SAPIENZA
UNIVERSITÀ DI ROMA

Evaluation of Vision Transformer for Face Verification under attacks and appearance variations

BIOMETRICS SYSTEMS

Professor:
Maria De Marsico

Student:
Alessia Infantino
1922069

Academic Year 2025/2026

Contents

1 Introduction

Face verification plays an important role in many biometric applications. Modern systems must correctly identify a person even when the face is affected by variations such as aging, makeup, illumination, or camera quality. They also need to detect possible spoofing attempts, such as deepfakes or printed photos, which can reduce the reliability of the system.

In this project, the performance of a Vision Transformer (ViT) model combined with the ArcFace loss is studied, which is one of the most popular and widely used approaches in modern face verification.

Then the model is evaluated through a custom dataset that includes a gallery of identities and a test set with both natural user variations and spoofing attacks. After extracting embeddings from the ViT-ArcFace model, the similarity scores between each probe image and all gallery identities are computed.

Through this evaluation, it is possible to see how robust the model is in real-world scenarios and how different factors affect recognition accuracy. The results also show the main strengths and limitations of ViT–ArcFace architectures in biometric systems.

2 Model Architectures

In this section, the architectures used in the face verification pipeline are described. The system combines RetinaFace for detection and alignment with a Vision Transformer trained with the ArcFace loss for feature extraction. Together, they provide the aligned inputs and discriminative embeddings needed for the verification stage.

2.1 RetinaFace

RetinaFace is a single-stage face detector that performs several prediction tasks in parallel [1]. For each anchor, the model estimates:

- **Face classification**: probability of face / non-face;
- **Bounding-box regression**: the position and size of the detected face;
- **Five facial landmarks**: the coordinates of the eyes, nose tip, and mouth corners.

The network architecture (Fig. 1) is composed of the following stages:

- **Backbone**. A ResNet-50 extracts feature maps at multiple depths (C2,...,C5).
- **Feature Pyramid Network (FPN)**. The backbone features are converted into a multi-scale pyramid (P2,...,P6), in order to analyze faces at different resolutions and to handle both small and large faces.
- **Anchors**. RetinaFace places predefined anchor boxes on each pyramid level. During training, an anchor is labeled *positive* if it overlaps a ground-truth face (IoU > 0.5), and *negative* otherwise. The model learns to classify each anchor and adjust it through box and landmark regression.
- **Context Modules**. Each level of the pyramid is processed by a dedicated context module (a short sequence of convolutional layers) that enlarges the receptive field and incorporates additional spatial context.
- **Prediction Head**. A shared prediction head is applied to all pyramid levels to produce the final outputs for each anchor: face/background classification (2 values), bounding-box regression (4 values), and five facial landmarks (10 values).

2.1.1 Training and loss

RetinaFace is trained on the WIDER FACE dataset, which provides a large number of faces with strong appearance variations and includes annotated facial landmarks. The model uses an anchor-based training strategy: positive anchors (IoU > 0.5) are

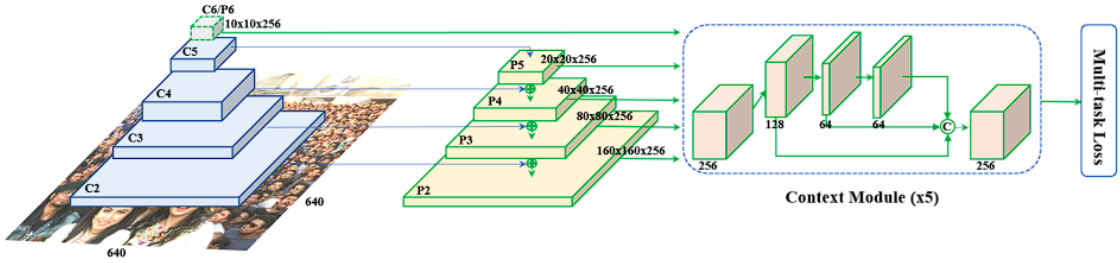


Figure 1: RetinaFace architecture overview.

supervised with the full multi-task loss (classification, bounding-box regression, and five-landmark regression), while negative anchors (IoU < 0.3) contribute only to the classification loss. Because the vast majority of anchors are negative, the training employs Online Hard Example Mining (OHEM) to maintain a balanced 3:1 ratio between negative and positive samples.

2.2 The Vision Transformer (ViT) for Face Verification

The main system adopted in this project is based on a pre-trained Vision Transformer (ViT) specifically designed for face Verification. The model retrieved from HuggingFace [2], was trained on the large-scale MS1MV3 dataset using the ArcFace loss. It outputs a 512-dimensional feature embedding for each face image, which is then used for verification through cosine similarity.

The Vision Transformer (ViT), introduced by Dosovitskiy et al. in 2020, adapts the Transformer architecture—originally developed for Natural Language Processing, to computer vision tasks [3]. Compared to classical CNNs that operate around a local receptive field, a ViT model split the original image into patches and find a global relationship between them, considering the entire image overall. It has the following structure:

- **Patch Embedding:** The original transformer works on 1D token sequences, so the input image is divided into fixed-size patches. Then each patch is flattened and linearly projected into a space of dimension D, to obtain patch embeddings. In this model, the input image is resized to 112×112 and divided into non-overlapping patches of size 8×8.

- **Positional Encoding:** a positional embedding is added to each patch vector to retain information about its position in the original image.

- **Transformer Encoder:** The sequence of patch embeddings is processed by several encoder layers composed of Multi-Head Self-Attention (MHSA) and Feed-Forward Neural Networks (FFN). This mechanism enables the model to learn global

dependencies between distant regions of the image. Each block also includes Layer Normalization (LN) and residual connections.

- **Global Pooling:** Unlike standard ViTs, this model does not rely on a [CLS] token. Instead, a global pooling operation aggregates information from all patch tokens to produce a single image representation, which is more suitable for metric-learning tasks such as face recognition (Fig. 2).

Compared to traditional CNNs, ViTs offer greater flexibility in modeling long-range spatial relationships; however, they generally require larger datasets and more computational resources for training (Fig. 3) [4].

