Adversarial Attacks on Face Recognition Systems

1st Shambhavi Jha
220953634
Department of ICT
Manipal Institute of Technology
Email: shambhavi.mitmpl2022@learner.manipal.edu

2nd Archit Singh
220911464
Department of ICT
Manipal Institute of Technology
Email: archit2.mitmpl2022@learner.manipal.edu

3rd Lakshay Prasher
220911011
Department of ICT
Manipal Institute of Technology
Email: lakshya.mitmpl2022@learner. manipal.edu

4th Krish A Manchanda
220953660
Department of ICT
Manipal Institute of Technology
Email: krrish.mitmpl2022@learner. manipal.edu

***Abstract*—Systems for face recognition have diverse applications, ranging from security to surveillance to identity verification. Such systems, however, can be attacked by opponents who may add small, imperceptible perturbations to the images in order to mislead the AI models. This project tackles the weakness of deep learning models—in this case, facial recognition models such as FaceNet [1]—to adversarial attacks. The main issue is that small, imperceptible to humans perturbations applied to input images can greatly impair the model's performance or have it misclassify inputs entirely. Such adversarial examples are serious security vulnerabilities for biometric authentication systems.**

The architecture being proposed makes use of the pretrained FaceNet [1] model as a feature extractor to create face embeddings. Adversarial samples are designed through Fast Gradient Sign Method (FGSM) [2], DeepFool [3], Projected Gradient Descent (PGD) [4] and Basic Iterative Method(BIM) [5] attacks using the Foolbox library [6]. The goal is to modify the face image in a subtle way so that the modified one varies semantically for the model even though it appears almost the same to humans. The findings illustrate that FGSM,PGD and BIM successfully decrease embedding similarity, asserting successful attacks. For example, FGSM [2] decreased similarity to 0.94, PGD [4] further deteriorated it and BIM [5] attack decreased it down to 0.85. DeepFool [3] had negligible effect with implementation problems or slight perturbation strength. These results demonstrate that even strong face recognition systems can be broken, highlighting the necessity of adversarial robustness in real-world applications.

***Index Terms*—adversarial attacks, face recognition, MTCNN, DeepFool, FGSM, BIM, PGD**

I. Introduction

Face recognition technology has become a central component of contemporary computer vision, allowing for secure and effective identification in many real-world scenarios. From mobile phone unlocking and social media photo tagging to surveillance, law enforcement, and security at airports, these technologies depend on sophisticated deep neural network models to identify and match distinctive facial characteristics. One of the best-known models in this area is FaceNet [1], which maps facial images into short embeddings inside a high-dimensional space. These embeddings encode the distinctive identity of a face, and similarity can be calculated in terms of geometric distance, often through the use of metrics like Euclidean distance or cosine similarity [7].

Although they have high accuracies in ideal conditions, face recognition systems are becoming increasingly susceptible to a new and insidious class of threat called adversarial attacks. Adversarial attacks consist of crafted perturbations introduced into input images, usually too small to be noticed by the human eye but capable of severely misleading deep neural networks. Adversarial examples, in the context of facial recognition, have the potential to make systems misrecognize people or enable intruders to impersonate genuine users, with far-reaching security and privacy implications.

A number of adversarial attacks have been created, each with varying mechanisms and effects. Fast Gradient Sign Method (FGSM) [2] is a simple, one-step attack that uses the model's gradient to generate perturbations. Projected Gradient Descent (PGD) extends FGSM by using it iteratively with projection constraints, typically making it stronger. Basic Iterative Method (BIM) [5] is another iterative version of FGSM that progressively increases perturbations over steps. DeepFool [3] operates in a different manner by computing the smallest perturbation needed to shift an input past the classifier's decision boundary. Aside from these,

more sophisticated attacks like Carlini-Wagner (CW), AutoAttack, and One-Pixel Attack also show how innovative and varied adversarial techniques can be.
This wider weakness in neural networks highlights the necessity of robustness testing of face recognition systems. Knowing how such models behave when exposed to various types of adversarial attacks is necessary, particularly in high-risk settings where safety and privacy are crucial.
With this in mind, the objective of this project is to explore systematically the impact of various adversarial attacks on a face recognition model, in this case FaceNet [1]. The research investigates how various attack methods influence both the visual integrity of the facial images as well as the internal embeddings the model uses to recognize them. It also employs a suite of quantitative measures—like similarity scores, perceptual quality judgments, and perturbation values—to evaluate and compare the efficacy of each attack. This analysis aids in determining the most effective attacks in being able to mislead the model and sheds light on the vulnerabilities of facial recognition systems when in adversarial settings.

The present paper is divided into seven sections: A literature survey, Section 2, reviews previous works in the adversarial field of face recognition, analyzing attack methodologies, and their effect on many applications using deep learning. Section 3: Works objectives outline the principal objectives of the project and provide an overview of examing each adversarial attack in face recognition. Section 4: Dataset illustrates this dataset, from CelebA, with preprocessing techniques such as resizing, normalization, and alignment. The basis of operation is presented in Section 5: Methodology, in a step-wise implementation of the whole project including detection and recognition of faces, adversarial attack generation using FGSM [2] and DeepFool [3], and evaluation of the performance of the model under attack. Section 6 deals with findings, exposes vulnerabilities detected in face recognition models, and proposes defenses. Finally, Section 7 lists all research papers, books, and other internet sources consulted for this study.

