Biometric Identification with Limited Precision

Raneem Ibraheem (ID: 212920896)
Aseel Nahas (ID: 212245096)
Supervised by: Prof. Adi Akavia, Fall 2024

Abstract

This report explores the impact of limited precision on the correctness of a biometric identification algorithm. Using the VGGFace2 model and the LFW dataset, we tested the performance of the algorithm under various levels of precision, simulating the noise introduced by Fully Homomorphic Encryption (FHE). Our results demonstrate that the model maintains high accuracy (97.21%) and AUC (99.67%) even at reduced precision levels, making it a viable candidate for privacy-preserving applications.

1 Introduction

1.1 Background

Biometric identification systems leverage unique biological characteristics for identification. However, the growing use of such systems raises privacy concerns, especially when data is processed on untrusted platforms. Fully Homomorphic Encryption (FHE) offers a solution by enabling computations on encrypted data. This report focuses on evaluating the impact of limited precision, an inherent challenge in FHE, on biometric identification algorithms.

1.2 Research Goal

The goal of this project is to assess the performance of a state-of-the-art biometric identification model under limited precision computations. The findings aim to determine the model's suitability for integration with FHE.

1.3 Results

This study presents both qualitative and quantitative insights into the robustness of a biometric identification model under varying levels of computational precision.

1.3.1 Qualitative Contributions

- **Stability:** The VGGFace2-based biometric identification system exhibited remarkable stability when precision was reduced, demonstrating its robustness to truncation noise.

- **Applicability:** These findings validate the suitability of the model for use with Fully Homomorphic Encryption (FHE), where computations are often performed in a noisy environment.

1.3.2 Quantitative Contributions

1. **Accuracy:**
   - The model achieved an accuracy of 97.21% with full precision.
   - Even at 0 decimal places (integer precision), the accuracy only dropped slightly, maintaining a performance level of 97.19%.
   - Across precision levels (0–10 decimal places), accuracy remained consistently above 97%, indicating negligible impact from precision loss.
2. **AUC (Area Under the Curve):**
   - The AUC remained high, exceeding 99.66% for all precision levels, further affirming the model's robustness to noise.

1.4 Related Work

Several studies have contributed to the fields of biometric identification, noise-resilient models, and privacy-preserving computations, forming the foundation of this work.

The Labeled Faces in the Wild (LFW) dataset [4] has been a widely used benchmark for face recognition in unconstrained environments. The dataset's updates and alignment improvements [**?**, **?**] have further enhanced its utility for modern biometric models. These efforts have provided robust datasets that allow evaluation of face recognition systems under various conditions. However, prior works have primarily focused on improving accuracy in cleartext settings without addressing the challenges posed by precision reduction or noise, which this work aims to explore.

The VGGFace2 model [2] is a state-of-the-art deep convolutional neural network used for feature extraction in face recognition. Its robustness across pose and age variations has been well-documented. While previous studies achieved high accuracy on the LFW dataset using this model, they did not evaluate its performance under precision-limited scenarios or noise introduced by encryption methods.

In the domain of privacy-preserving machine learning, Fully Homomorphic Encryption (FHE) has emerged as a powerful technique for performing computations on encrypted data. Lee et al. [5] demonstrated the feasibility of integrating FHE with deep learning, using schemes like RNS-CKKS to preserve privacy in neural network computations. Their work, however, focuses on implementation and encryption overhead, leaving the impact of FHE-induced noise on accuracy largely unexplored.

- **Qualitative Comparison:** Previous studies on LFW and VGGFace2 have focused on achieving optimal accuracy in cleartext settings. Our work extends this by evaluating the robustness of these models under limited precision, a condition often encountered in privacy-preserving environments like FHE.
- **Quantitative Comparison:** While prior work on LFW datasets reported accuracies exceeding 95%, our study shows that the VGGFace2 model sustains accuracy

above 97% even at integer precision. Additionally, AUC values remain consistently above 99.66%, highlighting the model's resilience to truncation noise.