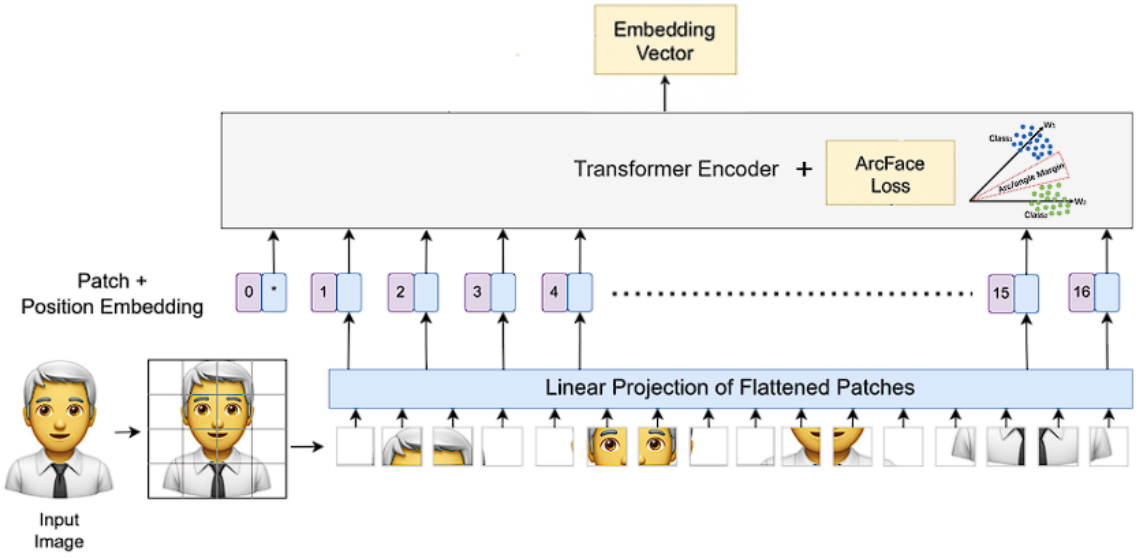


Figure 2: Model overview. The image is split into fixed-size patches, each patch is linearly embedded, positional embeddings are added, and the resulting sequence of vectors is fed into a standard Transformer encoder. The result is embedding vector that represent the entire image.

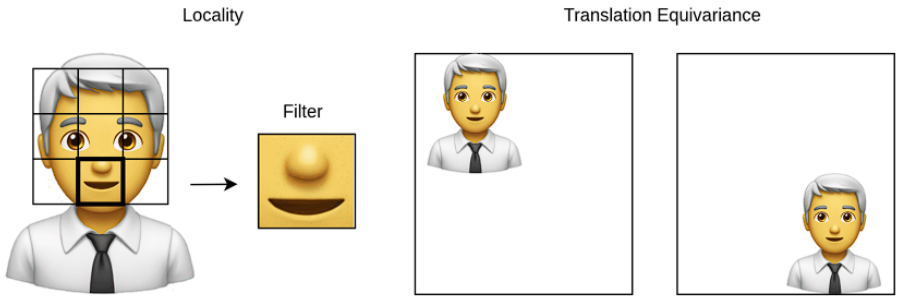


Figure 3: Illustration of inductive biases, locality, and translation equivariance present in CNNs for face recognition.

2.2.1 Training Dataset: MS1MV3

The model was fine-tuned on the MS1MV3 dataset [5], a cleaned and refined version of the MS-Celeb dataset consisting of more than 5 million aligned face images across 93,431 identities. All images in MS1MV3 are aligned using the standard 5 facial landmarks provided by the dataset (eyes, nose, mouth corners), resized to 112×112, and frontalized. This detail is important because the model also expects aligned faces during inference.

2.2.2 ArcFace Loss: Angular Margin for Discriminative Embeddings

The model is trained using ArcFace [6], an additive angular margin loss specifically designed for face recognition. While a standard softmax classifier aims to increase intra-class similarity and decrease inter-class similarity, it does not explicitly enforce a clear separation between classes. As a result, the learned embeddings may have poor discriminative margins, and the feature space may not be suitable for open-set face recognition.

To address this limitation, ArcFace introduces an *additive angular margin*. The idea is to enforce:

- small angular distances between samples of the same identity (intra-class compactness),
- large angular distances between different identities (inter-class separability).

Formally, ArcFace normalizes both the feature vectors and the class weight vectors, and adds an angular margin m to the target logit before applying the softmax [7] (Fig. 4). The resulting loss function is:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\,s\cdot\cos(\theta_{y_i}+m)}}{e^{\,s\cdot\cos(\theta_{y_i}+m)} + \sum_{j\neq y_i} e^{\,s\cdot\cos(\theta_j)}}.$$

where:

- x_i is the normalized feature vector of sample i:

$$\tilde{x}_i = \frac{x_i}{\|x_i\|},$$

- W_j is the normalized weight vector of class j:

$$\tilde{W}_j = \frac{W_j}{\|W_j\|},$$

- θ_j is the angle between $\tilde{x}_i$ and $\tilde{W}_j$:

$$\cos(\theta_j) = \tilde{W}_j^{\top}\tilde{x}_i,$$

- m is the additive angular margin,
- s is a scaling factor.

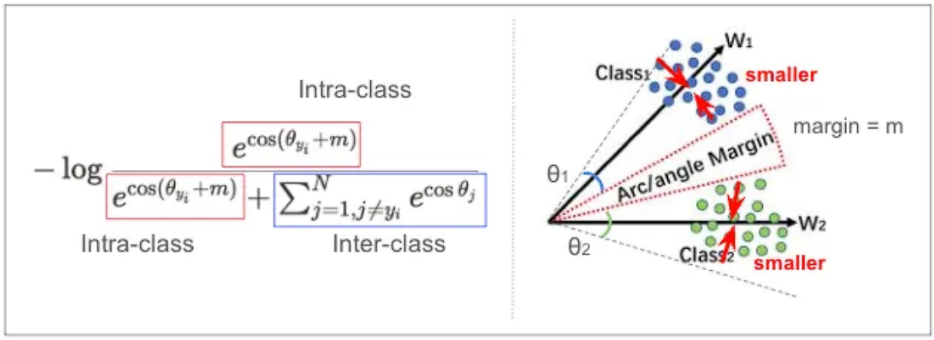


Figure 4: Visual explanation of the ArcFace loss. The left side shows the modified softmax formulation, where an additive angular margin m is applied to the target logit to enforce stronger intra-class compactness and inter-class separability. The right side illustrates how the margin increases the angular distance between class centers while pulling samples of the same class closer together.

3 Dataset

3.1 Gallery

The gallery consists of 212 images of well-known individuals, with three images per person captured from different angles or showing varied facial expressions, under controlled and uncontrolled environments. This allowed the system to become more robust to input variations (Fig. 5).

Figure 5: Three examples of the same subject in the gallery (Keanu Reeves).

3.2 Testing set

The testing set consists of 12 images per 7 category, with the exception of the interpersonal one, which contains 27 images. The images were carefully selected to challenge the system in different ways and to evaluate its ability to manage user variations and attacks.

3.2.1 User variations

To evaluate the robustness of the system under different real-world conditions, several types of user-related variations were considered. These variations reflect common changes in appearance that may influence the recognition process:

- **Plastic Surgery:** Represents the impact of cosmetic or reconstructive procedures that may reshape facial features (e.g., nose, eyes...). Such transformations represent a major challenge for maintaining consistent identity verification.
- **Make Up:** Illustrates how the use of make up can modify essential facial traits such as skin tone, texture and contours. These alterations make it harder for the system to preserve accuracy.