A. Rationale and Research Aims

- To emphasize weaknesses in facial recognition systems by showing how minor adversarial perturbations can profoundly change the embeddings utilized for identity verification.
- To compare and analyze the performance of various adversarial attack methods in changing FaceNet embeddings using a uniform similarity metric.
- To lay the groundwork for creating stronger and more secure face recognition systems based on how current models react to adversarial inputs and where there might be potential areas of defense.

II. Related work

Several studies have progressed face recognition with deep learning, but more recent studies have shown just how susceptible such systems are to adversarial attacks. This section summarizes briefly the advancements in face recognition models and the emerging work around adversarial robustness.

A. Face Recognition Models

Face recognition has progressed incredibly, largely due to breakthroughs in deep learning. FaceNet [1] transformed the industry by placing facial images within a small Euclidean space with a triplet loss, allowing for precise recognition by employing basic distance metrics. DeepFace [8] provided near-human precision prior to that with a deep network trained on large numbers of images. VGGFace [9] and ArcFace [10] increased the performance even further, with ArcFace adding angular margin loss to enhance discrimination. In spite of their performance, Dong et al. [11] in their research indicated that such models are extremely susceptible to adversarial attacks, which can use minute pixel-level variations to mislead the system—demonstrating an urgent need for robustness in security-critical applications.

B. Adversarial Attacks in Computer Vision

Adversarial attacks have become a serious weakness in computer vision systems, where minor, well-designed perturbations are able to deceive even state-of-the-art deep neural networks. Goodfellow et al. originally presented the Fast Gradient Sign Method (FGSM) [2], showing how linearity in high-dimensional space can be leveraged to deceive classifiers. Since then, there have been several more effective attacks, such as Basic Iterative Method [5], Projected Gradient Descent [4], and DeepFool [3], all of which advance optimization methods to make attacks stronger while still imperceptible to humans. Such attacks have revealed fundamental vulnerabilities in vision systems applied in security, healthcare, and autonomous driving. Current studies highlight not just the design of attacks but also the effects of these attacks on embeddings and feature representations, especially in applications such as facial recognition where small distortions can result in identity misclassification or unauthorized access.

C. Adversarial Attacks on Face Recognition

Face recognition systems are especially susceptible to adversarial attacks because they depend on fine-grained, subtle facial features. Even tiny perturbations suffice to drastically change the resulting face embeddings and result in misidentification or impersonation. Sharif et al.'s [12] research showed how attackers might employ accessories such as eyewear to deceive face recognition models in the physical world. Follow-up studies have revealed that such attacks as FGSM, PGD, and DeepFool can downgrade recognition performance or compel systems to misclassify identities. This presents grave concerns for security-critical use cases where facial authentication is prevalent, underlining the imperative need for resilient face recognition models that are unsusceptible to adversarial tampering.

D. Evaluation Metrics

To evaluate systematically the effect of adversarial perturbations on model performance, a range of evaluation metrics have been utilized in the literature. Cosine similarity has been applied to measure semantic consistency between original and adversarial embeddings, reflecting whether identity-preserving characteristics are being preserved [12]. Visual quality deterioration is usually assessed based on parameters like Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR), which indicate how imperceptible the perturbations are to human vision [13]. Furthermore, L2 distance quantifies embedding displacement in the feature space, while L-infinity norm (maximum perturbation value) represents the most important pixel-level variation [14]. In combination, these measurements provide a complete picture of both visual quality and representational accuracy under threat.

III. Addressed Research Gaps

A. Limited Evaluation on Embedding-Level Impact

While the majority of current studies on adversarial attacks in face recognition target either the classification performance or decision boundary deception, few explore how the perturbations impact the underlying face embeddings generated by models such as FaceNet [1]. The embeddings are the underlying representation in face verification and clustering applications, and any distortion here can cause identity mismatches across systems. Previous researches chiefly quantified the misclassification rate but did not investigate how the adversarial noise alters the space of semantic representations. This piece of work aims to fill that gap by evaluating the impact of attacks on representation vectors directly and providing a more detailed picture of adversarial weakness in representation learning.

B. Underuse of Multi-Metric Image Similarity

Most existing work concentrates on one evaluation measure, usually classification accuracy or cosine similarity [7] between the embeddings, that is not comprehensive enough to evaluate the complete impact of adversarial noise [15]. By contrast, this work employs a complete set of evaluation metrics—Cosine Similarity, Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), L2 norm, and L-infinity norm—to quantitatively measure both visual quality degradation and embedding disruption. This multi-metric framework allows for more in-depth and balanced assessment of attack efficacy across perceptual and semantic spaces.

C. Lack of Comparative Analysis

Though multiple attack methods like FGSM, DeepFool, and PGD have been introduced, they are generally compared in isolation or in classification problems (Carlini Wagner, 2017; Madry et al., 2018). No comparative study exists to compare and contrast the impact that these attacks, with their different optimization methods and respective strengths, place on the output embeddings and visual quality in a face recognition scenario. This work bridges that gap by comparing and contrasting four of the most common attacks—FGSM, PGD, DeepFool, and BIM—on the same model in the same settings, providing an integrated empirical analysis of their impacts.

IV. Dataset

We are using CelebA (CelebFaces Attributes Dataset) for this project, one of the most extensive face datasets, which has gained prominence in research tasks involving computer vision, facial recognition, and adversarial research. The dataset contains more than 200,000 face pictures belonging to 10,177 celebrities and annotated as regards 40 facial attributes such as gender, age, facial expressions, and accessories (like glasses, hats). These were collected from various online sources to assure diversity in illumination, poses, backgrounds, and expressions, making them perfectly fit the robustness testing of face recognition models.

