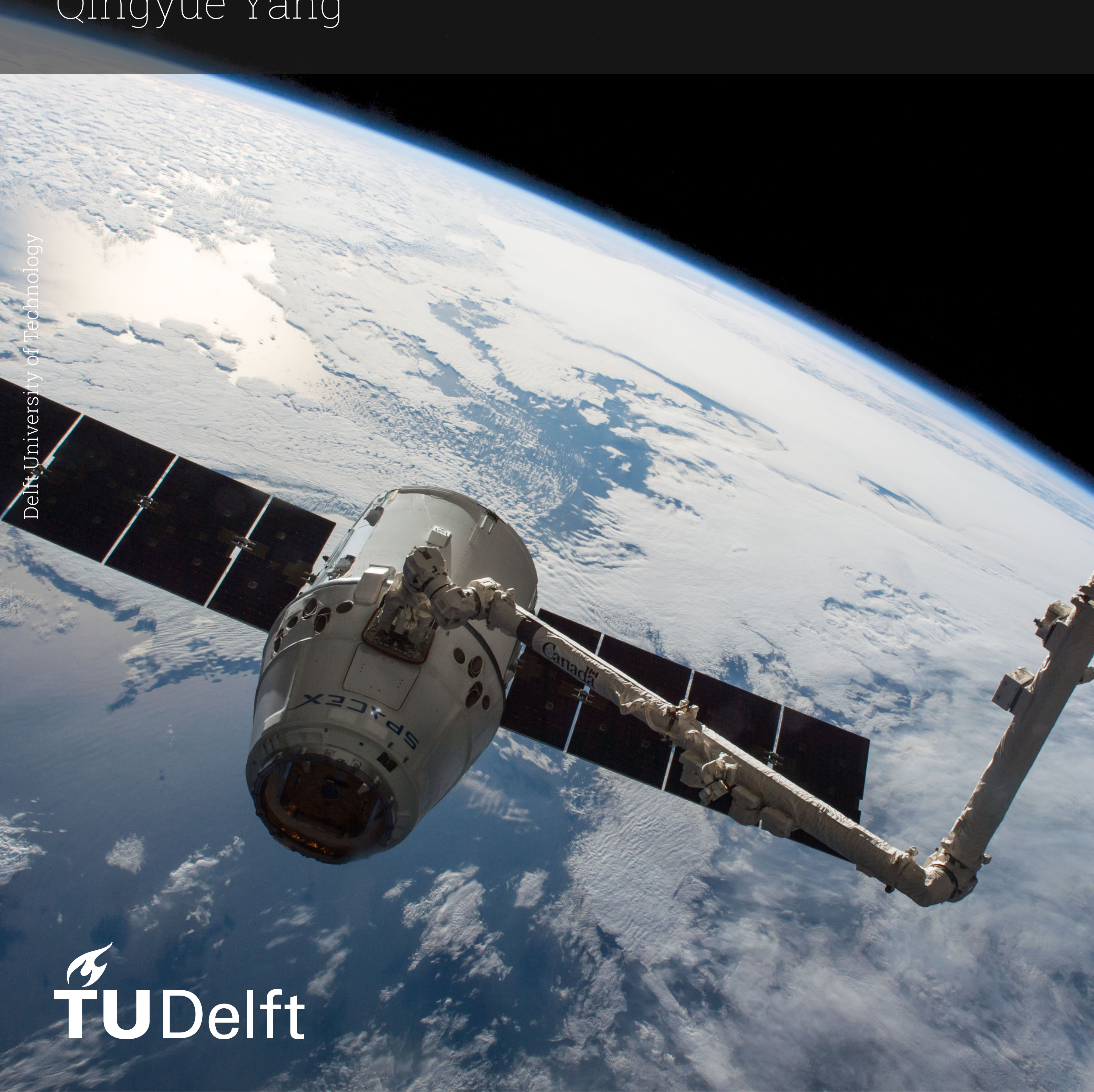
TU Delft Master Thesis

Privacy Protection via Imperceptible Face Masking: A Dynamic Approach based on HyperNet.

Faculty: EEMCS

Qingyue Yang



Delft University of Technology

TU Delft Master Thesis

Privacy Protection via Imperceptible Face Masking: A Dynamic Approach based on HyperNet.

by

Qingyue Yang

Student Name	Student Number
Qingyue Yang	5722675

Advisor:	Koen Langendoen
Daily Supervisor:	Qun Song
Project Duration:	11, 2023 - 6, 2024
Faculty:	Faculty of EEMCS, Delft

Abstract

The proliferation of video recording devices and facial recognition technology has led to significant privacy concerns, as surveillance systems can capture and identify individuals without their consent. Traditional facial obfuscation systems, which introduce pixel-level perturbations to images, aim to protect privacy by preventing unauthorized facial recognition. However, these systems are vulnerable to inversion attacks, where attackers can reverse the perturbations to restore original images, thereby compromising privacy. This thesis addresses these vulnerabilities by proposing HyperObf, a novel approach utilizing HyperNet technology to generate unique obfuscation networks for each user. HyperObf ensures that each user's images are distinctly protected, making it challenging for attackers to reverse-engineer the obfuscations. Our experiments demonstrate that inversion attacks can significantly degrade the protection offered by static obfuscation systems, with restored images achieving face recognition accuracy close to that of original images. In contrast, HyperObf effectively mitigates these attacks, reducing the attack success rate to 30% compared to 60% for existing methods. Additionally, HyperObf can generate 100 personalized MaskNets in 0.2 seconds using high-performance computing resources. These findings highlight the potential of HyperObf to enhance privacy protection against unauthorized facial recognition and inversion attacks in the digital age.

Contents

1 Introduction

The widespread availability of video recording devices and cameras has resulted in their extensive utilization for security and law enforcement purposes. For instance, the United States has over 50 million cameras installed across the nation for monitoring activities. The United States is not alone in this trend; other nations such as Germany possess 5.2 million CCTV cameras. This marks Germany as the country with the highest number of surveillance cameras in European countries [1]. While these technological advancements undoubtedly assist government and private organizations in tasks such as law enforcement and bioterrorism surveillance, they also pose a significant threat to the privacy of innocent individuals inevitably captured by these surveillance systems. Moreover, even small businesses or individuals can easily deploy cameras, further exacerbating privacy concerns among the public.

The proliferation of facial recognition technology compounds these privacy concerns. Studies indicate that facial recognition systems scan millions of individuals across the UK every day, often without their knowledge [2]. What is more concerning is the accessibility of online face recognition services like Amazon Rekognition Face [3] or Face++ [4], which empower individuals with moderate computing resources and a basic understanding of deep learning to create highly accurate facial recognition models. By simply obtaining a few photos of a person, anyone can potentially build a model capable of identifying them online. These two factors combine to create a significant privacy hazard. Whether we are walking down the street or entering a private building, there is a possibility of being captured by surveillance cameras. Individuals with malicious intent, such as stalkers or private investigators, can exploit facial recognition technology to track and monitor us without our consent. Once a person is captured on camera, they can be easily identified by these facial recognition models, enabling privacy invaders to locate and monitor them. With the ability to automate the creation of these models, the scale of potential targets increases exponentially, posing a great threat to personal privacy in the digital age.

Many efforts have been made to protect individuals from unauthorized facial recognition models. Most of the existing approaches focus on introducing perturbations to facial images. For example, in [5] and [6], such obfuscation systems utilize neural networks to generate pixel-level changes on the original images. These alterations, imperceptible to the human eye, modify the features learned by any facial recognition model trained on such images. Consequently, when presented with an unaltered image of the individual, like a photo taken by a smartphone or a streetlight camera, the facial recognition model would misclassify them. Therefore, individuals seeking to safeguard their privacy on social media or other public platforms can use such face obfuscation systems to alter their photos before uploading them. By incorporating these imperceptible alterations, the images become resistant to unauthorized facial recognition, as shown in Figure 1.1. However, there is a notable weakness in these obfuscation systems. Such perturbations can be removed through an inversion attack, which involves reconstructing the input data (such as an image, text, or other data type) from the output data or the internal representations of a neural network. This type of attack can pose significant privacy risks, especially when the input data is sensitive. All the existing obfuscation systems utilize static networks to generate perturbations, which means all the users share the same network. People who want to invade privacy can pretend to be users seeking protection and then access the obfuscation system. They might easily be able to perform an inversion attack to undo the changes made to other users' images.

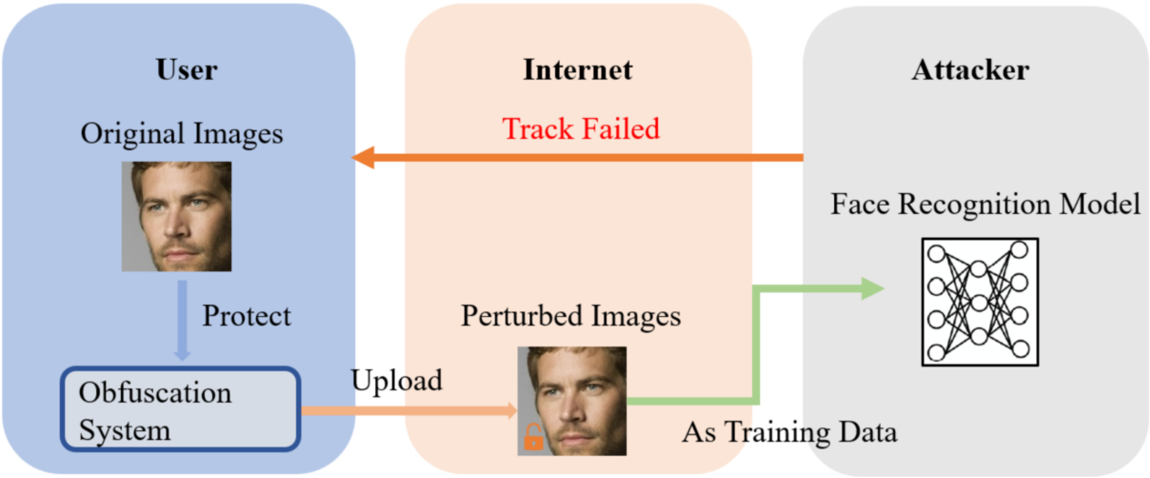


Figure 1.1: System overview of face obfuscation system: Users utilize the obfuscation system to perturb the images they upload to the Internet, to avoid being tracked by attackers using their original images.

Recognizing this weakness in static obfuscation networks, so our main research goal is to answer these two questions:

- Can attackers effectively restore perturbed images through inversion attacks and use the restored images to train a valid face recognition model?
- Is there an efficient way to generate perturbations that are hard to remove through inversion attacks?

Our experiments demonstrated that the inversion attack effectively removes perturbations, resulting in classification accuracy for the face recognition model trained on restored images comparable to that achieved with the original images. Therefore, a stronger solution is necessary. This thesis proposes creating unique obfuscation networks for each user to address this issue. Even if attackers obtain one or some of the MaskNets, they will struggle to execute the inversion attack due to the varied obfuscation systems among users. Our system, called HyperObf uses HyperNet [7] technology to generate personalized networks dynamically for each user. HyperNets employ a network to generate the weights for target networks. Consequently, all users share the same network structure of the obfuscation system, but with different weights. This unique configuration ensures that each user's facial images are distinctly protected, making it challenging to reverse the obfuscation. Through HyperObf, we offer users strong defenses against unauthorized facial recognition and privacy violations.

Our work produces several key findings:

- We demonstrate the existing static image obfuscation system is vulnerable to reverse engineering through the training of inversion networks. As our experiment results show, this vulnerability results in the near-perfect elimination of protection, with face recognition classification accuracy remaining high. The accuracy for original images is approximately 83% and the accuracy for restored images is around 81%.

- We introduce HyperNet into the obfuscation system and show its effectiveness in providing tailored protection for a wide range of users. The HyperNet enables the generation of 100 MaskNets in 0.2 seconds on SURF HPC Cloud with CUDA A10-1 GPU[1].
- Our experimental results demonstrate HyperObf's effectiveness in preventing inversion attacks. Restored images created with HyperNet have a significantly lower attack success rate against inversion attacks. HyperObf's average attack success rate is only 30%, while baselines like ensemble training and dropout have rates close to 60%.

[1] https://www.surf.nl/en

2
Background and Problem Statement

Visual obfuscation is widely used to preserve privacy. Many obfuscation methods have been proposed, such as blurring, painting [8], and adversarial perturbations [5]. These systems are aimed at protecting the public's privacy against potential unauthorized surveillance.

Research in visual obfuscation can be divided into two primary categories: evasion obfuscation and poisoning obfuscation. In the former, studies typically focus on directly manipulating images that malicious actors could potentially access. For instance, this might involve altering raw facial images captured by cameras. Techniques employed here might include inpainting to obscure faces or adding perturbations to the images [8]. However, such approaches often result in noticeable alterations to the original images or are tailored only to specific target users [9]. Conversely, poisoning obfuscation aims to safeguard the data utilized for training facial recognition models [5]. This involves users employing obfuscation systems to protect their original labeled images before sharing them. When attackers attempt to use these protected images to train a facial recognition model and subsequently test it with unaltered images, the model misclassifies the user due to the altered features in the perturbed images. Given its broader applications and lesser visual impact, training data obfuscation appears more suitable for our intended application scenario.

In this section, the problem definitions of image obfuscation systems are first discussed, followed by the introductions of preliminaries and system design.

2.1. Problem Definition

2.1.1. Problem Description

The misuse of facial recognition models has raised widespread concern among the public. Attackers exploit social media platforms such as LinkedIn and Instagram to gather facial images and identity information. The attackers can be the government, illegal detectives, and malicious extortionists. With sufficient data, they can train models or leverage commercial face recognition APIs like Microsoft Face Azure or Face++ easily. This poses a significant threat to privacy as attackers can potentially invade street cameras, security cameras, or even deploy private cameras in public spaces. Consequently, individuals risk being tracked wherever they go, as cameras capture their facial images.

2.1.2. System Objectives

The potential solution to mitigate the threat described in 2.1.1 is to protect all images against facial recognition models before sharing them online. In this context, users are playing the role of attackers, implementing data poisoning attacks, while the potential targets are facial recognition models. By introducing subtle perturbations or modifications to facial images, users can make the model misclassify with clean images, thereby enhancing privacy protection. This proactive approach can help individuals protect their privacy and mitigate the risks associated with unauthorized surveillance and tracking.

The design objectives of the visual obfuscation system are:

- When testing with clean and right-labeled images, the facial recognition model trained on the perturbed images should have **low classification accuracy**.
- The perturbed images should **maintain their utility**, namely, the perturbed picture should look the same as the original pictures.

2.1.3. Assumptions

User. In this scenario, users are individuals with access to limited to moderate computing resources, such as smartphones and personal laptops. They engage with the obfuscation algorithm that operates locally on their devices. Importantly, the obfuscation algorithm remains consistent across all users, ensuring uniform privacy protection measures.