- **Age:** Includes natural ageing effects, such as wrinkles and other time-related facial changes. Although these differences develop gradually, they can significantly influence the system's ability to recognize the same person over time.

(a) Katy Perry (b) Zooey Deschanel

Figure 6: Example of Interpersonal variation in the test set: two distinct individuals with highly similar facial features.

3.2.2 Potential attacks

The last three categories aim to reproduce potential attack scenarios, allowing an assessment of the system's robustness when exposed to different spoofing attempts. The following types of attacks were considered:

- **Print Attack:** Refers to an attempt where a printed photo of a subject's face is presented to the system in place of a real person. This scenario helps verify whether the model can differentiate between live human faces and two-dimensional reproductions.

- **Camouflage:** This category includes cases where users intentionally modify their appearance using elements such as masks, heavy makeup or face paint. These alterations make it harder for the system to correctly identify individuals, as several distinctive facial features are either hidden or distorted.

- **Interpersonal:** Refers to similarities between distinct individuals who share close physical traits (e.g., siblings or look-alikes). This category highlights the system's difficulty in distinguishing between faces with comparable biometric characteristics (Fig. 6).

- **Deep Fake:** Involves synthetically generated images through artificial intelligence techniques, capable of creating realistic yet fake faces and designed to imitate real people. This test evaluates how well the system can detect and reject manipulated and fully fabricated content (Fig. 7).

(a) Real

(b) Deep Fake

Figure 7: Example of deepfake image in the test set of Scarlett Johansson.

It should be noted that some images present mixed characteristics. For example, a sample classified under "plastic surgery" may also include "makeup", or an "age" sample may show additional appearance changes. These overlaps better reflect real-world conditions, where multiple variations often occur simultaneously.

4 Project pipeline

To assess the performance of the face verification system, a complete evaluation pipeline was designed.

Starting from the detection of facial landmarks, each face is aligned following the ArcFace standard and then passed through the Vision Transformer to obtain a 512-dimensional embedding. Finally, the embeddings are compared against the gallery using cosine similarity, and different metrics are computed to analyse the behaviour of the system under several variations and attack scenarios.

The full evaluation procedure is summarized in Figure 8, and each component is described in detail in the following subsections.

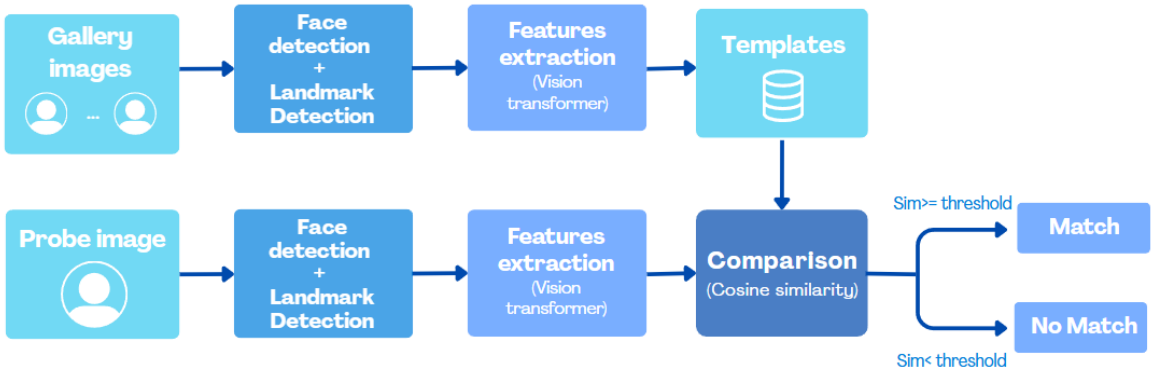


Figure 8: Summary of the pipeline followed in this project.

4.1 Landmarks detection and Face Alignment

The Vision Transformer used in this work was trained with an ArcFace-style loss function. Because ArcFace models expect a fixed geometric arrangement of facial landmarks, each input face must be aligned before feature extraction. In particular, the face is normalized using five keypoints: the two eyes, the nose tip and mouth corners. The standard ArcFace keypoints are reported in Table 1.

Landmark	**x**	**y**
Left eye	38.2946	51.6963
Right eye	73.5318	51.5014
Nose tip	56.0252	71.7366
Left mouth	41.5493	92.3655
Right mouth	70.7299	92.2041

Table 1: Standard ArcFace 5-point template (coordinates in pixels for a 112×112 crop).

To detect faces and extract the five landmarks, this work uses the `insightface` library [8] and in particular the "buffalo_l" model pack. This pack includes a RetinaFace-based detector (RetinaFace-10GF), which finds the face and directly predicts the five keypoints. Other modules included in the pack, such as 2D/3D landmark refinement or face recognition, are not used in this project.

Once the five keypoints are detected, they are matched to the ArcFace template by estimating an affine transformation with OpenCV's `estimateAffinePartial2D`. The image is then warped using `warpAffine` to obtain a normalized 112×112 face, ready for feature extraction (Fig. 9) [9].
An extract of the code used to compute this phase is shown in Fig. 10.

(a) Original image with detected landmarks using Buffalo_l model pack (RetinaFace-10GF).

(b) Aligned image using ArcFace 5-point template.

Figure 9: ArcFace Facial allignment example using 5 detected landmarks.

4.2 Features Extraction

After the face is aligned and normalized, the final 112×112 image is passed to the ViT model, which produces a 512-dimensional embedding. This feature vector is then L2-normalized.
For the gallery, all images of each identity are processed separately and stored as a set of individual embeddings. During testing, each probe image is processed in the same way.
An extract of the code used to compute this phase is shown in Fig. 11.

4.3 Similarity Computation

After computing the embeddings of the test set images (probes) and the enrolled gallery templates, similarity scores are computed. Given two embeddings x and y, their similarity is computed using the cosine similarity:

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

Then for each probe image, its embedding is compared against the embeddings of all gallery identities.

Consequently, for each probe, the evaluation framework generates:

- **One genuine comparison:** the match between the probe and its corresponding gallery templates.
- **N-1 impostor comparisons:** the matches between the probe and all remaining gallery identities, which act as impostors.

This approach ensures a good sampling of impostor scores, resulting in a reliable estimate of the False Acceptance Rate (FAR).

4.4 Evaluation metrics

The comparison outcomes are classified as:

- **Genuine Acceptance:** This occurs when the system correctly detects a match between the probe image and one image in the gallery, meaning it identifies the same person within the chosen threshold.
- **False Acceptance:** This happens when the system wrongly considers two different individuals as the same person, treating a non-matching image as if it were a correct match.
- **Genuine Rejection:** This is when the system correctly recognises that two images belong to different people and therefore rejects the match.
- **False Rejection:** This refers to cases where the system fails to identify a real match and mistakenly classifies the same person as a non-match.

