applied sciences

MDPI

Article

KRT-FUAP: Key Regions Tuned via Flow Field for Facial Universal Adversarial Perturbation

Xi Jin [1], Yong Liu [1], Guangling Sun [1], Yanli Chen [2], Zhicheng Dong [3] and Hanzhou Wu [1,*]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; jinxi@shu.edu.cn (X.J.); liuyongresearch@163.com (Y.L.); sunguangling@shu.edu.cn (G.S.)
[2] School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550025, China; yanli_027@163.com
[3] School of Information Science and Technology, Tibet University, Lhasa 850000, China; dongzc666@163.com
* Correspondence: hanzhou@shu.edu.cn

Abstract: It has been established that convolutional neural networks are susceptible to elaborate tiny universal adversarial perturbations (UAPs) in natural image classification tasks. However, UAP attacks against face recognition systems have not been fully explored. This paper proposes a spatial perturbation method that generates UAPs with local stealthiness by learning variable flow field to fine-tune facial key regions (KRT-FUAP). We ensure that the generated adversarial perturbations are positioned within reasonable regions of the face by designing a mask specifically tailored to facial key regions. In addition, we pay special attention to improving the effectiveness of the attack while maintaining the stealthiness of the perturbation and achieve the dual optimization of aggressiveness and stealthiness by accurately controlling the balance between adversarial loss and stealthiness loss. Experiments conducted on the frameworks of IResNet50 and MobileFaceNet demonstrate that our proposed method achieves an attack performance comparable to existing natural image universal attack methods, but with significantly improved stealthiness.

Keywords: face recognition; facial universal adversarial perturbation; flow field; facial key regions

check for updates

Citation: Jin, X.; Liu, Y.; Sun, G.; Chen, Y.; Dong, Z.; Wu, H. KRT-FUAP: Key Regions Tuned via Flow Field for Facial Universal Adversarial Perturbation. *Appl. Sci.* **2024**, *14*, 4973. https://doi.org/10.3390/app14124973

Received: 12 April 2024
Revised: 1 June 2024
Accepted: 4 June 2024
Published: 7 June 2024

1. Introduction

With the rapid advancement of technology and continuous improvement in computational hardware, the training and optimization of deep neural networks (DNNs) [1] have received support from data resources and computational capabilities. DNNs have played a vital role in various fields, particularly in technologies such as speech recognition [2], natural language processing [3], image classification [4], and face recognition [5], which have been widely applied in practical scenarios. As an efficient means of individual authentication and biometric identification, face recognition technology leverages DNNs to extract and compare facial image features, enabling the rapid identification of individuals. Thanks to the proliferation of DNN technology, the accuracy and convenience of face recognition have significantly improved, leading to widespread applications in various fields such as security authentication and identity verification.

Despite the remarkable achievements of DNNs in computer vision tasks, their security issues cannot be ignored. Attacks targeting DNNs, such as data poisoning [6], backdoor attacks [7], and adversarial attacks [8], pose serious threats to their stability and reliability. Adversarial attacks, first proposed by Szegedy et al. [9], involve introducing carefully crafted perturbations into input images, resulting in incorrect outputs from DNNs. These attacks pose a significant challenge to the security and accuracy of face recognition. Currently, there are several adversarial attack schemes targeting face recognition, revealing vulnerability in face recognition models and providing important insights for enhancing model security.

Adversarial attacks can be classified into two categories based on their target: one is single adversarial perturbations targeting specific images, which require generating new perturbations for each image, and the other is universal adversarial perturbations. Moosavi-Dezfooli et al. [10] discovered that by designing a single perturbation image, it can be widely applied to a set of data images with similar distributions, significantly reducing the recognition accuracy. However, current adversarial attacks in face recognition are specifically designed for an individual target identity, requiring the regeneration of perturbations for every application. For instance, in generating a tiny patch on the face [11], the adversarial samples can mislead the face recognition network. Additionally, there are adversarial samples of facial makeup generated using generative adversarial networks (GANs) [12], which exploit facial features to generate adversarial samples with specific makeup to deceive the target model.

