

Investigating Robustness of Face Recognition System against Adversarial Attacks

by

SARVAIYA MAULIK KARSHANBHAI
202111025

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

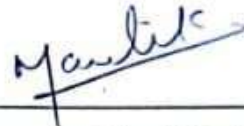


May, 2023

Declaration

I hereby declare that

- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.



Sarvaiya Maulik Karshanbhai

Certificate

This is to certify that the thesis work entitled Investigating Robustness of Face Recognition System against Adversarial Attacks has been carried out by Sarvaiya Maulik Karshanbhai for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



Prof. Shruti Bhilare
Thesis Supervisor

Acknowledgments

I extend my deepest gratitude and appreciation to the individuals and institutions who have supported me throughout the development of the adversarial bandage for my thesis. Their invaluable contributions and unwavering support have made this project possible. First and foremost, I would like to express my heartfelt thanks to my supervisor, Prof. Shruti Bhilare, for her guidance, expertise, and constant encouragement. Her insightful feedback and mentorship have been instrumental in shaping the direction of this research. I am truly grateful for her patience and dedication. I am also indebted to my colleagues who have provided their expertise, assistance, and fruitful discussions. Their collaboration and commitment have greatly enriched this project. Their input and suggestions have been invaluable in refining the design and functionality of this thesis.

Lastly, I would like to acknowledge the love, support, and understanding of my family and friends throughout this research endeavor. Their unwavering encouragement and belief in me have been a constant source of motivation. I am truly grateful to all those mentioned above and countless others who have played a role, no matter how small. Your support, guidance, and contributions have been instrumental, and I am humbled by your presence in my academic journey. Thank you all for your invaluable support and encouragement.

Contents

Abstract	v
List of Tables	vi
List of Figures	vii
List of Acronyms	ix
1 Introduction	1
1.1 What is an adversarial attack?	2
1.2 Types of adversarial attacks	3
1.2.1 Adversarial knowledge	3
1.2.2 Adversarial aim	3
1.2.3 Realizability	5
1.2.4 Scope of perturbation	5
1.3 Attack deployment strategies	5
1.3.1 Poisoning attack	6
1.3.2 Evasion attack	6
1.3.3 Extraction attack	6
1.4 Motivation	7
1.5 Objective	7
1.6 Contributions	8
1.7 Thesis outline	8
2 Literature Review	9
3 Methodology	19
3.1 Threat model	19
3.1.1 Attacker’s goal	19
3.1.2 Attacker’s knowledge	19
3.1.3 Attacker’s capabilities	20

3.1.4	Attacker’s strategy	20
3.2	Facenet	20
3.3	Proposed approach	22
3.3.1	Algorithm	23
3.3.2	Loss function for targeted	24
3.3.3	Loss function for untargeted	26
4	Experiments	28
4.1	Dataset description	28
4.2	Evaluation metrics	29
4.3	Experimental setup	30
4.4	Location and size of bandage	30
4.5	Results	31
4.5.1	Targeted attack	32
4.5.2	Untargeted attack	33
5	Conclusion	35
5.1	Conclusion	35
5.2	Future work	35
	References	36

Abstract

Facial Recognition (FR) systems based on deep neural networks (DNNs) are widely used in critical applications such as surveillance and access control necessitating their reliable working. Recent research has highlighted the vulnerability of DNNs to adversarial attacks, which involve adding imperceptible perturbations to the original image. The presence of these adversarial attacks raises serious concerns about the security and robustness of deep neural networks. As a result, researchers are actively exploring and developing strategies to strengthen the DNNs against such threats. Additionally, the object used should look natural and not draw undue attention. Attacks are carried out in white-box targeted as well as untargeted settings on Labeled Faces in the Wild (LFW) dataset. Attack success rate of 97.76% and 91.78% are achieved in untargeted and targeted settings, respectively demonstrating the high vulnerability of the FR systems to such attacks. The attacks will be evaluated in the digital domain to optimize the adversarial pattern, its size and location on the face.

List of Tables

- 2.1 Type of attacks and categorization of adversarial attacks on ML models 18
- 4.1 Attack success rate in impersonation and dodging attacks on FaceNet 31

List of Figures

1.1	Overview of the adversarial attack	2
1.2	Targeted attack [5]	4
1.3	Untargeted attack [5]	4
1.4	Types of adversarial attacks	5
2.1	Adversarial patch off-and-around a car [8]	10
2.2	Optimisation process for adversarial attack on an aerial imagery object detector [8]	11
2.3	Laser beam attack [9]	12
2.4	Adversarial patch [3]	12
2.5	The pipeline of the adversary objective function. [11]	13
2.6	Overall pipeline of Universal physical camouflage (UPC) [12]	14
2.7	Effects of triggers with different locations [21]	15
	(a) Trigger placement	15
	(b) Attack success rate	15
2.8	Impersonation using eyeglass frame [17]	16
	(a) Original image	16
	(b) Perturbed image	16
	(c) Misclassified class	16
2.9	The physical-world attacks on FR systems using adversarial stick- ers crafted by FaceAdv [18]	17
2.10	Samples of dodging and impersonating attacks [18]	17
3.1	Triplet loss [13]	21
3.2	Architecture of Inception Resnet v1	22
4.1	LFW dataset	29
4.2	Targeted adversarial bandage	29
4.3	Original bandage	31
4.4	Bandage mask	31
4.5	Epochs vs. Attack Success Rate	32

4.6	Targeted attack on the class of LFW dataset	33
4.7	Targeted attack on the class of LFW dataset	33
4.8	Untargeted adversarial bandage	34
4.9	Untargeted attack	34
4.10	Untargeted attack	34

List of Acronyms

FR Facial Recognition

DNNs deep neural networks

LFW Labeled Faced in Wild

SVM Support Vector Machine

UPC Universal physical camouflage

TV Total Variation

FGSM Fast Gradient Sign Method

ASR Attack Success Rate

NPS Non-printability score

CHAPTER 1

Introduction

The ability of DNNs to achieve high classification accuracy is the primary motivation in image classification applications. DNNs have proven to be exceptionally effective at reliably classifying images, making them popular in various applications. Although DNNs have obtained respectable prediction accuracy, they are still susceptible to flaws that attackers can use to craft adversarial examples. An image that has been altered by adding deliberate noise to the original input image is referred to as an *adversarial example*. The perturbed input has slight aesthetic differences from the original image but is purposefully altered to produce misclassification by a DNN. An adversarial attack is the process of producing such adversarial examples. Various conditions or traits can be used to categorize adversarial attacks. These attacks take advantage of DNNs' sensitivity to minute changes in input data to exploit their weaknesses and limitations. In the context of enhancing the security and robustness of DNN-based image classification systems, understanding adversarial attacks and their various types is crucial. However, the scope of adversarial attacks is not limited to image classification alone. *To generate an adversarial bandage that misleads the face recognition model under targeted and untargeted attacks in white-box settings*, researchers must delve into the intricacies of face recognition technology. This involves understanding the unique challenges posed by face recognition systems and the potential vulnerabilities that can be exploited. Building on this knowledge, researchers can develop innovative defense strategies to protect against adversarial manipulation, ultimately contributing to the broader field of computer vision and machine learning. To improve the security and robustness of DNN-based image classification and face recognition systems, researchers can create techniques to understand the effects of various adversarial attacks and construct more resilient models. The discussion will go into greater detail about adversarial attacks in both image classification and face recognition and provide an overview of various types of attacks in the next segments.

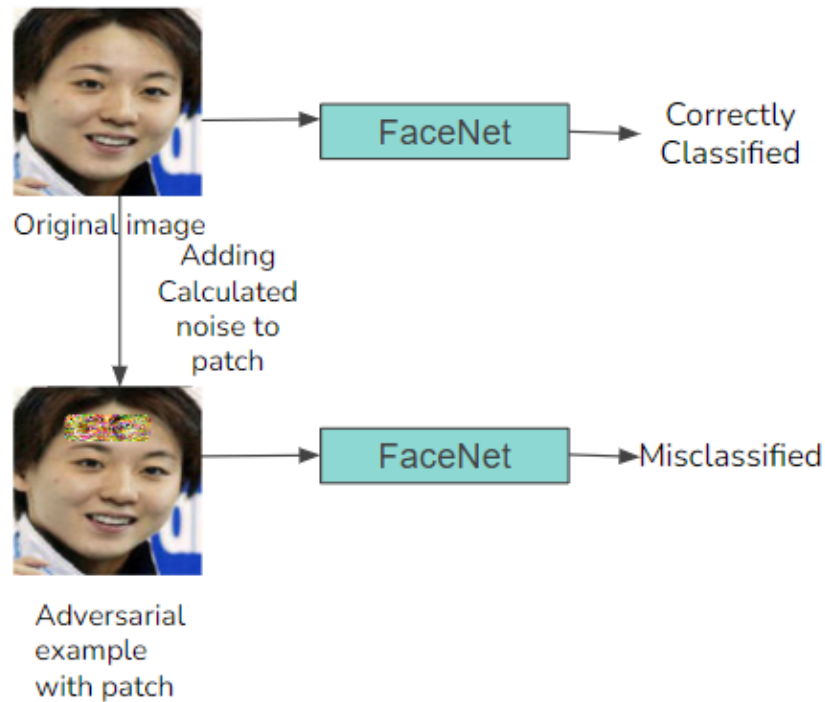


Figure 1.1: Overview of the adversarial attack

1.1 What is an adversarial attack?

An *adversarial attack* occurs when data inputs are purposefully altered or perturbed to trick or mislead a machine learning model. Adversarial attacks aim to make the model generate inaccurate or unexpected outputs by taking advantage of weaknesses in its decision-making process. These attacks typically make minute adjustments to the input data but significantly impact the model's predictions, as shown in Figure 1.1. The adjustments can involve changing specific photo pixels, introducing undetectable noise to audio signals, or altering particular elements in textual data. They are meticulously designed to take advantage of the model's flaws. The attack goal can change depending on the attacker's objectives. It might be done to misclassify a picture, fool a spam filter, tamper with sentiment analysis, or even fool autonomous vehicles by changing traffic lights or road signs. The security and resilience of machine learning systems have come under scrutiny due to adversarial assaults, particularly in areas as important as healthcare, banking, and autonomous systems. Researchers are working hard to understand the mechanisms behind these attacks better and create strategies to counter them.