Finally, various metrics are computed to test the robustness of the model under different variations, attacks and thresholds:

- **False Acceptance Rate (FAR):** The FAR represents the percentage of times the system incorrectly accepts an unauthorized person. A low FAR means the biometric system is effective at blocking impostors. It is defined as:

$$\text{FAR} = \frac{\text{False Acceptances}}{\text{False Acceptances} + \text{Genuine Rejections}}$$

- **False Rejection Rate (FRR):** The FRR indicates how often the system wrongly rejects a legitimate user. Reducing FRR is important for making the system more user-friendly and reliable. It is calculated as:

$$\text{FRR} = \frac{\text{False Rejections}}{\text{False Rejections} + \text{Genuine Acceptances}}$$

- **Genuine Acceptance Rate (GAR):** The GAR measures the system's ability to correctly recognize authorized users. It can be expressed as:

$$\text{GAR} = 1 - \text{FRR} = \frac{\text{Genuine Acceptances}}{\text{Genuine Acceptances} + \text{False Rejections}}$$

  A higher GAR shows that the system correctly identifies genuine users more frequently.

- **Genuine Rejection Rate (GRR):** The GRR is the counterpart of FAR and shows how well the system rejects impostors. It is defined as:

$$\text{GRR} = 1 - \text{FAR} = \frac{\text{Genuine Rejections}}{\text{Genuine Rejections} + \text{False Acceptances}}$$

- **Margin:** The Margin is the absolute difference between FAR and FRR:

$$\text{Margin} = |\text{FAR} - \text{FRR}|$$

  A smaller margin indicates a more balanced system, where both error rates are low. A larger margin suggests that the system is uneven in how it handles false acceptances and false rejections.

- **Error Rate of Recognition (ERR):** The ERR identifies the point where the system's incorrect acceptance rate equals its incorrect rejection rate. Using linear interpolation helps estimate this point between discrete data values. Formally:

$$\text{ERR} : \text{FAR} = \text{FRR}$$

- **Receiver Operating Characteristic (ROC):** The ROC curve displays the relationship between the Genuine Acceptance Rate (GAR) and the False Acceptance Rate (FAR) at various decision thresholds, showing how the system performs under different conditions.

- **Detection Error Tradeoff (DET):** The DET curve illustrates the tradeoff between FRR and FAR. It is useful for understanding how changing the threshold affects both types of errors and helps to find the most suitable operating point for the system.

An example of the code used to compute these metrics is shown in Fig. 12.

```python
#https://github.com/gau-nernst/timm-face/blob/main/test_ijb.py

# ArcFace 5-point template
ARCFACE_KPTS = np.array([
    [38.2946, 51.6963],
    [73.5318, 51.5014],
    [56.0252, 71.7366],
    [41.5493, 92.3655],
    [70.7299, 92.2041]
], dtype=np.float32)

def align_face(image_path, size=(112, 112)):
    # Load image
    img = cv2.imread(image_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Detect face + landmarks
    faces = app.get(img)
    if len(faces) == 0:
        raise ValueError(f"No face detected in {image_path}")

    kps = faces[0].kps.astype(np.float32)

    # Estimate affine transform to ArcFace template
    M, _ = cv2.estimateAffinePartial2D(kps, ARCFACE_KPTS, method=cv2.LMEDS)

    # Apply transform
    aligned = cv2.warpAffine(img, M, size, flags=cv2.INTER_CUBIC)

    return Image.fromarray(aligned)
```

Figure 10: Code illustrating how the alignment phase was computed.

```python
model = timm.create_model(
    "hf_hub:gaunernst/vit_tiny_patch8_112.arcface_ms1mv3",
    pretrained=True
).eval().to(device)

transform = T.Compose([
    T.ToTensor(),
    T.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
])

def get_embeddings(image_path):
    aligned = align_face(image_path)
    x = transform(aligned).unsqueeze(0).to(device)
    with torch.no_grad():
        emb = model(x)
        emb = F.normalize(emb, dim=1)
    return emb.squeeze(0).cpu()
```

Figure 11: Code illustrating how the features extraction phase was computed.

```python
# Loop over all thresholds
for threshold in tqdm(thresholds, desc=f"Thresholds {category}"):

    GA = FR = FA = GR = 0

    # Loop over all probe images
    for probe_path in probe_paths:

        probe_name = probe_path.stem.lower()  # exact name of the subject

        # Compute the embedding of the probe image
        try:
            probe_emb = get_embeddings(probe_path)
        except Exception as e:
            print(f"⚠ Skipping probe {probe_path.name}: {e}")
            continue

        # Compare probe with all gallery identities
        for gallery_name, template_list in gallery_embeddings.items():

            gallery_name_clean = gallery_name.lower()

            # Compute similarity with all templates of this identity
            try:
                sims = [
                    F.cosine_similarity(probe_emb, t, dim=0).item()
                    for t in template_list
                ]
            except Exception as e:
                print(f"⚠ Skipping comparison {probe_path.name} vs {gallery_name}: {e}")
                continue

            max_sim = max(sims)

            # Determine if this comparison is genuine or impostor
            is_genuine = (gallery_name_clean == probe_name)

            # Check match based on current threshold
            match = (max_sim >= threshold)

            # Update the confusion matrix counters
            if match and is_genuine:
                GA += 1  # Genuine Accept
            if match and not is_genuine:
                FA += 1  # False Accept
            if not match and is_genuine:
                FR += 1  # False Reject
            if not match and not is_genuine:
                GR += 1  # Genuine Reject

    # Compute metrics for this threshold
    FAR = FA / (FA + GR) if (FA + GR) else 0
    FRR = FR / (FR + GA) if (FR + GA) else 0
    GAR = 1 - FRR
```

Figure 12: Code illustrating how the evaluation metrics were computed.

5 Results

5.1 Detection and alignment

Before analyzing the results, it is important to report that a few images in the dataset could not be processed because the face detector failed to identify a valid facial region. This happened only in 3 cases, when the image presented is too close, have low resolution or showed digital artifacts (Fig. 13). In such conditions, the RetinaFace detector was unable to predict the five landmarks required for ArcFace alignment, and therefore the sample are excluded from the evaluation.

(a) (b) (c)

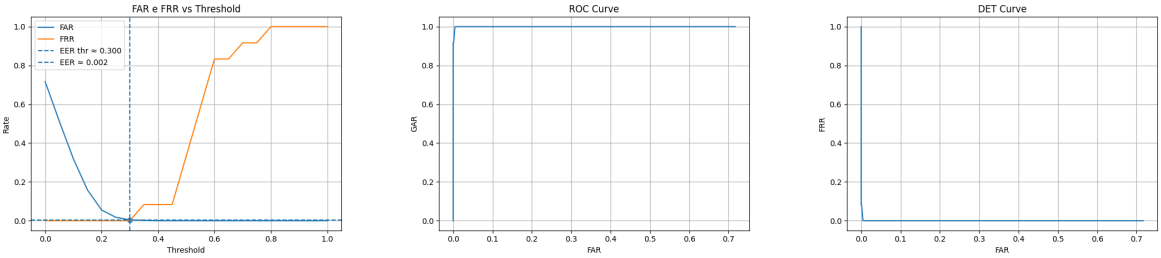
Figure 13: Images where the face detector did not produce any bounding box.