This paper proposes a universal adversarial perturbation generation method for face recognition, inspired by the domain of natural images. As shown in Figure 1, after applying universal adversarial perturbation to facial images, the extracted facial features no longer resemble those of the original images.

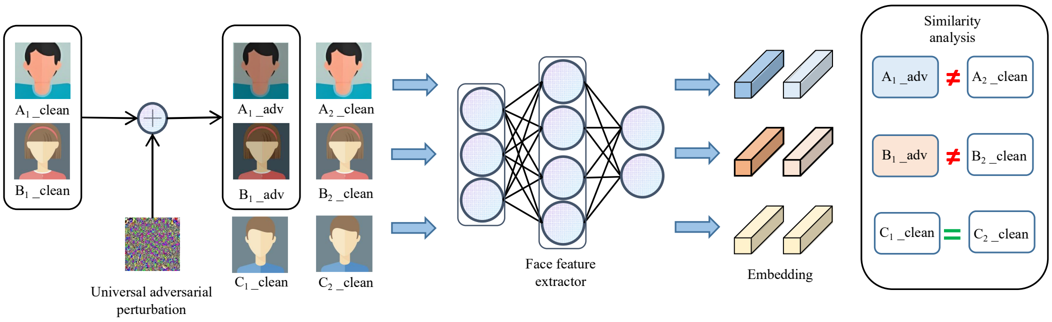


Figure 1. Diagram of universal adversarial perturbations against face recognition. During the normal recognition process, the feature extractor obtains clean face embedding features, which can be used to determine if it is the same face. However, during the adversarial attack process, a universal adversarial perturbation is superimposed on a set of images, causing the feature extractor to incorrectly recognize the embedding features and achieve the attack effect.

Specifically, the image classification task focuses on distinguishing between different categories and emphasizes the extraction of global features, while face recognition focuses more on local features and small differences in the face region. The interpretability analysis of face recognition [13] reveals that after face alignment, the approximate positions of facial features remain fixed. Face recognition primarily focuses on the features within these key regions, and the model achieves identity recognition by comparing distances and similarity features between facial features. Based on this, this paper proposes a method for adding universal adversarial perturbation to facial key regions. By obtaining the key point positions of the face and employing the convex hull algorithm to calculate the approximate regions of facial features, this study overlayed perturbations with different weights on different regions and designed a reasonable loss function to iteratively update the perturbations in the spatial domain. Ultimately, this study achieved universal adversarial perturbations with improved visual stealthiness.

In summary, the main contributions of this study are as follows:

- We apply the concept of universal adversarial perturbations from natural images to the face recognition system, proposing a universal adversarial perturbation attack for face recognition.
- We explore the impact of facial key regions on recognition accuracy. We use learnable flow field to fine-tune the key regions and overlay perturbations with different weights on these regions. This approach not only maintains attack effectiveness but also enhances attack stealthiness.

- We directly control the optimization direction of adversarial facial feature vectors as the adversarial loss, while employing features from a shallow layer of the Visual Geometry Group (VGG) as the stealthiness loss. Through this approach, we generate universal adversarial perturbations in two dimensions.

The subsequent components of this paper are as follows. The rest of the paper is arranged as follows. Section 2 discusses related works, followed by the proposed approach for generating facial universal adversarial perturbation in Section 3. In Section 4, we validate the effectiveness of our method through extensive experiments and evaluate the results of the experiments. Finally, we summarize the entire paper and discuss future research directions.