In many algorithms, the obfuscation system incorporates a feature extractor, which plays a key role in calculating the feature space loss and optimizing it throughout the protection process. Although the feature extractor utilized by users may differ from that of the attacker's model, the effectiveness of protection is independent of their consistency. However, there is potential for simplification if the feature extractors are identical.

Users should protect all images they share online, ensuring that attackers have access only to perturbed label images. It is assumed that users are proactive in applying protection to all shared images, thus providing consistent protection against recognition by attackers. For future work, a stronger attacker will be considered who has access to some clean images for training.

Attacker. Attackers in this context are regarded as third-party entities such as government agencies, corporations, or individuals lacking direct access to personal images. Nonetheless, they possess the capability to gather users' images from social media platforms along with associated identity information. With this data, attackers endeavor to develop face recognition models capable of simultaneously tracking multiple users.

Attackers may possess substantial computational resources or rely on commercial face recognition APIs to train their facial recognition models. Despite their resources, attackers are restricted to accessing only perturbed labeled images and are unable to differentiate between perturbed and clean images. This limitation underscores the effectiveness of the protecting mechanism in obfuscating identities and maintaining user privacy.

2.2. Preliminaries and System Design

2.2.1. Preliminaries

Notation. The notations in our discussions are listed as follows:

- x: the unprotected image of the target user
- Φ_{face}: the face recognition model trained by attackers
- y : target user label
- y_{pre} : the predicted target user
- $\delta(x)$: the computed image-specific perturbations of the target user
- $x \oplus \delta(x)$: the perturbed image of the target user

- Φ: the feature extractor used in obfuscation system
- $\Phi(x)$: feature vector extracted from an input x

In the face obfuscation system, the object is to make the face recognition model invalid, which means:

$$\begin{gathered}\max_{\delta} \mathsf{Dist}(y_{\mathsf{pre}}, y) \\ while\, y_{\mathsf{pre}} = \Phi_{\mathsf{face}}(x_p)\end{gathered} \tag{2.1}$$

Explanation. A face recognition model is frequently trained using DNN models. It typically consists of the classification layer and the feature extractor. The features of face photos are identified and extracted using feature extractors, which then map the features into a vector. A fully connected layer uses the vector as input to perform classification. If perturbations are added to maximize the difference between the feature space of protected images and the original images, the model's view of the feature space will be influenced therefore causing misclassification.

Therefore, intuitively the goal is to maximize the feature difference while keeping the images visually as same as possible by adding appropriate perturbations.

For a given image x_o the user wants to share online, the system modifies the original images to maximize the feature space loss, which is calculated as:

$$max_{\delta}Dist(\Phi(x), \Phi(x \oplus \delta(x))) \tag{2.2}$$

where $Dist(.)$ is the distance between the two feature vectors.

The limitation of perturbations value added to the original images is represented as

$$|\delta(x)| < \rho \tag{2.3}$$

$|\delta(x)|$ measures the perturbations value after protection. ρ is the budget value of the perturbations per pixel, limiting the maximum perturbations added to the images.

In addition, the maximum perturbations added to each pixel are restricted to ρ_{pixel}, therefore the budget for each pixel is represented as:

$$|\delta_{pixel}| < \rho_{pixel} \tag{2.4}$$

Given the input image x_u of user U, the feature extractor Φ, the budget value of perturbations ρ, and the clipping value of perturbations each pixel ρ_{pixel}. The obfuscation system randomly generates some perturbations and calculates the feature space loss using feature extractor Φ, following the Equation (2.2), subject to Equation (2.3).

The $|\delta_{pixel}|$ is defined using DSSIM(Structural Dis-Similarity Index). It is a metric for measuring how different two images are from one another that is based on the Structural Similarity Index Measure (SSIM). DSSIM determines the dissimilarity between two images and provides a measure of how different they are from one another, whereas SSIM measures how similar two images are. By introducing the perturbations budget, ensuring perturbations are accepted and the perturbed images are visually similar to the original images.

The penalty method to formulate the optimization equation is represented as follows:

$$\boldsymbol{min\, \lambda \cdot max(|\delta| - \rho, 0) - Dist(\Phi(x), \Phi(x \oplus \delta(x)))} \tag{2.5}$$

Where the $Dist(\Phi(x), \Phi(x \oplus \delta(x)))$ denotes to the feature space loss and $max(|\delta| - \rho, 0)$ denotes to the dissimilarity loss. λ controls the visual impact of the perturbations. To limit the visual difference to be imperceptible, we set $\lambda \to \infty$, therefore the Equation (2.3) is achieved. Additionally, to guarantee that Equation (2.4) is satisfied, the pixel change is clipped within the range $([-\rho_{pixel}, \rho_{pixel}])$.

2.2.2. System Design

As illustrated in figure 2.1, our system features a Privacy Service Provider (PSP) that allows mobile users to access MaskNet. Users seeking privacy protection send a request to the PSP. After verification, the PSP distributes MaskNet to the users, who then store it on their local devices.

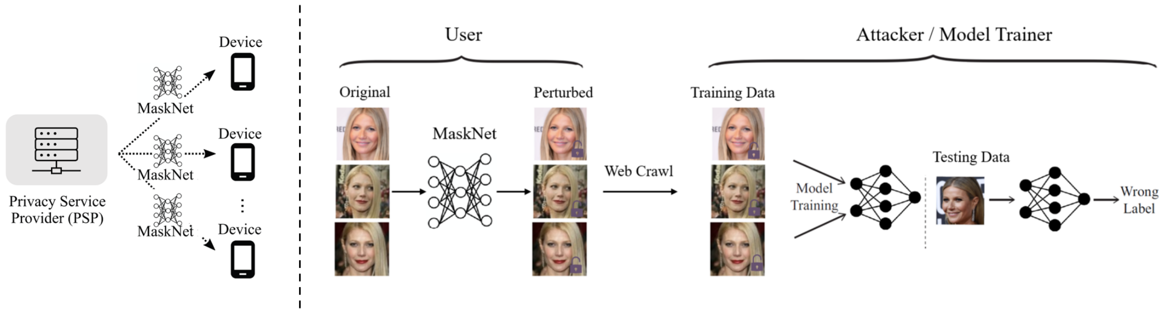


Figure 2.1: Overview of obfuscation system: The PSP distributes MaskNet for users. By introducing the obfuscation system, the attacker/model trainer can not effectively track the target users.

MaskNet refers to a specific neural network designed to generate perturbations, or alterations, to original images. Its purpose is to provide a protective layer for images before they are uploaded to the internet. Users employ the MaskNet as a precautionary measure to safeguard their images from potential threats.

By applying MaskNet to their images before sharing them online, users effectively introduce subtle changes that can disrupt the ability of unauthorized individuals, or attackers, to effectively utilize these images. If attackers attempt to gather these protected images from online sources and utilize them to train a face recognition model, the resulting model would be rendered invalid or unreliable. This is because the perturbations introduced by MaskNet alter the visual features of the images in such a way that the model's training process is disrupted. As a result, the face recognition model would likely produce incorrect identifications when tested with real-world images.

3

Inversion Attack and Threat Model

In Chapter 2, we introduced the face obfuscation system, designed as a proactive measure to safeguard public privacy. In this Chapter, we conduct a comprehensive investigation of such an obfuscation system. Through this investigation, we aim to derive specific research goals addressing the identified vulnerabilities. Our objective is to contribute to the ongoing efforts in privacy protection to ensure its effectiveness against real-world potential privacy attacks by training the face recognition model.

3.1. Inversion Attack

In this section, we delve into the inversion attack concerning image perturbation. In this context, an inversion attack involves the training of a model capable of restoring original images from their perturbed counterparts. This attack poses a significant threat as it enables attackers to undermine the protective measures implemented by obfuscation systems such as those used in Fawkes [5]. By successfully training an inversion model, adversaries gain the ability to reverse the perturbation process, thereby using the restored for face recognition training and tracking.

Notations The notations in our discussion are listed as follows:

- x: the original image from a public dataset
- x_p: the perturbed version of the image from the public dataset
- Φ_{in} : the inversion attack network
- Φ_p : the obfuscation system

In the inversion attack, the attacker's goal is to train the inversion network Φ_{in} using a public dataset. In the training phase, the input images are the x_p, generated from the obfuscation system Φ_p. The loss function is to minimize the SSIM of the output images and the original images. Once the training is done, the inversion network can be used to restore perturbed images of other users.

The training workflow can be represented as Figure 3.1:

3.2. Threat Model

The problem of system 2 lies in the fact that the obfuscation system is shared with all users, which makes it vulnerable to exploitation by attackers. These attackers could potentially reverse engineer the perturbation process by having access to the same obfuscation system.

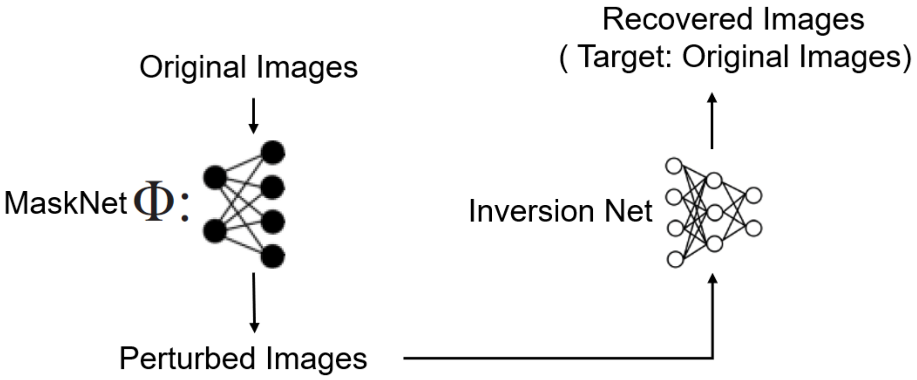


Figure 3.1: The training Workflow of inversion attack: attackers utilize the MaskNet to generate perturbed images, and then use the perturbed images and original images to train an inversion network.

This means they could train neural networks to undo the perturbations and restore the original images, completely defeating the protection.

The system's threat model is stated as follows:

3.2.1. Attacker Capability

The attackers possess several capabilities that enable them to carry out their malicious activities effectively. Firstly, they have the ability to crawl label-perturbed images, indicating their proficiency in accessing and retrieving images that have undergone obfuscation processes. Furthermore, the attackers have access to high computation resources, which are essential for training an inversion network—a sophisticated tool used to reverse the effects of image perturbations. Additionally, they possess knowledge about deep learning methodologies, particularly in training inversion networks, indicating a certain level of expertise in the field. Moreover, the attackers have access to public open-source face datasets commonly utilized for training inversion networks, further enhancing their capabilities in carrying out their objectives.

3.2.2. Attacker Knowledge

The attackers possess comprehensive knowledge about the obfuscation system employed to generate perturbated images. While they may lack insight into the internal workings or specific algorithms of the obfuscation system, they possess the necessary skills to utilize it effectively for generating both original and perturbed versions of images. Additionally, the attackers have access to open-source face datasets, which they can leverage in conjunction with the perturbed images obtained from other users. This knowledge empowers them to manipulate and analyze image data effectively, facilitating their malicious activities.

3.2.3. Attacker Goal

The primary objective of the attackers is to restore the perturbed images through the training of an inversion network. By achieving this goal, they aim to obtain high-quality, undistorted images that can be subsequently utilized for training a face recognition model. This reconstructed dataset enables the attackers to develop a robust face recognition model capable of accurately identifying individuals. Ultimately, the attackers intend to exploit this model for tracking and surveillance purposes, highlighting the malicious nature of their objectives.

3.3. Research Goals

Given the vulnerability to potential inversion attacks on perturbed images, the research aims to propose a new method to strengthen privacy protection mechanisms. This involves creating unique perturbation patterns or feature extractors for individual users. By customizing these patterns, the goal is to make it harder for attackers to inverse the perturbed images. To guide this research, two main questions are posed:

- Can attackers effectively restore perturbed images to their original versions using inversion attacks? This question explores the feasibility of such attacks in bypassing existing privacy protections.
- Is there an efficient way to create personalized neural networks or perturbation methods for each user to counter potential inversion attacks? This question seeks to investigate methods for customizing perturbation strategies based on individual user characteristics.

By addressing these questions, the research aims to develop stronger privacy protection mechanisms capable of resisting inversion attacks, thereby enhancing individuals' privacy and security in the digital age.

4

HyperObf

To address the first question mentioned in section 3.3, we conducted experiments as described in section 5.1. The findings indicate that current obfuscation systems are susceptible to inversion attacks. To address this issue, we propose utilizing a HyperNet to dynamically generate MaskNets. This chapter will first introduce the concept of HyperNet and outline our HyperNet training framework. Finally, an overview of the HyperObf system will be presented.

4.1. HyperNet Design

4.1.1. HyperNet preliminaries

HyperNetworks (also HyperNet) [7] is an approach of using small networks to generate weights for another larger target network. This paper will use the HyperNet to generate the MaskNets for different users. Suppose for an n-layer convolution network, the HyperNet should contain n generators. Each weight generator outputs the parameters of a layer of the target neural network, as shown in Figure 4.1. To guarantee the connectivity between different layers, a procedure called a mixer is introduced before the generators. The mixer takes a random vector z sampled from a multi-dimensional Gaussian distribution as the input. Then it transforms it into n vectors denoted by $\{v_i | i = 1, 2, ..., n\}$, and then be fed into n different generators. Once the HyperNet training is completed, the inference process of generating the parameters of one ensemble of target networks can be finished in a short time. Each time fed with a different random input vector z, the HyperNet generates different target networks.

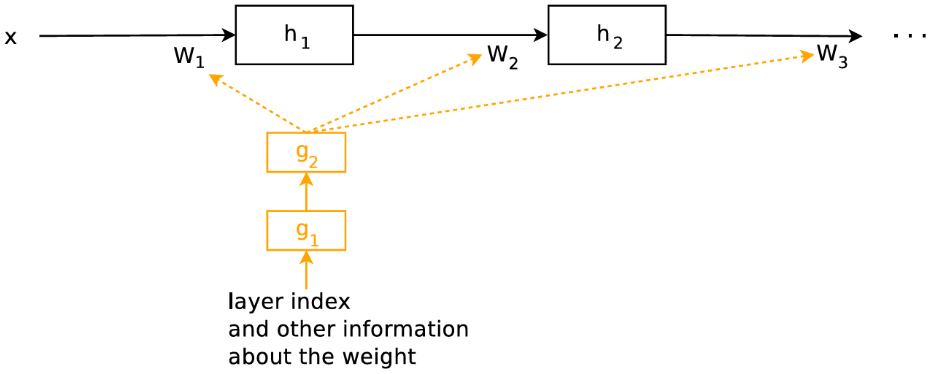


Figure 4.1: HyperNet generates the weights for the target network.

4.1.2. HyperNet Training Framework

Training Pipeline. In this thesis, we introduce HyperNet for the generation of MaskNets. During the training phase, we adopt a distinctive strategy whereby the HyperNet is exclusively tasked with generating the MaskNet, while a classification layer is shared and trained together within the whole batch. This methodological choice allows us to streamline the training process by isolating the optimization of the MaskNets from that of the classification layer therefore avoiding training overhead.