1.2 Types of adversarial attacks

Adversarial attacks can be broadly classified according to adversarial aim, adversarial knowledge, realizability, and scope of perturbation, as discussed below and shown in Figure 1.4. [4]

1.2.1 Adversarial knowledge

- **White-box attack:** In a white-box attack, an attacker has full access to or knowledge of the internal workings of a model. This details the model's parameters, training data, and architecture. Using this information, the adversary can create adversarial inputs, or adversarial examples, in a white-box attack that can trick the machine learning model [14].
- **Black-Box attack:** In a black-box attack, an attacker has limited knowledge of how a machine learning model functions internally. The attacker doesn't have access to the machine learning model's parameters, architecture, or training data, which makes it a black-box in this situation. Typically, the attacker can access the input-output pairs, which can be leveraged to produce adversarial examples. The attacker uses methods like gradient estimation, query-based attacks, or transferability to carry out a black-box attack [14].

1.2.2 Adversarial aim

Given an image x of class j , we want to misclassify it to target class t using discriminant function $g(x)$.

- **Targeted attack:** *Targeted attacks* in adversarial machine learning is intentional and malicious attempts to trick or influence machine learning models using adversarial examples. Adversarial examples are deliberately produced inputs that are slightly modified from regular data samples to deceive the machine learning model into producing incorrect predictions. In targeted attacks, the attacker aims to make the model produce the desired inaccurate output by manipulating it to do so. Following Equation (1.1) [5] denotes targeted attack. Here, Ω represents set.

$$\Omega = \{x | \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0\} \quad (1.1)$$

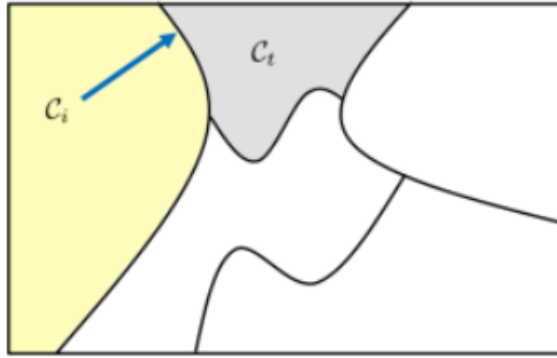


Figure 1.2: Targeted attack [5]

- **Untargeted attack:** *Untargeted attacks* aim to trick a machine learning model without explicitly aiming at a particular output or class label. The attacker seeks to produce adversarial examples, which are samples of input deliberately created to lead the model to make false predictions. No matter which specific incorrect prediction is made in an untargeted attack, the attacker's goal is to get the model to make a mistake. In order to trick the model, the attacker usually manipulates the input data in a human-imperceptible manner. The perturbations are deliberately designed to exploit weaknesses or blind spots in the model's decision-making process. Equation (1.2) [5] for untargeted attack can be defined as:

$$\Omega = \{x | g_i(x) - \max_{j \neq i} \{g_j(x)\} \leq 0\} \quad (1.2)$$

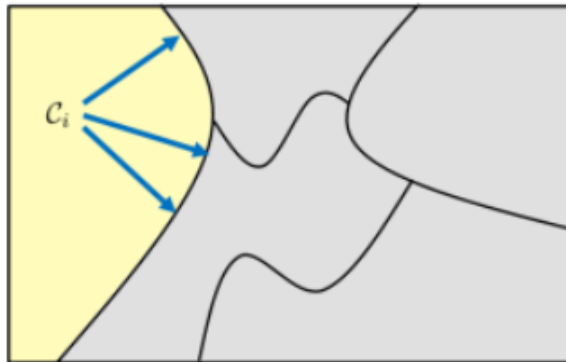


Figure 1.3: Untargeted attack [5]

1.2.3 Realizability

- **Digital attack:** They manipulate the original images present in the dataset and generally performed in controllable lab environments which focus on improving the performance of attack algorithms.
- **Physical attack:** They mostly focus on physical world deployed DNN models. Such attacks are comparatively more challenging to carry out due to complex physical environments like brightness, occlusion, viewpoints, etc. [18]

1.2.4 Scope of perturbation

- **Individual attack:** The optimization problem is solved for each sample, and the perturbation for each sample is different.
- **Universal attack:** The optimization problem is solved for the whole dataset, and the same perturbations can be used for different input samples to misclassify them to the wrong class. [25]

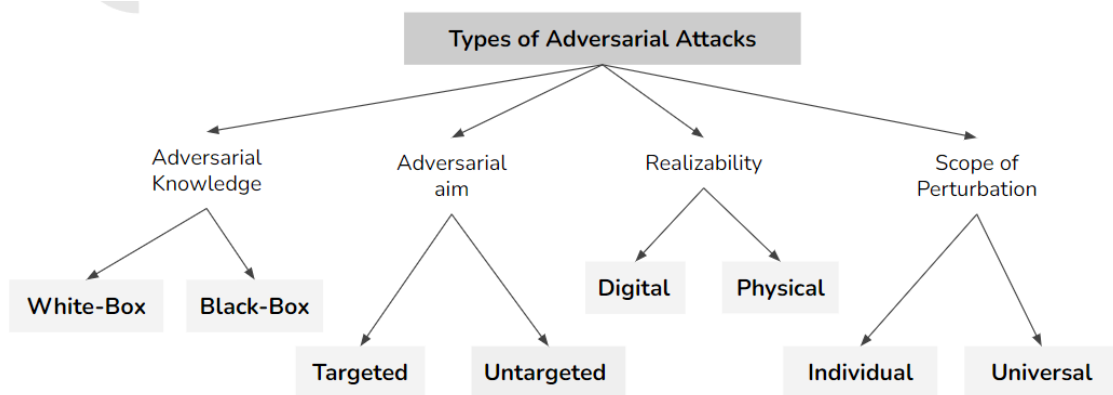


Figure 1.4: Types of adversarial attacks

1.3 Attack deployment strategies

Adversarial attacks can also be classified into the three following categories: Poisoning attacks, Extraction attacks, and Evasion attacks. Each attack type targets different stages of the machine learning pipeline, from the training phase to the testing or deployment phase.

1.3.1 Poisoning attack

The attacker can strategically alter the training data by inserting malicious samples specifically designed to fool the model. These samples might have subtle modifications or perturbations that are meant to fool the learning algorithm. By adding these contaminated samples into the training procedure, the attacker aims to bias the model towards making incorrect predictions or exhibiting undesirable behavior when deployed in practical situations. The fact that Machine Learning systems frequently have the ability to incorporate data obtained during operation for re-training raises serious concerns about data poisoning attacks. This implies that the attacker inserted malicious samples may affect the re-training procedure, continuing the negative effect even after the first deployment. This emphasizes the significance of strong data poisoning defenses, as the presence of poisoned samples during the training phase of model can have a great impact on the performance of the model.

1.3.2 Evasion attack

In evasion attacks, the attacker tampers with the data during the deployment phase to deceive trained classifiers. By manipulating the classifier's decision-making process, these attacks aim to force it to make erroneous or unwanted predictions [2]. Evasion attacks are common in many situations, but they are especially common in intrusion and malware detection. Attackers use evasion tactics to change spam emails or malicious code so that they can avoid being detected by deployed classifiers. Attackers try to evade detection by security systems that rely on classification algorithms by altering the content or structure of malware. Evasion attacks concentrate on perturbing samples during inference time instead of poisoning attacks targeting the training data.

1.3.3 Extraction attack

An attacker engages in model extraction when they want to learn more about how a machine learning model functions internally but are not readily accessible [1]. In order to learn more about the model's architecture, parameters, or decision boundaries, the attacker probes the black box system with carefully prepared queries or inputs. Model extraction can have a variety of motivations. Attackers may occasionally be motivated to recreate the model, which has many implications. For instance, the stolen model may represent confidential information.

1.4 Motivation

The motivation behind generating adversarial bandages is rooted in the necessity of understanding and addressing the vulnerabilities of deep neural networks (DNNs) to enhance defense mechanisms. Creating potent adversarial attacks aids in uncovering DNN weaknesses and fosters the development of robust defenses. Adversarial bandages serve two key purposes: impersonation, enabling attackers to mimic high-authority target persons, and evasion, allowing identity concealment. With the increasing prevalence of face recognition technology, privacy concerns arise, making adversarial bandages a vital tool to protect one's identity from unauthorized recognition. Furthermore, ethical concerns regarding face recognition in various applications, including surveillance and law enforcement, underscore the need for adversarial bandages as a means for individuals to assert control over their representation and guard against potential bias and discrimination. The development of adversarial bandages not only enhances privacy and security but also drives innovation in computer vision and machine learning by challenging researchers to bolster the robustness of face recognition algorithms. Consequently, robust adversarial attacks and defense mechanisms play a pivotal role in advancing the security and reliability of deep neural networks in real-world scenarios.