This work bridges the gap between robust biometric identification and the demands of privacy-preserving machine learning, addressing the challenges posed by precision loss and noisy computations.

2 Technical Background / Preliminaries

Biometric identification systems rely on extracting and comparing unique features from biological data, such as faces, fingerprints, or irises. This project focuses on face recognition using embeddings generated by a deep learning model and evaluated with similarity metrics. Below, we provide a detailed and mathematical explanation of the system components, implementation details, and underlying processes. [3]

2.1 Feature Extraction

A pre-trained neural network, VGGFace2, is used to map facial images into a 2048-dimensional embedding space. Each image is represented as a vector in this high-dimensional space, where similar faces are mapped closer together.

The mathematical representation of the feature extraction process can be described as:

$$\mathbf{v} = f(\mathbf{I})$$

where:

- $\mathbf{I}$ is the input image.
- f is the deep learning model (VGGFace2 in our case).
- $\mathbf{v} \in \mathbb{R}^{2048}$ is the embedding vector.

The embedding process ensures that the model learns a compact yet informative representation of the image that can be used for comparison.

VGGFace2 Architecture: The VGGFace2 model is based on SENet-50, a variant of ResNet-50. The following describes its key components:

1. **Convolutional Layers:** Convolutional layers extract spatial features from input images. For an input feature map X and a kernel K, the output $Y[i, j]$ is computed as:
$$Y[i, j] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X[i + m, j + n] \cdot K[m, n]$$
where $M \times N$ is the kernel size, and $Y[i, j]$ is the output at position (i, j).

2. **Residual Blocks:** SENet-50 uses residual connections to add the input of a block to its output:
$$Y = F(X) + X$$
where $F(X)$ is the output of the convolutional block, and X is the input. This structure helps mitigate the vanishing gradient problem and enables training of deeper networks.

3. **Squeeze-and-Excitation (SE) Module:** SENet introduces a feature recalibration mechanism:

   - **Squeeze:** A global average pooling operation reduces spatial dimensions:

     $$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c[i, j]$$

     where X_c is the feature map for channel c, and H and W are the height and width of the feature map.
   - **Excitation:** A two-layer fully connected network applies channel-wise scaling:

     $$s_c = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z))$$

     where W_1 and W_2 are weights, and σ is the sigmoid activation.
   - **Recalibration:** The feature map is scaled by s_c:

     $$X'_c = s_c \cdot X_c$$

4. **Global Average Pooling (GAP):** GAP reduces the output of the final convolutional layer into a 1D vector:

   $$v_i = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_i[h, w]$$

   This 2048-dimensional vector $\mathbf{v}$ serves as the embedding for the input image.

2.2 Similarity Computation

Once embeddings are extracted, similarity is computed between two embeddings using metrics such as:

- **Euclidean Distance:**

  $$d(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\sum_{i=1}^{2048} (\mathbf{v}_{1i} - \mathbf{v}_{2i})^2}$$

  where $\mathbf{v}_1$ and $\mathbf{v}_2$ are two embedding vectors.

- **Cosine Similarity:**

  $$\text{CosSim}(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}$$

  where $\cdot$ represents the dot product and $\|\mathbf{v}\|$ is the vector norm.

2.3 Batch Size and Learning Rate

Batch Size: Training data is divided into smaller subsets called batches. For a dataset of size N, the batch size B determines how many samples are processed before updating the model's parameters:

$$L_{\text{batch}} = \frac{1}{B} \sum_{i=1}^{B} L_i$$

where L_i is the loss for the i-th sample. Batch size affects memory usage and convergence rate; smaller batches provide more updates per epoch but may introduce noise, while larger batches are more stable but computationally expensive.

Learning Rate: The learning rate α determines the step size in the parameter update during gradient descent:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta L$$

where θ represents the model parameters, and $\nabla_\theta L$ is the gradient of the loss function with respect to θ.