2. Related Works

2.1. Background of Face Recognition

Face recognition is a biometric technology that utilizes facial information for identification. Currently, the face recognition process typically involves four steps: face detection, face alignment, feature extraction, and feature recognition. The face recognition task is generally divided into two subtasks: face verification and identification. Face verification aims to determine whether a pair of face images belong to the same identity, while identification aims to directly identify the specific identity of a single face image. Presently, feature extraction is predominantly performed using neural network models. With the rapid development of deep learning technology, numerous excellent image classification models have emerged, such as VGG [14], GoogleNet [15], ResNet [16], and MobileNet [17]. These models provide a superior and more flexible network structure for the face recognition task, enabling the construction of deeper networks to handle larger-scale face datasets without concerns about gradient disappearance. Besides network structure, the choice of loss function plays a crucial role in assessing the model's recognition capability. Selecting appropriate loss functions facilitates the separation of face images in different feature spaces, thereby enhancing recognition accuracy. Triplet loss [18] is a common metric learning method that aims to train the model by separating the distances between positive and negative pairs by a certain margin. In addition, other loss functions such as center loss [19] and Arcface [20] are used to train high-precision face recognition models.

2.2. Adversarial Attacks

Szegedy et al. [9] were the first to introduce the concept of adversarial examples. They demonstrated that adding small and deliberately crafted adversarial perturbations to natural images can deceive deep neural networks, causing them to make incorrect predictions. The Fast Gradient Sign Method (FGSM), proposed by Goodfellow et al. [21], is a fast attack strategy. Many subsequent works are based on improvements to FGSM. For instance, the Basic Iterative Method (BIM) [22] is an iterative version of FGSM that generates adversarial examples by iteratively modifying one-step operations. The DeepFool algorithm [23] obtains the adversarial perturbation by calculating the minimum distance of the sample across the decision boundary. Universal adversarial attacks attempt to generate a single perturbation that, when added to any sample, causes the model to make incorrect decisions. Building upon the DeepFool algorithm, Moosavi-Dezfooli et al. [10] demonstrated the existence of UAPs in DNNs. Inspired by GANs, Mopuri et al. [24] utilized a generator to model the distribution of UAPs and implemented diversity in perturbations using this approach. Poursaeed et al. [25] proposed a unified framework called GAP based on a generative model to generate UAPs. Mopuri et al. introduced Fast Feature Fool (FFF) [26] and GD-UAP [27], both of which do not require the use of training data and are applicable to data-independent attack scenarios. Zhang et al. [28] proposed a new perspective on the relationship between the carrier image and UAPs, suggesting that perturbations possess key features that dominate model decisions. They designed a feature-guided UAP algorithm based on this insight. Dai et al. [29] enhanced the computational efficiency of UAPs by proposing a strategy to select the vector direction closest to the previous perturbation direction during each iteration update.

2.3. Adversarial Attacks for Face Recognition

Currently, face recognition has widespread applications across various fields, and there have been numerous adversarial attack methods developed for face images. Existing face recognition attack methods are mainly categorized into two types: physical domain attacks and digital domain attacks. Physical attacks primarily target real-world scenarios, where adversaries convert adversarial perturbations into various wearable items. For example, Sharif et al. [30] proposed an adversarial eyeglass attack, where the eyeglass frame helps adversaries evade face recognition systems. Adv-hat [31] utilized a printed sticker with a pattern that is applied to a hat to attack face recognition models. Ibsen et al. [32] printed specially crafted facial images on T-shirts to confuse face recognition systems. Digital domain attacks involve directly modifying digital images using computer programs. These modifications can range from small pixel-level changes to complex image transformations. Rozsa et al. [33] selected a target face as the goal for adversarial face image samples and minimized the Euclidean distance between the target face and the adversarial sample to conduct targeted attacks. Dabouei et al. [34] altered the positional information of clean facial features to generate adversarial face images, thereby attacking face recognition systems. Dong et al. [35] introduced an evolutionary attack method in a decision-based black-box scenario, improving the efficiency of black-box attacks. Most of the aforementioned adversarial attack studies were aimed at single-face images, with the goal of causing the misidentification of individual faces. In this study, based on the UAP attack approach for natural images, we attempted to generate a UAP to alter face images across the entire dataset.