1.5 Objective

The objective of this research is to address the critical problem statement: *To generate an adversarial bandage that misleads the face recognition model under targeted and untargeted attacks in white-box settings.* This multifaceted problem statement encompasses several key challenges and objectives. The primary aim of this research is to design and develop adversarial bandages, specialized image overlays, or modifications with the capacity to deceive face recognition models. These bandages should be capable of effectively misguiding the model's predictions, leading to instances of incorrect or unauthorized identification. The research will investigate two fundamental modes of adversarial attacks: targeted and untargeted. In targeted attacks, the adversarial bandage will be tailored to specifically deceive the model into recognizing a predetermined individual. Conversely, untargeted attacks involve the creation of bandages aimed at causing general misidentification without a specific target in mind. This differentiation will enable a comprehensive understanding of the bandages' versatility and effective-

ness. The research will operate within white-box settings, implying that the attacker has full access to the architecture and parameters of the face recognition model. This setting allows for a deeper exploration of vulnerabilities and the development of highly effective adversarial bandages.

1.6 Contributions

The main contributions of the our thesis are listed as follows:

- The primary contribution of this research lies in assessing the vulnerability of the face recognition systems to patch-based adversarial attacks.
- Secondly, adversarial bandages were generated in the white-box attack setting for targeted and untargeted attacks.
- Reconstruction loss and total variation loss was incorporated in the attack framework to minimize patch conspicuousness and maximize smoothness.

1.7 Thesis outline

The rest of the thesis is organised as follows. Chapter 2 discusses the literature survey on similar works. Chapter 3 discusses about the architecture and loss functions which are used in FaceNet [15]. We are using FaceNet as the state-of-the-art architecture for facial recognition to perform attacks. It also describes the possible threats and the proposed algorithms to attack on FaceNet. Chapter 4 discusses the experiments performed to craft adversarial bandages. We conclude in Chapter 5.

CHAPTER 2

Literature Review

The development in adversarial machine learning by Szegedy et al. [10] was significant in exposing the flaws in state-of-the-art DNNs. Even extremely accurate DNNs have blind spots or flaws that adversaries can use to create adversarial samples. One important finding is that a machine learning model, particularly a DNN, can be easily tricked by making small changes to the distribution of input data. This property of DNNs is used in adversarial attack techniques to create adversarial samples, which are purposely produced inputs intended to trick the model and yield false or unexpected results. Important concerns regarding the reliability and security of machine learning systems are raised by the prevalence of vulnerabilities in DNNs and the ease with which adversarial samples can be produced.

One such attack was proposed by Cheng et al. [6] on emails to bypass spam detection model. Authors converted email texts into feature vectors using Word2vec, Doc2vec, or Term Frequency - Inverse Document Frequency (TF-IDF) vectorization methods. They used projected gradient descent on the set of spam email feature vectors. Perturbations are obtained, which are later converted back into words named *magic words*. These words are then added to spam emails and fed to spam detectors to verify the effectiveness of bypassing the detection. They have also discussed a black-box scenario where they obtain magic words using Support Vector Machine (SVM) classifier and use it on another classifier where the type and weights of classifiers are unknown.

Similarly, Yoshida and Okuda [24] proposed an approach to produce adversarial examples to fool image cropping systems that Twitter and Netflix use. They used a gradient-based attack on the model, which predicts a saliency map to shift the cropping images. They also introduced a novel method to evaluate the effectiveness of the image cropping model.

Xu et al. [23] proposed an efficient attack against scene text recognition. This attack includes operations on character and word levels. For instance, the word

Tiger is manipulated by insertion (*Timger*), substitution (*Tigar*) and deletion (*Tigr*). Previous research made changes in random pixels, while Du et al. [8] suggested a patch-based physical adversarial attack on aerial imagery object detectors. They proposed an on-body attack, which is physically realizable, where the generated patch is applied on the roof of a car to stay undetected from detectors. Figure 2.2 shows optimisation process to generate the patch. They have even proposed novel attack which surrounds the target object with generated adversarial patterns as shown in Figure 2.1. They also devised novel metrics to evaluate the efficiency of physical attacks. The loss function is defined as:

$$L_i(p) = \max \left(\mathcal{S}_i^{\mathcal{U}} \right) + \delta \cdot NPS(P) + \gamma \cdot TV(P) \quad (2.1)$$

where $\mathcal{S}_i^{\mathcal{U}}$ is the predicted objectness score, and δ, γ are weights for the *Non – printability score* (NPS) and Total Variation (TV) of P respectively. Here P denotes patch. The NPS term is used to enforce colors in the patch P to be as similar as possible to colors that a printing device can accurately reproduce as shown in Equation (2.2). It encourages spatially smooth and printable colors in the output image.

$$NPS(P) = \sum_{u,v} \left(\min_{c \in C} \|p_{u,v} - c\|_2 \right) \quad (2.2)$$

Here, $p_{u,v}$ is the pixel at (u, v) index in P , and colour vector from the set of printable colours is denoted by c . TV contributes to the physical realizability of P by penalizing sharp changes or abrupt transitions between neighboring pixel values as shown in Equation (2.3).

$$TV(P) = \sum_{t,u,v} \sqrt{(p_{t,u,v} - p_{t,u+1,v})^2 + (p_{t,u,v} - p_{t,u,v+1})^2} \quad (2.3)$$

where $p_{t,u,v}$ represents the pixel value of channel t (red, green, or blue) at position (u, v) in P .



Figure 2.1: Adversarial patch off-and-around a car [8]

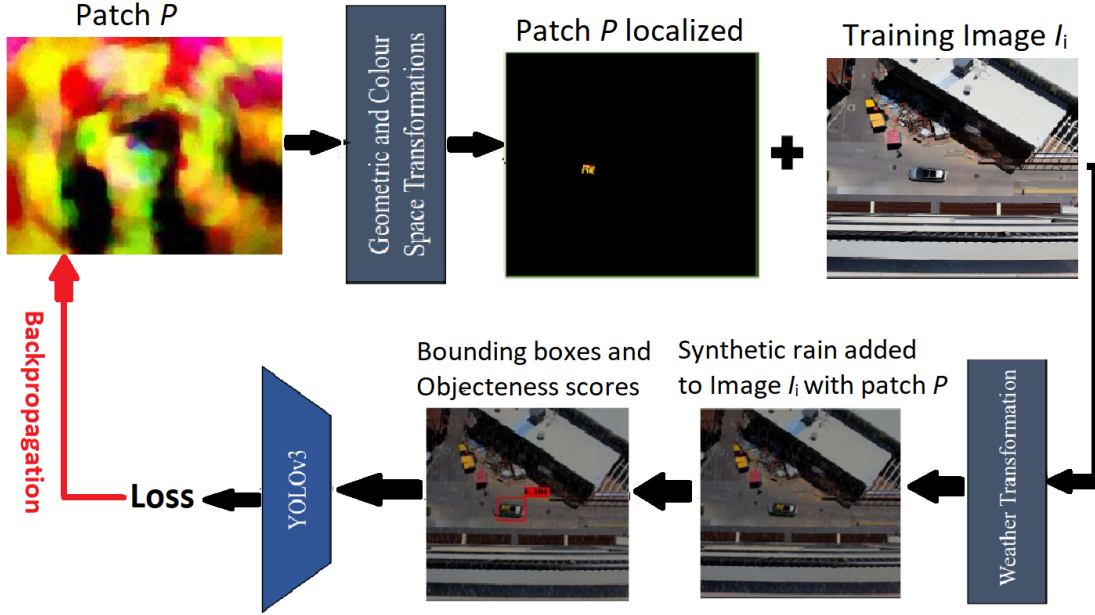


Figure 2.2: Optimisation process for adversarial attack on an aerial imagery object detector [8]

Although adversarial patches off and on the car are physically realizable, carrying them out requires a lot of effort by the attacker. On the other hand, attack proposed by Duan et al. [9] is the first light-based attack which employs laser beams to fool DNNs within a blink. They trained the laser’s wavelength, layout, width, and intensity to attack as shown in Figure 2.3. This attack introduces a dominant feature in the acquired image that leads to the prediction of classes related to lighting such as candle and lamps. They have used 1000 correctly classified classes from ImageNet and crafted adversarial examples for each image with a simulated laser beam for which the success rate is 95.1% for digital settings and 77.43% for real-world scenarios. However, their approach requires using different laser beams for different input images.

Contrary to several previous works that involve generating input-specific adversarial patterns, Brown et al. [3] worked on creating a universal adversarial patch that can be used in the real world simply by pasting it near the input image. They used the following objective function shown in Equation (2.4) to generate adversarial patches:

$$\hat{p} = \arg \max_p \mathbb{E}_{x \in X, t \in T, l \in L} [\log \Pr(\hat{y} | A, p, x, l, t)] \quad (2.4)$$

where X is a training set of images, T is a distribution over transformations of

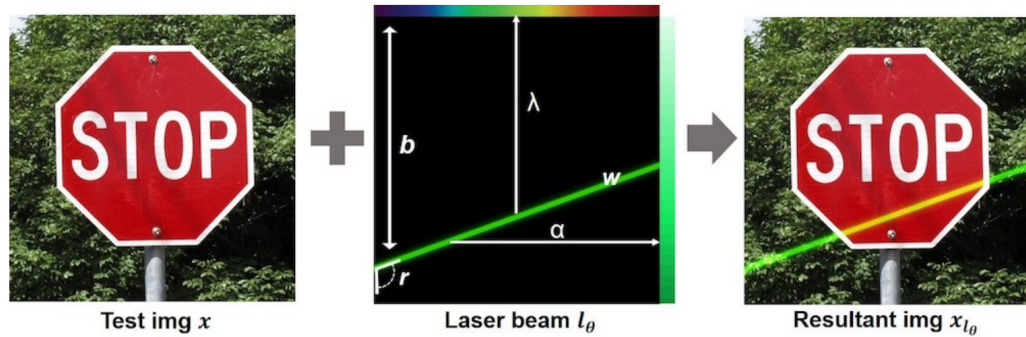


Figure 2.3: Laser beam attack [9]

the patch, L is a distribution over locations in the image, $A(p, x, l, t)$ is a patch application operator where x is input image, \hat{y} is target class, p is patch, l is patch location and t is transformation. They showed that even small patches caused misclassification. A trained patch \hat{p} is shown in Figure 2.4.