5.2 Verification Performance

5.2.1 Age

The results obtained on the age category show that the model performs very well under age-related variations. Both FAR and FRR quickly approach zero and the Equal Error Rate (EER) is extremely low (0.002). The ROC curve reaches almost perfect values (GAR close to 1 for nearly all thresholds), while the DET curve confirms that false accepts and false rejects are rare (Fig. 14).

These results are consistent with the expected behavior of the model. The ViT used in this work achieved very high accuracy on the AgeDB-30 benchmark, which evaluates the robustness to changes in age. In Fig. 15 and Fig. 16 the best and worst matches are shown in this category. In the best-case example, the model successfully recognizes the subject despite an age gap of approximately 20 years. Conversely, the worst-case match involves a much larger age difference (around 60–70 years), which explains the drop in similarity and confirms that extreme age variation remains a challenging scenario.



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 14: Evaluation results on the *age* variation subset.

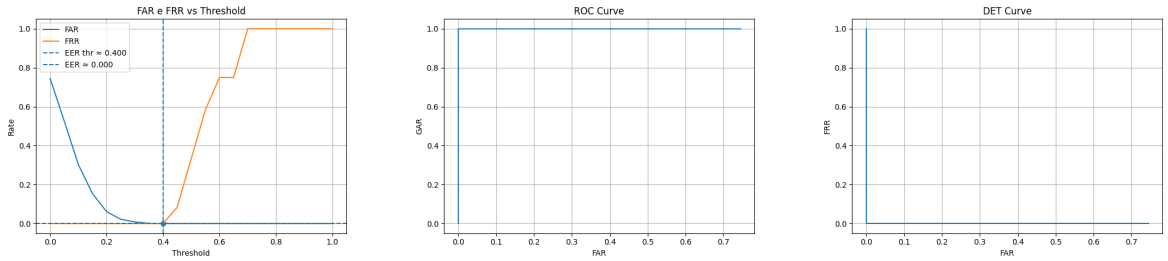
Figure 15: Age Best Match: Zac Efron; similarity score: 0.7523.

Figure 16: Age Worst Match: Queen Elizabeth; similarity score: 0.2571.

5.2.2 Camouflage

The results on the camouflage category show that the model remains stable even when the face is partially covered or subject to color alterations. Both FAR and FRR decrease rapidly, and the Equal Error Rate (EER) is close to zero. The ROC curve presents a high GAR across all FAR values, while the DET curve confirms that errors remain limited (Fig. 17). It is worth noting that the camouflage subset represents an

attack scenario, where the user actively attempts to alter their appearance to mislead the verification system. Despite this intent, the model consistently distinguishes genuine users, showing robustness against appearance manipulation and partial occlusions (Fig. 18). However, when the face is almost entirely covered, as in the worst-case example, the similarity score drops significantly (Fig. 19). This indicates that the model loses discriminative power when key facial regions are occluded, confirming that it mostly rely on that.



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 17: Evaluation results on the *camouflage* variation subset.

Figure 18: Camouflage Best Match: Harry Styles; similarity score: 0.6961.

5.2.3 Make up

The make up category shows that the model remains reliable. The FAR decreases quickly and remains close to zero for most thresholds, while the FRR drops more slowly but still reaches low values. The Equal Error Rate (EER) is around 0.10–0.15, which indicates that the model is slightly more sensitive to this type of variation compared to the age or camouflage categories. The ROC curve confirms that the system can correctly recognize most identities while the DET curve also shows a moderate number of false rejections at low thresholds, but the error becomes small as the threshold increases (Fig. 20).

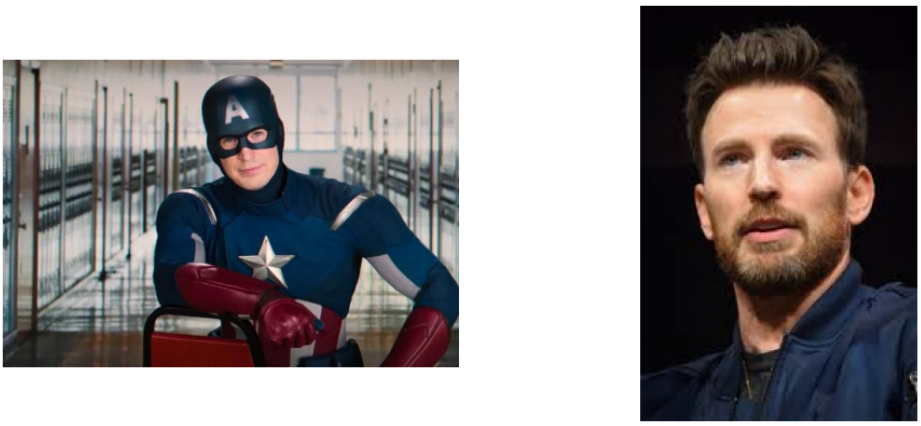
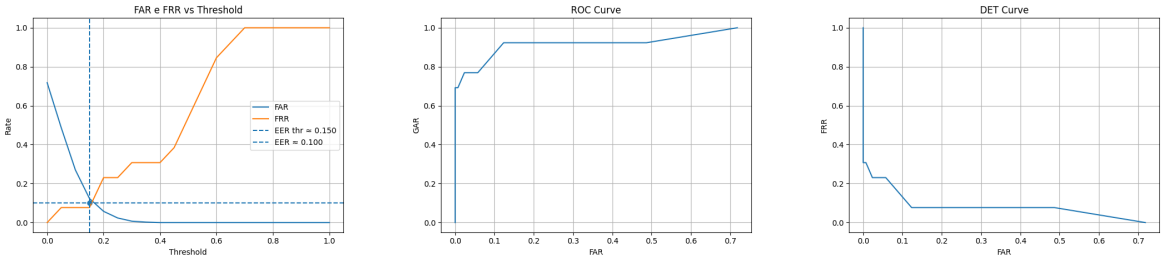


Figure 19: Camouflage Worst Match: Chris Evans; similarity score: 0.3594.

Among the categories tested, make-up appears to be the most challenging. In fact, make-up can drastically alter multiple facial attributes at once, such as skin texture, shading, color tones and even the apparent shape of the face. These changes affect several features that deep models use to build embeddings, increasing intra-class variability.

The best-case example (Fig. 21) shows that the model can still recognize the subject when make-up remains within natural ranges. In contrast, the worst-case (Fig. 22) example involves extreme make-up and costuming, which heavily modifies the entire facial appearance, resulting in a very low similarity score. Such cases highlight the limitations of face verification under strong aesthetic transformations.



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 20: Evaluation results on the *make up* variation subset.