3. Methodology

3.1. Overview

Distinct from the perturbation task in image classification, facial UAPs aim to optimize perturbations so that the similarity between adversarial samples and clean samples exceeds a predefined threshold. The corresponding formula is as follows:

$$\mathrm{Similarity}_{\mathrm{x}\in\mathrm{X}}\{\mathrm{F}(\mathrm{x}+\mathrm{v}),\mathrm{F}(\mathrm{x})\} < \mathrm{t}, \tag{1}$$

where X denotes the distribution of images, $\mathrm{F}(\cdot)$ is defined as the target feature extractor, which outputs a feature vector $\mathrm{F}(\mathrm{x})$ for each input image x, and t denotes the designated threshold for discriminative similarity, while v represents the generated UAP. Our aim was to find a v for almost all data samples x sampled from the distribution X to deceive the neural network, which represents a fixed, image-independent perturbation that significantly alters the feature vectors extracted by the neural network from the original vectors, thus accomplishing the purpose of fooling the face recognition network.

As an open-set classification task, the interpretability analysis of face recognition focuses on explaining the similarity between embedding vectors rather than predicting categories. Due to the high similarity in appearance among different individuals' faces, the key aspect of face recognition lies in distinguishing the subtle differences within these highly overlapping features. When performing interpretability analysis, researchers focus on the specific impact of different face regions on the embedding vector extraction process. Therefore, in the context of UAP research for face recognition, the key lies in identifying and leveraging common features among faces of different identities in the dataset. The architecture diagram of KRT-FUAP is shown in Figure 2.

Existing methods for generating universal adversarial perturbations typically rely on directly training in the spatial domain globally. This approach depends entirely on the automatic learning process during training, leading to perturbations that may be somewhat lacking in both attack efficacy and stealth. In our approach, facial key regions are used as prior regions for training, allowing for a more precise identification of advantageous locations for embedding perturbations. This study focused on the key regions of the face, as the features of these regions play a crucial role in determining the similarity of the final embedding vectors. We utilized learnable flow field to fine-tune a mask of facial key regions. By finely adjusting the mask, we can generate perturbations with different weights in the spatial domain, ensuring that the perturbations are concentrated in the regions that have the greatest impact on face recognition. To balance the attack efficacy and stealth of the perturbations, we incorporate corresponding adversarial and stealth loss functions, jointly optimizing the final perturbations. This approach not only enhances the effectiveness of the attack but also increases the stealthiness of the perturbations, enabling attacks on face recognition systems without raising suspicion.

