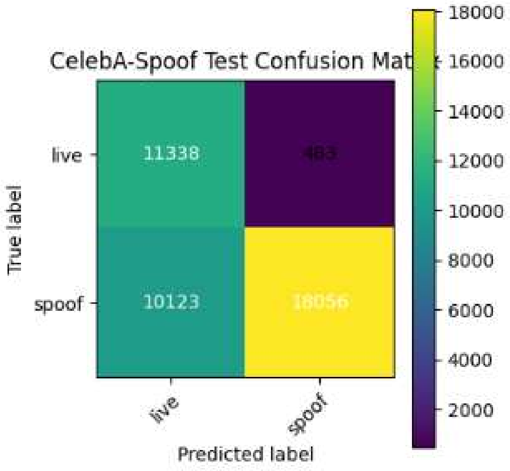


Figure 9. Anti-spoof Confusion Matrix

The confusion matrix (Figure 9) illustrates this imbalance. Out of 11,821 live samples, the model correctly predicted 11,338, misclassifying 483 as spoof. For spoof samples, the detector correctly identified 18,056 but misclassified 10,123, showing that certain spoof types, especially high-quality digital screens, remain difficult and can lead to false acceptances. These cases often contain high-resolution, low-noise facial textures that mimic the appearance of real faces.

```
ROC AUC (from curve): 0.8991
Approx. EER: 19.12% at threshold ≈ 0.0326
```

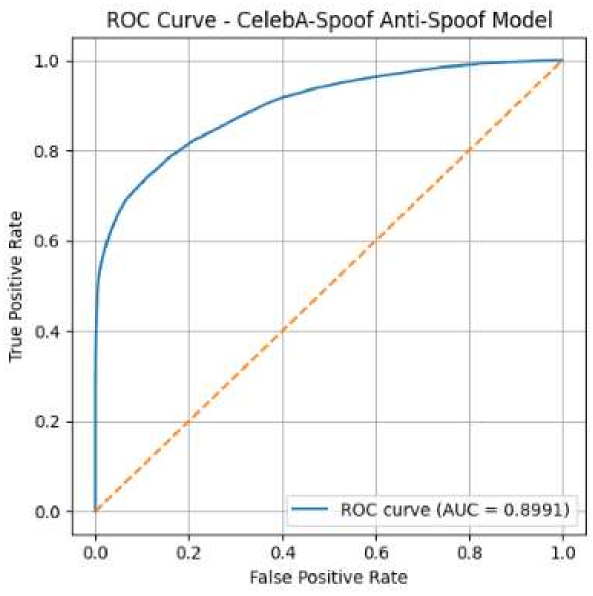


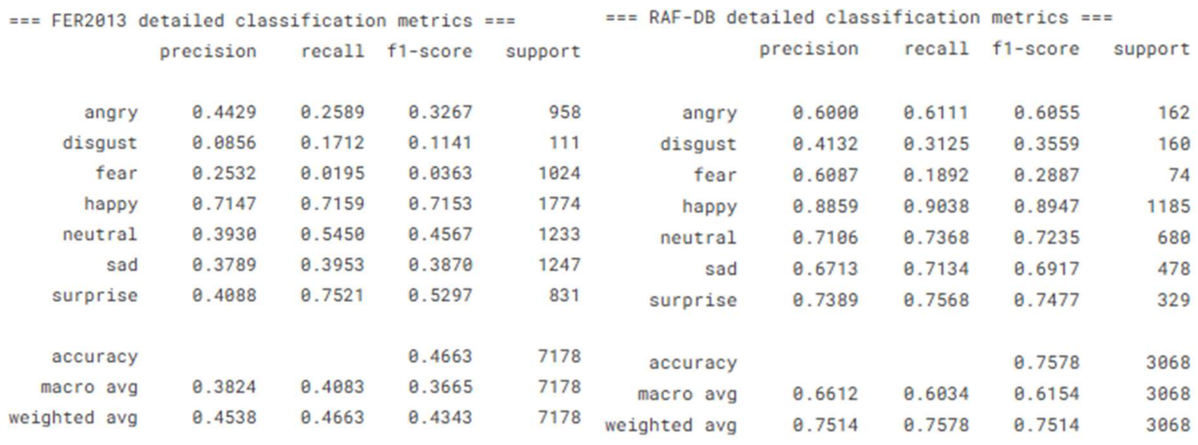
Figure 10. Anti-spoof ROC curve

To further assess discriminability, we plotted the ROC curve (Figure 10), which yielded an AUC of 0.8991, indicating strong overall separation between live and spoof classes. The Equal Error Rate (EER) of 19.12% at a threshold of 0.0326 reflects the point where false acceptance and false rejection rates are balanced. Although a lower EER is generally preferred, this value is reasonable given the complexity and variability of spoof attacks in CelebA-Spoof.