5.2.4 Plastic Surgery

The plastic surgery category is more challenging because the person's appearance can change significantly. Even in this case, the model keeps a very low error rate. The Equal Error Rate (EER) is around 0.01, which means that the system is still able to recognise most identities correctly. FAR remains close to zero for almost all thresholds, while FRR decreases steadily and reaches very low values.

The ROC curve shows a high GAR and the DET curve confirms that false rejections

Figure 21: Make Up Best Match: Kylie Jenner; similarity score: 0.6662.

Figure 22: Make Up Worst Match: Robin Williams; similarity score: 0.0232.

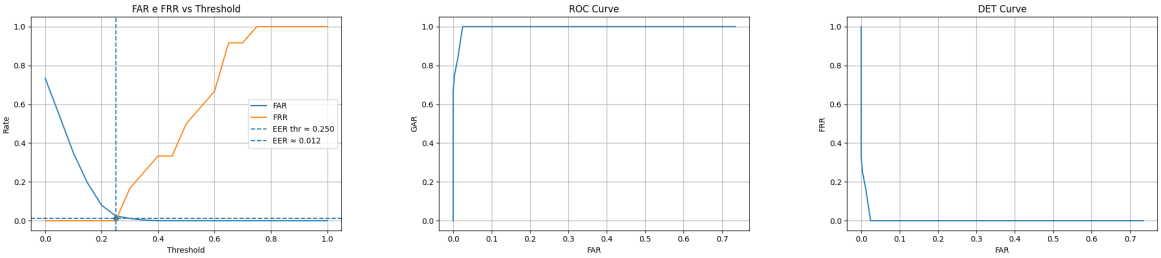
occur only at lowest thresholds. Overall, the model handles plastic surgery variations better than expected, thanks to the strong discriminative power of the ArcFace- style embeddings (Fig. 23).

The best-case match (Fig. 24) shows that when surgical interventions are subtle and do not alter the global facial structure, the model maintains a highly stable representation. The two images share the same overall geometry, including jawline, eye spacing and mid-face proportions, which allows the embedding to remain coherent despite possible minor cosmetic procedures. As a result, the similarity score remains very high.

In the worst-case match (Fig. 25), the subject exhibits noticeable modifications around key facial regions (skin tension, volume...). These changes can alter the geometry of the face and the distribution of fine-grained textures, producing a larger shift in the embedding space and resulting in a significantly lower similarity score.

5.2.5 Inter-Personal

The inter-personal category is one of the most difficult, because it includes different people who may look similar to each other. Despite this, the model shows strong



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 23: Evaluation results on the *plastic surgery* variation subset.

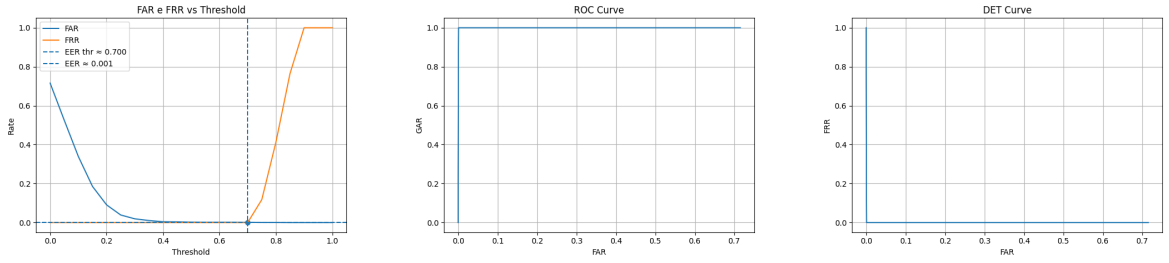
Figure 24: Plastic Surgery Best Match: Zac Efron; similarity score: 0.7205.

Figure 25: Plastic Surgery Worst Match: Lindsay Lohan; similarity score: 0.1889.

performance. Both FAR and FRR stay very close to zero across most thresholds, and the Equal Error Rate (EER) is extremely low (0.001). The ROC curve reaches GAR values near 1 almost everywhere, and the DET curve confirms that both types of errors are rare (Fig. 26).

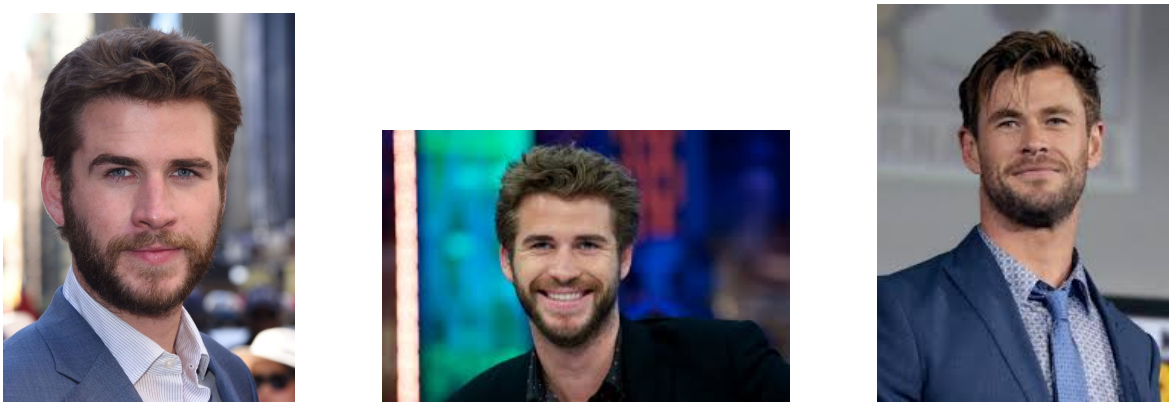
These results suggest that the model can clearly separate different identities, even when they share similar facial traits. This is consistent with the behaviour of ArcFace-style embeddings which are trained to maximise inter-class separation.

Qualitative examples confirm this behaviour. When the probe identity is present in the gallery (Fig. 27), even subjects who share strong familial resemblance produce clearly separated embeddings. Conversely, when the probe identity is absent from the gallery (Fig. 28), the model still assigns the highest similarity to the most visually similar subject, demonstrating how inter-personal resemblance can reduce the margin between genuine and impostor scores.



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 26: Evaluation results on the *interpersonal* variation subset.



Probe
Liam Hemsworth

Genuine gallery identity
Liam Hemsworth

Top impostor match
Chris Hemsworth

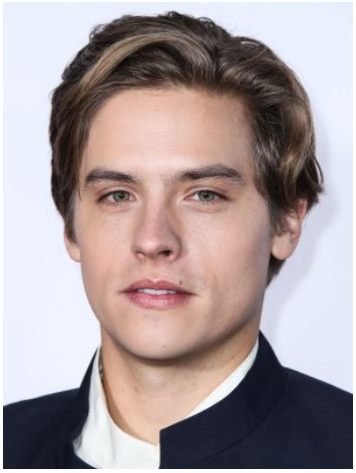
Figure 27: Inter-personal example where the probe identity is present in the gallery. For the probe sample, the highest genuine similarity with its gallery template is 0.8281. The highest impostor similarity obtained with a different gallery identity (brother) is 0.4920.