Figure 2.4: Adversarial patch [3]

Although attack success rate of physically realizable attacks such as [3] is high, it is dependent on the viewpoint. On the other hand, Hu et al. [11] proposed a cloth-based multi-angle attack to evade person detector. The textural pattern of the cloth is adversarially generated. Authors have proposed a novel Toroidal-Cropping-based Expandable Generative attack, which consists of two main stages. Firstly, adversarial patterns are generated using a fully convolutional network, and in the second stage, the best latent pattern is selected which is repeatedly printed on cloth and can be used in real life to fool a person detector.

Previous works have focused on fooling either a detection or recognition model. However, Lifeng et al. [12] proposed a UPC attack that consists of attacks on both. The first stage of UPC fools the Region Proposal Network (RPN) to reduce fore-

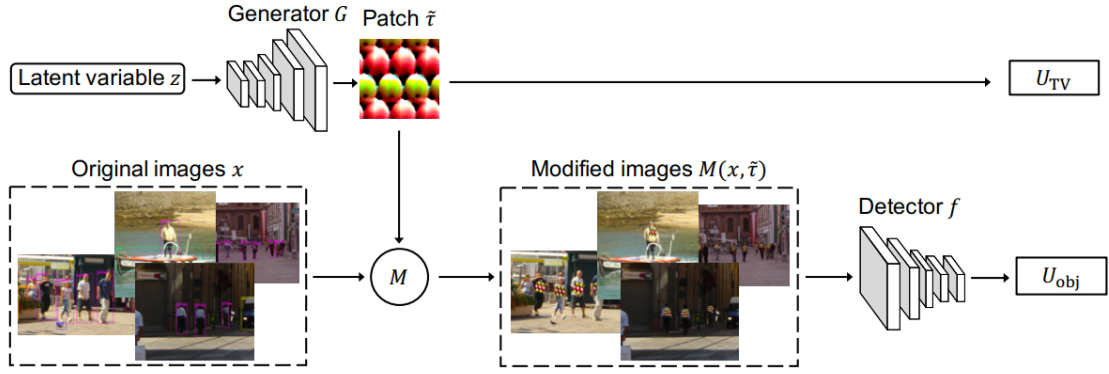


Figure 2.5: The pipeline of the adversary objective function. [11]

ground proposals. The loss function for fooling the RPN is given as:

$$L_{rpn} = \mathbb{E}_{p_i \sim P} (\mathcal{L}(s_i, y^t) + s_i \|\vec{d}_i - \Delta \vec{d}_i\|_p) \quad (2.5)$$

where y^t is the target score, and they set y^1 for background and y^0 for foreground; \mathcal{L} is the Euclidean distance; P is the output proposals; s_i is the confidence score for the i -th bounding box; \vec{d}_i represents the coordinates of i -th bounding box; $\Delta \vec{d}_i$ is a pre-defined vector, which is used for attacking proposals by shifting the center coordinate and corrupting the shape of original proposals.

In the second stage, these proposals are misclassified by fooling the classifier and regressor as shown in given Figure 2.6. Equation (2.6) and (2.7) are to fool the classifier and regressor respectively are as below:

$$L_{cls} = \mathbb{E}_{x \sim \hat{P}} C(x)_y + \mathbb{E}_{x \sim P^*} L(C(x), y') \quad (2.6)$$

$$L_{reg} = \sum_{x \sim P^*} \|R(x)_y - \Delta \vec{d}\|_l \quad (2.7)$$

where L is the cross-entropy loss, $C(x)$ and $R(x)$ are the outputs of the classifier and the regressor, respectively. \hat{P} is the subset containing top k proposals after applying non-maximum suppression on the output of region proposal network. P^* is the set of proposals corresponding to the true label y , and y' is the target label. $\Delta \vec{d}$ denotes the distortion offset. Additionally, they employ TV loss to make the adversarial patch look natural. After combining all the above loss functions and adding TV loss, final objective function is as shown in Equation (2.8).

$$\arg \min_{\Delta \delta} \mathbb{E}_{\hat{x} \sim \hat{X}} (L_{rpn} + \lambda_1 L_{cls} + \lambda_2 L_{reg}) + \text{TV}(\delta^t) \quad (2.8)$$

where δ and \hat{X} denotes the universal pattern and the set of perturbed images, respectively.

$$\text{TV}(r) = \sum_{i,j} ((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2)^{\frac{1}{2}} \quad (2.9)$$

Equation (2.9) represents the total variation loss (TV) where $r_{i,j}$ indicates pixel intensity at the $(i, j)^{th}$ location in the patch. It ensures minimal Euclidean distance between adjacent pixel values so that the overall bandage looks more natural and becomes natural.

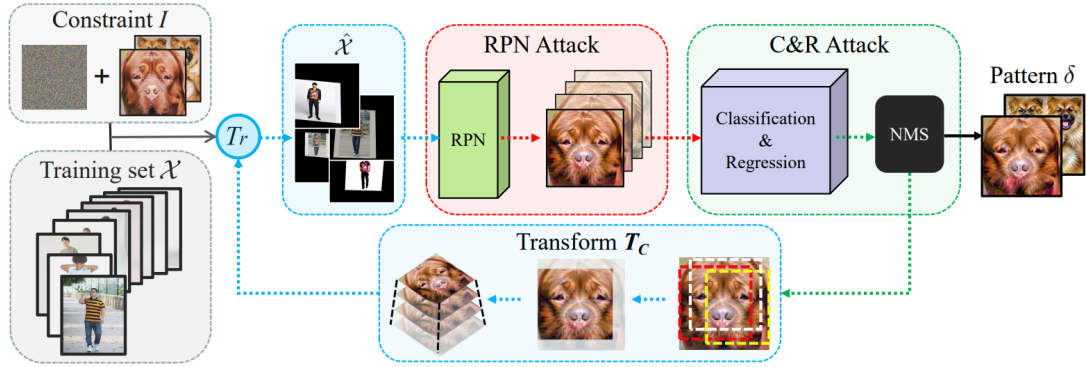


Figure 2.6: Overall pipeline of UPC [12]

Authors also contributed the first standardized virtual database namely, *AttackScenes* to make a fair comparison between attacks on object detectors.

Majority of the prior works focus on attacking the models after deployment. On the other hand, Wenger et al. [21] proposed a poisoning attack which involves corrupting the database at the time of training the model. Specifically, they fooled a face recognition model by considering seven physical objects including dots, tape, bandana and earrings as triggers. During training of model, they appended the trigger objects on benign inputs digitally and mislabeled them to the target class. Particularly, they assigned a single target label to all images of different subjects containing a particular trigger. In this manner, they trained the models for all the triggers. During testing, to realize a physical attack, an object similar to the trigger is worn by the subject resulting in misclassification to the target class assigned to the trigger. Additionally, they found out in their research that their attack efficiency is reduced when trigger objects are used far from the center of the face as shown in Figure 2.7.

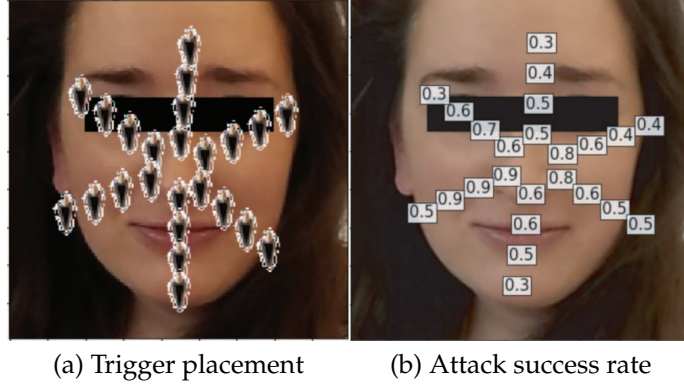


Figure 2.7: Effects of triggers with different locations [21]

Another attack on face recognition model proposed by Sharif et al. [17] is based on generating a perturbed eyeglass frame as shown in Figure 2.8. The eyeglass frame with the adversarial pattern is 3D printed and is worn by the attacker at the time of testing. To generate the adversarial pattern, authors first digitally rendered solid colored frames onto the subject, trying to attack and update frame color iteratively using Gradient Descent. They employed the loss function for targeted attack Equation (2.10) and untargeted attack Equation (2.11), respectively:

$$\arg \min_{\delta} (\text{softmaxloss}(f(x + \delta), y_t)) \quad (2.10)$$

$$\arg \min_{\delta} (-\text{softmaxloss}(f(x + \delta), y_x)) \quad (2.11)$$

where x is the input image, δ is the perturbation and, y_t and y_x denote the target and original class, respectively. here, softmaxloss is,

$$\text{softmaxloss}(a, b) = -\log \left(\frac{e^{a,b}}{\sum_{c=1}^N e^{c,b}} \right) \quad (2.12)$$

In order to accommodate the errors introduced by the movement in the physical world, authors moved the frame up to three pixels horizontally or vertically and rotated it slightly.