Despite the limitations observed in certain spoof categories, the anti-spoofing module performs reliably in most practical scenarios and integrates smoothly with the full attendance system. When combined with rPPG-CHROM signals and multi-frame smoothing, both of which stabilize predictions over time, the system becomes more robust against high-quality screen-based spoof attempts that may initially confuse the CNN classifier.

3. Emotion Detection Results

The emotion recognition module was evaluated on two complementary datasets: FER2013, a large but noisy benchmark with low-resolution images, and RAF-DB, a smaller yet higher-quality dataset with clearer annotations. This combination allowed the model to learn both coarse robustness and fine-grained facial expression cues.



```
=== FER2013 detailed classification metrics ===
              precision    recall  f1-score   support

       angry     0.4429    0.2589    0.3267       958
     disgust     0.0856    0.1712    0.1141       111
        fear     0.2532    0.0195    0.0363      1024
       happy     0.7147    0.7159    0.7153      1774
     neutral     0.3930    0.5450    0.4567      1233
         sad     0.3789    0.3953    0.3870      1247
    surprise     0.4088    0.7521    0.5297       831

    accuracy                         0.4663      7178
   macro avg     0.3824    0.4083    0.3665      7178
weighted avg     0.4538    0.4663    0.4343      7178
```

```
=== RAF-DB detailed classification metrics ===
              precision    recall  f1-score   support

       angry     0.6000    0.6111    0.6055       162
     disgust     0.4132    0.3125    0.3559       160
        fear     0.6087    0.1892    0.2887        74
       happy     0.8859    0.9038    0.8947      1185
     neutral     0.7106    0.7368    0.7235       680
         sad     0.6713    0.7134    0.6917       478
    surprise     0.7389    0.7568    0.7477       329

    accuracy                         0.7578      3068
   macro avg     0.6612    0.6034    0.6154      3068
weighted avg     0.7514    0.7578    0.7514      3068
```

Figure 11. Emotion Recognition Classification Report

On the FER2013 test set (Figure 11), the model obtained an accuracy of 46.63%, which aligns with typical performance on this challenging dataset. The classifier achieved strong recall for *surprise* (0.7521) and *happy* (0.7159), but struggled with *fear* and *disgust*, reflecting the ambiguity and class imbalance inherent in FER2013. The confusion matrix shows that many misclassifications occur among negative-valence expressions such as fear, sadness, and disgust, while happy and neutral dominate predictions. Despite these limitations, the model still captures meaningful emotion patterns under FER2013's noisy conditions.

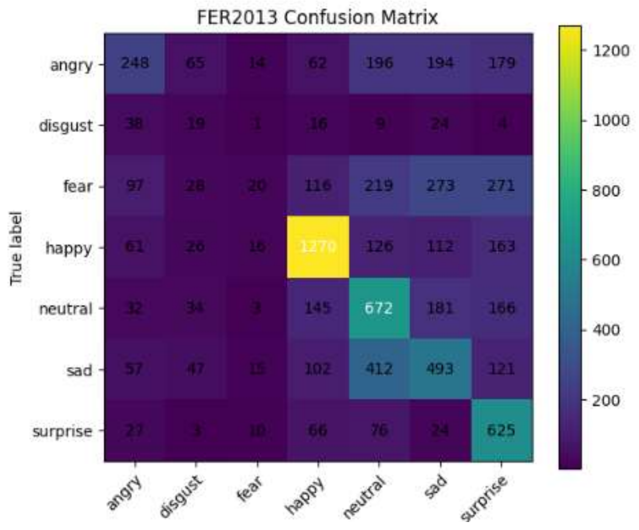


Figure 12. Emotion Recognition Confusion Matrix (FER2013)

In contrast, evaluation on RAF-DB yielded a substantially higher accuracy of 75.78%, demonstrating the model's effectiveness when given cleaner input data. Performance improved across almost all classes, with *happy* achieving an F1-score above 0.89 and *neutral* above 0.72. The RAF-DB confusion matrix shows far clearer separability between classes, particularly for happy, neutral, and surprise, indicating successful fine-tuning on higher-quality images.