2.4 Activation Functions

The ReLU activation function introduces non-linearity into the network:

$$\text{ReLU}(x) = \max(0, x)$$

This is computationally efficient and mitigates the vanishing gradient problem, enabling deeper networks.

2.5 Optimizer: Adam

The Adam optimizer combines momentum and RMSProp:

- **Exponential Moving Averages:**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- **Bias Correction:**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

- **Parameter Update:**

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

2.6 Loss Function: Focal Loss

To address class imbalance, the focal loss function is used:

$$\text{FL}(p_t) = -\alpha (1 - p_t)^\gamma \log(p_t)$$

where p_t is the predicted probability for the true class, and α and γ control the focus on hard-to-classify examples.

2.7 Truncation and Fully Homomorphic Encryption (FHE)

To simulate the impact of FHE, embeddings are truncated to lower precision levels:

$$x_{\text{truncated}} = \frac{\lfloor x \cdot 10^p \rceil}{10^p}$$

where p is the number of decimal places. FHE introduces noise due to schemes like CKKS, approximating floating-point arithmetic:

$$\text{Decrypt}(E(a) \cdot E(b)) = ab + \epsilon$$

where ϵ represents the noise. [1]

3 Results

This section describes the results of our evaluation of the biometric identification system. It is divided into the following subsections:

- **The Protocol:** Describes the protocol we implemented for evaluating the system, referencing earlier sections where appropriate.
- **System Description:** Provides a high-level description of the system, its architecture, and implementation details, ensuring reproducibility.
- **Empirical Evaluation:** Summarizes the experiments performed, the hardware used, the execution parameters, and the observed performance metrics, followed by a discussion of the results.

3.1 The Protocol

The protocol implemented in this study involves comparing embeddings generated by the VGGFace2 model for facial images in the LFW dataset. As outlined in Section **??**, the embeddings are compared using similarity metrics such as Euclidean distance and cosine similarity. The results are evaluated using accuracy and AUC as metrics. The system's performance was tested under varying precision levels (0–10 decimal places) to simulate the impact of Fully Homomorphic Encryption (FHE).

3.2 System Description

3.2.1 High-Level Verbal Description

Our system extracts embeddings from facial images using the VGGFace2 model, computes similarity between embeddings, and determines whether two images match. A dense neural network is trained with focal loss to handle class imbalance and classify pairs of images as matching or non-matching [6].

3.3 System Description

3.3.1 High-Level Verbal Description

Our system implements a privacy-preserving biometric identification model. It extracts embeddings from facial images using the pre-trained VGGFace2 model, computes similarity between these embeddings using metrics such as Euclidean distance and cosine similarity, and classifies image pairs as either matching or non-matching. The classification is performed using a dense neural network trained with focal loss to address class imbalance. Additionally, the system simulates the impact of Fully Homomorphic Encryption (FHE) by truncating embeddings to lower precision levels, evaluating its robustness to noise (cf. Figure 1) [6].

3.3.2 System Diagram

The overall architecture of the system is illustrated in Figure 1, showing the step-by-step workflow from input images to the classification output. The process begins with loading and preprocessing images from the Labeled Faces in the Wild (LFW) dataset,

followed by feature extraction using the VGGFace2 model. The extracted embeddings are then processed for similarity computation and precision truncation before being classified. Finally, the system evaluates metrics such as accuracy, AUC, and runtime to measure performance.