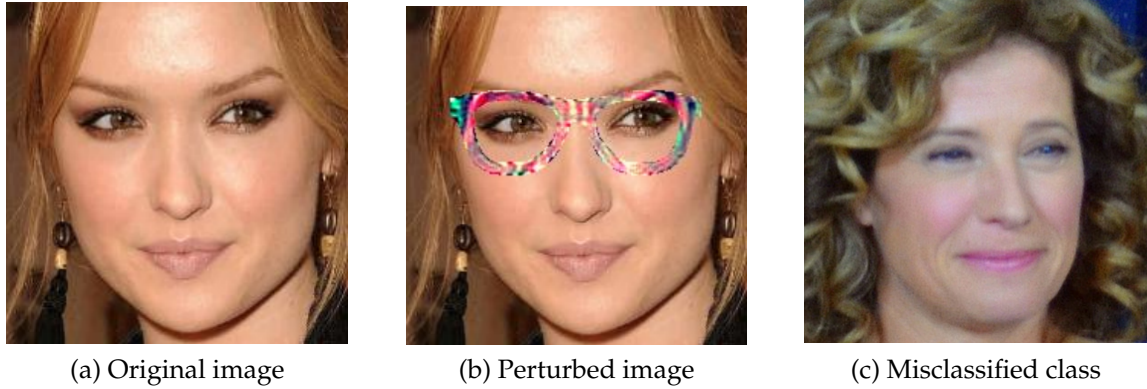


Figure 2.8: Impersonation using eyeglass frame [17]

Unlike previous adversarial patches that relied on designing perturbations, Wei et al. [20] used real stickers that already exist in our daily life. The position and rotation angle of the sticker is less affected by printing loss and color distortion, providing a key advantage in maintaining the attacking performance in the physical world. The attacks were conducted in a black-box setting with limited information about the targeted system, making the adversarial stickers more practical. To effectively determine the sticker’s parameters, the authors proposed the Region-based Heuristic Differential Evolution Algorithm. This algorithm utilized regional aggregation of effective solutions and an adaptive adjustment strategy for evaluation criteria. The method is verified in face recognition and later extended to image retrieval and traffic sign recognition tasks. In contrast, our proposed method is fixing the position of a patch and training it to look off-the-shelf bandage that is used in day-to-day life.

Moreover, Xiao et al. [22] suggested a method that evaluates the robustness of face recognition models against adversarial patches using transferability with limited attacker access to the target models. The effectiveness of the proposed method in a black-box setting is showcased through extensive experiments conducted in the digital world. The authors extend transfer-based attack techniques to generate transferable adversarial patches. They observed that transferability is sensitive to initialization and degrades with large perturbation magnitudes, indicating overfitting to substitute models. To overcome this, they proposed a regularization approach that utilizes a low-dimensional data manifold represented by generative models pre-trained on legitimate human face images. By optimizing face-like features as adversarial perturbations on the manifold, they successfully reduced the gaps between substitute responses of models and target models, enhancing transferability.

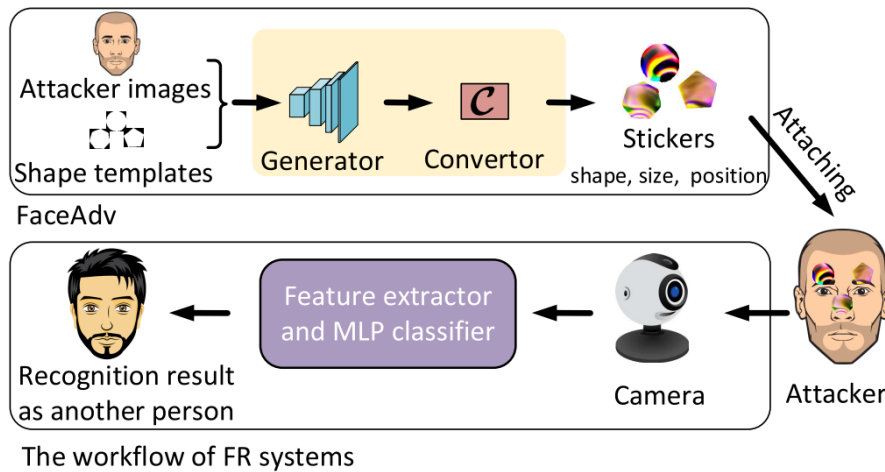


Figure 2.9: The physical-world attacks on FR systems using adversarial stickers crafted by FaceAdv [18]

Unlike existing attacks that primarily operate in the digital realm or rely on specialized equipment, Shen et al. [18] proposed a novel approach FaceAdv involving the creation of adversarial stickers. The attack consists of two main components: a sticker generator and a converter as shown in Figure 2.9. The sticker generator is responsible for crafting stickers with various shapes, while the converter digitally attaches these stickers to human faces as shown in Figure 2.10 and provides feedback to the generator for enhancing its effectiveness. To evaluate the performance of FaceAdv, the authors conducted extensive experiments targeting three typical face recognition systems: ArcFace, CosFace, and FaceNet. The results indicate that FaceAdv outperforms a state-of-the-art attack by significantly improving the success rates of both dodging (untargeted) and impersonating attacks.


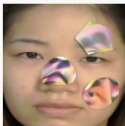
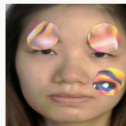
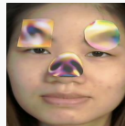


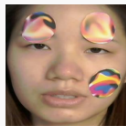
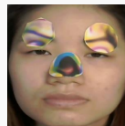
Attacker	Mode	Target	Target Model		
			ArcFace	CosFace	FaceNet
	Dodging	Another person			
	Impersonating				

Figure 2.10: Samples of dodging and impersonating attacks [18]

The literature survey is summarised and compared in the following Table 2.1 using Adversarial knowledge, Adversarial aim, and type of Realizations.

Table 2.1: Type of attacks and categorization of adversarial attacks on ML models

Papers	Application	Adversarial knowledge	Adversarial aim	Realizations
[12]	Object detection	White-box	-	Both
[11]	Person detection	White-box	-	Both
[21]	Person Recognition	White-box	Targeted	Both
[17]	Person Recognition	Both	Both	Both
[24]	Image cropping	White-box	Targeted	Both
[9]	Object recognition	Both	Untargeted	Both
[8]	Aerial imagery object detection	White-box	-	Physical
[6]	Email Spam Detection	Both	-	Digital
[23]	Text Recognition	Both	Both	Both
[3]	Object Recognition	White-box	Targeted	Both
[20]	Object Recognition	Both	Untargeted	Both
[22]	Transferability	Both	-	Both
[18]	Person Recognition	Both	Both	Both

CHAPTER 3

Methodology

Prior work has fooled face detectors using adversarial patches. Unlike previous work that mainly focused on fooling person detectors, we explored more about fooling person recognition models [20]. We generated an adversarial patch in such a manner that by imposing it on any images, the classifier will output as chosen target class. We used weights of classifiers during training of the adversarial patch, which will come under white-box settings. Initially, we experiment by generating a general adversarial patch. Later, during the training phase of a patch, we restricted its shape and size to make the patch look more like bandages. We added deformations and transformations on adversarial patches during training, making patches robust against different viewpoints or locations of patches.

In this chapter, the methodology adopted in our work is discussed. Section 3.1 discusses the threat model, Section 3.2 describes the target model's architecture, and the proposed approach is presented in Section 3.3.

3.1 Threat model

3.1.1 Attacker's goal

The purpose of creating an adversarial bandage is to trick or deceive a facial recognition system like FaceNet, ArcFace [7], or CosFace [19] into misclassifying or failing to recognise the person wearing the bandage. The objective of the attacker is to take advantage of flaws or weaknesses in the decision-making or algorithmic processes of the facial recognition system.

3.1.2 Attacker's knowledge

The target machine learning model's architecture, parameters, and access to training data are all believed to be known to the attacker. They might also be familiar

with the precise distribution of the input data or have access to a subset of the training data. This assumption is true, for instance, where the targeted detector is known to be derived from an open-source implementation, and the attacker has access to and can reverse-engineer an implementation of a black-box detector. However, the most significant benefit of this assumption is that it reflects the worst-case situation for the defender, allowing us to calculate the greatest harm the attacker may inflict.

3.1.3 Attacker’s capabilities

The attacker is adept at manipulating images and is aware of the aspects of facial images that affect facial recognition. The targeted facial recognition system’s limitations and weaknesses are known to the attacker. An attacker can take advantage of particular vulnerabilities with this knowledge and may employ strategies to guarantee that the changed bandage looks similar to off-the-shelf bandages making it difficult for the system to distinguish between the real bandage and the adversarially crafted bandage.

3.1.4 Attacker’s strategy

To produce the desired adversarial effect, the attacker may employ several methods to perturb or modify the small region of the image referred to as an adversarial patch. This may include methods such as generative models, evolutionary algorithms, or gradient-based optimisation. In order to increase the effectiveness of the adversarial patch, the attacker iteratively improves it by modifying its parameters and optimisation strategies. The patch must be regularly assessed and modified during this process to overcome any defences put in place by the facial recognition system. The attacker wants to render the adversarial patch visually not too conspicuous and resistant to changes in light, image transformations, and noise. This makes sure that the patch is still reliable and challenging to detect in practical situations.