CelebA is selected for this project primarily because it boasts such a rich collection of high-resolution facial images, with major variations in pose, background, and lighting conditions, which would allow us to study the effect of adversarial attacks on face-recognition models under realistic settings, and also secondarily, because it contains aligned cropped facial images that may be straightened to simplify face detection and feature extraction.

V. Methodology

The goal of this research is to assess the resilience of the FaceNet face recognition model against different types of adversarial attacks. These attacks make slight changes to input images to deceive deep learning models without any apparent changes to the human eye. The approach includes a number of important steps: data preparation, generation of adversarial attacks, model testing, and effect assessment using various metrics.

The initial step is to upload a public facial dataset, in this example CelebA, with labeled face images. These face images are then converted into high-dimensional embeddings by the FaceNet model [1]. Then, we create adversarial examples through techniques such as FGSM, PGD, BIM, and DeepFool. These attacks add small but carefully placed perturbations to deceive the model. After generating the adversarial images, we measure their effect through quantitative and qualitative metrics. These measures enable us to evaluate the semantic and visual degradation resulting from the attacks, measuring the distortion in the embeddings and image quality. Finally, the adversarial samples are tested against the original image, and the outcomes are examined to determine how each attack impacts the model's performance and image quality. This approach is useful in deriving insights into FaceNet's weaknesses and vulnerability to adversarial attacks, which aid in the enhancement of model robustness in security-critical tasks.As shown in Figure 1 following are the detailed steps:

A. Face Recognition System Setup

This project employs the FaceNet model [1] for face recognition, which embeds images of faces into a small embedding space where the points are close together if the faces are similar. FaceNet [1] uses a deep convolutional neural network (CNN) architecture trained on a triplet loss function that pushes embeddings of the same identity close to each other and those of different identities away from each other. First, the image is required to pass through face detection and alignment from MTCNN (Multi-task Cascaded Convolutional Networks), a deep face detector used to detect principal facial landmarks for centering the face and resizing it appropriately before passing the face image through the network to output a 128-dimensional embedding vector as shown in Algorithm 1.

Algorithm 1 Generate Face Embedding using FaceNet

```
Require: Face image tensor face_tensor $\in \mathbb{R}^{3\times H\times W}$
Load pre-trained model: facenet $\leftarrow$ InceptionResnetV1('vggface2').eval()
Convert input to float32: face_tensor $\leftarrow$ face_tensor.to(torch.float32)
function GetFaceEmbedding(face_tensor)
    Disable gradient: with torch.no_grad()
        emb $\leftarrow$ facenet(face_tensor)
        return emb.numpy().flatten()
end function
embedding $\leftarrow$ GetFaceEmbedding(face_tensor)
Print first 10 values: embedding[:10] =0
```

B. Dataset and Preprocessing

CelebA dataset is utilized in this project that holds more than 200,000 images of celebrities with varied facial features, angles, and lighting. Before being fed into the FaceNet model, the images undergo a series of preprocessing operations. MTCNN [16] first detects and aligns faces in each image so that the orientation and size of faces are consistent. The face areas cropped are resized to 160×160 pixels, the input size FaceNet expects. The pixel values are then normalized to the range [-1, 1], which aids in model performance and convergence by normalizing the input distribution as described in Algorithm 2. This neat and uniform input format guarantees that the embeddings produced by FaceNet are accurate and meaningful.

Algorithm 2 Preprocess Input Image for FaceNet

```
Require: RGB or grayscale image at path image_path
Load image: image $\leftarrow$ Image.open(image_path)
if image mode $\neq$ 'RGB' then
    Convert to RGB: image $\leftarrow$ image.convert('RGB')
end if
Define transform: Resize $\rightarrow (160, 160)$, ToTensor, Normalize
Apply transform: face_tensor $\leftarrow$ transform(image)
Add batch dimension: face_tensor $\leftarrow$ face_tensor.unsqueeze(0) =0
```

C. Adversarial Attacks Methods and Sample Generation

To measure the face recognition model's vulnerability, a number of popular adversarial attack techniques are used as shown in Algorithm . These techniques produce small, imperceptible perturbations to input images that lead the model to misclassify them.

- Fast Gradient Sign Method (FGSM): FGSM perturbs the image in the direction of the loss function gradient with respect to the input as . It performs a single step to calculate:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \tag{1}$$

where ϵ controls the intensity of perturbations.

- DeepFool Attack: DeepFool calculates the smallest perturbation required to alter the classification output. It presumes a locally linear decision boundary and iteratively perturbs the image until the decision of the classifier is altered.

$$x_{i+1} = x_i - \frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \cdot \nabla f(x_i) \tag{2}$$

  where x_i and x_{i+1} are the input and updated adversarial examples respectively, $f(x_i)$ is the classifier's output, and $\nabla f(x_i)$ is its gradient with respect to x_i, with $|\nabla f(x_i)|_2$ denoting the L2 norm of the gradient.

- PGD Attack: It is a multi-step extension of the FGSM attack, where small steps of FGSM are applied iteratively and the resulting adversarial example is projected back onto an L_∞ ball of radius ϵ around the original image x:

$$x_{t+1} = \Pi_{B_\epsilon(x)} \left(x_t + \alpha \cdot \text{sign} \left(\nabla_x J(\theta, x_t, y) \right) \right) \tag{3}$$

  Here, $\Pi_{B_\epsilon(x)}(\cdot)$ denotes the projection operator onto the ϵ-ball centered at x, and α is the step size. PGD is widely regarded as one of the strongest first-order adversarial attacks.

