Detection of Face Recognition Adversarial Attacks

Aakash Loyar
EP21BTECH11001

Rithika Kallu
AI22BTECH11010

Muskan Jaiswal
CS21BTECH11037

Shakshi Shukla
MA23MCST11020

Prasham Walvekar
CS21BTECH11047

November 29, 2024

1. Abstract

Adversarial attacks in image recognition are common problems in today's world. However, developments on solutions to safeguard against these attacks have also emerged in the last few years. The domain of Face Recognition is fairly unexplored in terms of adversarial attacks, and it is an important standalone domain in the modern era. Face recognition systems are widely used in security and authentication applications, but their vulnerability to adversarial attacks presents a significant security risk which cannot be detected by naked eyes, but fools systems based on neural networks. This paper presents a dual approach to detect and mitigate adversarial attacks on FR systems along with creating adversarial images using methods such as FGSM. For detection, we propose the use of a Support Vector Machine Classifier, trained to identify adversarial examples based on feature discrepancies. For mitigation, we introduce a selective dropout technique combined with Adaptive Noise Reduction, aimed at the impact of adversarial perturbations on the model's decision-making process. Our experimental result shows an accuracy of about 98% in detecting these attacks, while the selective dropout and the Adaptive Noise Reduction (as image processor) combination perform slightly better than the technique used by the authors in their work "Unravelling Robustness of Deep Learning Based Face Recognition against Adversarial Attacks" in rectification/mitigation. This dual approach improves the robustness of FR systems, offering a more effective defense against adversarial threats.

2. Problem Statement

Face recognition system have become an essential part of modern tech, used in security systems, mobile devices and surveillance cameras. From unlocking smartphones to enhancing security in sensitive areas, these systems have proven to be highly accurate and efficient. However, as these technologies grow in importance, they also face increasing challenges. One of the most concerning threats is adversarial attacks, which exploit the weaknesses in AI models by introducing small, often imperceptible changes known as perturbation to images. These changes can cause face recognition systems to make incorrect predictions, such as misidentifying individuals or failing to recognize them altogether. These changes are often so subtle that naked human eyes cannot notice them, but they can severely impact the performance of face recognition models.

Adversarial attacks represent a significant risk to the security and reliability of AI systems. They can undermine trust in applications that rely on face recognition, such as biometric authentication, surveillance, and identity verification. The consequences of a successful attack could range from unauthorized access to confidential information to misuse of personal identities. These attacks pose a serious threat to face recognition systems, making them less reliable in critical applications. Thus, defense against these attacks is a necessity for the success of Deep Learning in practice.

2.1. Objective

To create a detection and restoration pipeline that ensures adversarially attacked images are cleaned and classified correctly by the VGG16 model, addressing both the detection and defense aspects of adversarial attacks.

2.2. Proposed Solution

Detection of Adversarial Images: Implemented a Support Vector Machine (SVM) classifier to detect and differentiate adversarial images from clean ones.

Image Restoration for Robust Classification: Developed a combined approach for improving the quality of adversarial images: Adaptive Noise Reduction: To reduce the perturbations introduced by adversarial attacks. Selective Dropout: To further enhance image clarity and assist in restoring the image features

needed for correct classification.

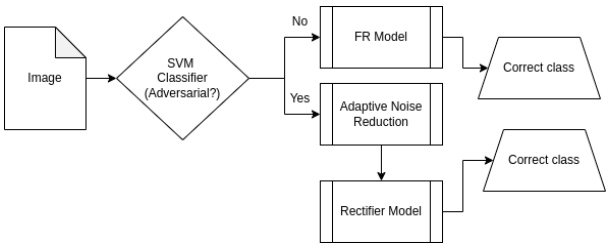

Figure 1. Overall framework

2.3. Assumptions

- We mainly focus on White-Box attacks.
- Our detection and mitigation framework requires full knowledge/access of the targeted face recognition neural network, including the model's architecture, parameters, and gradients.
- We omit adversarial retraining of the model considering it to be expensive, and also to preserve the already-existing model. However our rectifier model is assumed to be a copy of the existing model, whose parameters will be modified (but no training is done)
- We focus on attacks that just aim to misclassify the image from the original class, and not towards a specific target class

3. Literature Review

3.1. Literature Review - Adversarial Attacks Against Face Recognition: A Comprehensive Study

This literature was helpful to us to gain a good understanding of the domain, about adversarial attacks, how they are performed and current solutions. Adversarial attacks are deliberate perturbations applied to input data, such as images, to deceive machine learning models into making incorrect predictions. These attacks are particularly effective in deep learning models, where even small, imperceptible changes can cause significant misclassification.