3.2 Facenet

Google researchers created FaceNet [15], a deep-learning network with the goal of producing high-dimensional embeddings for facial images. It is made to extract and represent facial features so that face recognition and verification tasks

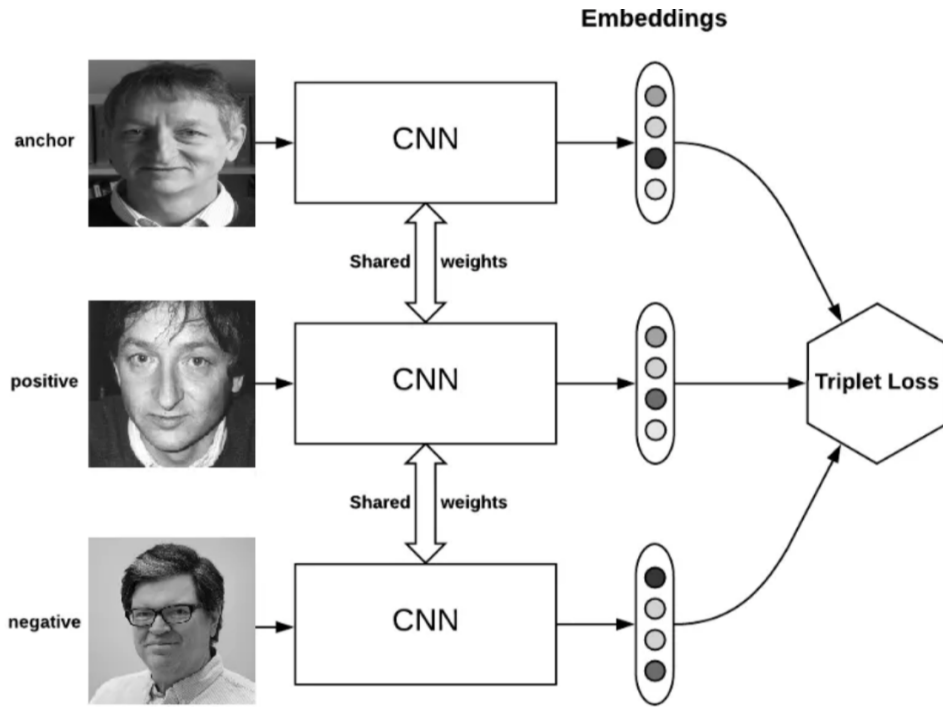


Figure 3.1: Triplet loss [13]

are made easier. FaceNet’s main goal is to develop a mapping from facial images to a small, continuous vector space, where the distances between the vectors represent similarity or dissimilarity between the associated faces. A deep convolutional neural network architecture is used to learn this mapping. FaceNet gains the ability to train itself to optimize the embedding space so that similar faces are mapped close together, and dissimilar faces are mapped far apart. This is accomplished by using triplet loss, which promotes the network to maximise the distance between an anchor face and a negative face (of a different identity) while minimising the distance between the anchor face and a positive face (of the same identity) as shown in Figure 3.1. The triplet loss L is defined as follows:

$$L = \sum_{i=1}^N \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + \alpha \right] \quad (3.1)$$

where $f(x_i^a)$, $f(x_i^p)$, $f(x_i^n)$ are embeddings of anchor, positive and negative samples, respectively, and α is the margin that is enforced to differentiate between positive and negative pairs. Here, N denotes total number of images present in the dataset. The aim is to ensure that the embeddings of positive images (photos of the same person as the anchor) are closer to the anchor embedding than the embeddings of negative images (images of different persons). An anchor image is chosen as a reference. Several convolutional and pooling layers are commonly used in the FaceNet architecture to extract hierarchical features from facial images,

and then fully connected layers are used to create the final embedding vector as shown in Figure 3.2. After being trained, FaceNet embeddings can be applied to a variety of tasks involving faces, including face verification (figuring out if two faces belong to the same person), face clustering, and face recognition. The compact and discriminative character of the embeddings makes large-scale face matching across databases possible.

After FaceNet creates the embeddings, they can be used as feature vectors to represent an individual’s facial traits. This enables the use of common methods from diverse fields, such as SVM or clustering. The produced embeddings can be used as feature vectors for face recognition and compared using cosine distance or Euclidean distance. The system may determine the closest neighbours (matching faces) based on the distances between the embeddings by using SVM or comparable methods. Similar to face verification or authentication, it is possible to specify a threshold value to assess whether or not two embeddings belong to the same person. The faces are a match if the distance between two embeddings is less than the threshold, indicating that they belong to the same individual.

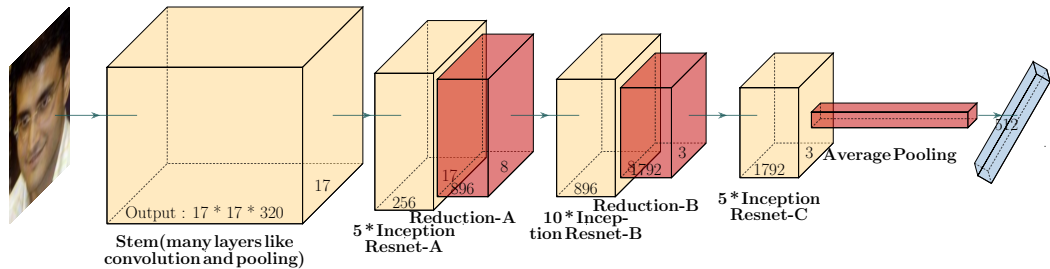


Figure 3.2: Architecture of Inception Resnet v1

3.3 Proposed approach

Based on the threat model, this section discusses optimisation of adversarial bandages. The goal of Fast Gradient Sign Method (FGSM) is to perturb the input image by adding a small noise vector to it in such a way that the resulting perturbed image causes the FaceNet model to misclassify the image, while still being visually similar to the original image. FGSM works by perturbing the input data in a way that is most likely to cause misclassification. It does this by computing the gradient of the model’s loss function with respect to the input data and then adjusting the input data in the direction that increases the loss the most for a given magnitude of perturbation. The idea is that by making a small modification to the

input data, the model’s decision boundary can be crossed, resulting in a different prediction. The FGSM attack on FaceNet involves the following steps: Given an input image x , patch p is appended on x using mask m , and the output of the FaceNet model for that image is computed. The gradient of the loss function with respect to the obtained embedding is calculated. The loss function is the Euclidean distance between input and output embedding. The sign of the gradient is computed to determine the direction in which to perturb the input image. The perturbation is a small value multiplied by the sign of the gradient. The perturbation is added to the input image to generate the adversarial example. Patch is extracted from the resulting adversarial example and is used for the next images in the dataset to make it universal.

3.3.1 Algorithm

Algorithm 1: Universal Targeted Adversarial Bandage

Input: Image $x \in X$; target image t ; embedding of t t_embed ; $epoch = 20$;
 $max_iter = 500$; $step_size = 0.3$; $model =$ Inception ResNet v1;
Classifier $cls =$ SVM ;patch p ; mask m

Output: patch

```

1 Initialize patch  $p$ 
2  $t\_embed = model(t)$ 
3 for  $i = 1$  to  $epoch$  do
4     for each  $x$  in  $X$  do
5         for  $i = 1$  to  $max\_iter$  do
6              $\hat{x} = (1 - m) \odot x + m \odot p$ 
7              $\hat{x}\_embed = model(\hat{x})$ 
8             if  $cls(t\_embed) = cls(\hat{x}\_embed)$  then
9                  $break$ 
10             $L = \|\hat{x}\_embed - t\_embed\| + \delta \cdot RC(p) + \gamma \cdot TV(p)$ 
11             $\nabla L = sign(grad(L, \hat{x}\_embed))$ 
12             $\nabla L = \nabla L \odot m$ 
13             $p = p + step\_size * \nabla L$ 
14             $p = clip(p, 0, 1)$ 
15        end
16    end
17 end
18 return  $p$ 

```

The provided algorithm is a procedure for generating a patch image that can be embedded into a target image in such a way that it is classified as belonging to the same class as the target image by a pre-trained classifier. This algorithm aims to optimize the patch in terms of its similarity to the target image and certain constraints. The algorithm takes several input parameters, including an input image x , a target image t , an embedding of the target image t_embed , the number of training epochs, maximum iterations per epoch, step size for updates, a pre-trained model (Inception ResNet v1), a classifier (SVM), an initial patch p , and a mask m . It initializes the patch p and computes the embedding of the target image t using the specified model. The algorithm iterates through a specified number of training epochs, and for each epoch, it iterates through all images x in the dataset X . For each x , it conducts an inner loop for a maximum number of iterations. Within this loop, it blends the input image x with the patch p using the mask m to create a modified image \hat{x} . It computes the embedding of \hat{x} using the same model. If the classifier's prediction on \hat{x} matches that on the target image t , the loop breaks. Otherwise, it calculates a loss function L that balances the embedding similarity between \hat{x} and t_embed with additional terms that encourage regularization (represented by $RC(p)$ for some regularization term and $TV(p)$ for total variation). It computes the gradient of L with respect to \hat{x}_embed and scales it by the $mask$ to focus on the patch region. The patch p is updated using gradient ascent with a specified step size and is clipped to ensure pixel values remain within a valid range (0 to 1). This process repeats for the specified number of epochs, and the final optimized patch p is returned as the output. The algorithm aims to generate a patch that can be added to an input image in a way that fools the classifier into classifying the modified image as belonging to the same class as the target image t while adhering to certain constraints.

3.3.2 Loss function for targeted

The main aim of the loss function is to reduce the distance between input embedding and target embedding. Thus, loss function obtained is as shown below in Equation (3.2):

$$L = \|f(x^o) - f(x^t)\| + \delta \cdot RC(p) + \gamma \cdot TV(p) \quad (3.2)$$

where $f(x^o)$ and $f(x^t)$ denote embeddings of original and target images, respectively and p denotes patch. Reconstruction RC ; Total variation loss TV ; γ and δ are regularization parameters.