- BIM attack: It is an iterative variant of FGSM that applies the gradient sign method multiple times with small step sizes. Unlike PGD, BIM uses clipping instead of projection to ensure that the perturbation stays within the valid pixel range and within an ϵ-neighborhood of the original image:

$$x_{t+1} = \text{Clip}_{x,\epsilon} \left(x_t + \alpha \cdot \text{sign} \left(\nabla_x J(\theta, x_t, y) \right) \right) \tag{4}$$

  The clipping function $\text{Clip}_{x,\epsilon}(\cdot)$ ensures that the perturbed image remains within ϵ of the original image x and that all pixel values stay within the allowable image intensity range (e.g., [0,1]).

Adversarial samples are created by using these techniques on preprocessed facial images. Perturbed versions are created for each clean image using FGSM, BIM, PGD, or DeepFool. Even though they look similar to human eyes, these adversarial examples greatly misalign the FaceNet embeddings, causing them to be recognized incorrectly. All perturbed samples are stored with clean labels, allowing for direct comparison between clean and adversarial embeddings. This supports both quantitative assessment and visualization of the perceptual and semantic deterioration caused by every attack technique. Algorithm 3 demonstartes the same.

Algorithm 3 Adversarial Attack Tensor Generation using Foolbox

```
Input: Preprocessed image tensor $x$, ground truth label $y$
Initialize Foolbox model $fmodel$ with FaceNet and preprocessing parameters
Define attack strength $\epsilon$
function GENERATE_FGSM_ATTACK($x$, $y$, $\epsilon$)
    Initialize FGSM attack
    $(x_{\text{raw}}, x_{\text{adv}}, success)$ ← FGSM($fmodel$, $x$, $y$, $\epsilon$)
    Print success status and tensor difference
    return $x_{\text{adv}}$
end function
function GENERATE_DEEPFOOL_ATTACK($x$, $y$, $\epsilon$)
    Initialize DeepFool attack
    $(x_{\text{raw}}, x_{\text{adv}}, success)$ ← DeepFool($fmodel$, $x$, $y$, $\epsilon$)
    Print success status and tensor difference
    return $x_{\text{adv}}$
end function
function GENERATE_PGD_ATTACK($x$, $y$)
    Initialize PGD attack with $\epsilon = 0.1$
    Set criterion to Misclassification($y$)
    $(x_{\text{raw}}, x_{\text{adv}}, success)$ ← PGD($fmodel$, $x$, criterion, $\epsilon$)
    Extract $x_{\text{adv}}[0]$ for the given $\epsilon$
    Print success status and tensor difference
    return $x_{\text{adv}}[0]$
end function
function GENERATE_BIM_ATTACK($x$, $y$, $\epsilon$)
    Initialize BIM attack
    $(x_{\text{raw}}, x_{\text{adv}}, success)$ ← BIM($fmodel$, $x$, $y$, $\epsilon$)
    Print success status and tensor difference
    return $x_{\text{adv}}$
end function
Generate adversarial examples:
$x_{\text{fgsm}}$ ← GENERATE_FGSM_ATTACK($x$, $y$, 0.04)
$x_{\text{deepfool}}$ ← GENERATE_DEEPFOOL_ATTACK($x$, $y$, 100.0)
$x_{\text{pgd}}$ ← GENERATE_PGD_ATTACK($x$, $y$)
$x_{\text{bim}}$ ← GENERATE_BIM_ATTACK($x$, $y$, 0.1)
=0
```

D. Evaluating Attack Effectiveness

To assess how well the adversarial attacks affected the face recognition system, several metrics were employed, such as recognition accuracy and various measures of similarity between original and adversarial images and embeddings.

- Cosine similarity: Cosine similarity [7] as demonstrated in Algorithm 4 , was employed to measure the 128-dimensional FaceNet embeddings of original and adversarial images. Lower cosine similarity reflects more deviation in facial representation by adversarial perturbations.

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{5}$$

,where A and B are the FaceNet embedding vectors for the original and adversarial images, respectively.

Algorithm 4 Cosine Similarity Computation Between Original and Adversarial Embeddings

```
Extract embedding $E_{orig} \leftarrow$ get_face_embedding(face_tensor)
Extract embedding $E_{fgsm} \leftarrow$ get_face_embedding(fgsm_tensor)
Extract embedding $E_{deepfool} \leftarrow$ get_face_embedding(deepfool_tensor)
Extract embedding $E_{pgd} \leftarrow$ get_face_embedding(pgd_tensor)
Extract embedding $E_{bim} \leftarrow$ get_face_embedding(bim_tensor)
function CosineSimilarity($e_1, e_2$)
    return $\frac{e_1 \cdot e_2}{\|e_1\| \cdot \|e_2\|}$
end function
Compute $S_{fgsm} \leftarrow$ CosineSimilarity($E_{orig}, E_{fgsm}$)
Compute $S_{deepfool} \leftarrow$ CosineSimilarity($E_{orig}, E_{deepfool}$)
Compute $S_{pgd} \leftarrow$ CosineSimilarity($E_{orig}, E_{pgd}$)
Compute $S_{bim} \leftarrow$ CosineSimilarity($E_{orig}, E_{bim}$)
Print "Original vs FGSM Attack Similarity:", $S_{fgsm}$
Print "Original vs DeepFool Attack Similarity:", $S_{deepfool}$
Print "Original vs PGD Attack Similarity:", $S_{pgd}$
Print "Original vs BIM Attack Similarity:", $S_{bim}$ =0
```

- Structural Similarity Index (SSIM): It measures perceived loss in image quality by assessing luminance, contrast, and structural similarity between original and adversarial images as shown in Algorithm 5. It returns a score between 0 and 1, where 1 signifies perfect copies.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{6}$$

,where μ_x, μ_y: means; σ_x^2, σ_y^2: variances; σ_{xy}: covariance; C_1, C_2: stability constants.