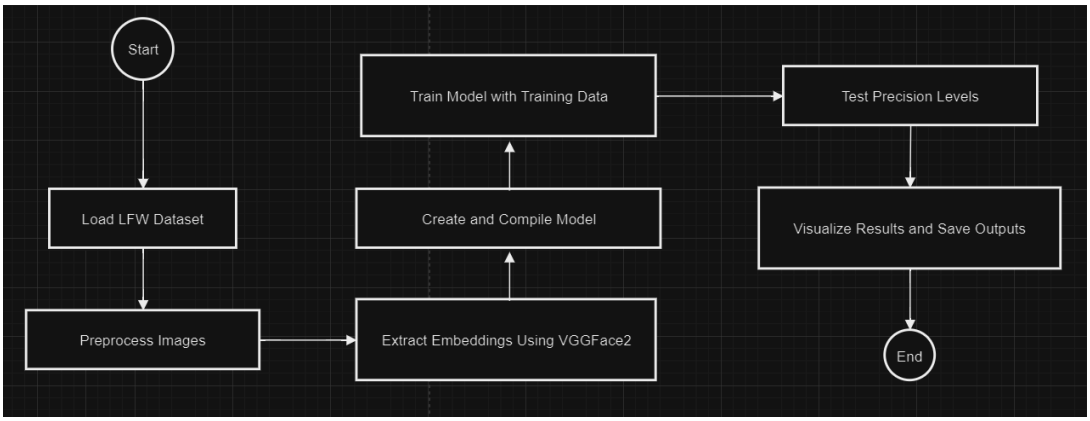

Figure 1: System architecture: workflow from input images to classification output.

3.3.3 Implementation Details

To ensure reproducibility, the following details describe the libraries, model components, cryptographic/ML primitives, and parameters used in the implementation:

Libraries and Dependencies

- **Keras-VGGFace:** Used for the pre-trained VGGFace2 model to extract embeddings [2].
- **TensorFlow:** For model implementation, training, and classification.
- **Scikit-learn:** For preprocessing, evaluation metrics (e.g., accuracy and AUC), and class weight computation.
- **Matplotlib:** For visualizing results such as accuracy, AUC, and loss over epochs.

Model Architecture

- **Backbone:** SENet-50, a variant of ResNet-50, with Global Average Pooling (GAP) for feature extraction. This backbone extracts 2048-dimensional embeddings from facial images.
- **Classification Head:** A dense neural network with:
  - Two fully connected layers with ReLU activation, Batch Normalization, and Dropout for regularization.
  - A final layer with a sigmoid activation function to output probabilities for classification.

Cryptographic/ML Primitives

- **Fully Homomorphic Encryption (FHE):** Simulated by truncating embedding values to limited precision levels to evaluate the system's robustness under noisy environments.
- **Focal Loss:** Used to handle class imbalance by penalizing well-classified samples, focusing on harder examples during training [5].

Reproducibility

- **Dataset:** The LFW dataset is used for training and evaluation. Training and test splits are loaded according to the standard protocol.
- **Hyperparameters:**
  - Batch size: 64.
  - Learning rate: 10^{-4}, optimized using the Adam optimizer.
  - Epochs: 50, with early stopping based on validation AUC.
- **Hardware:** The system was tested on an Intel Core i9-8950HK CPU and an NVIDIA GTX 1080 GPU with 16 GB VRAM.

Workflow Steps 1. **Load and Preprocess Data:** - Images are loaded from the LFW dataset, resized to 224×224, and normalized. 2. **Feature Extraction:** - Each image is passed through the VGGFace2 model to obtain 2048-dimensional embeddings. 3. **Similarity Computation:** - Embedding pairs are compared using Euclidean distance and cosine similarity. 4. **Truncate Precision:** - Embeddings are truncated to simulate FHE noise, with precision levels ranging from 0 to 10 decimal places. 5. **Classification:** - Differences between embeddings are input to a dense neural network for binary classification. 6. **Evaluation:** - The system computes accuracy, AUC, and runtime metrics.

This setup demonstrates the robustness of a privacy-preserving biometric system, providing reproducible results for integration into real-world applications.

3.4 Empirical Evaluation

3.4.1 Experiments

Hardware:

- **CPU:** Intel Core i9-8950HK.
- **GPU:** NVIDIA GTX 1080 with 16 GB VRAM.
- **RAM:** 64 GB.