5.2.6 Print attack

The print attack category contains printed images of the correct identity. From an identity verification perspective, the model performs extremely well: FAR and FRR remain near zero, the EER is 0.0, and both ROC and DET curves indicate almost no errors (Fig. 29).

These results, however, expose a clear security weakness. Although the system correctly matches printed photos, such behaviour is undesirable in real biometric

Probe
Cole Sprouse (not in gallery)

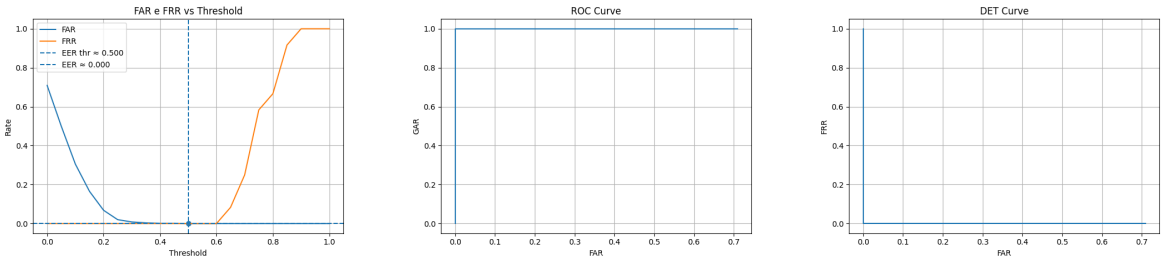


Top impostor match
Dylan Sprouse

Figure 28: Inter-personal example where the probe identity is not enrolled in the gallery. The highest similarity is obtained with another gallery subject (twin brother) with a similarity of 0.6659, resulting in a pure impostor match.

applications, where a printed image should be rejected as a spoof. This confirms that the model focuses solely on identity similarity and must therefore be complemented by dedicated liveness or anti-spoofing modules.

The best-case example (Fig. 30) shows a printed image that is almost identical to the reference one in terms of facial structure, expression and global appearance, leading to a very high similarity score. Conversely, in the worst-case match (Fig. 31), the printed photo differs more from the reference image due to age-related changes and alterations, which lowers the similarity.



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 29: Evaluation results on the *print attack* variation subset.

5.2.7 Deep Fake

The deep fake category contains synthetic images that still represent the same identity of the gallery. For the purpose of identity verification, the model works very well: both FAR and FRR quickly approach zero, and the EER is extremely low (0.001). This

Figure 30: Print Attack Best Match: Leonardo Di Caprio; similarity score: 0.8629.

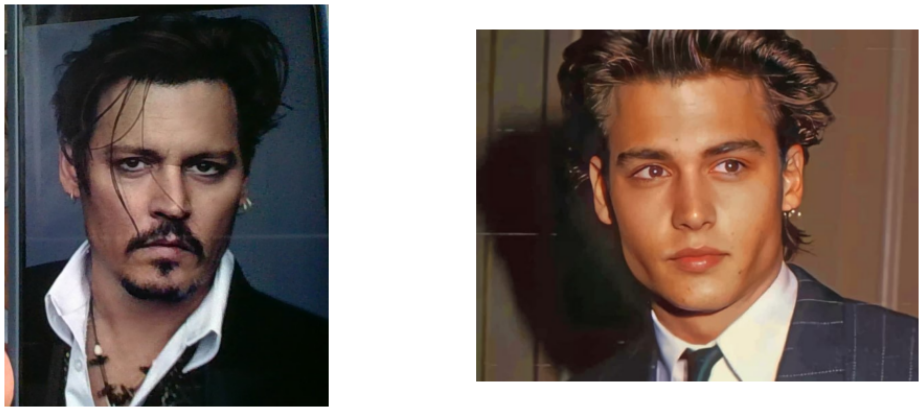


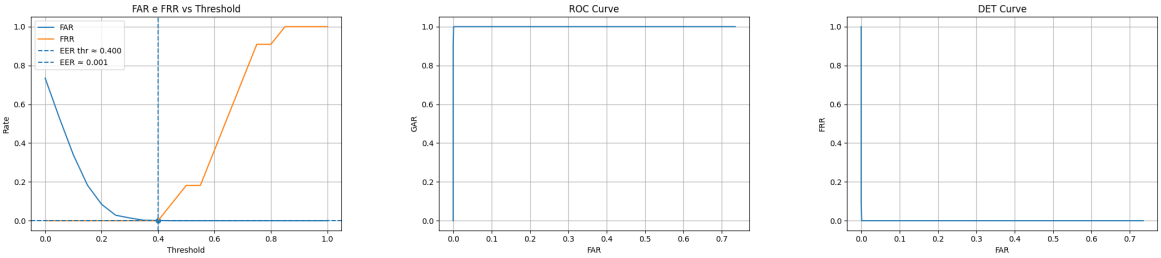
Figure 31: Print Attack Worst Match: Johnny Depp; similarity score: 0.4295.

means that the system is able to correctly recognise the person even when the image is artificially generated. The ROC and DET curves confirm this stable behaviour (Fig. 32).

However, these results also reveal an important vulnerability. Although the model recognises the identity correctly, a deepfake image should be considered a spoofing attack in real-world biometric systems. Since the model accepts synthetic faces of the correct identity, an attacker could exploit a deepfake to impersonate a user. So face verification alone is not sufficient for security, and it should be combined with dedicated anti-spoofing or liveness detection methods. As in the print attack category, the best (Fig. 33) and worst (Fig. 34) matches are mainly determined by natural appearance variations, rather than by the synthetic nature of the images.

5.2.8 Threshold variability across subsets

An important aspect emerging from the experiments is that the optimal decision threshold is not the same across the different subsets. Each category introduces its own type of appearance variation and this directly influences the distribution of genuine



(a) FAR vs FRR (b) ROC Curve (c) DET Curve

Figure 32: Evaluation results on the *deep fake* variation subset.

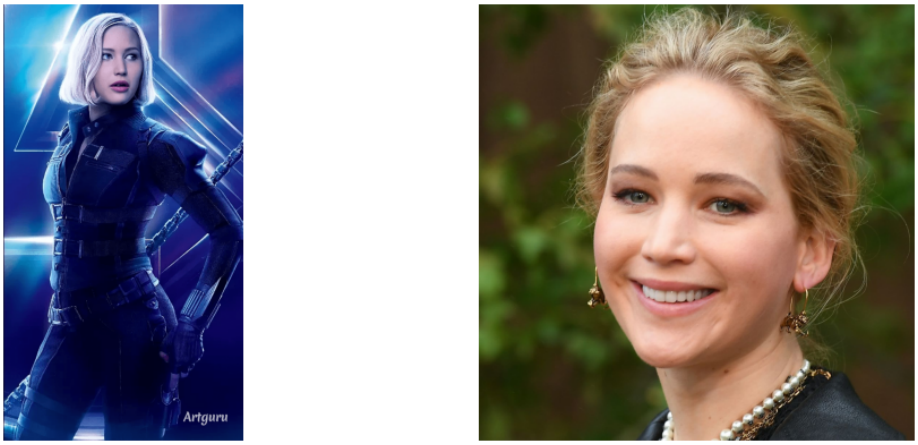


Figure 33: Deep Fake Best Match: Jennifer Lawrence; similarity score: 0.8629.