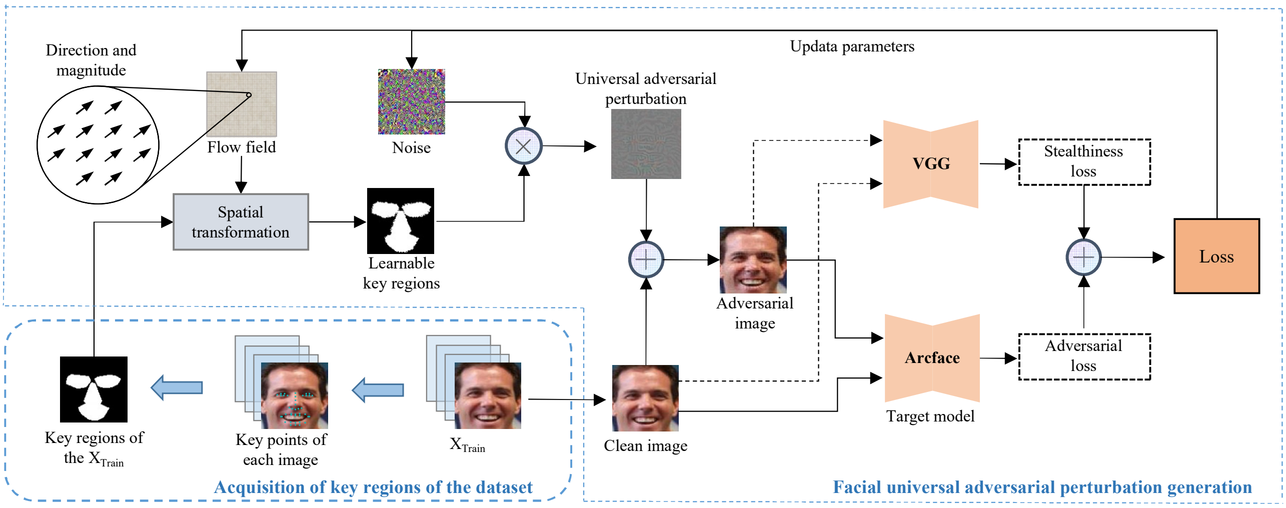


Figure 2. Architecture diagram of KRT-FUAP. First, keypoint detection is employed to acquire the positions of keypoints on the facial images. The convex hull algorithm is then utilized to obtain the key regions of these facial images, and the intersection of these key regions is taken to obtain the key regions mask tailored to the dataset. Subsequently, spatial transformation flow field and noise are initialized, and the flow field is utilized to control the spatial transformation of the key regions mask, thereby obtaining learnable key regions. Afterward, perturbation weights are adjusted based on the positions of these regions to obtain universal adversarial perturbation. The perturbation is superimposed onto clean images, and adversarial loss and stealthiness loss are computed separately using a target facial recognition model and a VGG model. The iteration continues until a certain criterion is fulfilled.

3.2. Facial Key Regions

Image classification is coarse-grained classification, where the differences between categories are substantial, and classification is achieved by focusing on and learning the distinct differences between these categories. In contrast, the face recognition task is fine-grained classification, characterized by minimal differences between identities, necessitating the differentiation of subtle variations between different faces. The structure of the face comprises multiple feature regions, each containing numerous unique characteristics, such as the shape of the eyes, the width of the nose, and the contour of the mouth. These features are crucial for distinguishing between different individuals. Existing deep learning-based methods rely on extracting various features from images, and semantic key regions provide highly discriminative features that enhance the discriminative power of feature vectors.

In the interpretability task of face recognition, Mery et al. [13] investigated the impact of different facial key regions on the extraction of face recognition embedding vectors by conducting regional occlusion on facial data. Their results show that different facial key regions exhibit regional characteristics in different dimensions of the embedding vector, implying that the facial key region features of different faces are the primary factors affecting face recognition accuracy.

To align with the characteristics of UAPs, it is necessary to identify common patterns across the entire dataset of facial images. Therefore, we focused on the facial key regions. Face recognition typically conducts alignment preprocessing on facial images, resulting in a consistent distribution of facial features in the dataset. These regions with fixed distributions of facial features are suitable for the application of UAP. To validate this idea, we conducted experiments using different masks and tested the model's accuracy, as shown in Figure 3. Detailed experimental procedures and results can be found in Section 4. The results indicate that the impact of key regions on face recognition accuracy is indeed more significant than that of non-key regions.

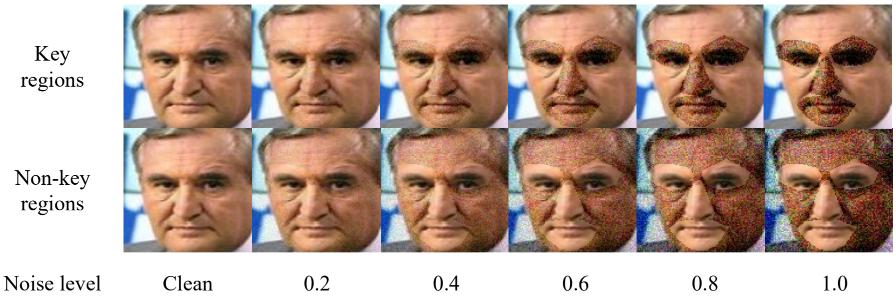


Figure 3. Visualization of noise superimposed on key and non-key regions.