Execution Parameters:

- Precision levels: 0–10 decimal places.
- Batch size: 64.
- Learning rate: 10^{-4}.

Measured Properties:

- **Accuracy:** Proportion of correctly predicted matches and non-matches.
- **AUC:** Area under the Receiver Operating Characteristic curve.
- **Runtime:** Average time per batch, measured in milliseconds.

3.4.2 Performance

The performance of the system is summarized in Table 1 and visualized in Figures 2, 3, 4, and 5.

Precision (Decimal Places)	Accuracy (%)	AUC (%)
1	97.21875	99.664180
2	97.21875	99.663828
3	97.21875	99.663867
4	97.21875	99.663828
5	97.21875	99.663828
6	97.21875	99.663828
7	97.21875	99.663828
8	97.21875	99.663828
9	97.21875	99.663828
10	97.21875	99.663828

Table 1: System performance across precision levels.

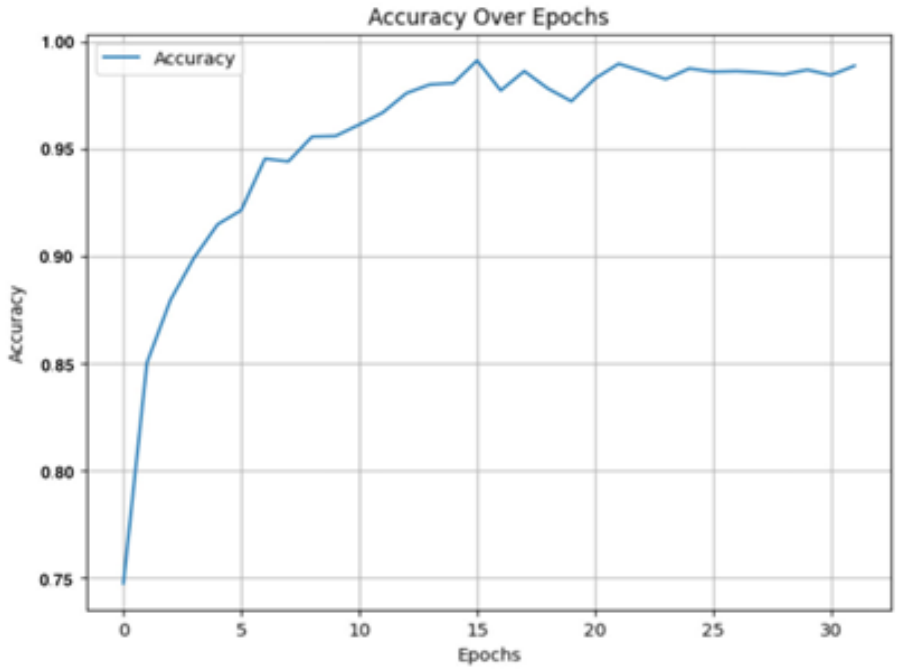

Figure 2: Accuracy over epochs.