Algorithm 2: Universal Untargeted Adversarial Bandage

Input: Image $x \in X$; embedding of x x_embed ; embedding of \hat{x} \hat{x}_embed ;
 $epoch = 20$; $max_iter = 500$; $step_size = 0.3$; $model =$ Inception
ResNet v1; Classifier $cls =$ SVM; $patch$ p ; mask m

Output: patch

```
1 Initialize patch  $p$  as shown in Figure 4.3
2 for  $i = 1$  to  $epoch$  do
3   for each  $x$  in  $X$  do
4      $x\_embed = model(x)$ 
5     for  $i = 1$  to  $max\_iter$  do
6        $\hat{x} = (1 - m) \odot x + m \odot p$ 
7        $\hat{x}\_embed = model(\hat{x})$ 
8       if  $cls(x\_embed) \neq cls(\hat{x}\_embed)$  then
9          $break$ 
10       $L = -\|\hat{x}\_embed - x\_embed\| + \delta \cdot RC(p) + \gamma \cdot TV(p)$ 
11       $\nabla L = sign(grad(L, \hat{x}\_embed))$ 
12       $\nabla L = \nabla L \odot m$ 
13       $p = p + step\_size * \nabla L$ 
14       $p = clip(p, 0, 1)$ 
15    end
16  end
17 end
18 return  $p$ 
```

The algorithm presented is designed for generating adversarial patches to deceive a pre-trained image classification model. It begins by taking several inputs, including an image x from a set of images X , embeddings of the original image x_embed , embeddings of the adversarial image \hat{x}_embed , and various hyperparameters like the number of training epochs, maximum iterations per image, step size for gradient descent, the pre-trained image classification model (Inception ResNet v1), a classifier (SVM), an initial patch p , and a mask m . To start, the algorithm initializes the adversarial patch p . This patch is a modification that will be applied to the original image to craft an adversarial image that can fool the model. The initial patch p is formed based on some reference or pre-defined pattern, illustrated in Figure 4.3. The core of the algorithm is a nested loop structure. It first iterates through each epoch, and within each epoch, it processes each image x in the dataset X . For each image x , it creates an adversarial image \hat{x} by blending the original image x and the patch p , with the blending controlled by the mask m .

This blending gradually alters the original image to craft an adversarial example. Next, the algorithm computes embeddings for both the original image x and the adversarial image \hat{x} using the pre-trained model. These embeddings represent the underlying features of the images in a lower-dimensional space, capturing essential characteristics. Within the inner loop, it iterates a maximum number of times specified by max_iter . During each iteration, it checks if the predictions of the classifier on the original image x and the adversarial image \hat{x} differ. If they do, it exits the loop, indicating that the adversarial patch is successful. Otherwise, it calculates a loss function L , which comprises terms related to the difference between embeddings, regularization on the patch p (controlled by the $RC(p)$ term), and total variation (TV) of the patch (controlled by the $TV(p)$ term). The algorithm then computes the gradient of the loss with respect to the embedding of the adversarial image, \hat{x}_{embed} , and multiplies it by a mask. This gradient is used to update the patch p . The step size for this update is controlled by $step_size$, and the patch values are clipped to ensure they remain within a valid range (typically between 0 and 1). The training process continues for multiple epochs, with the goal of finding a patch p that, when applied to an image, makes the classification model misclassify it. The final adversarial patch p is returned as the output of the algorithm.

3.3.3 Loss function for untargeted

The main aim of the loss function is to increase the distance between input embedding and embeddings generated after applying a perturbed bandage. Reconstruction loss (RC) is added to penalize large differences between the original input and its reconstructed version, where reconstruction is performed using the adversarial example as input. Total variation loss (TV) is employed to make the bandage look natural and imperceptible. Thus, loss function obtained is as shown below in Equation (3.3):

$$L = - \|f(x^o) - f(x^a)\| + \delta \cdot RC(p) + \gamma \cdot TV(p) \quad (3.3)$$

where $f(x^a)$ denotes embedding of input image before applying bandage.

Reconstruction Loss

$$RC(p, \hat{p}) = \|p - \hat{p}\| \quad (3.4)$$

Here p denotes the original patch, and \hat{p} denotes the updated patch after adding perturbations. The reconstruction loss, as shown in Equation (3.4) is computed by comparing the modified or adversarial input data with the original input data. It quantifies the differences between the two and shows the amount of distortion that was caused by the attack. The attack becomes more undetectable to human observers as the reconstruction loss decreases.

Total Variation Loss

$$\text{TV}(p) = \sum_{i,j} ((p_{i,j} - p_{i+1,j})^2 + (p_{i,j} - p_{i,j+1})^2)^{\frac{1}{2}} \quad (3.5)$$

The TV loss, as shown in Equation (3.5) calculates the degree of variation or gradient between adjacent pixels. It captures the overall variations in brightness or color values throughout the image. The TV loss is minimal, indicating a smoother region when adjacent pixels have values that are similar to one another. One can encourage the resulting images to have smoother differences between nearby pixels by including the TV loss in the optimization or training process. The TV loss can be employed as a regularizer in adversarial bandage production to urge the resulting bandage picture to have smooth transitions and prevent sharp artifacts. It facilitates the development of coherent and plausible adversarial patches that seamlessly integrate with the original image.

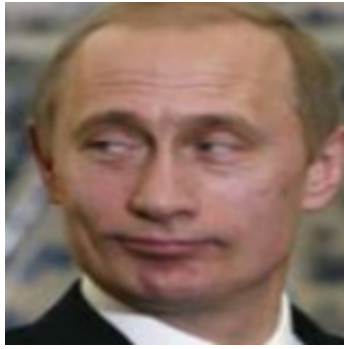
CHAPTER 4

Experiments

4.1 Dataset description

The LFW dataset is a collection of human face photos gathered from various sources on the web as shown in Figure 4.1. It consists of 13,229 images, comprising 5,749 different classes with associated labels. These images exhibit a wide range of variations in poses, expressions, lighting conditions, ethnicities, ages, and genders. To conduct specific experiments, a subset of the dataset was created, consisting of 5,885 images from 483 classes. The selection criteria for this subset involved choosing classes with more than 5 images. In the LFW dataset, each face photo is linked to the identity of the person depicted, resulting in pictures of more than 5,000 different individuals. The images are in JPEG format and come in various resolutions. For consistency in experiments, the images are standardized to a size of 160x160 pixels. Typically, the LFW dataset is divided into a training set and a test set. The training set is used to train facial recognition algorithms and typically includes 4 photographs per individual. The test set is then utilized to evaluate the performance of the facial recognition algorithm.

The LFW dataset poses significant challenges for face recognition systems due to the diverse variations present in the photos. These challenges include differences in lighting conditions, facial poses (frontal vs. non-frontal faces), facial expressions, occlusions, and image quality. Successfully addressing these challenges and achieving accurate face recognition on the LFW dataset requires the development of robust algorithms capable of effectively handling these variations. As a benchmark dataset, LFW plays a crucial role in evaluating the performance of face recognition algorithms in real-world scenarios.



(a) Vladimir Putin



(b) Michael Jackson



(c) Tom Cruise



(d) Tiger Woods

Figure 4.1: LFW dataset

After adding perturbations to the bandage, the adversarial bandage for targeted attack looks like below, as shown in Figure 4.2.



Figure 4.2: Targeted adversarial bandage

4.2 Evaluation metrics

The *attack success rate* is a measure of the efficiency of adversarial attacks on machine learning models. It measures the proportion of adversarial examples that the model incorrectly classifies. While a lower success rate means that the model is more resistant to adversarial attacks, a greater success rate suggests that the model is more susceptible to such attacks. There are two different types of Attack Success Rate (ASR) as shown in Equation (4.1) and (4.2).

$$ASR_{targeted} = \frac{N_{classified}^t}{N_{dataset}} \quad (4.1)$$

Here, $N_{classified}^t$ denotes the total number of images that are classified as target class t , and $N_{dataset}$ denotes the total number of images present in the dataset.

$$ASR_{untargeted} = \frac{N_{misclassified}}{N_{dataset}} \quad (4.2)$$

Here, $N_{misclassified}$ is the total number of misclassified images.

4.3 Experimental setup

The chosen model for face recognition was FaceNet, and a SVM classifier was utilized. The training dataset consisted of 1692 samples, while the test dataset contained 4293 samples. The initial accuracy achieved on the test set was recorded at 99.03%. The adversarial attacks employed in the experiment aimed to achieve both targeted and untargeted results. The attacks were carried out with white box attack knowledge, meaning the attackers had access to the internal workings of the model. Universal perturbations were applied, affecting the entire dataset. The experiment spanned 10 epochs, with a total of 4000 iterations. A main loss weight of 0.6 was assigned to emphasize the importance of the primary loss function. Additional parameters, such as gamma, delta, and epsilon, were set to 0.2, 0.2, and 0.03, respectively, influencing the adversarial perturbations. By examining these attributes and their corresponding values, the experiment's setup and parameters can be better understood, providing insights into the approach and techniques used for face recognition in the given context.

4.4 Location and size of bandage

The face image is resized to $160\text{px} \times 160\text{px}$ and the bandage image is considered to be of size $30\text{px} \times 80\text{px}$. Consequently, the total area occupied by the adversarial stickers is 9.375% of the original image. Larger stickers generally improve attack success rates but they will look less inconspicuous. So, there is a trade-off between the size of sticker and inconspicuousness. The location of the bandage is fixed at such that the bandage usually covers part of the lower forehead. The upper left corner of the bandage (40,18). The bandage is initialized with the following Figure 4.3.



Figure 4.3: Original bandage

We have generated a binary mask of dimension $160\text{px} \times 160\text{px}$ using the fixed coordinates and image of the bandage. The mask of the bandage is as shown in Figure 4.4.