There are two primary types of adversarial attacks:

- **White-box attacks**: The attacker has full access to the model's architecture, parameters, and gradients, allowing precise manipulation to generate adversarial examples.
- **Black-box attacks**: The attacker has no direct access to the model and instead relies on querying the model to generate adversarial examples by observing its outputs.

The goal of adversarial attacks is often to cause the model to either:

- **Misclassify an input** (non-targeted attack), where the model outputs an incorrect class different from the true label.
- **Classify an input into a specific incorrect class** (targeted attack), where the attack aims to mislead the model into predicting a particular wrong label.

3.1.1 Adversarial Attack Strategies for General Image Classifiers: Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a widely-used technique for generating adversarial examples. Mathematically the perturbation is computed as:

$$n = \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathcal{J}(\theta, \mathbf{x}, \mathbf{y})\right)$$

where ϵ is the perturbation magnitude, $\mathcal{J}$ is the cost function, and $\text{sign}(\cdot)$ is the sign function. The adversarial example $\mathbf{x}'$ is then generated by:

$$\mathbf{x}' = \mathbf{x} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathcal{J}(\theta, \mathbf{x}, \mathbf{y})\right)$$

FGSM performs a one-step update in the direction of the gradient sign at each pixel. We mainly use FGSM for our project because it is fast and effective in creating adversarial images, as compared to the other traditional methods, such as CW-attack, Deepfool, etc.

3.1.2 Adversarial Attack Strategies for General Image Classifiers: DeepFool

The **DeepFool attack**, works iteratively to find a minimal norm perturbation that pushes an image across decision boundaries. It begins by approximating the classifier's decision boundary as affine and gradually adds small perturbations to move the image closer to this boundary. This process is repeated iteratively until the image is misclassified. **DeepFool** typically generates smaller perturbations compared to other attacks like FGSM and JSMA while achieving comparable fooling success.

3.2. Literature Review: Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks

3.2.1 Methodology

This was the main paper that we implemented in our detection framework, with a slight modification. Adversarial distortions are detected by analyzing layer-wise deviations in intermediate network representations. The detection pipeline computes the *mean representation* of layer outputs across the training data, mathematically expressed as:

$$\mu_i = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \phi_i(I_j) \quad (1)$$

where $\phi_i(I_j)$ represents the activations at the i-th layer for the j-th image, and N_{train} is the total number of training images.

During detection, the *Canberra distance* quantifies deviations between activations for an input image and these mean representations:

$$\Psi_i(I, \mu) = \sum_z \frac{|\phi_i(I)_z - \mu_{i,z}|}{|\phi_i(I)_z| + |\mu_{i,z}|} \quad (2)$$

This yields a feature vector of distances across layers, used to train a Support Vector Machine (SVM) classifier for distinguishing adversarial from normal inputs.

3.2.2 Mitigation Strategies

Mitigation involves preprocessing techniques like *median filtering* to denoise input images and *selective dropout*, which disables the most distortion-sensitive filters. Sensitivity of filters is scored as:

$$\epsilon_{ij} = \sum_{k=1}^{N_{\text{dis}}} ||\phi_{ij}(I_k) - \phi_{ij}(I'_k)|| \quad (3)$$

where N_{dis} is the number of distorted images, and $\phi_{ij}(I_k)$ denotes the response of the j-th filter in the i-th layer for the k-th image.

3.2.3 Results

This approach demonstrates high detection accuracy using Canberra-based SVM classifiers, particularly for large-scale datasets like CMU Multi-PIE. Mitigation strategies restore recognition performance by rejecting or denoising adversarial inputs and modifying network pipelines selectively. These results affirm the robustness of combined detection and mitigation strategies in safeguarding face recognition systems from adversarial threats.

3.3. Literature Review: Detecting Adversarial Image Examples in DNNs with Adaptive Noise Reduction

3.3.1 Introduction

This was an important image-denoising technique that we chose over traditional methods because of its adaptive nature. Adversarial examples exploit small perturbations to mislead deep neural networks (DNNs). This paper proposes an adaptive noise reduction technique to detect such examples, using quantization and smoothing guided by image entropy. The method neutralizes perturbations without requiring prior knowledge of the attack type.

3.3.2 Methodology

1. **Entropy-Based Categorization**: Images are categorized into low, medium, and high entropy based on complexity. Noise reduction intensity is adjusted accordingly.
2. **Quantization**: Maps pixel values to discrete levels to reduce perturbations. The number of levels depends on the image's entropy.
3. **Smoothing**: High-entropy images are further processed using Gaussian or median filters to reduce noise.
4. **Classification Comparison**: Predictions of original and denoised images are compared to detect adversarial examples.