The rationale behind this strategy lies in its ability to optimize the training process, enabling us to focus HyperNet's efforts solely on generating MaskNets that are both diverse and effective. This separation of concerns not only enhances the adaptability and robustness of the resulting MaskNets but also contributes to streamlining the overall training pipeline.

Loss Function Formulation for HyperNet Training. To train the HyperNet effectively in generating diverse and efficient MaskNets, we define a comprehensive loss function comprising two key components: the effectiveness loss (J_1) and the diversity loss (J_2).

- **Effectiveness Loss (J_1):** The primary objective of the HyperNet training is to ensure the efficacy of the generated MaskNets in their classification tasks. This is encapsulated by the effectiveness loss, defined as:

  $$J_1 = L(f(x; G(E(z_H; \phi_E); \phi_G; \phi_F)), y), \tag{4.1}$$

  Here, $E(z_H; \phi_E)$ represents the latent code generated by the mixer, $G(E(z_H; \phi_E))$ denotes the weights generated by the HyperNet, ϕ_F signifies the fully-connected layer weights, and $f(.)$ indicates the classification result of the MaskNet for input x. The effectiveness loss J_1 is calculated using the cross-entropy loss function L between the predicted output and the ground truth labels y.

- **Diversity Loss (J_2):** In addition to effectiveness, promoting diversity among the generated Mask-Nets is crucial for enhancing adaptability and robustness. This is achieved through the diversity loss, formulated as:

  $$J_2 = exp(-Var(G(E(z_H; \phi_E); \phi_G))), \tag{4.2}$$

  Here, Var$(G(E(z_H; \phi_E); \phi_G))$ denotes the average variance of the parameters of convolution layers within one batch. The exponential function applied to the negative variance encourages a wider distribution of parameters, thus fostering diversity among the MaskNets.

- **Total Loss Function (J_{total}):** Combining the effectiveness and diversity objectives, we define the total loss function as a weighted sum:

  $$J_{total} = \lambda \cdot J_1 + J_2 \tag{4.3}$$

  Where λ serves as a hyperparameter controlling the trade-off between effectiveness and diversity during training.

By optimizing the total loss function J_{total}, we aim to train the HyperNet to generate MaskNets that not only excel in classification tasks but also exhibit diversity in their parameterizations, thereby enhancing their adaptability and robustness in various scenarios.

4.1.3. HyperNet training detail

The training of HyperNet and hyper-parameter setting are motivated by two requirements. First, the generated MaskNet should be effective for protection, which is achieved by the loss function of effectiveness loss. Second, the generated MaskNets should be as diverse as possible, this is achieved by setting the diversity loss function, which calculates the MSE of network parameters within one batch. By carrying out experiments to balance these two requirements, the total loss function is designed as:

$$J_{total} = \lambda_{diversity} \cdot max(J_1 - \alpha_{threshold}, 0) + J_2 \tag{4.4}$$

Where the $\lambda_{diversity}$ and the $\alpha_{threshold}$ are the hyper-parameters. J_1 is the diversity loss and J_2 is the effectiveness loss. Both of these two hyper-parameters control the effectiveness and diversity of the generated MaskNets.

4.1.4. KL Divergence

Kullback-Leibler (KL) divergence is a fundamental concept in information theory that measures the difference between two probability distributions. Specifically, it quantifies how one probability distribution Q diverges from a second, reference distribution P. The formula for KL divergence is:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{4.5}$$

For continuous distributions, it's expressed as:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \tag{4.6}$$

Intuitively, KL divergence measures the inefficiency of assuming that the distribution Q is the true distribution when P is actually the true distribution. It is always non-negative and zero if P and Q are identical.

In machine learning, KL divergence is used to evaluate model performance, especially in probabilistic models and variational inference. Lower KL divergence indicates a model that closely approximates the true data distribution. It is also pivotal in training generative models like Variational Autoencoders (VAEs), where minimizing KL divergence helps align the generated data distribution with the true data distribution. Despite being asymmetrical, its role in measuring informational discrepancy makes it a crucial tool in data science and statistics. In our system evaluation, the KL divergence is used to evaluate the diversities of the generated models.

4.2. Approach Overview

Therefore, in order to dynamically generate MaskNets for different users, as illustrated in Figure 4.2, we examine a setup introducing HyperNet in PSP. The PSP is assumed to have ample computational resources and assumes the responsibility of training the HyperNet. Further details regarding the training methodology will be discussed in Section 4.1.2. Once the training phase is over, the HyperNet is ready to be deployed and start supporting mobile devices. When a smartphone indicates that it wants to remain private and use the service, the PSP starts the HyperNet inference process using a random seed. This creates a smaller neural network called the MaskNet, which is then distributed to the smartphones. The parameters of the MaskNets used by mobile devices differ, despite their share the same architecture. When users want to protect their images, they can provide MaskNet with the images that need to be protected, which will result in the creation of perturbed versions. For each mobile, the MaskNet should be unique and confidential.

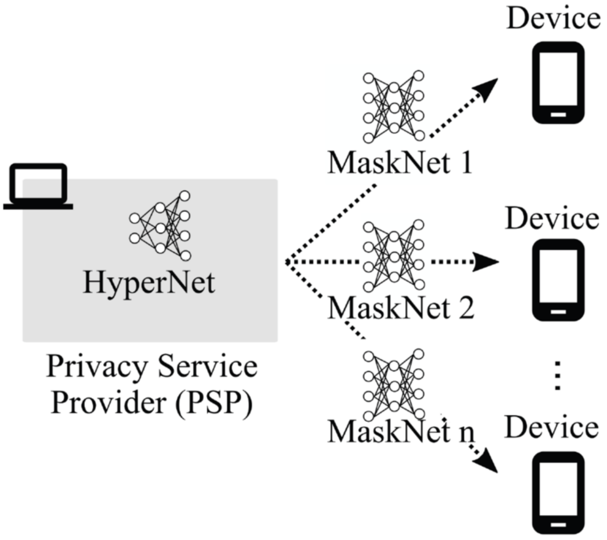


Figure 4.2: HyperNet is introduced in PSP to dynamically generate MaskNets.

The overall system design of HyperObf involves the use of a MaskNet for image perturbation and a Privacy Service Provider to ensure user privacy. Here's an elaboration of the system components and their roles:

- **MaskNet:** The MaskNet is a neural network designed for image perturbation. It takes an input image and applies perturbations to it, creating a modified version of the original image. This network is utilized by both users and attackers for generating perturbed images.
- **Privacy Service Provider (PSP):** The Privacy Service Provider is responsible for ensuring the privacy of user data. It generates unique MaskNets for each user. These MaskNets are essentially masks or filters that can be applied to images to perturb them in a unique way. The MaskNets are stored locally on the user's device, ensuring that the perturbation process can be performed quickly and without the need to transmit sensitive data over the network.
- **User:** Users interact with the system by utilizing the MaskNet provided by the Privacy Service Provider to perturb their images. This process is performed locally on the user's device, allowing them to maintain control over their data and ensuring privacy.
- **Attacker:** Attackers are entities that may attempt to surveil users without authorization, compromising user privacy. In this system, attackers are also allocated a unique MaskNet by the Privacy Service Provider. However, even with access to a MaskNet, attackers cannot effectively reconstruct perturbed images to their original versions. This is because MaskNets are unique to each user, meaning the perturbations applied to images are specific to that user's MaskNet. As a result, any attempt by an attacker to reconstruct perturbed images would be unsuccessful, as the MaskNet used for perturbation is different from those used by other users.

In summary, the Privacy Service Provider plays a crucial role in generating unique MaskNets for users, allowing them to perturb their images locally while maintaining privacy. Even if attackers obtain a MaskNet, they cannot effectively reverse the perturbations to reconstruct original images, ensuring user data remains protected.

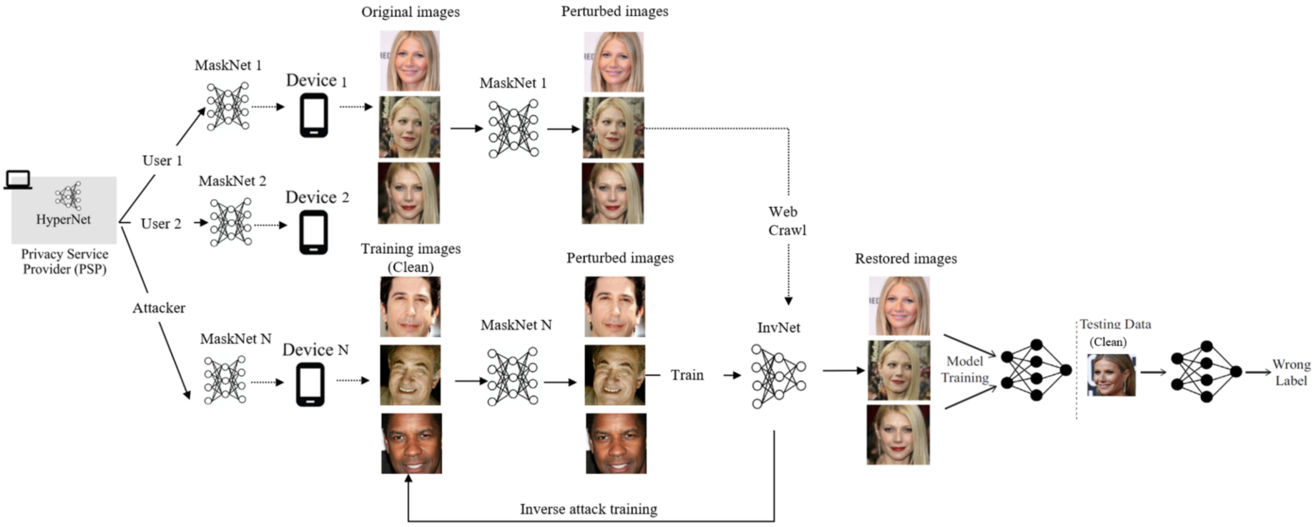


Figure 4.3: HyperObf system overview: HyperNet generates the weights for the MaskNets. Users use their unique MaskNet for protection while the attackers use their MaskNets to train an inversion attack network. However, the attacker can not use their trained inversion attack network to successfully restore the perturbed images and the face recognition model trained on the restored images should be invalid.

5 Experiments

In this section, we begin by presenting the results of the inversion attack experiments, which highlight the necessity of a dynamic approach. We then evaluate the effectiveness of HyperObf against inversion attacks. The setup of the experiments is explained first, followed by a comparison between HyperObf and other potential dynamic approaches. For clarity, within this chapter, the term “static system” refers to the existing approach where all users share the same static obfuscation system, whereas the “dynamic system” refers to our method in which MaskNets are dynamically generated.

5.1. Experiments on Static System

5.1.1. Experiment Setup

To assess the efficacy of the inversion attack, a series of experiments were conducted, focusing on 10 individual users. Each experiment was dedicated to a particular user, where their dataset comprised 25 images designated for face recognition training. These images underwent perturbation and subsequent restoration. Other clean testing images are used for evaluation.

Throughout the experiments, the training images were augmented with another set of 50 class training images. Each class in this supplementary set also consisted of 25 images for training purposes. This augmented dataset was then utilized to train the face recognition model.

The face recognition model employed in these experiments was adapted and trained using a pre-existing Resnet-18 network, as it is renowned for its ability to achieve classification accuracies surpassing 95%. Following this adaptation, the model’s accuracy in identifying faces belonging to the target user class was compared under two scenarios: one where the images were subjected to perturbation and another where they were restored using an inversion network.

5.1.2. Implementation Details

The privacy threat arises due to the potential for attackers to employ MaskNet in implementing an inversion attack. This technique, outlined in the study by [10], leverages an inversion network (InvNet) to recover inference data under the constraints of a black-box setting. The training methodology for this inversion attack is detailed as follows:

- **Data Collection:** The attacker begins by assembling a dataset consisting of human face images. These images are then fed into the MaskNet, yielding their corresponding perturbed versions. Subsequently, the outputs from MaskNet are collected to form the input training dataset for the inversion attack.

- **Design of Inversion Network (InvNet):** An appropriate network structure is chosen to serve as the inversion network. Typically, the architecture of the InvNet mirrors that of the MaskNet, albeit in reverse. The objective of the InvNet is to learn the mapping from the output of the MaskNet back to its original input.
- **Training of Inversion Network:** The InvNet is trained using the collected dataset of protected images and their corresponding original inputs. During training, the InvNet learns to reverse the transformations applied by the MaskNet, effectively restoring the original images from their protected versions.

Training Dataset For the training of the Inversion Network (InvNet), the attackers are assumed to utilize a dataset of 50 distinct classes of images from a public dataset. In our experiments, we use the FaceScrub dataset and extract a subset from the FaceScrub dataset[11].

InvNet Structure and Training We proposed to novel method following an autoencoder structure for image reconstruction. The InvNet comprises two essential components: the encoder and the decoder. The encoder is responsible for compressing the input images into a lower-dimensional latent space, while the decoder aims to reconstruct the original images from this compressed representation. By leveraging InvNet, the pixel-level changes introduced by the protecting process can be precisely calculated and applied to the perturbed images. This process effectively counteracts the perturbations added to the original images, resulting in the restoration of the original images. Thus, InvNet serves as a powerful tool in mitigating privacy concerns and ensuring the integrity of the image data in applications where image obfuscating is employed.

5.1.3. Static System Experiment Result

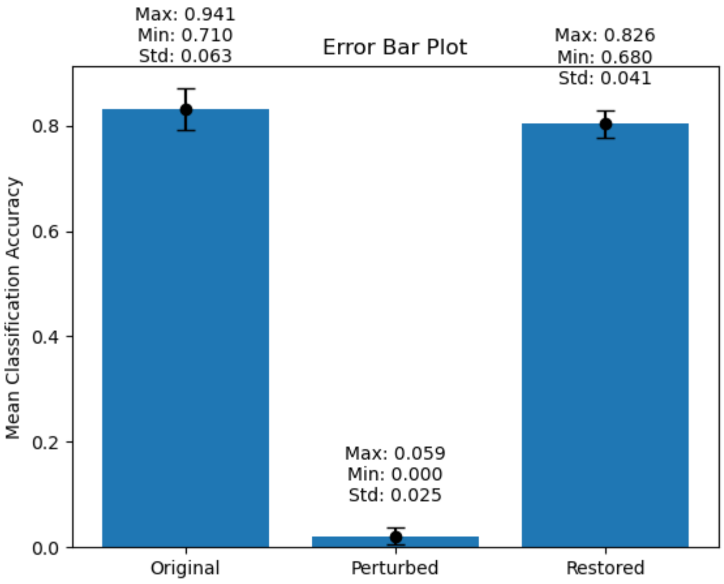


Figure 5.1: The trained face recognition model classification accuracy comparison using three types of training datasets: (1) The original images, (2) the perturbed images, and (3) the restored images after inversion attack.