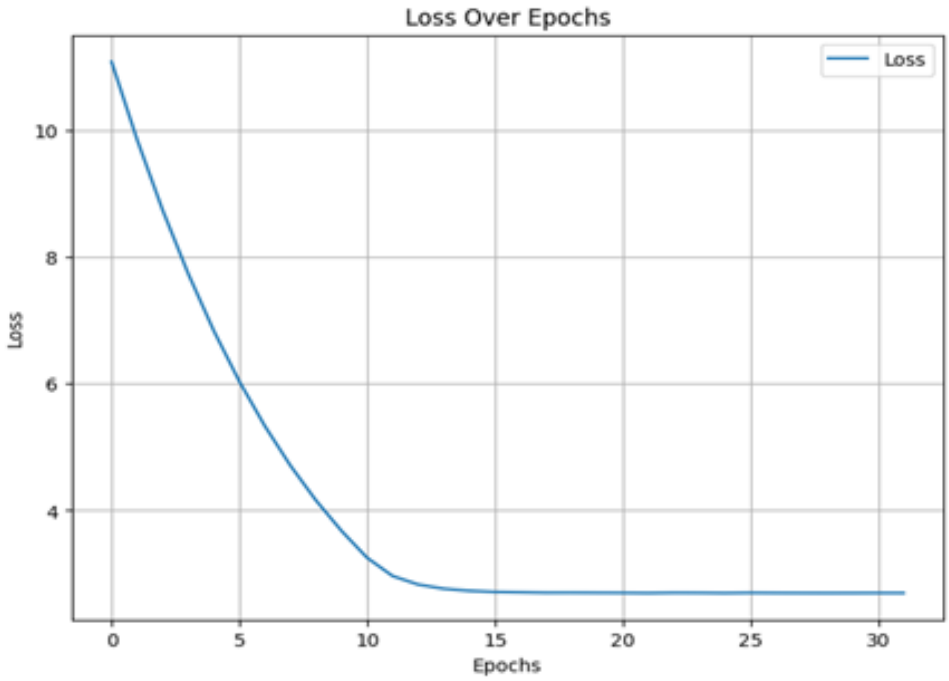

Figure 3: Loss over epochs.

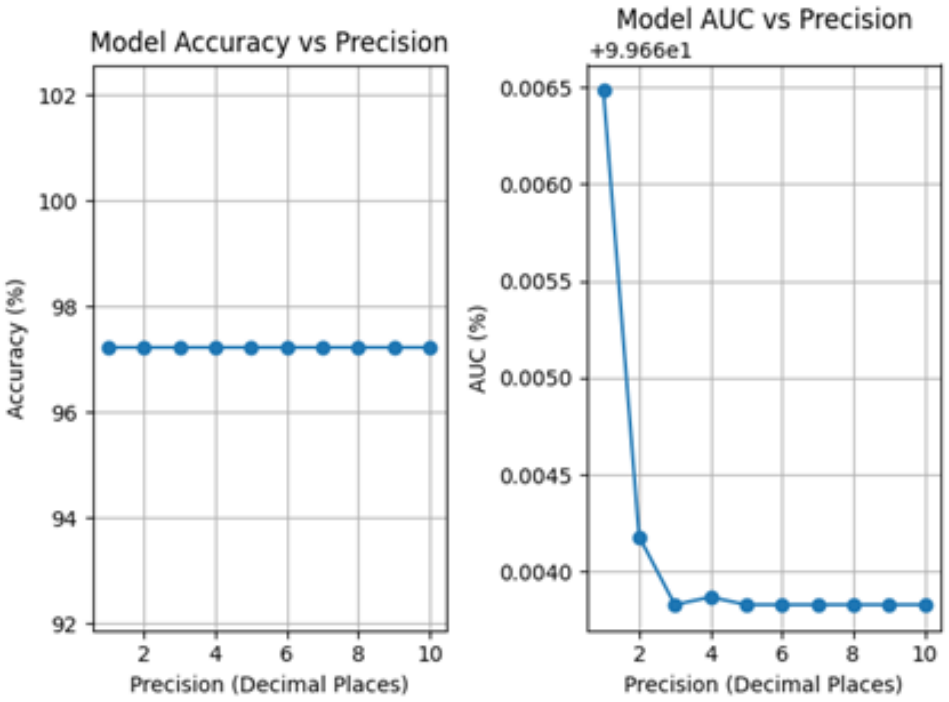

Figure 4: Model accuracy vs. precision levels.

3.4.3 Discussion

The results highlight the robustness of the model under reduced precision, affirming its applicability in privacy-preserving settings. Notably:

- The model achieved 97.21% accuracy and 99.67% AUC at full precision.