Figure 34: Deep Fake Worst Match: Johnny Depp; similarity score: 0.4295.

and impostor similarity scores (Fig. 35).

For example, subsets such as *age*, *camouflage* and *inter-personal* show a good separation between genuine and impostor pairs, which results in higher optimal thresholds (0.300, 0.400 and 0.700 respectively). In contrast, the *make-up* subset produces strong changes in texture and colour that reduce the genuine similarity scores, leading to a much lower threshold (0.150). Intermediate categories such as *plastic surgery* (0.250) fall in between these behaviors.

Spoofing subsets like *print attack* (0.500) and *deepfake* (0.400) also show high

thresholds, since the model still recognizes the identity and does not detect liveness.

Overall, the thresholds span a wide range (0.150–0.700), confirming that a single fixed value would not be reliable across heterogeneous conditions and motivating the use of adaptive thresholds or dedicated anti-spoofing modules.

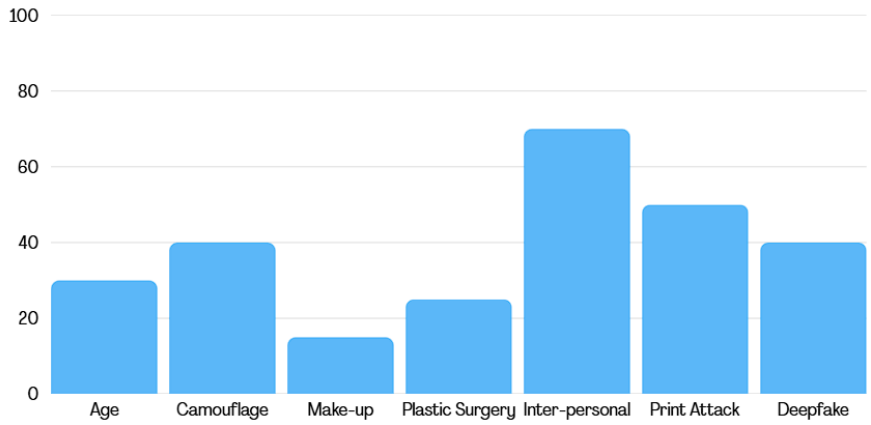


Figure 35: EER thresholds for each variation subset. The large variability (0.150–0.700) confirms that a single fixed threshold is not suitable across heterogeneous conditions.

5.3 Relation to Doddington's Zoo

The performance differences observed across the variation subsets can also be explained in terms of Doddington's Zoo, a classical model used to describe how different users influence the score distributions of a biometric system.

In this taxonomy, "goats" are users who tend to produce low genuine scores, making them harder to verify. This behavior appears in the *make-up* and *extreme age* cases, where the model sometimes struggles to match the probe image with its gallery template. These conditions increase the intra-class variability and temporarily turn normal subjects into goat-like users.

On the opposite side, Doddington's Zoo defines "lambs" as users who can be easily impersonated and therefore tend to produce high impostor scores. Although spoof images cannot be classified as "lambs" in the strict sense, their effect on the impostor score distribution resembles lamb-like behaviour and highlights an increased vulnerability to presentation attacks.

"Wolves" are users capable of producing high impostor similarities, often due to strong resemblance to others. This is reflected in the *inter-personal* subset, where look-alike individuals (e.g., siblings or actors with similar features) occasionally obtain relatively high impostor scores. Although the model handles these cases well, the reduced separation between genuine and impostor scores is consistent with wolf-like matching behaviour.

Finally, the majority of samples in the *age*, *camouflage* and *plastic surgery* subsets behave like "sheep": they generate stable and well-separated scores, allowing the model to achieve very low error rates. These cases show that the system works reliably.

6 Conclusions

The experiments show that a Vision Transformer combined with ArcFace embeddings can achieve very good face verification performance, even when the appearance of the person changes. In most categories, the error rates are very low, and both the ROC and DET curves indicate that the system clearly separates genuine and impostor samples. This confirms that the combination of proper face alignment and ArcFace-style features is very effective.

Looking at the different subsets, the model works especially well with natural variations such as age, moderate make-up, and camouflage. In these cases, the similarity scores remain stable, and the optimal thresholds are low. More difficult conditions, such as heavy make-up or major plastic surgery, reduce the similarity between images of the same person, but the performance remains good overall. This shows that the embeddings are still reliable even when the face changes significantly.

The spoofing categories (print attack and deepfake) reveal a different issue. Even though the model recognizes the correct identity, it also accepts printed photos and synthetic faces as genuine. This means that the system cannot detect presentation attacks and should not be used alone in security-critical scenarios. Dedicated liveness or anti-spoofing methods are necessary to prevent these attacks.

In conclusion, this work shows that Vision Transformers with ArcFace alignment are strong feature extractors for face verification. However, the results also highlight that additional components, such as spoof detection, image quality checks and adaptive thresholds, are essential to make the system robust and secure in real-world biometric applications.

References

[1] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

[2] Gau-Nernst. vit_tiny_patch8_112.arcface_ms1mv3. `https://huggingface.co/gaunernst/vit_tiny_patch8_112.arcface_ms1mv3`, 2023. Accessed: 2025-11-20.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Ander Galván, Mariví Higuero, Ane Sanz, Asier Atutxa, Eduardo Jacob, and Mario Saavedra. Comparing cnn and vit for open-set face recognition. *Electronics*, 14(19), 2025.

[5] Jiankang Deng, Jia Guo, Debing Zhang, Yafeng Deng, Xiangju Lu, and Song Shi. Lightweight face recognition challenge. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2638–2646, 2019.

[6] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, October 2022.

[7] Tee Kai Feng. Understanding arcface loss: Intuitive insights and its application for representation learning. `https://medium.com/@teekaifeng/understanding-arcface-loss-intuitive-insights-and-its-application-for-representat` April 2024. Accessed: 2025-11-20.

[8] DeepInsight and Contributors. Insightface: 2d and 3d face analysis project. `https://github.com/deepinsight/insightface`, 2023. Accessed: 2025-11-20.

[9] Gau-Nernst. test_ijb.py from the timm-face repository. `https://github.com/gau-nernst/timm-face/blob/main/test_ijb.py`, 2021. Accessed: 2025-11-20.