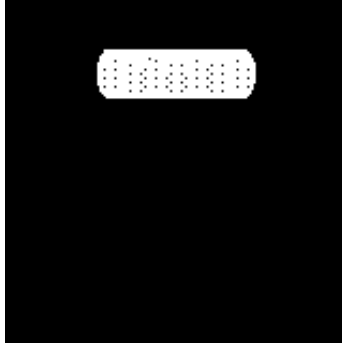


Figure 4.4: Bandage mask

The bandage is superimposed using mask as shown in the given Equation (4.3). Here, m denotes the mask, x denotes the input image, and p denotes the bandage image.

$$\hat{x} = (1 - m) \odot x + m \odot p \quad (4.3)$$

4.5 Results

In this subsection, we will discuss the experimental results to evaluate the effectiveness of our method. The experiments have been carried out in two settings namely, targeted attack and untargeted attack. as discussed below.

Table 4.1: Attack success rate in impersonation and dodging attacks on FaceNet

Mode	Method	ASR(%)
Targeted attack (Impersonation)	Ours	97.51
	FaceAdv [18]	94.56
	AGNs [16]	60.33
Untargeted Attack (Dodging)	Ours	91.78
	FaceAdv [18]	100.00
	AGNs [16]	96.00

4.5.1 Targeted attack

Goal of a targeted attack or impersonation attack is to deceive the system into recognizing the attacker as a specific target individual. We have considered two targets namely, *Adrien Brody* and *Colin Powell* in our study, Table 4.1 compares the performance of different methods for impersonation in terms of the ASR on the FaceNet face recognition system. As can be seen, the proposed method achieves the highest ASR compared to FaceAdv and AGNs methods. For both target individuals, our method achieves ASR of 97.76% and 97.26%, respectively, outperforming the other methods. Figure 4.6 shows the adversarial example for impersonating against FaceNet.

The generation of bandages involves extracting a patch at the end of each execution on the entire dataset and reapplying it for the next epoch. The results, as depicted in the provided bar graph in Figure 4.5 show the ASR for each epoch. Interestingly, the ASR does not exhibit significant changes across epochs, indicating consistent performance. However, it is noteworthy that the number of iterations required to achieve misclassification decreases as the epochs progress. This implies that the bandages become more efficient in achieving their intended purpose over time. Overall, the experimental findings suggest that the bandages maintain a stable ASR while demonstrating improved efficiency in achieving misclassification with each subsequent epoch.

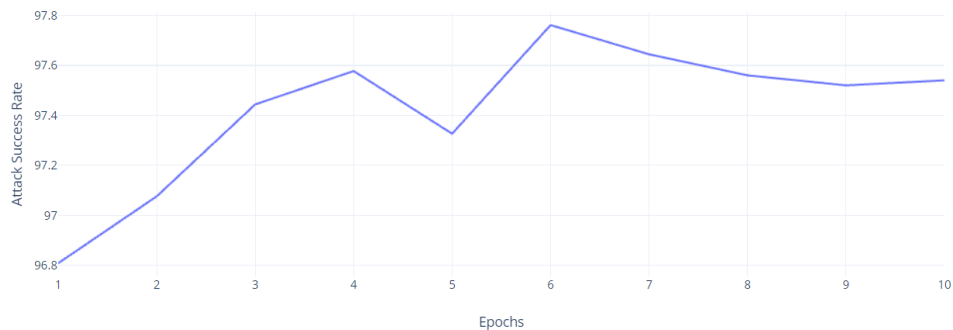


Figure 4.5: Epochs vs. Attack Success Rate

In Figure 2.10, the authors [18] demonstrate their approach of applying three noticeable stickers on the face, occupying a substantial portion of the facial area. The stickers exhibit discernible shapes and sizes, and their generated patterns are easily distinguishable. In contrast, our method utilizes a single patch as shown in Figure 4.6 resembling a bandage, making it an off-the-shelf bandage commonly used in everyday situations. The generated patterns on our patch maintain a more

natural and inconspicuous appearance. Furthermore, it is worth noting that certain classes in the dataset remain unaffected by the bandage, implying a higher Euclidean distance between the target class and the original class. This observation indicates that some classes may possess distinct features that are resilient to the perturbation caused by the bandage, potentially requiring further investigation to improve the effectiveness of the approach on those specific classes.



Figure 4.6: Targeted attack on the class of LFW dataset

After adding perturbations to the bandage, the adversarial bandage for targeted attack looks like below, as shown in Figure 4.7.



Figure 4.7: Targeted attack on the class of LFW dataset

4.5.2 Untargeted attack

The goal of the untargeted attack or dodging attack is to evade recognition altogether. FaceAdv achieves a perfect ASR of 100%, indicating that it successfully

evades detection by the FaceNet system. The proposed method achieves an ASR of 91.78%, while AGNs performs slightly better with an ASR of 96.00%. The reason behind comparatively lower ASR for dodging attacks is that we generate universal bandage while other methods are examples of Individual attacks. Based on the ASRs, the given approach consistently shows competitive performance across both impersonation and dodging attacks, with high ASRs in each category. FaceAdv also performs well in dodging attacks but lags behind in impersonation attacks compared to the given proposed method. AGNs has the lowest ASRs in all attack scenarios. Figure 4.9 and 4.10 show the adversarial example for impersonating against FaceNet. For untargeted attack, bandage looks like below as shown in Figure 4.8.

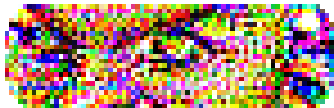


Figure 4.8: Untargeted adversarial bandage



(a) Adversarial image: Salma Hayek



(b) Misclassified as: Wayne Ferreira

Figure 4.9: Untargeted attack



(a) Adversarial image: Gonzalo Sanchez



(b) Misclassified as: Clint Eastwood

Figure 4.10: Untargeted attack

CHAPTER 5

Conclusion

5.1 Conclusion

The targeted adversarial bandages are designed to strategically place specific patterns or designs on bandages to exploit vulnerabilities in facial recognition systems. When attached, these bandages can deceive facial recognition systems, causing misclassification or failure to recognize the attacker's identity. Attack success rate of 97.76% and 91.78% are achieved in untargeted and targeted settings, respectively. This poses a significant challenge to the widespread use and reliability of facial recognition systems. The achieved adversarial success rates through targeted adversarial bandages emphasize the pressing need for robust defenses and countermeasures to protect facial recognition systems from such attacks. To ensure the reliable and trustworthy operation of facial recognition algorithms in critical applications like security, access control, and law enforcement, it is essential to develop mitigation measures that enhance their robustness and accuracy. Acknowledging that the development of targeted adversarial bandages raises ethical concerns regarding surveillance, privacy, and the potential misuse of facial recognition systems.

5.2 Future work

One typically needs access to the database of images to execute the attack, making it an unrealistic assumption in practical situations. On the other hand, physically realizable attacks involve generating adversarial examples in a more realistic manner, assuming no prior access to the original input. Investigate methods to make adversarial targeted bandages universal is significant as they can be deployed on multiple individuals, amplifying the impact and effectiveness of the adversarial attack. We can apply various transformations during the training phase to make it robust for physical attack.

References

- [1] B. G. Atli, S. Szyller, M. Juuti, S. Marchal, and N. Asokan. Extraction of complex dnn models: Real threat or boogeyman? *Engineering Dependable and Secure Machine Learning Systems*, page 42–57, 2020.
- [2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 387–402. Springer, 2013.
- [3] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.
- [4] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021.
- [5] S. Chan. Chapter 3: Adversarial attacks, 2021.
- [6] Q. Cheng, A. Xu, X. Li, and L. Ding. Adversarial email generation against spam detection models through feature perturbation. In *2022 IEEE International Conference on Assured Autonomy (ICAA)*, pages 83–92. IEEE, 2022.
- [7] J. Deng, J. Guo, N. Xue, and S. 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.
- [8] A. Du, B. Chen, T.-J. Chin, Y. W. Law, M. Sasdelli, R. Rajasegaran, and D. Campbell. Physical adversarial attacks on an aerial imagery object detector. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1796–1806, 2022.
- [9] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang. Adversarial laser beam: Effective physical-world attack to dnns in a blink. In *Proceedings*

- of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16062–16071, 2021.
- [10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [11] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, and X. Hu. Adversarial texture for fooling person detectors in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13307–13316, 2022.
- [12] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020.
- [13] D. Kumar. Introduction to facenet: A unified embedding for face recognition and clustering, 2019.
- [14] I. H. Sarker. Machine learning for intelligent data analysis and automation in cybersecurity: Current and future prospects. *Annals of Data Science*, pages 1–26, 2022.
- [15] F. Schroff, D. Kalenichenko, and J. 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.
- [16] M. Sharif, S. Bhagavatula, L. Bauer, and M. Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security*, 22:1–30, 06 2019.
- [17] M. Sharif, S. Bhagavatula, L. Bauer, and M. 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.
- [18] M. Shen, H. Yu, L. Zhu, K. Xu, Q. Li, and J. Hu. Effective and robust physical-world attacks on deep learning face recognition systems. *IEEE Transactions on Information Forensics and Security*, 16:4063–4077, 2021.
- [19] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

- [20] X. Wei, Y. Guo, and J. Yu. Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [21] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6206–6215, 2021.
- [22] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. Gao, X. Zhang, J. Zhou, and J. Zhu. Improving transferability of adversarial patches on face recognition with generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11845–11854, 2021.
- [23] X. Xu, J. Chen, J. Xiao, L. Gao, F. Shen, and H. T. Shen. What machines see is not what they get: Fooling scene text recognition models with adversarial text images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12304–12314, 2020.
- [24] M. Yoshida and M. Okuda. Adversarial examples for image cropping in social media. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4898–4902. IEEE, 2022.
- [25] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, and I. S. Kweon. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498*, 2021.