Algorithm 5 SSIM Between Original and Adversarial Images

```
Require: Original image tensor $I_{\text{orig}}$, adversarial tensors $I_{\text{fgsm}}, I_{\text{deepfool}}, I_{\text{pgd}}, I_{\text{bim}}$
function ComputeSSIM($I_1, I_2$)
    $I_1^{np} \leftarrow$ Convert $I_1$ to NumPy format and reshape to $(H, W, C)$
    $I_2^{np} \leftarrow$ Convert $I_2$ to NumPy format and reshape to $(H, W, C)$
    Clip both images to $[0, 1]$
    return SSIM($I_1^{np}, I_2^{np}$) with $channel\_axis = -1$
end function
$S_{\text{fgsm}} \leftarrow$ ComputeSSIM($I_{\text{orig}}, I_{\text{fgsm}}$)
$S_{\text{deepfool}} \leftarrow$ ComputeSSIM($I_{\text{orig}}, I_{\text{deepfool}}$)
$S_{\text{pgd}} \leftarrow$ ComputeSSIM($I_{\text{orig}}, I_{\text{pgd}}$)
$S_{\text{bim}} \leftarrow$ ComputeSSIM($I_{\text{orig}}, I_{\text{bim}}$)
Print "SSIM (Original vs FGSM):", $S_{\text{fgsm}}$
Print "SSIM (Original vs DeepFool):", $S_{\text{deepfool}}$
Print "SSIM (Original vs PGD):", $S_{\text{pgd}}$
Print "SSIM (Original vs BIM):", $S_{\text{bim}}$ =0
```

- Peak Signal-to-Noise Ratio:PSNR quantifies the quality of the adversarial image compared to the original as demonstrated in Algorithm 6. Smaller perturbation and greater visual similarity are represented by a larger PSNR value.

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{\text{MSE}}\right) \tag{7}$$

,where MAX_I is the maximum possible pixel value of the image (e.g., 255 for 8-bit images), and MSE is the Mean Squared Error between the original and adversarial images.

- L2 (Euclidean) Distance:It calculates the Euclidean distance between the original and adversarial FaceNet embeddings to measure the size of change in representation as shown in Algorithm 7.

$$\|x - x_{\text{adv}}\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - x_{\text{adv},i})^2} \tag{8}$$

,where x and x_{adv} are the embedding vectors of the original and adversarial images, respectively.

Algorithm 6 PSNR Between Original and Adversarial Images

```
Require: Original image tensor $I_{\text{orig}}$, adversarial tensors $I_{\text{fgsm}}, I_{\text{deepfool}}, I_{\text{pgd}}$
function ComputePSNR($I_1, I_2$)
    $I_1^{np} \leftarrow$ Convert $I_1$ to NumPy format and reshape to $(H, W, C)$
    $I_2^{np} \leftarrow$ Convert $I_2$ to NumPy format and reshape to $(H, W, C)$
    Normalize $I_1^{np} \leftarrow (I_1^{np} + 1)/2$
    Normalize $I_2^{np} \leftarrow (I_2^{np} + 1)/2$
    return PSNR($I_1^{np}, I_2^{np}$) with $data\_range = 1.0$
end function
$P_{\text{fgsm}} \leftarrow$ ComputePSNR($I_{\text{orig}}, I_{\text{fgsm}}$)
$P_{\text{deepfool}} \leftarrow$ ComputePSNR($I_{\text{orig}}, I_{\text{deepfool}}$)
$P_{\text{pgd}} \leftarrow$ ComputePSNR($I_{\text{orig}}, I_{\text{pgd}}$)
Print ”PSNR (FGSM):”, $P_{\text{fgsm}}$
Print ”PSNR (DeepFool):”, $P_{\text{deepfool}}$
Print ”PSNR (PGD):”, $P_{\text{pgd}}$ =0
```