3.3.3 Results

- Achieved an average **F1 score of 98.80%** for CW L2 attacks.
- Recovered correct classifications for **93.94%** of adversarial examples.
- Introduced minimal computational overhead (<1 second per image).

3.4. Literature Review: Universal Adversarial Perturbations and Perturbation Rectifying Networks

This paper tackles **Universal Adversarial Perturbations (UAPs)**, image-agnostic modifications that mislead classifiers across diverse inputs, posing significant risks in critical applications. The authors introduce the **Perturbation Rectifying Network (PRN)**, a modular preprocessing layer that removes adversarial noise without altering classifier architectures.

The PRN features:

1. A **Perturbation Detector** leveraging frequency analysis to identify adversarial inputs.
2. A **Perturbation Rectifier** that cleans inputs to restore correct classifications.

Using a ResNet-based architecture, the PRN achieves a 97.5% success rate against unseen UAPs on models like GoogLeNet and VGG-F, offering a practical and robust defense.
This paper was included because it motivated us a bit to think on similar terms, although majority of our work is based off the two papers mentioned before.

4. Methodology

Our methodology consists of 4 main parts/stages (shown in Figure 2):

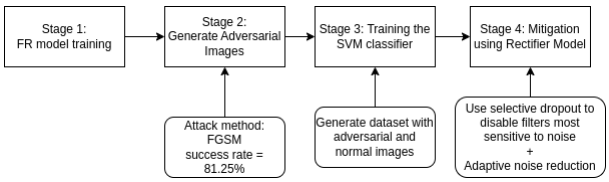

Figure 2. The development stages of the proposed framework

4.1. The Face Detection Model

To build a model for the face detection task, VGG16, a widely-used convolutional neural network architecture was selected as the base model. The choice of VGG16 was motivated by its proven capability to extract high-quality features from images, owing to its deep architecture of 16 layers with small convolutional filters. We chose VGG16 over other image classifiers because of its simple architecture, and because comparatively it requires lesser data to fine-tune compared to other classifiers like ResNet, InceptionV3, etc.

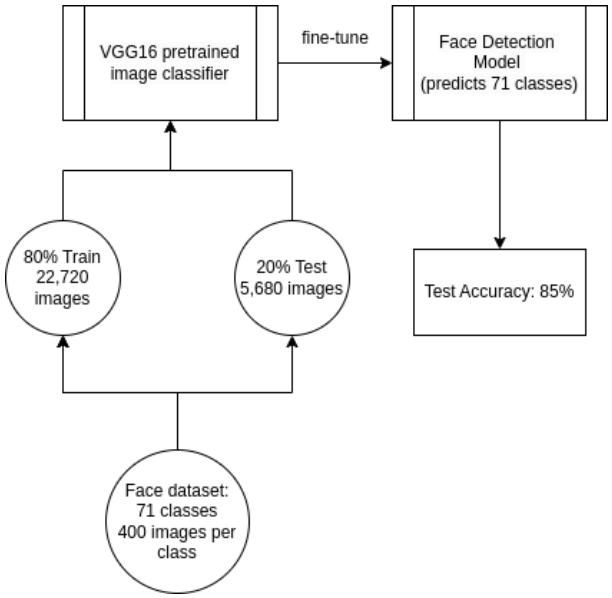

Figure 3. The FR model

4.1.1 VGG16 Model Architecture

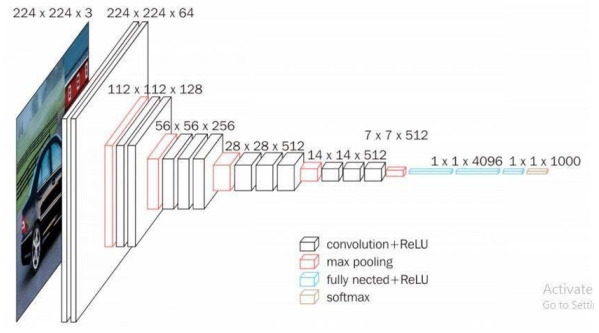

Figure 4. VGG16 model architecture

VGG16 model was fine-tuned to perform face recognition on a facial image dataset comprising 71 classes, each with 400 images. The primary objective was to adapt the pre-trained model to classify these facial images.