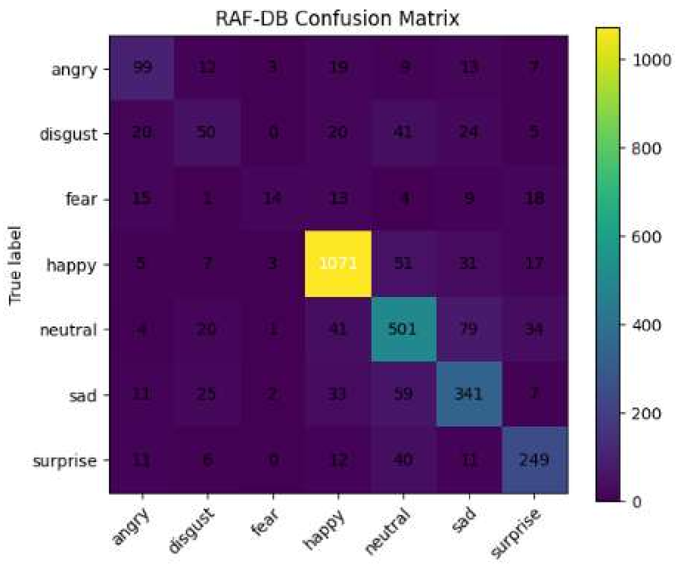


Figure 13. Emotion Recognition Confusion Matrix (RAF-DB)

Hence, the results confirm the effectiveness of the two-stage training strategy, where the hybrid EfficientNetB4 + CBAM + Tiny ViT backbone is pre-trained on FER2013 before being fine-tuned on RAF-DB. In real-time deployment, the model performs consistently under frontal, well-lit conditions but may degrade in cases of occlusion or extreme lighting. Nonetheless, it remains sufficiently stable for integration into the GUI, where a single emotion label per frame enhances user interaction and provides useful sentiment context for the attendance system.

IV. Limitation

The system performs well in controlled conditions but remains sensitive to lighting changes, motion blur, and facial occlusions, which can reduce verification and emotion accuracy. Anti-spoofing may misclassify high-quality screen or print attacks not represented in the training data, and rPPG signals weaken under poor illumination. Running multiple models simultaneously also lowers FPS on CPU-only devices. These issues highlight the need for better lighting robustness, more diverse spoof data, and stronger temporal modeling for real-world deployment.

V. Conclusion

The system successfully integrates face verification, liveness detection, and emotion recognition into a functional real-time attendance application. The supervised hybrid model delivers the most stable verification performance, the anti-spoofing module provides strong defense against common attacks, and the emotion classifier performs reliably on high-quality data. Although the system still faces limitations with lighting, occlusion, and computational load, it remains effective for controlled environments. Future improvements may include stronger feature normalization, depth-based spoof detection, and more efficient temporal modeling for real-world robustness.

Reference

[1] W. Wang, S. Stuijk, and G. de Haan, “A Novel Algorithm for Remote Photoplethysmography: Spatial Subspace Rotation,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 9, pp. 1974–1984, Sep. 2016, doi: https://doi.org/10.1109/tbme.2015.2508602.

[2] C. Lugaresi *et al.*, “MediaPipe: A Framework for Building Perception Pipelines,” *arXiv:1906.08172 [cs]*, Jun. 2019, Available: https://arxiv.org/abs/1906.08172

[3] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *arXiv.org*, May 28, 2019. https://arxiv.org/abs/1905.11946

[4] S. Woo, J. Park, J.-Y. Lee, and I. Kweon, “CBAM: Convolutional Block Attention Module.” Available: https://arxiv.org/pdf/1807.06521

[5] V. Sharma, A. K. Mishra, A. Kumar, N. Shakti, M. Sharma, and S. M, “Improving Image Classification Using Attention Mechanisms in Deep Learning Models,” *2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE)*, pp. 376–380, Nov. 2024, doi: https://doi.org/10.1109/aece62803.2024.10911830.

[6] S. Ghaffarian, J. Valente, M. van der Voort, and B. Tekinerdogan, “Effect of Attention Mechanism in Deep Learning-Based Remote Sensing Image Processing: A Systematic Literature Review,” *Remote Sensing*, vol. 13, no. 15, p. 2965, Jul. 2021, doi: https://doi.org/10.3390/rs13152965.

[7] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *arXiv:2010.11929 [cs]*, Oct. 2020, Available: https://arxiv.org/abs/2010.11929

[8] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “CoAtNet: Marrying Convolution and Attention for All Data Sizes,” *arXiv:2106.04803 [cs]*, Sep. 2021, Available: https://arxiv.org/abs/2106.04803

[9] X. Liu, J. Xu, and T. Sun, “PsyCounAssist: A Full-Cycle AI-Powered Psychological Counseling Assistant System,” *arXiv.org*, 2025. https://arxiv.org/abs/2504.16573

[10] Z. Peng *et al.*, “Conformer: Local Features Coupling Global Representations for Recognition and Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 9454–9468, Aug. 2023, doi: https://doi.org/10.1109/tpami.2023.3243048.

[11] "A Hybrid Fully Convolutional CNN-Transformer Model for Inherently Interpretable Medical Image Classification," *Arxiv.org*, 2020. https://arxiv.org/html/2504.08481v1