Algorithm 7 L2 Distance Between Original and Adversarial Face Embeddings

```
Require: Original embedding vector $\mathbf{e}_{\text{orig}}$, and adversarial embeddings $\mathbf{e}_{\text{fgsm}}, \mathbf{e}_{\text{deepfool}}, \mathbf{e}_{\text{pgd}}, \mathbf{e}_{\text{bim}}$
function L2_Distance($\mathbf{e}_1, \mathbf{e}_2$)
    return $\|\mathbf{e}_1 - \mathbf{e}_2\|_2$
end function
$\mathbf{e}_{\text{orig}} \leftarrow$ GetFaceEmbedding(face_tensor)
$\mathbf{e}_{\text{fgsm}} \leftarrow$ GetFaceEmbedding(fgsm_tensor)
$\mathbf{e}_{\text{deepfool}} \leftarrow$ GetFaceEmbedding(deepfool_tensor)
$\mathbf{e}_{\text{pgd}} \leftarrow$ GetFaceEmbedding(pgd_tensor)
$\mathbf{e}_{\text{bim}} \leftarrow$ GetFaceEmbedding(bim_tensor)
$D_{\text{fgsm}} \leftarrow$ L2_Distance($\mathbf{e}_{\text{orig}}, \mathbf{e}_{\text{fgsm}}$)
$D_{\text{deepfool}} \leftarrow$ L2_Distance($\mathbf{e}_{\text{orig}}, \mathbf{e}_{\text{deepfool}}$)
$D_{\text{pgd}} \leftarrow$ L2_Distance($\mathbf{e}_{\text{orig}}, \mathbf{e}_{\text{pgd}}$)
$D_{\text{bim}} \leftarrow$ L2_Distance($\mathbf{e}_{\text{orig}}, \mathbf{e}_{\text{bim}}$)
Print ”L2 Distance FGSM:”, $D_{\text{fgsm}}$
Print ”L2 Distance DeepFool:”, $D_{\text{deepfool}}$
Print ”L2 Distance PGD:”, $D_{\text{pgd}}$
Print ”L2 Distance BIM:”, $D_{\text{bim}}$ =0
```

- Perturbation Magnitude:The magnitude of perturbation quantifies the total amount or norm of the noise vector that is applied to the original image to obtain the adversarial version, demonstrated in Algorithm 8

$$\|\delta\|_2 = \sqrt{\sum_{i=1}^{n} \delta_i^2} \tag{9}$$

,where $\delta = x_{\text{adv}} - x$ is the adversarial perturbation.

Algorithm 8 Perturbation Magnitude between Original and Adversarial Images

```
Require: Original tensor $T_{\text{orig}}$, adversarial tensors $T_{\text{fgsm}}, T_{\text{deepfool}}, T_{\text{pgd}}$
function ComputePerturbationStats($T_1, T_2$)
    $D \leftarrow |T_2 - T_1|$ {Absolute difference}
    $L_\infty \leftarrow \max(D)$ {Maximum (L-infinity) norm}
    $\mu \leftarrow \text{mean}(D)$ {Mean difference}
    $L_2 \leftarrow \|D\|_2$ {L2 norm}
    return $L_\infty, \mu, L_2$
end function
$L_\infty^{fgsm}, \mu^{fgsm}, L_2^{fgsm} \leftarrow$ ComputePerturbationStats($T_{\text{orig}}, T_{\text{fgsm}}$)
$L_\infty^{deepfool}, \mu^{deepfool}, L_2^{deepfool} \leftarrow$ ComputePerturbationStats($T_{\text{orig}}, T_{\text{deepfool}}$)
$L_\infty^{pgd}, \mu^{pgd}, L_2^{pgd} \leftarrow$ ComputePerturbationStats($T_{\text{orig}}, T_{\text{pgd}}$)
Print ”FGSM - Max Diff:”, $L_\infty^{fgsm}$, ”Mean Diff:”, $\mu^{fgsm}$, ”L2 Diff:”, $L_2^{fgsm}$
Print ”DeepFool - Max Diff:”, $L_\infty^{deepfool}$, ”Mean Diff:”, $\mu^{deepfool}$, ”L2 Diff:”, $L_2^{deepfool}$
Print ”PGD - Max Diff:”, $L_\infty^{pgd}$, ”Mean Diff:”, $\mu^{pgd}$, ”L2 Diff:”, $L_2^{pgd}$ =0
```

In the Figure 1, SSIM, PSNR, and Perturbation Magnitude are derived from comparisons amongst the raw input images (prior to FaceNet processing) since they measure visual and pixel-level alterations : SSIM measures structural similarity, PSNR does pixel fidelity, and Perturbation Magnitude measures direct pixel differences (δ = x_adv - x). These measures necessitate the original image space. On the other hand, Cosine Similarity and L2 Distance work upon FaceNet's 128-D embedding vectors, calculating semantic distortions in the feature space in which facial recognition truly takes place. This differentiation explains that image-quality metrics work upon pre-embedding data, whereas similarity metrics assess post-embedding relationships.

VI. Results

This section displays the outcomes of using four types of adversarial attacks—Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD), and DeepFool—on a face recognition system designed with the FaceNet architecture. Each attack add small perturbations to facial images with the intention of interfering with the model's capacity to properly identify or verify individuals. The efficacy of these attacks is measured using quantitative measures as well as qualitative comparisons by visual inspection between clean and adversarial images. The findings point to the differing extents to which each approach affects both the semantic embedding space and the perceptual quality of input images.