The results of these experiments revealed that both the obfuscation protection and inversion attack were successful. Specifically, the original images had an average classification accuracy of 0.82 and a maximum accuracy of 0.941. When these images were perturbed, the classification accuracy dropped significantly, with an average of 0.02 and a maximum of 0.059, demonstrating the effectiveness of the obfuscation. However, the inversion attack managed to restore the images to near-original quality, with the restored images achieving an average classification accuracy of 0.81 and a maximum of 0.826. This outcome compromises the integrity of the obfuscation measures and underscores the importance of robust defenses against such attacks. It also highlights the need for further research and development in the area of image security to counteract sophisticated inversion techniques.

5.2. Experiments on Dynamic System

5.2.1. Experimental Setup

The experiment setup combines Ubuntu 20.04 with a CUDA A10-1 GPU, hosted on the SURF HPC Cloud platform, providing a robust environment for GPU-accelerated computations. This setup runs on an x86_64 architecture, featuring an Intel Xeon Gold 6342 CPU clocked at 2.80GHz, with 11 CPU cores, a substantial cache hierarchy including a 44 MiB L2 cache and a 176 MiB L3 cache, and VT-x virtualization support for efficient resource allocation. Additionally, it includes an NVIDIA A10 GPU with CUDA Version 12.2, offering dedicated CUDA cores for accelerated parallel processing and 23028 MiB of memory for data-intensive tasks. The code is written With PyTorch 2.2.2.

5.2.2. Implementation Details

MaskNet Structure. In the baseline experiment, the target MaskNet consists of two convolutional layers followed by a fully connected layer. The first convolutional layer has 16 output channels, while the second has 32. Both convolutional layers use a 3x3 kernel size with a stride of 1 and padding of 1. The fully connected layer receives the output from the convolutional layers and produces a 256-dimensional output.

Face Recognition Model Structure. In our scenario, the attackers have no knowledge about MaskNet, so they are assumed to use a different face recognition model. According to [5], the face recognition model may have an impact on the performance of perturbations. The performance is worse when the face recognition model is different from the MaskNet structure. For evaluation purposes, a pre-trained ResNet-18 model serves as the face recognition model. ResNet-18 is a popular convolutional neural network architecture that consists of 18 layers, featuring residual connections to alleviate the vanishing gradient problem and enable the training of deeper networks more effectively. It has demonstrated remarkable performance in various computer vision tasks, particularly in image classification and feature extraction. For the testing, 20 epochs are run and early stopping is used when the accuracy converges.

InvNet Structure. Structurally, the InvNet comprises a decoder section that mirrors the architecture of the MaskNet in reverse. It initiates with transposed convolutional layers, strategically positioned to upsample the feature maps obtained from the output of the MaskNet. These transposed convolutional layers progressively augment the spatial dimensions of the feature maps, facilitating the reconstruction of higher-resolution images. Throughout the decoder section, Rectified Linear Unit (ReLU) activation functions are thoughtfully integrated between the convolutional layers to introduce non-linearity, thereby capturing intricate image features more effectively. Eventually, the final convolutional layer within the decoder section reduces the number of channels to align with the original input images' channel count. Subsequently, a hyperbolic tangent (Tanh) activation function is applied to ensure the output pixel values are confined within the range [-1, 1].

5.2.3. Datasets

In our experiments, we utilize three distinct datasets, each serving a specific purpose within our study framework. These datasets are derived from the Facescrub [11] database, ensuring consistency and facilitating a more comprehensive understanding of our evaluations. The FaceScrub dataset is a widely used benchmark dataset in the field of facial recognition and computer vision. Curated by researchers at the Massachusetts Institute of Technology (MIT), FaceScrub contains a diverse collection of facial images featuring a wide range of individuals from various demographics and backgrounds. The dataset comprises over 100,000 images of more than 500 individuals. Here's a detailed breakdown of each dataset:

- **HyperNet training dataset:** This dataset is tailored for training the HyperNet model and is composed of 20 classes, with each class comprising 50 images. These images are partitioned into separate sets for training and evaluation.
- **InvNet training dataset:** The InvNet training dataset comprises approximately 1000 images distributed across 20 distinct classes. This dataset is specifically curated for training the InvNet model. While smaller in size compared to the HyperNet training dataset, it is still representative of a diverse range of facial attributes and characteristics essential for effective adversarial training.
- **Face recognition testing dataset:** This dataset is designed for evaluating the performance of the face recognition model under various conditions. During testing, the training images of a specific class undergo perturbations, simulating adversarial attacks, while the testing images remain unaltered. To ensure the robustness and generalizability of our findings, we select five different users as target individuals for evaluation. This target user is trained together with the other 50 classes, with 25 images per class utilized for training. Multiple testing images are employed to comprehensively evaluate the model's performance across diverse classes, including the specific target class under investigation. We conduct experiments separately for each of the five selected users and calculate the mean value and variance across these experiments. This setup enables a thorough assessment of the model's performance across various scenarios and user profiles, enhancing the reliability and confidence in our results.

5.2.4. Baselines

To showcase the efficacy of HyperObf in enhancing the diversity of generated MaskNets and thereby bolstering resilience against inversion attacks, a comparative analysis was conducted. This aimed to study the impact of employing Hypernet against two alternative strategies: ensemble training and dropout regularization.

- **Ensemble Training:** In the ensemble training scenario, ten identical MaskNets were trained independently. Although sharing the same architecture, each MaskNet underwent training with distinct initialization. This approach aimed to use different learning paths independently, strengthening the collective robustness of the ensemble against adversarial manipulations.
- **Dropout Regularization:** Dropout regularization was introduced as an alternative method to augment diversity within the trained MaskNets. By stochastically dropping out units during training, dropout encouraged the MaskNets to adopt different representations and learned features, thereby fostering a more diverse variety of models within the ensemble.

5.2.5. Dynamic Approach Experiment Results

In our approach, HyperNet was trained and employed to generate 10 MaskNets simultaneously. Subsequently, for each of these ten MaskNets, perturbed images were generated, and corresponding inversion networks were trained accordingly. To assess the challenge of restoring perturbed images produced by different MaskNets, we perform mix-restoration

Table 5.1: Protection performance of five different users

Classification Accuracy	Original accuracy	After protection
User1	0.82	0.056
User2	0.81	0.129
User3	0.825	0.018
User4	0.804	0.133
User5	0.819	0.030

experiments.

Protection Performance in the Absence of Inversion Attacks. The table 5.1 presents the classification accuracy for five users before and after the application of a protective measure against unauthorized facial recognition. Initially, the original classification accuracies for User1 through User5 are relatively high, ranging from 0.804 to 0.825. However, after implementing the protection, there is a significant reduction in classification accuracy for all users. User1's accuracy drops from 0.82 to 0.056, while User2's accuracy decreases from 0.81 to 0.129. Similarly, User3's accuracy falls from 0.825 to 0.018, User4's from 0.804 to 0.133, and User5's from 0.819 to 0.030. This substantial decline in accuracy indicates the effectiveness of the protection measure in preventing unauthorized facial recognition, as it severely hampers the system's ability to correctly classify the protected faces. Such a drastic reduction in post-protection accuracy demonstrates the robustness of the MaskNets employed, highlighting their potential as reliable methods for safeguarding individual privacy against facial recognition technologies.

Figure 5.2 shows the original and perturbed images, demonstrating that the perturbation does not have a big visual impact on the original images, therefore maintaining the images' utility.

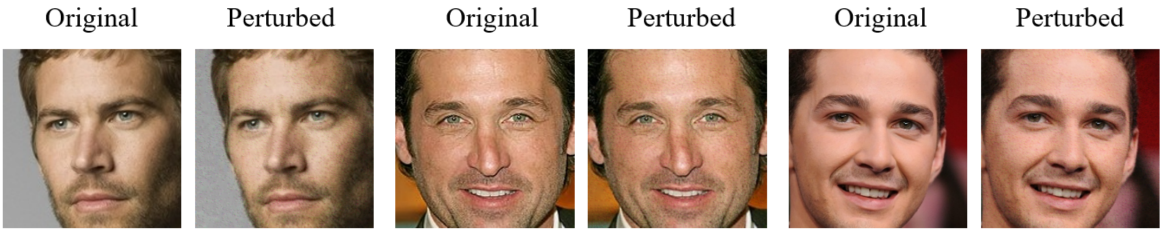


Figure 5.2: The comparison between original and perturbed images indicating the perturbations do not have a great visual impact on the images.

Protection Performance in the Presence of Inversion Attacks. Specifically, for each MaskNet, we employ 10 inversion networks that have been trained using 10 MaskNets generated within the same batch to restore the perturbed images. These restored images are then utilized for face recognition training. This comprehensive process results in a matrix that captures the restoration performance across different combinations of MaskNets and inversion networks, as depicted in Figure 5.3.

The matrix provides a detailed visualization of how well the inversion networks can restore the perturbed images, which is crucial for evaluating the effectiveness of our protection mechanism. Each cell in the matrix represents the classification accuracy of a model trained on the mix-inversion images, essentially indicating the success rate of the inversion attack. The value in each cell signifies the classification accuracy achieved when a particular MaskNet and inversion network combination is used. The higher the accuracy, the more effective the inversion attack is, meaning the perturbed images have been successfully restored to a recognizable form.

The matrix's diagonal elements are particularly noteworthy as they represent the scenarios where the same MaskNets are used for both protection and inversion attacks. However, in our setup, such a scenario is not possible, which inherently strengthens our protection mechanism. The off-diagonal elements, which show the mix-inversion accuracies, are of primary interest.

Our results indicate that the mean successful attack rate is 0.383. This is a significant reduction compared to the successful attack rate of 0.81 observed when using a static MaskNet. The decrease in the successful attack rate clearly demonstrates the efficacy of our approach. By introducing a variety of MaskNets through HyperNet, we significantly enhance the robustness of our protection mechanism against inversion attacks. This variability confounds the inversion networks, making it substantially more difficult for them to restore the perturbed images accurately.

This dramatic drop in mix-inversion accuracy underscores the strength of using HyperNet to generate diverse MaskNets. It shows that the dynamic nature of the generated MaskNets creates a formidable barrier against unauthorized facial recognition attempts. Consequently, the introduction of HyperNet in generating MaskNets not only improves the protection but also maintains the visual integrity of the images, ensuring that they remain useful for legitimate purposes while thwarting malicious inversion attacks.

In summary, the detailed analysis of the matrix reveals that HyperObf significantly diminishes the success rate of inversion attacks, confirming that the integration of HyperNet substantially enhances the protection mechanism.

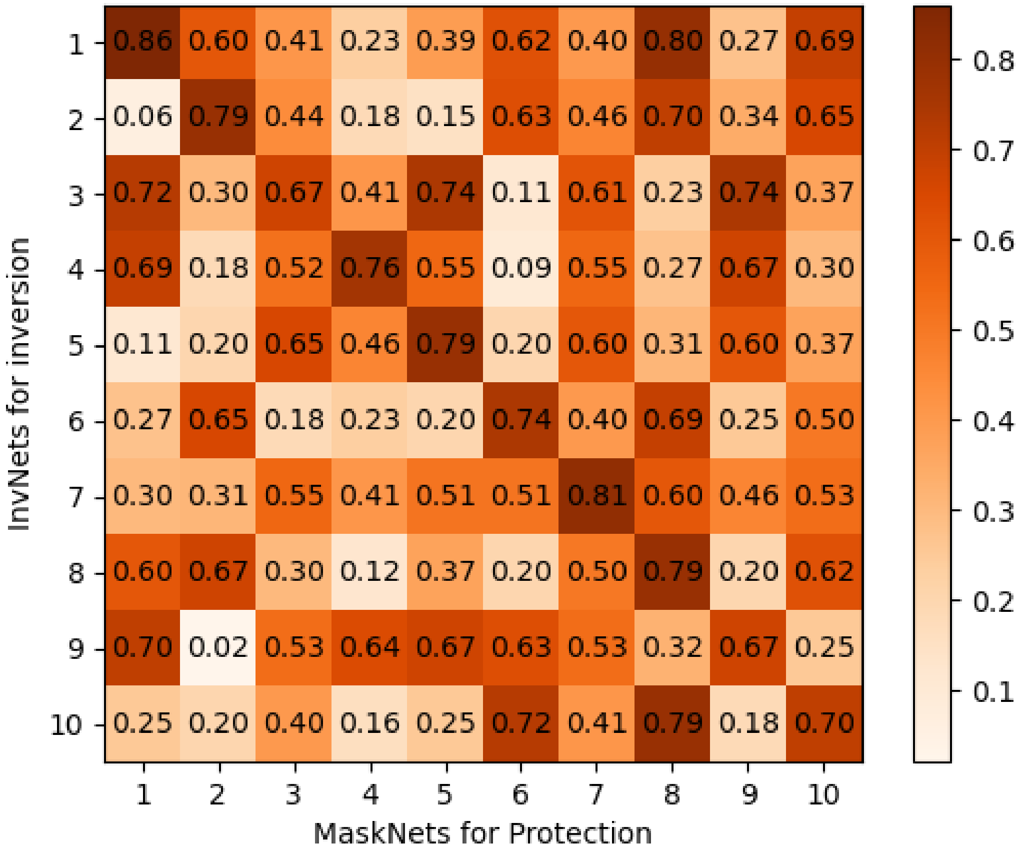
Comparison result with Baselines. Table 5.2 provides a detailed comparison of three dynamic approaches for enhancing image security: HyperNet, Ensemble Training, and MC Dropout. Each method exhibits distinct performance characteristics in terms of mean accuracy, variance, and range. Notably, in this context, lower accuracy indicates better security performance, as it reflects the method's effectiveness in obfuscating the images.

HyperNet achieves a mean accuracy of 0.383 with a variance of 0.024, operating within a range of 0.13 to 0.49. This suggests that HyperNet provides a consistent level of image obfuscation, successfully lowering the accuracy and thus enhancing security.

Ensemble Training, while demonstrating a higher mean accuracy of 0.611 and a variance of 0.116, covers a range from 0.36 to 0.79. Although this method has improved accuracy in general machine learning terms, it is less effective in this scenario as higher accuracy means less effective obfuscation. The increased variance also indicates greater fluctuations in its performance.

MC Dropout presents a mean accuracy of 0.586 and a variance of 0.018, with a range from 0.31 to 0.76. Despite offering lower accuracy compared to Ensemble Training, which implies better obfuscation, its accuracy is still higher than HyperNet, indicating less effectiveness in securing images.

Overall, HyperNet emerges as the most effective approach in this context, with the lowest mean accuracy and thus the best security performance. MC Dropout follows, offering a balance between accuracy and consistency, though it is slightly less effective than HyperNet. Ensemble Training, with the highest mean accuracy, is the least effective in terms of obfuscation. These results underscore the importance of selecting the appropriate approach based on the specific requirements for image security, with a focus on achieving lower accuracy for better protection.