4.1.2 Data overview

We used VGGFace2 dataset (link given in References). The dataset was split into training and validation subsets, with 80% of the data used for training and 20% reserved for validation. For the training set, data augmentation techniques such as random horizontal flipping, rotation, affine transformations, and perspective distortion were employed. These techniques were coupled with adjustments to brightness, contrast, saturation, and hue, as well as the application of Gaussian blur and random erasing. This preprocessing helped introduce variability and improve the model's robustness. For both training and testing data, images were resized to 224×224 pixels, converted to PyTorch tensors, and normalized to match the ImageNet standards.

4.1.3 Training Phase

To adapt the model to the face recognition task, the fully connected output layer was replaced with a custom classifier consisting of a dropout layer for regularization and a fully connected layer with 71 output units, matching the number of classes. While the earlier layers of the model were frozen to retain the pretrained features, the last eight layers of the feature extractor were unfrozen and fine-tuned to allow adaptation to the new task. The training process employed the Adam optimizer with a weight decay of 1×10^{-4} to prevent overfitting. A learning rate of 1×10^{-5} was assigned to the fine-tuned feature extractor layers, while a higher rate of 1×10^{-4} was applied to the classifier layers. Cross-entropy loss was used as the objective function, and a ReduceLROnPlateau scheduler dynamically adjusted the learning rate based on validation loss, reducing it by a factor of 10 if no improvement was observed for three consecutive epochs. Training was conducted over 40 epochs with a batch size of 64. Fine-tuning of the model required careful selection of layers to freeze or unfreeze, as well as the use of separate learning rates for different parameter groups to ensure efficient training. Despite these challenges, the model successfully adapted to the task, achieving strong performance.

4.2. Adversarial attack/adversarial dataset generation

We made use of FGSM to generate the adversarial images, taking epsilon = 0.01
The perturbations in the adversarial images generated were imperceptible to the human eye. The attack success rate was around 81.25%. In the end, we had a dataset of around 4-5 thousand adversarial images for our model. We took help of the torchattacks library in pytorch to perform the attack.

4.3. Detection

Our Detection Framework mainly consists of a SVM classifier at its core. The SVM classifier is trained on a dataset.
For creating the training dataset, the features were the canberra distances between the adversarial image and its corresponding unperturbed image, at each layer. The dataset consisted of roughly 38,000 data points.
The test dataset consisted of roughly 6400 images of both types- adversarial and normal, on which the SVM classifier accuracy is calculated.

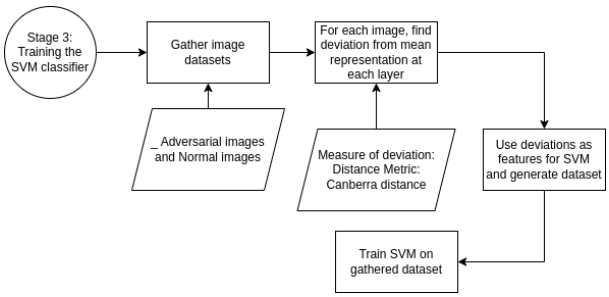

Figure 5. The detection framework

4.4. Rectification/ Mitigation

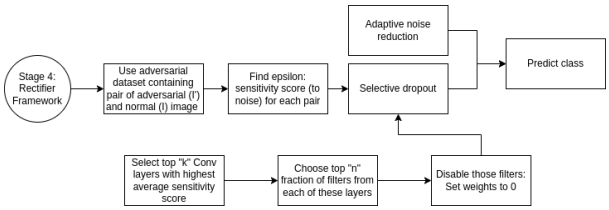

Figure 6. The rectification/mitigation framework

The rectification/mitigation framework has been depicted in Figure 6. The goal of this framework is to try to find the correct class of the adversarial image, after it has been classified as an adversarial image.
Our mitigation framework does the following:
First, it finds the most sensitive filters in all the Conv layers of the model. This is done by calculating the epsilon (sensitivity) score mentioned in the literature review. A pair of normal, adversarial image is passed and the difference between the activation of the normal vs. adversarial image as a result of the j'th filter in the i'th layer is calculated. Then, the top "k" layers which have the largest average epsilon values are chosen and the top "n" fraction of filters in each of these layers are disabled, i.e., their weights are set to 0 for the rectifier model. This method is called "selective dropout". The rectifier model is a copy of the original model, but with selective dropout applied.
Second, after the SVM classifies the image as adversarial before passing through the rectifier model, the image is denoised/clean, to make the perturbations less impactful. The original paper uses median filtering to denoise the adversarial image, but our method proposes the use of Adaptive Noise Reduction because it is a more robust method, which denoises to extents depending on the properties of the image (such as entropy of the image).
So our novelty is the combination of 2 methods described in the 2 papers into one method! i.e., mitigation using selective dropout + adaptive noise reduction.