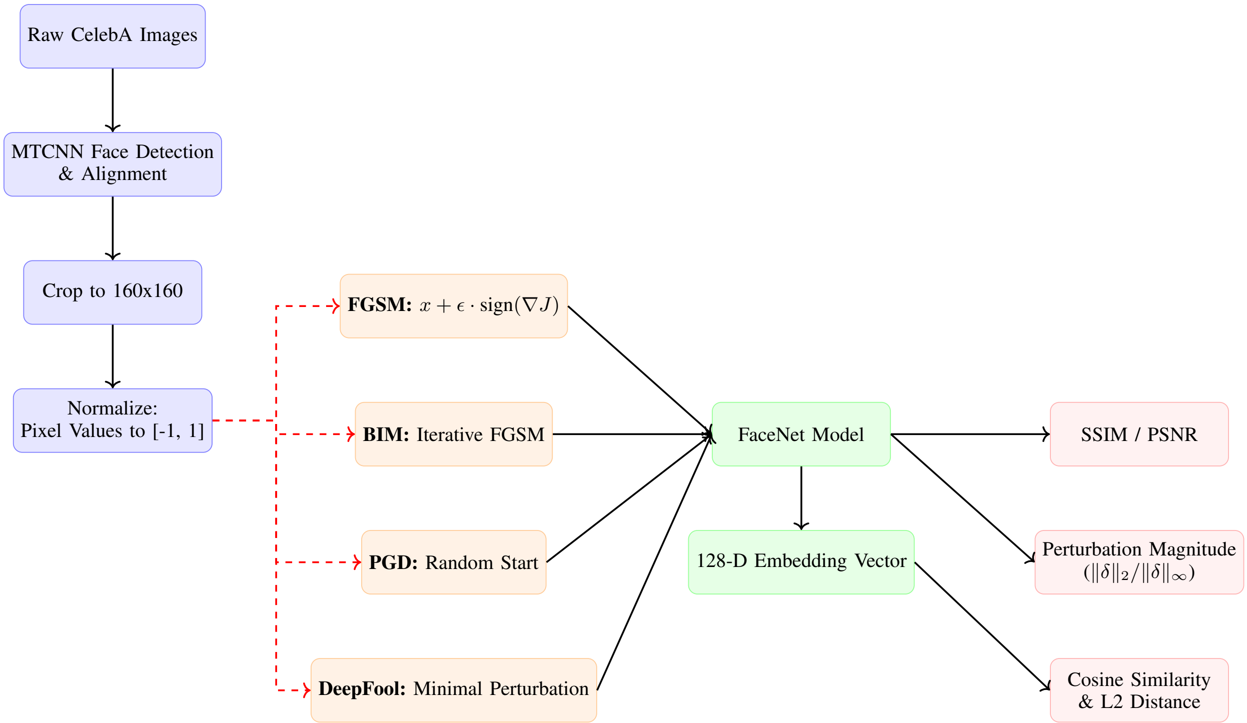

Raw CelebA Images
MTCNN Face Detection & Alignment
Crop to 160x160
Normalize: Pixel Values to [-1, 1]
FGSM: $x + \epsilon \cdot \text{sign}(\nabla J)$
BIM: Iterative FGSM
PGD: Random Start
DeepFool: Minimal Perturbation
FaceNet Model
128-D Embedding Vector
SSIM / PSNR
Perturbation Magnitude ($\|\delta\|_2/\|\delta\|_\infty$)
Cosine Similarity & L2 Distance

Fig. 1. A detailed flowchart of the adversarial attack pipeline on FaceNet using CelebA.

A. *Qualitative Results*

To visually evaluate the effect of adversarial attacks on face recognition, a side-by-side view of original and adversarial images for targeted identity is shown. The grid shows results for attacks like FGSM, BIM, PGD, and DeepFool. Each row corresponds to a different attack on a chosen identity, and images are captioned with their corresponding cosine similarity and L2 distance scores to demonstrate to what extent the model's perception has been changed through minimal visual differences. Surprisingly, although FGSM, BIM, and PGD introduce slight but significant perturbations that greatly impact the embedding output, the DeepFool attack, in most cases, does not introduce any visible or measurable perturbation. This is probably because the FaceNet model is considering the whole identity as one class in the embedding space (rather than discrete classes), and so untargeted attacks such as DeepFool are not effective. As DeepFool operates by pushing inputs over decision boundaries, and decision boundaries are less well defined in FaceNet's continuous embedding space, the attack has no effect.

B. *Quantitative Results*

To measure the semantic effect of adversarial attacks on the face recognition model quantitatively, we calculate the Cosine Similarity and L2 Distance between the original and adversarial embeddings produced by FaceNet for each attack strategy. Smaller cosine similarity and larger L2 distance values represent larger semantic deviation in the embedding space. Among the attacks, FGSM, BIM, and PGD exhibit a significant reduction in cosine similarity and a rise in L2 distance, validating their capacity to move embeddings away from the source class as shown in Table II. Surprisingly, DeepFool produces an average cosine similarity of 1.0 and L2 distance of 0.0, reflecting no measurable shift in embeddings. This is because the model is insensitive to the specific direction of the perturbation introduced by DeepFool or the subtlety of the perturbations themselves.

C. *Visual Quality Assessment*

To measure the visual quality degradation induced by adversarial attacks, we employ SSIM (Structural Similarity Index) and PSNR (Peak Signal-to-Noise Ratio), as well as perturbation magnitudes in terms of L norm (maximum pixel change) and mean pixel difference. These measures assist in determining the extent to which the adversarial image differs from the original, both perceptually and quantitatively. While FGSM, BIM, and PGD cause different amounts of perturbation observable by minimal falls in SSIM and PSNR, DeepFool on the other hand causes no deviation in all of these measures, suggesting imperceptible changes according to these assessment criteria.