	1	2	3	4	5	6	7	8	9	10
1	0.86	0.60	0.41	0.23	0.39	0.62	0.40	0.80	0.27	0.69
2	0.06	0.79	0.44	0.18	0.15	0.63	0.46	0.70	0.34	0.65
3	0.72	0.30	0.67	0.41	0.74	0.11	0.61	0.23	0.74	0.37
4	0.69	0.18	0.52	0.76	0.55	0.09	0.55	0.27	0.67	0.30
5	0.11	0.20	0.65	0.46	0.79	0.20	0.60	0.31	0.60	0.37
6	0.27	0.65	0.18	0.23	0.20	0.74	0.40	0.69	0.25	0.50
7	0.30	0.31	0.55	0.41	0.51	0.51	0.81	0.60	0.46	0.53
8	0.60	0.67	0.30	0.12	0.37	0.20	0.50	0.79	0.20	0.62
9	0.70	0.02	0.53	0.64	0.67	0.63	0.53	0.32	0.67	0.25
10	0.25	0.20	0.40	0.16	0.25	0.72	0.41	0.79	0.18	0.70

Figure 5.3: The confusion matrix represents the result of mix-inversion accuracy across different MaskNets and InvNets.

Table 5.2: Mix-inversion accuracy comparison of different dynamic approaches

Dynamic Approaches	Mean	Variance	Range
HyperNet	0.383	0.024	(0.13, 0.49)
Ensemble Training	0.611	0.116	(0.36, 0.79)
MC Dropout	0.586	0.018	(0.31, 0.76)

5.2.6. Latency Measurement

In addition to producing more diverse MaskNets, another significant advantage of the HyperObf approach is its ability to generate multiple MaskNets in a very short amount of time. Traditional methods, such as ensemble training or MC Dropout, require each MaskNet to be individually trained, which is computationally expensive and time-consuming. These conventional approaches involve extensive training processes for each MaskNet, necessitating significant computational resources and prolonged training periods. This complexity and resource demand limit the scalability and practical application of ensemble training and MC Dropout in real-world scenarios where rapid and efficient model deployment is crucial.

In contrast, HyperNets streamline this process by only requiring inference to generate multiple MaskNets. This inference process is substantially faster and more efficient than traditional training methods, leading to considerable savings in both computational resources and time. Hyperobf approach leverages a small, primary network to dynamically produce the weights for larger target networks, eliminating the need for separate, individual training sessions for each MaskNet. This efficiency is particularly beneficial in environments where quick response times and adaptability are essential.

The efficiency of HyperNets is demonstrated in empirical results. As shown in Figure 5.4, generating 10 MaskNets takes approximately 0.08 seconds, and generating 100 MaskNets takes around 0.19 seconds. This rapid generation capability highlights the remarkable speed and efficiency of HyperNets compared to traditional methods. Furthermore, the scalability of HyperNets is underscored when dealing with a much larger number of MaskNets. Figure 5.5 illustrates that generating 10^4 MaskNets simultaneously takes about 12 seconds. This scalability ensures that HyperNets can handle large-scale deployments without significant delays or resource burdens.

Overall, the HyperNet's ability to quickly generate a multitude of MaskNets not only enhances diversity and robustness in model outputs but also significantly optimizes the use of computational resources. This rapid generation capability allows for more frequent updates and adaptations to the MaskNets, ensuring that they remain effective against evolving threats. Moreover, the reduced computational burden allows for broader implementation of privacy protection measures across various platforms and devices, making HyperNets a powerful tool for the goal of privacy protection. By integrating HyperNets into privacy protection strategies, organizations can achieve a higher level of security and efficiency, ultimately enhancing the protection of sensitive information and user privacy in an increasingly digital world.

5.3. Sensitivity Parameter Discussion

As discussed in 4.1.3, the HyperNet training has a set of hyperparameters that will affect the performance of the HyperNet. We will mainly focus on the $\lambda_{diversity}$ and $\alpha_{threshold}$. and to investigate how these two hyper-parameter settings affect the effectiveness of protection mechanisms. Therefore a comprehensive series of experiments is conducted. This involves training multiple HyperNets with different hyper-parameter configurations and training epochs. After training, 10 MaskNets are generated within each batch for every HyperNet. The combined classification accuracy after protection is calculated for these MaskNets, along with their parameter variance, which measures their diversity. This systematic examination sheds light on how slight adjustments in hyper-parameter values impact the performance of protection mechanisms.

5.3.1. Training Epoch

According to Ratzlaff et al. (2020) [12], the diversity of the generated MaskNets decreases over time as training epochs progress, making early stopping crucial. However, insufficient training in the HyperNet can also reduce the protective effectiveness of the MaskNets. There-

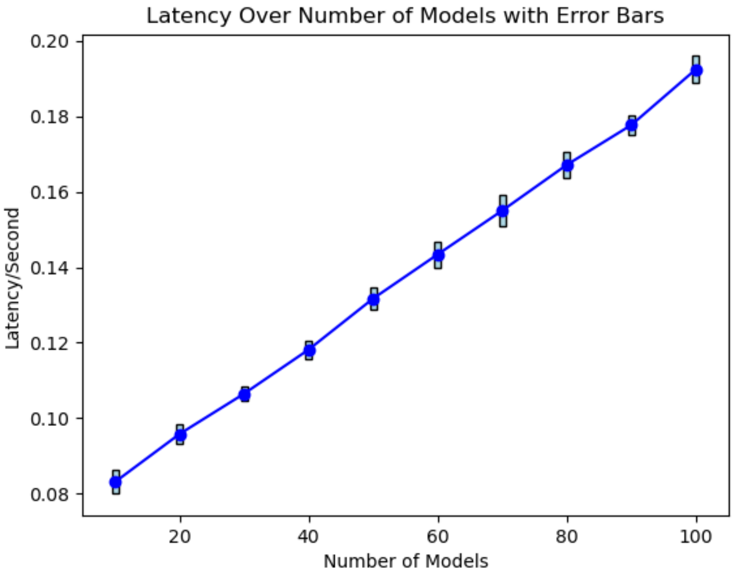


Figure 5.4: The measured latency for generating 10 to 100 MaskNets simultaneously indicates a clear advantage of the HyperNet in terms of time efficiency. Unlike the baselines, which require considerable time to train networks individually, the HyperNet significantly reduces the overall training time.

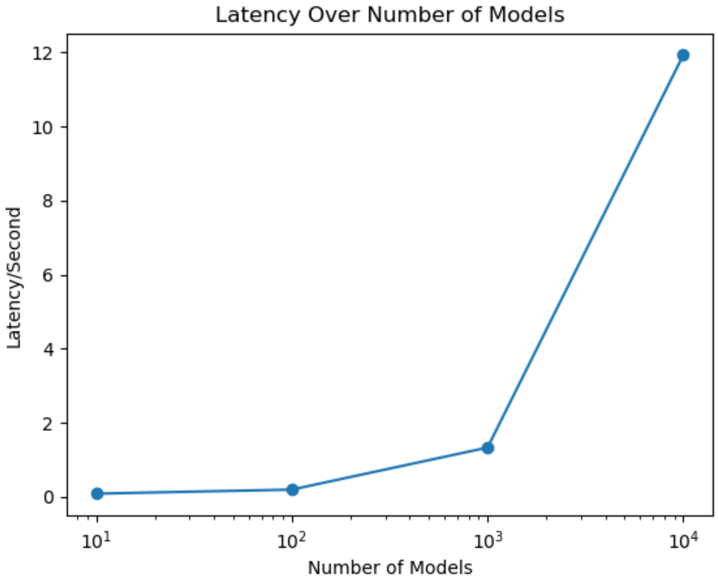


Figure 5.5: Ensemble latency vs. ensemble size when ensemble size is large (up to 10^4).

fore, identifying the optimal training epoch is vital for achieving a balance between diversity and protection.

In our study, we aimed to determine the most suitable training epoch by evaluating HyperNets trained from 500 to 1150 epochs. For each HyperNet, we generated 10 MaskNets and tested their average classification accuracy in scenarios both with and without inversion attacks. The rationale behind generating multiple MaskNets for each HyperNet is to assess the consistency and robustness of the protection across different generated models.

In our experiments, we observe that when training reaches 800 epochs, the accuracy in the absence of inversion attacks drops significantly to below 0.2, indicating robust protection. This drop signifies that the generated MaskNets are effectively obfuscating the input data, making it difficult for the classifier to achieve high accuracy, which is desirable in the absence of an inversion attack as it means the original data is well protected.

As training continues beyond 800 epochs, the accuracy converges to around 0.1. This convergence suggests that the models are stabilizing and consistently providing strong protection, but it also indicates a possible overfitting of the noise introduced during training, leading to a decrease in the effective diversity of the MaskNets.

Conversely, in the presence of inversion attacks, the accuracy starts to increase sharply from 850 epochs onwards. This increase implies that the MaskNets are becoming less diverse and more predictable, thereby making it easier for the inversion attack to reverse-engineer the original data. The sharp increase in accuracy in the presence of inversion attacks, as observed beyond 850 epochs, corroborates Ratzlaff et al.'s (2020) conclusion that diversity decreases over time with prolonged training.

Based on our results, we identified 850 epochs as the optimal training point. At this stage, the balance between maintaining diversity and ensuring effective protection against inversion attacks is achieved. The models trained up to this epoch offer strong protection by significantly lowering classification accuracy in the absence of attacks while still maintaining enough diversity to mitigate the effectiveness of inversion attacks. This balance is crucial for practical applications where both security and model performance need to be optimized.

5.3.2. Hyperparameter

Experiment Setup. To investigate the impact of various hyper-parameter settings shown in Equation 4.4 on the performance and diversity of a neural network system, nine different combinations of parameters were evaluated. The parameters tested include:

- **Diversity Threshold:** This threshold determines when the diversity loss will contribute to the total loss. Set at three different values: 0.01, 0.001, and 0.0001.
- **Lambda Diversity:** It controls the trade-off between diversity and effectiveness in the generated MaskNets. Set at three different values: 1 and 10.

This results in a total of 9 unique hyper-parameter pairs. For each pair, the following process was undertaken:

- **HyperNet Training:** A HyperNet was trained for 850 epochs using the specified pair of hyper-parameters.
- **MaskNet Generation:** Using the trained HyperNet, 16 MaskNets were generated for each hyper-parameter setting.

The outcomes of this experiment focused on two key metrics:

- **Protection Performance in the Absence of Inversion Attack:** This measures how well the network performs in protecting its data or functionality when no inversion attack is present.

- **Diversity of Generated MaskNets:** This assesses the variation among the 16 MaskNets produced by the same HyperNet, indicating the level of diversity introduced by the hyper-parameter settings.

The results obtained from this comprehensive evaluation provided insights into the relationship between hyper-parameter settings and the resulting performance and diversity of the MaskNets. By systematically varying the threshold and lambda diversity values, the study aimed to identify optimal settings that balance protection performance and diversity, which are crucial for the robustness and generalization of the neural network system.

Protection Performance Result. In terms of the $\lambda_{diversity}$, the $\lambda_{diversity}$ equals 10 showing the most satisfying protection performance. Under different $\lambda_{diversity}$ settings, the optimal threshold is also different. For example, under the lambda diversity 10, the optimal threshold is 0.0001. As shown in Figure 5.7, the hyper-parameter setting also has a great impact on the generated MaskNets' diversity. Considering both the protection performance and diversity, the best hyper-parameter setting should be $\lambda_{diversity}$ = 10 and threshold = 0.001.

As shown in Figure 5.6, both of the settings for $\lambda_{diversity}$ and threshold have a great impact on the protection performance. For the group $\lambda_{diversity}$ = 1 and threshold = 0.01 the accuracy values are relatively low, ranging from 0.041 to 0.161, indicating a moderate level of resistance to inversion attacks at this threshold. As the threshold decreases to 0.001 and 0.0001, we observe a notable reduction in accuracy values. This trend suggests that lower thresholds enhance the protection against inversion attacks, making it harder for the adversary to successfully recognize the images.

When examining the $\lambda_{diversity}$ = 10 group, we see similar trends. At the 0.01 threshold, accuracy values are between 0.0 and 0.116, indicating a decent level of protection. However, as the threshold decreases to 0.001 and 0.0001, the accuracy values further drop. This consistent decline across various divisions reinforces the idea that smaller thresholds are more effective at reducing the success of inversion attacks.

Overall, the data illustrates that lower perturbation thresholds generally lead to lower classification accuracies in mix-inversion scenarios, thereby enhancing protection against unauthorized facial recognition.

Diversity Result. The provided data represents the KL (Kullback-Leibler) divergence values, indicating the diversity of the networks under different perturbation thresholds and division strategies across various epochs. For the $\lambda_{diversity}$ = 1 group, at a threshold of 0.01, the KL divergence values range from 103.085 to 104.412, showing moderate diversity among the networks. As the threshold decreases to 0.001, the values fall to between 97.290 and 99.434, and further decrease to 81.445 to 82.359 at the 0.0001 threshold. This trend suggests that lower thresholds result in more consistent networks with less diversity.

In the $\lambda_{diversity}$ = 10 groups, the KL divergence values at a 0.01 threshold are relatively high, ranging from 102.944 to 103.879, indicating substantial diversity. When the threshold is reduced to 0.001, the values decrease slightly to between 100.661 and 101.841. At the 0.0001 threshold, the values further drop to a range of 100.239 to 101.212, indicating that the networks become more similar with lower thresholds.

Overall, the data demonstrates that both the division strategy and perturbation threshold significantly affect the diversity of the networks. Higher $\lambda_{diversity}$ generally maintains higher diversity at the same threshold compared to lower $\lambda_{diversity}$. However, as the threshold decreases, the diversity decreases for both division strategies, indicating a trend towards more uniform network behavior with tighter perturbation constraints. This suggests that careful adjustment of both division and threshold parameters is crucial for balancing network diversity and consistency.

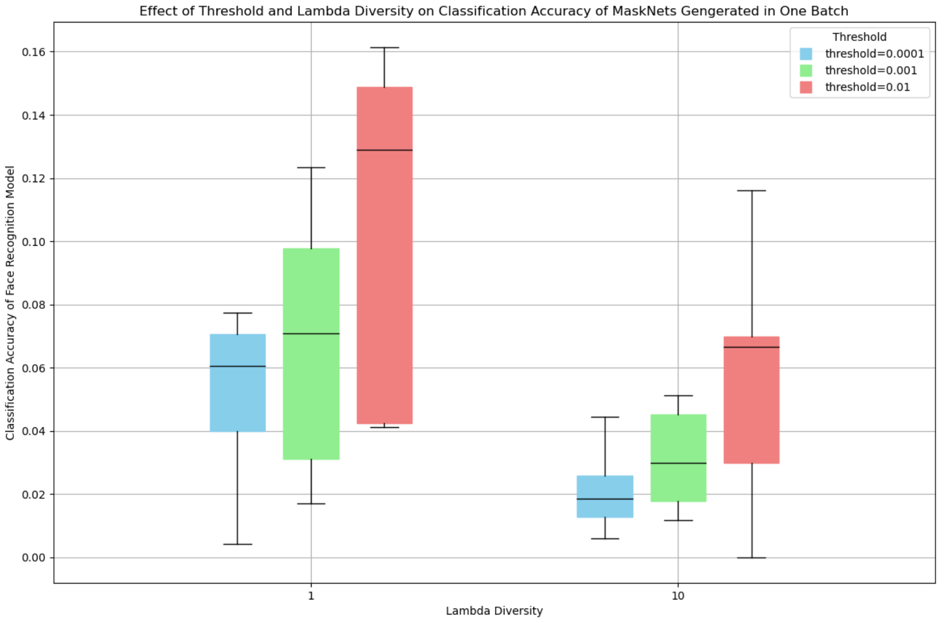


Figure 5.6: The average protection performance when the inversion attack is absent.