5. Results

5.1. Performance of the Face Detection Model

Throughout training, the model demonstrated improvements in both training and validation accuracy. The training accuracy reached its peak at 99.9% achieved at epoch 29, while the highest validation accuracy was 85.56% achieved at epoch 20. The final test accuracy of the model was 85.53%.

The visualization of the training and validation loss and training and validation accuracy during the training phase are as follows,

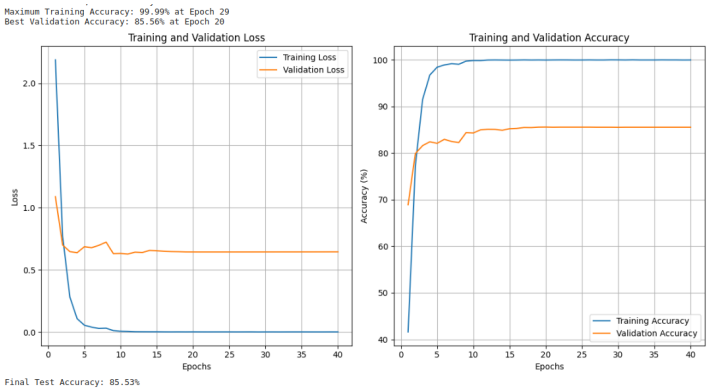

Figure 7. Visualization of the training vs validation loss and training vs validation accuracy during the training phase.

5.2. Adversarial Image generation

The adversarial attack (FGSM) had a success rate of around 80% on a dataset of about 5680 face images. The adversarial image and the normal image have imperceptible differences (to the human eye).

Figure 8. Adversarial Image example 1

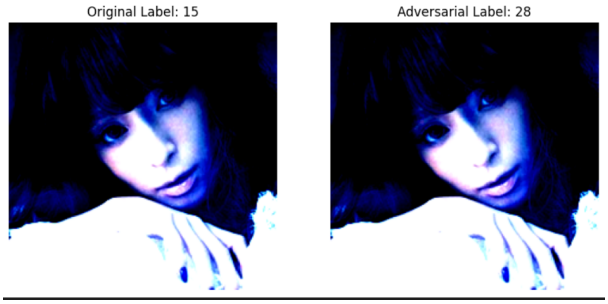

Figure 9. Adversarial Image example 2

5.3. Detection

The SVM classifier reported a high accuracy on the test dataset, generated from a mixture of the original adversarial image dataset that we had created, along with normal images. The test accuracy was reported as **98.75**% (Refer Figure 9)
Adaptive noise filtering (as a detection mechanism) failed for this use-case, with an accuracy of near 0.

```
Training the SVM classifier...
Training complete!
Accuracy: 98.75%

Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      5454
           1       0.97      0.94      0.96       946

    accuracy                           0.99      6400
   macro avg       0.98      0.97      0.97      6400
weighted avg       0.99      0.99      0.99      6400
```

Figure 10. Result of the SVM classifier on test data

5.4. Rectification/Mitigation

The accuracy reported (i.e., the fraction of adversarial images correctly classified by the rectifier) with median filtering was around 1.51% The accuracy reported (i.e., the fraction of adversarial images correctly classified by the rectifier) with adaptive noise filtering was around 2.33% Although these accuracies are really small, it still shows that our proposed method is slightly better than the one mentioned in either paper

6. Conclusion

Our proposed framework for detection of adversarial images in FR (using SVM classifier) has a very good accuracy, and hence it is a successful method. It performs better than just adaptive noise reduction as a detection strategy.
Our proposed framework for rectificaion of adversarial images in FR (using adaptive noise filtering + selective dropout) also performs slightly better than the method proposed in the paper, although the accuracy is less. Future work includes improvement on the accuracy of the rectification model, exploring different classifiers, distance metrics, denoising strategies and generalization for any model/data.

References

[1] *Gaurav Goswami, Nalini Ratha, Akshay Agarwal, Richa Singh, Mayank Vatsa "Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks"*. Available here

[2] *The VGGFace2 dataset we used to train our model* available here

[3] *Bin Liang; Hongcheng Li; Miaoqiang Su; Xirong Li; Wenchang Shi; Xiaofeng Wang "Detecting Adversarial Image Examples in Deep Neural Networks with Adaptive Noise Reduction"*. Available here

[4] *Adversarial Attacks Against Face Recognition: A Comprehensive Study*. Available here.

[5] *Torchattack documentation* Available here

[6] *Defense against Universal Adversarial Perturbations*. Available here.

[7] *N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in Proc. IEEE Eur. Symp. Secur. Privacy (EuroSP), Mar. 2016, pp. 372–387*