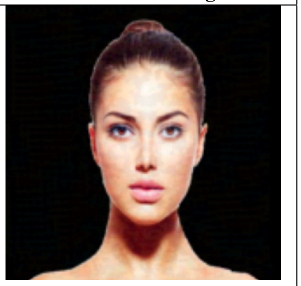
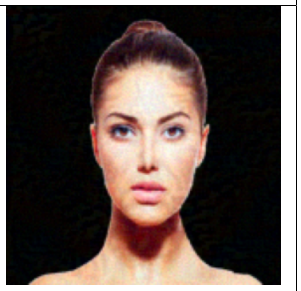
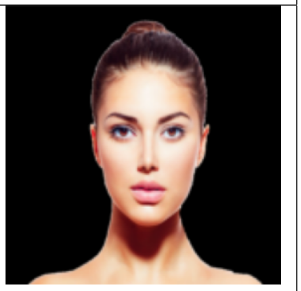
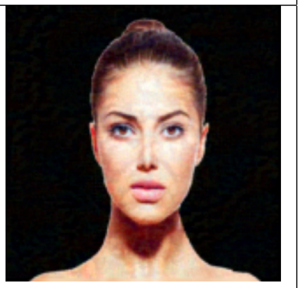
Adversarial Attacks	Original Image	Adversarial Image
FGSM		
PGD		
DeepFool		
BIM		

TABLE I
VISUAL COMPARISON OF ORIGINAL AND ADVERSARIAL IMAGES WITH SIMILARITY SCORES.

TABLE II
COMPARISON OF ATTACK METHODS BASED ON AVERAGE COSINE SIMILARITY AND L2 DISTANCE

Attack Method	Avg. Cosine Similarity	Avg. L2 Distance
FGSM	0.938	0.352
BIM	0.841	0.564
PGD	0.872	0.507
DeepFool	1.000	0.000

The results show dramatic disparities in the efficacy of the adversarial attacks. DeepFool, even though it was formulated to identify minimal perturbations, was unable to produce significant embedding changes, and the visual output was the same as the original—meaning that the attack did not effectively deceive the FaceNet model in this scenario. In contrast, attacks such as FGSM, BIM, and PGD made minute but powerful modifications that drastically varied the face embeddings,based them outside of recognition bounds. These outcomes demonstrate the weakness of deep face recognition systems: even unnoticeable modifications to input images lead to extreme drops in recognition rates, testifying to the importance

TABLE III
EVALUATION OF VISUAL AND PERTURBATION METRICS FOR ADVERSARIAL ATTACKS

Attack Method	**SSIM ↑**	**PSNR (dB) ↑**	**Perturbation Magnitude ↓**		
			Max Diff	**Mean Diff**	**L2 Diff**
FGSM	0.971	35.94	0.040	0.0255	8.844
BIM	0.951	31.48	0.100	0.041	14.781
PGD	0.942	31.21	0.100	0.0427	15.263
DeepFool	1.000	-	0.000	0.000	0.000

of powerful defenses against adversarial tampering.

VII. LIMITATIONS

Although the project successfully exploits the vulnerability of face recognition systems to adversarial attacks and assesses their effects on them with various perceptual and embedding-based measures, there are some limitations. The comparison is limited to a single face recognition model (FaceNet) and a single dataset (CelebA), which can restrict the generalizability of results to other models or real-world data with more varied conditions. Furthermore, the attacks are only conducted in white-box environments—where model parameters are accessible—while black-box attacks, the more realistic setting for deployed systems, is not considered. Lack of defense mechanisms or robustness test techniques also implies that the project does not offer nor validate solutions to counteract such adversarial weaknesses. Finally, the research approximates static lighting and pose conditions as a result of dataset limitations that might not entirely represent the wide variety of problems encountered in live face recognition systems.

VIII. CONCLUSION

This work emphasizes the vulnerability of contemporary face recognition systems, even with their high performance, to small adversarial perturbations. Through the use and testing of various attack techniques, the paper offers a holistic insight into how adversarial noise can undermine embedding consistency in the FaceNet system. Analysis through cosine similarity, L2 distance, SSIM, PSNR, and perturbation magnitude provides a dual perspective into semantic and perceptual degradation.

The findings emphasize the need for robust defense mechanisms to counter adversarial threats in real-world applications. Techniques such as adversarial training, input preprocessing, and defensive distillation could enhance model resilience against such attacks. This study provides a foundation for further research into improving the security of face recognition models, ensuring their reliability in critical applications such as surveillance, identity verification, and access control systems.

REFERENCES

[1] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[5] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[6] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607, 2020.

[7] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International conference on cyber and IT service management*, pages 1–6. IEEE, 2016.

[8] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[9] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[11] Huoyuan Dong, Jialiang Dong, Shuai Yuan, and Zhitao Guan. Adversarial attack and defense on natural language processing in deep learning: A survey and perspective. In *International conference on machine learning for cyber security*, pages 409–424. Springer, 2022.

[12] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[13] Alan C Brooks, Xiaonan Zhao, and Thrasyvoulos N Pappas. Structural similarity quality metrics in a coding context: exploring the space of realistic distortions. *IEEE Transactions on image processing*, 17(8):1261–1273, 2008.

[14] Xiaowei Huang, Daniel Kroening, Marta Kwiatkowska, Wenjie Ruan, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. Safety and trustworthiness of deep neural networks: A survey. *arXiv preprint arXiv:1812.08342*, page 151, 2018.

[15] Guillermo Ortiz-Jiménez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Optimism in the face of adversity: Understanding and improving deep learning through adversarial robustness. *Proceedings of the IEEE*, 109(5):635–659, 2021.

[16] Rongrong Jin, Hao Li, Jing Pan, Wenxi Ma, and Jingyu Lin. Face recognition based on mtcnn and facenet. In *AAAI Conference on Artificial Intelligence*, 2021.