5.4. Stronger Attackers

5.4.1. System Design

In previous experiments, we assumed attackers had access to only a single MaskNet instance within a batch. However, in real-world scenarios, attackers may gain access to multiple MaskNets by registering additional devices. Consequently, attackers could use two or more MaskNets for training the inversion attack instead of relying on just one. This section explores the implications of increasing the number of acquired MaskNets.

To investigate this, we conducted an experiment where 16 MaskNets were generated simultaneously in one batch. We then assumed attackers could randomly acquire 2 or 3 of these MaskNets. The primary focus was on assessing the classification accuracy after inversion using the InvNet trained with multiple MaskNets. By evaluating the performance with 2 or 3 MaskNets, we aimed to understand how the increased availability of MaskNets affects the success rate of inversion attacks.

The results of this experiment are crucial for understanding the potential vulnerabilities in scenarios where attackers have access to more than one MaskNet. If attackers can achieve higher classification accuracy with multiple MaskNets, it highlights a significant risk and the need for stronger protection mechanisms. This study not only reveals potential weaknesses but also informs the development of more robust defenses against sophisticated adversarial attacks in facial recognition systems.

5.4.2. Attack Performance with One HyperNet

While training with the same number of epochs, Table 5.3 presents the average mix-inversion classification accuracy for five target users under varying attacker capabilities. The baseline scenario features an attacker with access to only one MaskNet. In the other two experimental scenarios, the attacker possesses two and three MaskNets, respectively, to train the InvNet.

Table 5.3 details the mean mix-inversion accuracy across these different settings. In the base-

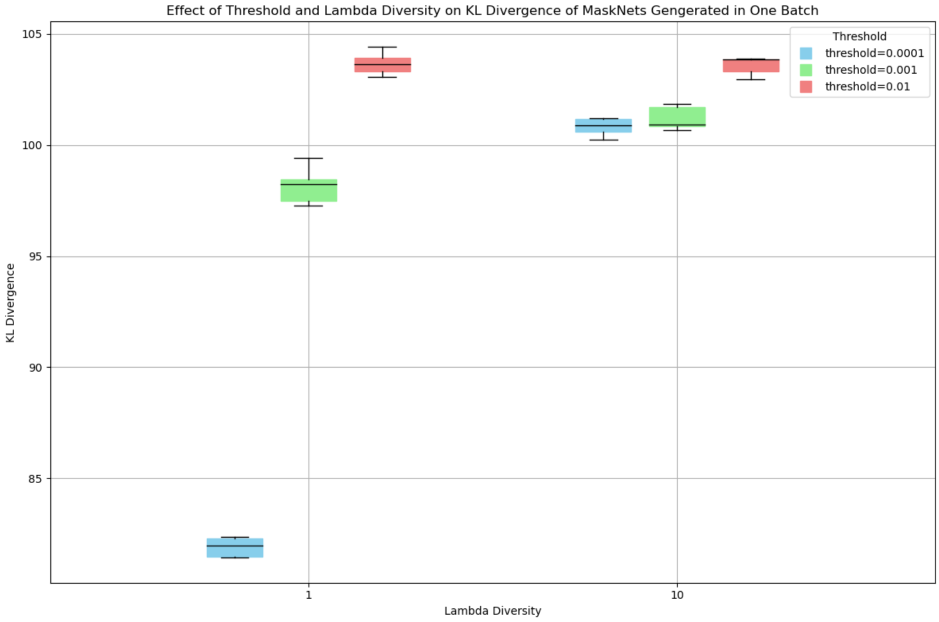


Figure 5.7: The KL divergence of 16 MaskNets generated in one batch.

line scenario, where the attacker has access to just one MaskNet, the average classification accuracy across all users is approximately 0.3502. This relatively low accuracy indicates that with only a single MaskNet, the attacker's ability to successfully invert the obfuscation and accurately classify the images is limited.

When the attacker uses two MaskNets for the inversion attack, the average classification accuracy significantly improves to approximately 0.55. This marked increase suggests that having access to an additional MaskNet provides the attacker with more information, thereby enhancing their capability to breach the obfuscation and achieve higher classification accuracy.

Further increasing the number of MaskNets available to the attacker to three results in a slight additional improvement, with the average classification accuracy reaching approximately 0.554. This marginal gain indicates that while adding a third MaskNet does provide some incremental benefit to the attacker, the most substantial improvement is observed when moving from one to two MaskNets.

These findings indicate that the effectiveness of inversion attacks generally improves with the availability of multiple MaskNets. The significant jump in accuracy when the attacker transitions from one to two MaskNets underscores the added value of having diverse sources of information for training the InvNet. However, the relatively smaller increase when moving from two to three MaskNets suggests diminishing returns with the addition of each new MaskNet.

Overall, these results highlight the importance of limiting the number of MaskNets that potential attackers can access. By restricting access, the robustness of obfuscation measures can be better maintained, reducing the efficacy of inversion attacks and thereby enhancing the security of the obfuscated images.

5.4.3. Attack Performance with Multiple HyperNet

In the preceding sections, it becomes evident that while introducing the HyperNet enhances resilience against inversion attacks, the accessibility of multiple MaskNets to an attacker heightens the risk of successful intrusion. To mitigate this vulnerability, one potential solution is to incorporate multiple HyperNets.

Table 5.3: Comparison of Mix-inversion Result for Stronger Attackers.

Mean Successful Attack Rate	1 MaskNets for inversion attack	2 MaskNets for inversion attack	3 MaskNets for inversion attack
User1	0.332	0.55	0.56
User2	0.341	0.59	0.58
User3	0.412	0.58	0.59
User4	0.534	0.67	0.68
User5	0.132	0.36	0.36

In the above discussion, only one HyperNet is trained to generate MaskNets every time. Therefore, all the MaskNets share the same network structure although with different parameters. However, for enhanced robustness, the Privacy-Preserving System (PSP) can adopt multiple HyperNets to generate diverse MaskNet structures. This approach offers two key advantages:

- **Diversification**: The introduction of additional HyperNets fosters greater diversity among the MaskNets for different users, thereby diminishing the likelihood of successful inversion attacks. Each HyperNet can be trained independently to produce a unique class of MaskNets. This variation ensures that even if an attacker manages to acquire multiple MaskNets, the diversity in their structures will make it considerably harder to develop a universally effective inversion strategy.

  For instance, consider a scenario where three HyperNets are used to generate MaskNets. Each HyperNet produces MaskNets with a unique architecture and set of parameters. When users employ these MaskNets to perturb their images, the resultant images will carry perturbations reflective of different structural nuances. An attacker, therefore, would need to develop multiple inversion networks, each tailored to a specific MaskNet structure, significantly increasing the complexity and resource requirements of the attack.

- **Redundancy**: With multiple HyperNets in place, users have the flexibility to use multiple MaskNets with distinct structures. When seeking to safeguard their images, users can randomly select different MaskNets to apply perturbations. This strategy complicates inversion attacks, as the attacker's collection of images from the target user may encompass various types of perturbations.

  Consider a user who has access to MaskNets from three different HyperNets. Each time they need to protect an image, they can randomly choose one of these MaskNets. Over time, the user's protected images will display a mix of perturbation patterns, each corresponding to a different MaskNet structure. This randomness introduces an additional layer of complexity for an attacker, who must now account for multiple perturbation types in their inversion attempts.

  Moreover, redundancy ensures continuity and resilience in the PSP. If one HyperNet's MaskNet is compromised, the system can continue to function effectively using MaskNets from the other HyperNets. This redundancy also facilitates seamless updates and improvements. New HyperNets with enhanced security features can be introduced without disrupting the existing protection mechanisms, allowing for continuous evolution of the PSP in response to emerging threats.

In summary, the integration of multiple HyperNets into the PSP significantly enhances its resilience against inversion attacks by introducing diversification and redundancy. This approach not only complicates the attacker's efforts but also ensures the system's robustness and adaptability. By fostering greater diversity among MaskNets and allowing users to randomly select different MaskNets for image perturbation, the PSP can effectively mitigate the risks associated with attackers accessing multiple MaskNets. The practical implementation of this strategy requires careful planning and continuous monitoring, but the benefits in terms of enhanced security and robustness make it a compelling solution for privacy-preserving facial recognition systems.

In our experiment here, we train three different HyperNets, generating three different structures of MaskNets. The three different structures of Masknets are as follows:

- **MaskNet 1:** The MaskNet 1 consists of two convolutional layers followed by a fully connected layer. The first convolutional layer has 16 output channels, while the second has 32. Both convolutional layers use a 3x3 kernel size with a stride of 1 and padding of 1. The fully connected layer receives the output from the convolutional layers and produces a 256-dimensional output.
- **MaskNet 2:** The MaskNet 2 has the same structure except the last output layer is 128-dimensional, which is suitable for less computation resource devices.
- **MaskNet 3:** The MaskNet 3 also consists of two convolutional layers followed by a fully connected layer. The first convolutional layer has 32 output channels, while the second has 32. Both convolutional layers use a 3x3 kernel size with a stride of 1 and padding of 1. The fully connected layer is 256-dimensional as well.

For each of the three HyperNets, we use them to generate 10 MaskNets, resulting in a total of 30 MaskNets with three distinct structures. From each structure, we randomly select one MaskNet to train an InvNet, as illustrated in Figure 5.8. This setup ensures that each InvNet is specifically tailored to a unique MaskNet structure generated by its corresponding HyperNet.

To assess the robustness of this approach, we conducted an experiment where each of these three InvNets is used to restore perturbed images generated by the MaskNets from the other two structures. The primary goal of this experiment is to verify that training an InvNet on one MaskNet structure does not generalize well to restoring images perturbed by MaskNets of different structures. This cross-structure evaluation is crucial for demonstrating the effectiveness of using multiple HyperNets to enhance protection against inversion attacks.

The results, as shown in Figure 5.8, indicates that the average classification accuracy under inversion attack when multiple HyperNets are introduced ranges between 0.2 and 0.4. This relatively low accuracy suggests that the InvNets struggle to successfully restore images perturbed by MaskNets with different structures, confirming that the introduction of multiple HyperNets significantly complicates the inversion process for attackers. Consequently, the system exhibits robust protection against inversion attacks, underscoring the effectiveness of using diversified HyperNets to generate MaskNets with varied structures. This strategy not only enhances the overall security of the system but also demonstrates the practical viability of implementing multiple HyperNets in privacy-preserving facial recognition systems.

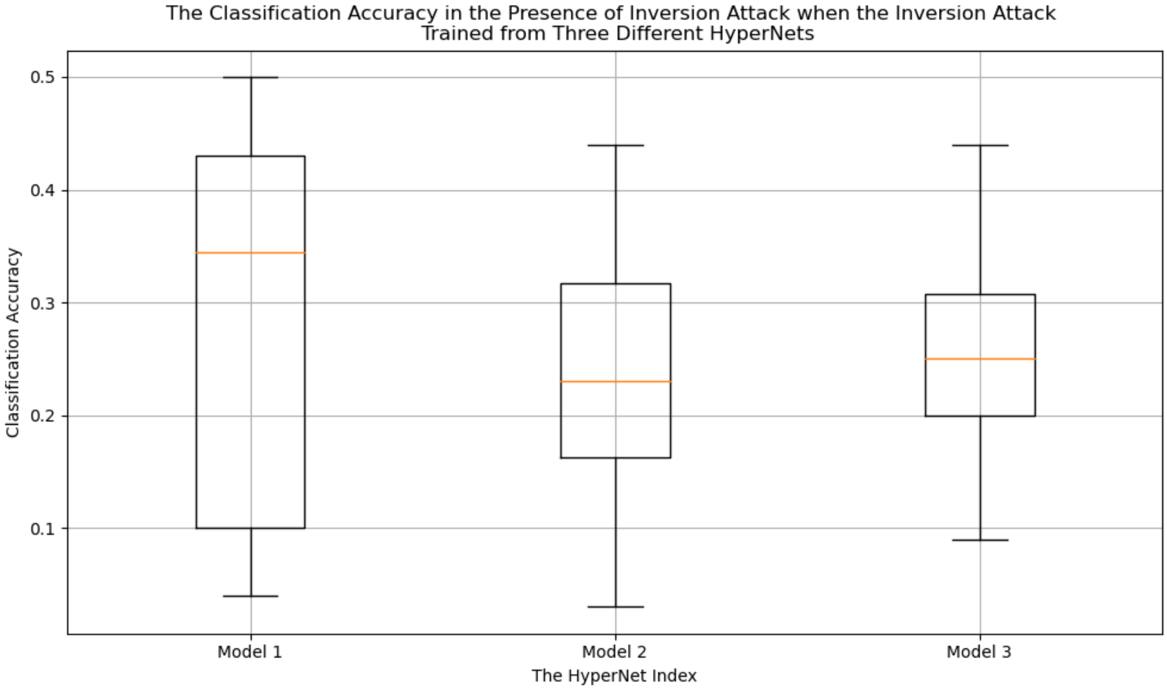


Figure 5.8: The successful attack rate when InvNet is trained using MaskNets from Model 1, Model 2, or Model 3.

6

Related Works

6.1. Image Privacy Protection

In conventional approaches to safeguarding face image privacy, methods such as blurring, masking, or entirely blacking-out facial images have been commonly employed [13]. While these techniques effectively obscure facial features, thereby preserving anonymity, they often result in a loss of important visual information. This loss can significantly diminish the usefulness of the image for various applications. Furthermore, some research has proven that even though the images are artificially blurred or masked, it is still possible to reconstruct the original images, which eliminates the protection [14]. As such, there is a growing need for more sophisticated privacy protection techniques that can effectively obscure sensitive information while preserving the utility and interpretability of the protected images. In this section, recent prior efforts to protect users' image privacy are discussed.

6.1.1. Images Alteration Techniques

K-anonymity Algorithms. The methods originate from the theory proposed by Sweeney etc. [15] in 2002. The common idea behind all k-anonymity-based methods is to provide identical or similar aggregated samples with the original examples, therefore de-identifying the original examples and ensuring a recognition rate of less than 1/k. Based on this idea, Newton et al. [16] proposed a k-same algorithm to calculate the similarity between faces and create new image components. This method successfully makes it difficult for the face recognition model to identify while maintaining the facial details. Lily et al. [17] further improve this algorithm by introducing new solutions to achieve a recognition rate of zero instead of just below the theoretical maximum of 1/k. In addition, more studies are carried out to make the de-identified faces more natural and preserve more non-identity-related information including GARP-face algorithms [18], K-Diff- furthest [19].