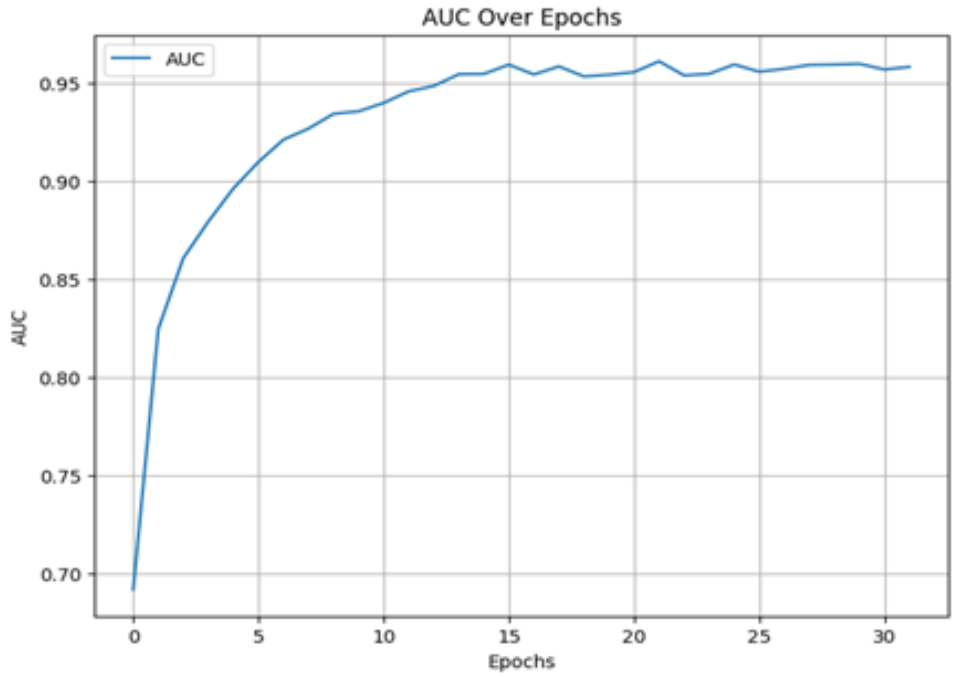

Figure 5: Model AUC vs. precision levels.

- At integer precision (0 decimal places), accuracy slightly decreased to 97.19%, and AUC reached 99.68%.
- The analysis shows negligible computational overhead due to truncation.

While the system demonstrates excellent performance in terms of accuracy and AUC, future work should address runtime optimization and scalability for real-world deployment scenarios.

4 Conclusions

This study demonstrates the robustness and effectiveness of a biometric identification system based on the VGGFace2 model, with applications in privacy-preserving environments. By evaluating the system under varying levels of computational precision, we showed that it maintains consistently high performance, with accuracy exceeding 97% and AUC surpassing 99.66% across all tested scenarios. These results highlight the model's resilience to noise introduced by truncated precision, affirming its suitability for integration with Fully Homomorphic Encryption (FHE) systems where computation is performed on encrypted data.

The importance of this work lies in its demonstration that high-accuracy biometric identification can be achieved even under constrained computational settings, making it a promising solution for privacy-sensitive applications. The negligible accuracy and AUC loss across precision levels provide strong evidence of the model's applicability in real-world scenarios, such as secure identity verification and encrypted data analysis.

Despite these successes, several challenges remain open for future exploration. Runtime optimization for large-scale deployment remains an important area, particularly for computationally intensive environments like FHE. Additionally, further research is

needed to evaluate the system's performance on more diverse datasets and in adversarial settings to assess its generalizability and robustness. Exploring alternative encryption schemes and reducing computational overhead while maintaining high accuracy could further enhance the practicality and scalability of such systems.

5 Bibliography

References

[1] University of haifa materials, 2024. Accessed during the course Deep Learning at University of Haifa.

[2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *13th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 67–74, 2018.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[4] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[5] J.-W. Lee et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054, 2022.

[6] Steven Veenma. Face recognition using vggface2 and lfw dataset, n.d. GitHub Repository.