Active Appearance Model. To further improve the K-same algorithms [16], Edwards et al. [20] demonstrated a method to interpret face images using the Active Appearance Model(AAM), and this was later used in dealing with the "ghosting" artifacts problem - the visual effects or distortions that may occur in images in K-same algorithms [16]. For instance, in a related study [21], an empirical approach was proposed to apply k-anonymity de-identification to AAM features, failing the identification of either human eyes or facial recognition algorithms. Although using the AAM can effectively alter the appearance of faces in videos, the algorithms of preprocessing of the AAM algorithm are time-consuming, making it less likely to be widely used [22]. Despite this, the assumption that each subject only appears once in the dataset is another limitation.

Neural Networks. Given the formidable feature extraction capabilities of deep learning models, considerable efforts have been dedicated to the generation of de-identified images utilizing neural networks. For example, In [23], a hybrid model consisting of two stages is proposed to generate de-identified images while maintaining their visual similarity. In this model, the identity encoder is trained in the first stage, combined with a U-Net-like neural network. Then in the second stage, keeping the trained neural network fixed, Gaussian noise is added directly to the identity embedding according to user demand. In this way, the face verification can be successfully misled. In [24], K Brkić et al. proposed a system based on a neural art algorithm that uses the responses of a deep neural network, to alter the appearance of people. It successfully de-identifies both biometric and non-biometric identifiers. Moreover, extensive research has demonstrated the efficacy of deep autoencoders in the realm of de-identification. In a study conducted by Nousi et al. [25], the authors showcased the potential of deep autoencoders for this purpose. By integrating supervised and unsupervised training methodologies, they successfully utilized deep autoencoders with fine-tuning to achieve effective de-identification.

GAN. Recently, Generative Adversarial Networks(GANs) have also proven to perform well in the de-identification of images. GANs are made up of a generator and a discriminator who compete in creating realistic data samples and evaluating them for authenticity. GANs excel at producing high-quality, diverse data with fine detail, making them useful for tasks such as image synthesis, text generation, and data augmentation. In [26], AdvGAN is proposed to generate perturbations efficiently and has proven to have a high success attack rate on various models and under state-of-the-art defenses. In [27], Wu et al. only focused on removing or altering biometric information and developed a novel Privacy-Protective-GAN to protect privacy. At the same time, Tao Li and Lei Lin also propose AnonymousNet to further balance usability and privacy from both measurable and intuitive perspectives. They also came up with countermeasures for facial privacy and ways to be manipulated to fulfill users' needs [28]. In [8], Sun et al. proposed an inpainting method to generate realistic personal images using GANs, largely improving the user experience and achieving satisfied de-identification performance again face recognition model.

Anti-FR Systems. To safeguard user images from unauthorized tracking, researchers have introduced Anti-Face Recognition (Anti-FR) Systems designed to introduce imperceptible perturbations to user images. These systems leverage advanced technologies to render classifiers ineffective against the modified images while preserving their utility. Notably, the Fawkes system, introduced by Shan et al. [5], offers an efficient and straightforward approach to privacy protection by generating cloaks on images. Fawkes has demonstrated wide-ranging applications in privacy-preserving systems, including location-based services and network systems. The image cloaks produced by Fawkes have been shown to exhibit robustness against cloak disruption and cloak detection attempts. Building upon the foundation laid by Fawkes, the LowKey system, as proposed by Cherepanova et al. [6], further enhances privacy protection capabilities. LowKey achieves a notable improvement by showcasing a satisfactory protection rate against face recognition systems deployed by popular commercial black-box APIs such as Microsoft Face Azure and Amazon Rekognition. This advancement is particularly significant as it addresses shortcomings observed in Fawkes, where effectiveness against such systems was limited.

Comparison. In this section, we have explored various efforts to protect individuals from unauthorized facial recognition technologies. To offer a clearer comparison of these approaches, we summarize the key techniques and their characteristics in 6.1. Among these methods, Anti-FR/obfuscation systems perform best, providing the most effective protection with the least

Table 6.1: Comparison of different image privacy protection methods.

	High-Protection Performance	Consistency With Original Version	Time Effeciency*	Anti-Inversion Attack
K-anonymity Algorithms	+	No	+	No
Active Appearance Model	+	No	+	No
Neural Networks	++	No	++	No
GAN	+++	++	+	No
Anti-FR / Obfuscation System	+++	+++	+++	No

Time efficiency*: It measures the time to protect images. The less time used, the higher the time efficiency.

counterproductive effects. However, none of these methods can defend against an inversion attack, which is the primary issue our system HyperObf aims to address.

6.1.2. Face Recognition Model Attack Approaches

Evasion Attack. One direct way to preserve privacy is to make the images difficult for face recognition models to classify. This is proven can be achieved with various techniques. These methods include leveraging techniques described above such as Generative Adversarial Networks (GANs).

By carefully studying the recognition model and applying imperceptible perturbations, it is possible to cause a face recognition model to misclassify an image [29]. The generated perturbed images are called adversarial examples. Various approaches are proposed using this idea. For instance, Sharif et al. devised a method to automatically implement attacks by integrating a pair of eyeglass frames onto facial images [29]. Similarly, another study by Komkov et al. discovered that affixing computed adversarial rectangular stickers onto hats can effectively diminish the accuracy of face recognition systems [30]. Additionally, Thys et al. proposed an approach to conceal individuals from person detectors by introducing an adversarial patch [31].

Poisoning Attack. Instead of evading a trained model through adversarial examples, an alternative strategy involves attacking the training stage by poisoning the training data. This method aims to subvert the performance of trained networks by introducing a set of malicious examples that induce classification errors. Poisoning attacks can be further classified into two types: the clean label attack and the model corruption attack.

The clean label attack is to keep all the labels correct but only change the content of the target class to control the classifier behavior of the specific class. Shafahi et al. [32] showed that in transfer learning only one poisoning data can effectively manipulate the classifier behavior. Besides, they proposed a method to inject around 50 training instances to make poisoning attacks reliable. In [33], by designing a model that generates poison images surrounding the target images in feature space, Zhu et al. achieved over 50% of attack success rate with only 1% of training data poisoned. However, the above clean-label attacks have several limitations. First, they can only successfully cause one selected image to be misclassified. In other words, they can promise to misclassify any images from the target user. This does not satisfy the requirements of privacy protection where the users want to hide from any illegal surveillance. Second, the performance is model-dependent. The above clean label attacks do not transfer to different models. In [33], the success attack rate drops dramatically to 30% for a different model trained on the same dataset.

Another method of data poisoning attack is to generate low-quality training data, to decrease the model accuracy. For example, in [34], a technique called TensorClog was presented and resulted in an increment of the converged training loss and test error by 300% and 272% for the model training on the poisoned data generated by it. However, since this method makes the loss converge to a high level, the privacy tracker may notice the abnormal data and filter them. And since many countermeasures of poisoning attack have been studied and proposed [35] [36], the model corruption attack is easy to detect and eliminate.

6.2. Privacy Preserving In Machine Learning

Privacy-preserving machine learning has become increasingly vital in data-driven applications. Various techniques have emerged to address the challenge of protecting data privacy while enabling effective machine learning processes. Privacy-preserving machine learning (PPML) and privacy-preserving inference (PPI) are both techniques aimed at protecting sensitive data during the process of machine learning and inference. Although they focus on different stages of the machine learning pipeline and address different aspects of privacy protection, the proposed techniques are similar. These techniques encompass encryption-based methods, model training strategies, data-sharing approaches, and other privacy-preserving methodologies. In this section, we delve into the diverse array of techniques employed to protect data privacy in machine learning, providing insights into their principles and applications.

6.2.1. Encryption-based Techniques

Secure Multiparty Computation. Secure Multiparty Computation (SMC) is a cryptographic technique that enables multiple parties to perform computations without revealing their private inputs. The implementation of SMC typically involves using cryptographic protocols to ensure the security and privacy of computations. [37] designs new and efficient protocols for privacy-preserving linear regression, logistic regression, and neural network training in the two non-colluding server models. DeepSecure was the first provably secure framework that simultaneously enabled accurate and scalable privacy-preserving deep learning execution based on Yao's Garbled Circuit (GC) protocol. The framework achieved up to 58-fold higher throughput per sample compared with the best prior solution. Although the GC protocols are effective and support both nonlinear and linear computations, the computation and communication overheads can be very high. Other protocols were also proposed such as Oblivious Transfer [38] and Secret Sharing [39]. The secret-sharing protocol is efficient in terms of additions and multiplications, and it does not require symmetric cryptographic operations in the online phase, making it more suitable for limited resource devices.

However, SMC is high-costly in terms of computational complexity and communication overhead. Thus, SMC is unsuitable for training complex models over big datasets implicating many clients[40].

Homomorphic Encryption. Homomorphic encryption(HE) enables calculations to be done without having a private key to decrypt it. This means HE schemes allow computations on encrypted data. The computed results are also encrypted; data decryption is not required during computation. Encrypting all stages of computation, from input of intermediate values to output, makes HE schemes an ideal choice for outsourcing computation. After decryption, the results of encrypted computation are identical to those obtained by performing the computation on plain data. This capability is highly significant for privacy-preserving computation, as it allows sensitive data to remain encrypted throughout processing, minimizing the risk of exposure.

Many state-of-the-art privacy-preserving systems are based on the HE. For example, Bourse et al. [41] proposed a method using small ciphertexts for NN inference to avoid high computation overhead. CryptoNets [42] was one of the first methods using Fully homomorphic encryption for NN inference, achieving both high throughput and accuracy. Other researchers improved the computation implementation to speed up computation time [43], [44]. The protection level was also considered for improvement, based on the system designed by Shokri and Shmatikov [45], researchers proposed a new system leaking no information over the honest-but-curious cloud server via additively homomorphic encryption [46].

As encrypted sensitive data can only be decrypted by the authorized data owner, approaches relying on HE typically facilitate prediction on encrypted data using existing trained models, rather than training models directly on encrypted data. This limitation arises from the confidentiality of computed results within HE, preventing their use alongside labels during the back-propagation phase for evaluation. Consequently, machine learning models must be initially trained on plaintext data, after which the trained model can be applied to encrypted data for prediction tasks. To tackle this problem, some studies also use emerging functional encryption instead of HE to support both the training and inference phases.

6.2.2. Model Training-based Techniques

Differential Privacy. In differential privacy, the randomized mechanism is applied to the data analysis process to prevent the leaking of sensitive information about individuals. This mechanism adds computed noise to the query results or aggregates the data in a way that obscures individual contributions, while still allowing useful statistical analysis to be performed on the dataset. The differential privacy was first proposed by Dwork in 2006 to provide strong privacy protection against statistical database disclosure. Then wide attention was raised and is used in practice. For example, it is used in auction mechanism design to protect equity and participants have limited effect on the outcome.

However, the accuracy may decrease with the increase of privacy protection at this stage. So researches are conducted to find the trade-off between accuracy and privacy [47]. In [48], a full decentralization technique was used to increase privacy protection while maintaining the utility. In [49], the individual differential privacy was proposed as an alternative differential privacy notion that offers the same privacy guarantees as standard differential privacy to individuals which has less impact on the data utility. Another way is to combine differential privacy with another technique such as memorization or adding a proxy server [50], [51].

Recently, studies about differential privacy have focused more on computation efficiency and applications. In 2016, Abadi et al. improved the computational efficiency of differentially private training by introducing algorithms like training examples and sub-dividing tasks into smaller batches to reduce memory consumption. As a result, they can train deep neural networks with non-convex objectives under acceptable privacy training overhead as well as the software complexity, training efficiency, and model quality. In [52], the differential privacy is further used to solve the privacy problem in deep reinforcement learning.

6.2.3. Data Sharing-based Techniques

Distributed Learning. In[53], the researchers proposed a novel system to let participants independently train their own datasets and selectively share subsets of their model's key parameters. In this system, the participants can still benefit from other participants' training while preserving the privacy of their own data. The PrivcyNet [54] was also proposed to split the DNN model to enable cloud-based training with control of control of privacy loss. The privacy and resource consumption trade-off was measured and compared. However, according to the [55], it is also possible to reconstruct an image using the leaking information from the output of the edge device in a distributed learning system and achieve 70% similarity.

Federated Learning. [56] Federated Learning is a machine learning paradigm in which the model is trained locally. This process is without data exchange across several decentralized edge devices or servers. Only model updates—typically in the form of gradients—are sent to a central server where they are combined to update the global model. Federated learning now is extensively studied for privacy protection.

Most privacy-preserving federated learning consists of encryption or noise addition. By training a personalized model with differential privacy, Hu et al. [57] proposed a learning approach that addresses user heterogeneity, with rigorous analysis of convergence and privacy guarantees, and demonstrates its robustness and efficacy through experiments on mobile sensing data. The noise can also be added before model aggregation to disturb local parameters. In [58], the privacy was further improved by directly adding noise to the parameters. Another way to ensure privacy is to use encryption. For example, Li et al. [59]proposed a system called chain-PPFL to achieve higher accuracy compared with differential privacy based on the chained SMC technique. In some studies, noise addition and encryptions are also combined [60].

7 Conclusion, Limitations and Future Works

This thesis has significantly contributed to the field of safeguarding individuals from unauthorized facial recognition systems. Our research builds upon the foundation of existing face obfuscation systems, which apply small, precisely calculated perturbations to protect images from being recognized. These face obfuscation techniques have been proven effective against state-of-the-art face recognition APIs, including those provided by Microsoft, Amazon, and Face++. In the initial phase of our research, we conducted an exhaustive evaluation of these existing face obfuscation systems. Our findings revealed that such systems could be compromised by the inversion attack, thereby highlighting the critical need for employing unique MaskNets tailored to individual users.

To address this issue, our thesis proposed HyperObf, which introduced HyperNet to dynamically generate MaskNets. This innovation ensures that while all MaskNets share the same structural framework, each possesses distinct weights, thus providing personalized protection. The deployment of HyperNet is managed by a Privacy Service Provider (PSP), which is responsible for training and maintaining the HyperNet. When a user requires protection, they submit a request to the PSP, which in turn generates a unique MaskNet specifically for that user and transmits it back to them.

Moreover, we delved into scenarios involving a more potent attacker, who might gain access to a larger segment of the MaskNet and utilize it for training purposes. Our research indicates that increasing the amount of MaskNet available for inversion attacks indeed raises the likelihood of a successful breach. To mitigate this risk, we explored the strategy of employing multiple MaskNets to generate a more diverse set of MaskNets with varying structures. Our results demonstrate that the introduction of multiple HyperNets significantly enhances the robustness of the protection against inversion attacks.

Our exploration addressed two central questions mentioned in Section 1:

- **Can attackers effectively restore perturbed images through inversion attacks and use the restored images to train a valid face recognition model?**
  As discussed in Section 5.1, our experiments revealed that an attacker, by training an autoencoder inversion network, can restore perturbed images even without knowing the implementation details of the obfuscation system. Once the attacker gains access to the obfuscation system, they can train the inversion network on the perturbed images. Upon completing this training, the inversion network can successfully restore all images protected by the same MaskNets. This demonstrates the potential vulnerability of static

obfuscation systems and underscores the necessity for more dynamic and diverse protection mechanisms, such as those provided by the HyperNet approach.

- **Is there an efficient way to generate perturbations that are hard to get rid of through inversion attacks?**
  As demonstrated in our experiments, introducing HyperNets allows for the generation of unique perturbations, significantly complicating their removal by inversion attacks. This added complexity stems from the diverse structures and parameters of MaskNets generated by different HyperNets, which hinder an attacker's ability to develop a universally effective inversion strategy. Furthermore, HyperNet outperforms other dynamic approaches by offering superior diversity and time efficiency. The ability to generate varied MaskNets quickly and effectively ensures that each user's data is protected with minimal processing time, enhancing the practicality and scalability of the system. The combination of increased diversity and efficiency makes HyperNet a robust solution for enhancing security in privacy-preserving facial recognition systems.

While the thesis highlights the benefits of introducing HyperNet for dynamic MaskNet generation and demonstrates its superiority over other dynamic approaches, it does not guarantee individual protection against inversion attacks. There are still possibilities that an attacker could obtain MaskNets similar to those used by the target user, leading to successful attacks. The thesis primarily focuses on overall performance rather than delving into protection for individual users. To ensure individual protection, users could adopt a strategy of using multiple MaskNets to protect their images, updating their MaskNets periodically to enhance security.

Moreover, like most obfuscation systems, HyperObf relies on the assumption that all images accessible to attackers are perturbed. This presupposes that users are highly privacy-conscious and that attackers cannot obtain clean images except those uploaded by users themselves. This assumption, however, may not always hold true in real-world scenarios. Consequently, future studies should explore more complex situations where attackers can access both perturbed and clean images for training. Developing more robust methods to handle such realities is crucial.

For example, as many studies have suggested, incorporating adversarial training during the HyperNet training phase could help in generating stronger defenses. Adversarial training involves creating and using adversarial examples during the training process to improve the model's robustness against attacks. This approach can make the MaskNets more resistant to inversion attacks by ensuring that the perturbations remain effective even when attackers have access to both perturbed and clean images. Additionally, combining multiple HyperNets and frequently updating MaskNets can further diversify the perturbations, making it more challenging for attackers to develop effective inversion networks.

In summary, while the HyperNet approach offers significant advantages, it does not fully address individual protection against inversion attacks. Future research should focus on enhancing the robustness of the system by considering more complex attack scenarios and employing advanced techniques like adversarial training. By continuously evolving and improving the protection mechanisms, we can better safeguard users' privacy in an increasingly adversarial environment.

References

[1] Steve White. *CCTV Cameras by Countries Cities (2023 Guide)*. Accessed on May 7, 2024. 2023. URL: https://upcomingsecurity.co.uk/security-guides/cctv-camera-guides/cctv-by-country/.

[2] Adam Satariano. *Police Use of Facial Recognition Is Accepted by British Court*. Accessed on May 7, 2024. 2019. URL: https://www.nytimes.com/2019/09/04/business/facial-recognition-uk-court.html.

[3] *Amazon Rekognition Face Verification API.* Accessed on May 7, 2024. URL: https://aws.amazon.com/cn/rekognition/.

[4] *Face++ Face Searching API*. Accessed on May 7, 2024. URL: https://www.faceplusplus.com/face-searching/.

[5] Shawn Shan et al. *Fawkes: Protecting Privacy against Unauthorized Deep Learning Models*. 2020. arXiv: 2002.08327 [cs.CR].

[6] Valeriia Cherepanova et al. *LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition*. 2021. arXiv: 2101.07922 [cs.CV].

[7] David Ha, Andrew Dai, and Quoc V. Le. *HyperNetworks*. 2016. arXiv: 1609.09106 [cs.LG].

[8] Qianru Sun et al. "Natural and effective obfuscation by head inpainting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5050–5059.

[9] Seong Joon Oh, Mario Fritz, and Bernt Schiele. "Adversarial Image Perturbation for Privacy Protection A Game Theory Perspective". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1491–1500. DOI: 10.1109/ICCV.2017.165.

[10] Zecheng He, Tianwei Zhang, and Ruby B. Lee. "Model inversion attacks against collaborative inference". In: *Proceedings of the 35th Annual Computer Security Applications Conference*. ACSAC '19. San Juan, Puerto Rico, USA: Association for Computing Machinery, 2019, pp. 148–162. ISBN: 9781450376280. DOI: 10.1145/3359789.3359824. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3359789.3359824.

[11] Hong-Wei Ng and Stefan Winkler. "A data-driven approach to cleaning large face datasets". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 343–347. DOI: 10.1109/ICIP.2014.7025068.

[12] Neale Ratzlaff and Li Fuxin. *HyperGAN: A Generative Model for Diverse, Performant Neural Networks*. 2020. arXiv: 1901.11058 [cs.LG].

[13] Slobodan Ribaric, Aladdin Ariyaeeinia, and Nikola Pavesic. "De-identification for privacy protection in multimedia content: A survey". In: *Signal Processing: Image Communication* 47 (2016), pp. 131–151. ISSN: 0923-5965. DOI: https://doi.org/10.1016/j.image.2016.05.020. URL: https://www.sciencedirect.com/science/article/pii/S0923596516300856.

[14] Ralph Gross et al. "Integrating utility into face de-identification". In: *Privacy Enhancing Technologies: 5th International Workshop, PET 2005, Cavtat, Croatia, May 30-June 1, 2005, Revised Selected Papers 5*. Springer. 2006, pp. 227–242.

[15] Latanya Sweeney. "k-anonymity: a model for protecting privacy". In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10.5 (Oct. 2002), pp. 557–570. ISSN: 0218-4885. DOI: 10.1142/S0218488502001648. URL: https://doi.org/10.1142/S0218488502001648.

[16] E.M. Newton, L. Sweeney, and B. Malin. "Preserving privacy by de-identifying face images". In: *IEEE Transactions on Knowledge and Data Engineering* 17.2 (2005), pp. 232–243. DOI: 10.1109/TKDE.2005.32.

[17] Lily Meng and Zongji Sun. “Face De-identification with perfect privacy protection”. In: *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2014, pp. 1234–1239. DOI: `10.1109/MIPRO.2014.6859756`.

[18] Liang Du et al. “GARP-face: Balancing privacy protection and utility preservation in face de-identification”. In: *IEEE international joint conference on biometrics*. IEEE. 2014, pp. 1–8.

[19] Zongji Sun, Li Meng, and Aladdin Ariyaeeinia. “Distinguishable de-identified faces”. In: *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. Vol. 4. IEEE. 2015, pp. 1–6.

[20] Iain Matthews and Simon Baker. “Active appearance models revisited”. In: *International journal of computer vision* 60 (2004), pp. 135–164.

[21] Hehua Chi and Yu Hen Hu. “Facial image de-identification using identiy subspace decomposition”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 524–528. DOI: `10.1109/ICASSP.2014.6853651`.

[22] Hehua Chi and Yu Hen Hu. “Face de-identification using facial identity preserving features”. In: *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2015, pp. 586–590. DOI: `10.1109/GlobalSIP.2015.7418263`.

[23] Yunqian Wen et al. “A Hybrid Model for Natural Face De-Identiation with Adjustable Privacy”. In: *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. 2020, pp. 269–272. DOI: `10.1109/VCIP49819.2020.9301866`.

[24] Karla Brkić, Tomislav Hrkać, and Zoran Kalafatić. “Protecting the privacy of humans in video sequences using a computer vision-based de-identification pipeline”. In: *Expert Systems with Applications* 87 (2017), pp. 41–55.

[25] Paraskevi Nousi et al. “Deep autoencoders for attribute preserving face de-identification”. In: *Signal Processing: Image Communication* 81 (2020), p. 115699.

[26] Chaowei Xiao et al. *Generating Adversarial Examples with Adversarial Networks*. 2019. arXiv: `1801.02610 [cs.CR]`.

[27] Yifan Wu, Fan Yang, and Haibin Ling. *Privacy-Protective-GAN for Face De-identification*. 2018. arXiv: `1806.08906 [cs.CV]`.

[28] Tao Li and Lei Lin. *AnonymousNet: Natural Face De-Identification with Measurable Privacy*. 2019. arXiv: `1904.12620 [cs.CV]`.

[29] Christian Szegedy et al. “Intriguing properties of neural networks”. In: (Dec. 2013).

[30] Stepan Komkov and Aleksandr Petiushko. “AdvHat: Real-World Adversarial Attack on ArcFace Face ID System”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, Jan. 2021. DOI: `10.1109/icpr48806.2021.9412236`. URL: `http://dx.doi.org/10.1109/ICPR48806.2021.9412236`.

[31] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. *Fooling automated surveillance cameras: adversarial patches to attack person detection*. 2019. arXiv: `1904.08653 [cs.CV]`.

[32] Ali Shafahi et al. *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*. 2018. arXiv: `1804.00792 [cs.LG]`.

[33] Chen Zhu et al. *Transferable Clean-Label Poisoning Attacks on Deep Neural Nets*. 2019. arXiv: `1905.05897 [stat.ML]`.

[34] Juncheng Shen, Xiaolei Zhu, and De Ma. “TensorClog: An Imperceptible Poisoning Attack on Deep Neural Network Applications”. In: *IEEE Access* 7 (2019), pp. 41498–41506. DOI: `10.1109/ACCESS.2019.2905915`.

[35] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. *Certified Defenses for Data Poisoning Attacks*. 2017. arXiv: `1706.03691 [cs.LG]`.

[36] Shiqi Shen, Shruti Tople, and Prateek Saxena. “A uror: defending against poisoning attacks in collaborative deep learning systems”. In: Dec. 2016, pp. 508–519. DOI: `10.1145/2991079.2991125`.

[37] Payman Mohassel and Yupeng Zhang. “SecureML: A System for Scalable Privacy-Preserving Machine Learning”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 19–38. DOI: 10.1109/SP.2017.12.

[38] Nitin Agrawal et al. *QUOTIENT: Two-Party Secure Neural Network Training and Prediction*. 2019. arXiv: 1907.03372 [cs.CR].

[39] Ilan Komargodski, Moni Naor, and Eylon Yogev. “How to Share a Secret, Infinitely”. In: *IEEE Transactions on Information Theory* 64.6 (2018), pp. 4179–4190. DOI: 10.1109/TIT.2017.2779121.

[40] Ahmed El Ouadrhiri and Ahmed Abdelhadi. “Differential Privacy for Deep and Federated Learning: A Survey”. In: *IEEE Access* 10 (2022), pp. 22359–22380. DOI: 10.1109/ACCESS.2022.3151670.

[41] Florian Bourse et al. *Fast Homomorphic Evaluation of Deep Discretized Neural Networks*. Cryptology ePrint Archive, Paper 2017/1114. https://eprint.iacr.org/2017/1114. 2017. URL: https://eprint.iacr.org/2017/1114.

[42] Nathan Dowlin et al. “CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy”. In: (Mar. 2016).

[43] Qian Lou and Lei Jiang. *SHE: A Fast and Accurate Deep Neural Network for Encrypted Data*. 2019. arXiv: 1906.00148 [cs.CR].

[44] Edward Chou et al. *Faster CryptoNets: Leveraging Sparsity for Real-World Encrypted Inference*. 2018. arXiv: 1811.09953 [cs.CR].

[45] Reza Shokri and Vitaly Shmatikov. “Privacy-Preserving Deep Learning”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. CCS ’15. Denver, Colorado, USA: Association for Computing Machinery, 2015, pp. 1310–1321. ISBN: 9781450338325. DOI: 10.1145/2810103.2813687. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2810103.2813687.

[46] Le Trieu Phong et al. “Privacy-Preserving Deep Learning via Additively Homomorphic Encryption”. In: *IEEE Transactions on Information Forensics and Security* 13.5 (2018), pp. 1333–1345. DOI: 10.1109/TIFS.2017.2787987.

[47] Muah Kim, Onur Günlü, and Rafael F. Schaefer. “Federated Learning with Local Differential Privacy: Trade-Offs Between Privacy, Utility, and Communication”. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 2650–2654. DOI: 10.1109/ICASSP39728.2021.9413764.

[48] Edwige Cyffers and Aurélien Bellet. *Privacy Amplification by Decentralization*. 2022. arXiv: 2012.05326 [cs.LG].

[49] Jordi Soria-Comas et al. “Individual Differential Privacy: A Utility-Preserving Formulation of Differential Privacy Guarantees”. In: *IEEE Transactions on Information Forensics and Security* 12.6 (2017), pp. 1418–1429. DOI: 10.1109/TIFS.2017.2663337.

[50] Xiaofeng Ding et al. “A Novel Privacy Preserving Framework for Large Scale Graph Data Publishing”. In: *IEEE Transactions on Knowledge and Data Engineering* 33.2 (2021), pp. 331–343. DOI: 10.1109/TKDE.2019.2931903.

[51] Bin Zhao et al. “Anonymous and Privacy-Preserving Federated Learning With Industrial Big Data”. In: *IEEE Transactions on Industrial Informatics* 17.9 (2021), pp. 6314–6323. DOI: 10.1109/TII.2021.3052183.

[52] Jun Li et al. “A Privacy-Preserving Online Deep Learning Algorithm Based on Differential Privacy”. In: *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 2023, pp. 559–564. DOI: 10.1109/CSCWD57460.2023.10152847.

[53] Reza Shokri and Vitaly Shmatikov. “Privacy-preserving deep learning”. In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2015, pp. 909–910. DOI: 10.1109/ALLERTON.2015.7447103.

[54] Vahid Mirjalili, Sebastian Raschka, and Arun Ross. “PrivacyNet: Semi-Adversarial Networks for Multi-Attribute Face Privacy”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9400–9412. DOI: 10.1109/TIP.2020.3024026.

[55] Hadjer Benkraouda and Klara Nahrstedt. “Image reconstruction attacks on distributed machine learning models”. In: *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*. DistributedML ’21. Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 29–35. ISBN: 9781450391344. DOI: 10.1145/3488659.3493779. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3488659.3493779.

[56] Kang Wei et al. “Federated Learning With Differential Privacy: Algorithms and Performance Analysis”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3454–3469. DOI: 10.1109/TIFS.2020.2988575.

[57] Rui Hu et al. “Personalized Federated Learning With Differential Privacy”. In: *IEEE Internet of Things Journal* 7.10 (2020), pp. 9530–9539. DOI: 10.1109/JIOT.2020.2991416.

[58] Yunlong Lu et al. “Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics”. In: *IEEE Transactions on Industrial Informatics* 16.3 (2020), pp. 2134–2143. DOI: 10.1109/TII.2019.2942179.

[59] Yong Li et al. “Privacy-Preserving Federated Learning Framework Based on Chained Secure Multiparty Computing”. In: *IEEE Internet of Things Journal* 8.8 (2021), pp. 6178–6186. DOI: 10.1109/JIOT.2020.3022911.

[60] Chunyi Zhou et al. “Privacy-Preserving Federated Learning in Fog Computing”. In: *IEEE Internet of Things Journal* 7.11 (2020), pp. 10782–10793. DOI: 10.1109/JIOT.2020.2987958.