The aligned face dataset $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}\}$ was collected, and 68 facial keypoints were extracted from small batches of data. The nose, eyes, and mouth regions are obtained based on the keypoint coordinates from the semantic regions. Mask regions are formed for each area using the convex hull algorithm, and the final mask M_i for each facial image is calculated as the overlay of these three regions. The equation for the corresponding ith face extraction mask M_i is as follows:

$$\mathrm{M}_i = \mathrm{H}_1(\mathrm{x}_i) + \mathrm{H}_2(\mathrm{x}_i) + \mathrm{H}_3(\mathrm{x}_i), \tag{2}$$

where $\mathrm{H}_1(\cdot)$, $\mathrm{H}_2(\cdot)$, and $\mathrm{H}_3(\cdot)$ denote the eyes, nose and mouth regions obtained by different convex hull algorithms. Because the mask regions obtained using different faces may not overlap, we calculate the intersection portion of the key regions of those face images as the feature region extracted from the entire face dataset, defined as the original key regions M_o. The formula is expressed as follows:

$$\mathrm{M}_\mathrm{o} = J\{\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_\mathrm{n}\}, \tag{3}$$

where $J\{\cdot\}$ denotes the intersection operation of masks, and the mask size matches the size of the aligned face. This study applied perturbations with different weights to both key and non-key facial regions in order to increase the effectiveness and stealthiness of the UAP.

3.3. Spatial Transformation via Learnable Flow Field

Spatial transformation adversarial samples were initially proposed by Xiao et al. [36], where small displacements were applied to input pixels to deceive the target neural network model. However, it is important to note that spatial transformations are only applicable to specific original clean samples. In the case of UAPw, there is no universal spatial transformation flow field that can be applied to the entire dataset. The key region mask is extracted from the entire face dataset. It serves as an indirect representation of the distribution of key regions across the dataset. We utilize a learnable spatial transformation flow field to induce minor positional variations in this key region mask, thereby indirectly achieving the purpose of generating UAP throughout the entire face dataset. The corresponding formula is as follows:

$$\mathrm{M}_\mathrm{f} = \mathrm{Flow}\left\{\hat{f}_{\mathrm{flow}}, \mathrm{M}_\mathrm{o}\right\}, \tag{4}$$

where $\mathrm{Flow}\{\cdot\}$ represents the spatial coordinate transformation function, $\hat{f}_{\mathrm{flow}}$ is the learned flow field, and $\mathrm{M_f}$ is the mask regions after spatial transformation.

We define the spatial coordinate transformation flow field as $f_{\mathrm{flow}} \in [-1,1]^{2\times h\times w}$, where h denotes the height of the image, and w denotes the width of the image; their sizes are consistent with the dimensions of the image, ensuring that each pixel in the image has a corresponding transformation rule. Specifically, $f^{(i)}_{\mathrm{flow}}$ represents the displacement rule for the *i*th pixel in the image, indicating the direction and magnitude of the change in its coordinates. Specifically, the coordinates of the *i*th pixel point in the original mask image can be represented as $\left(\mathrm{a}^{(i)},\mathrm{b}^{(i)}\right)$, and its corresponding position displacement in the flow field is $f^{(i)}_{\mathrm{flow}}$ = $\left(\Delta\mathrm{a}^{(i)},\Delta\mathrm{b}^{(i)}\right)$. $\left(\hat{\mathrm{a}}^{(i)},\hat{\mathrm{b}}^{(i)}\right)$ represents the coordinates of the *i*th pixel point in the transformed mask generated, and the relationship between its coordinates and those of the same position in the original image is as follows:

$$\left(\mathrm{a}^{(i)},\mathrm{b}^{(i)}\right) = \left(\hat{\mathrm{a}}^{(i)} + \Delta\mathrm{a}^{(i)}, \hat{\mathrm{b}}^{(i)} + \Delta\mathrm{b}^{(i)}\right). \tag{5}$$

As the computed displacement changes $\left(\Delta\mathrm{a}^{(i)},\Delta\mathrm{b}^{(i)}\right)$ may not be integers, and grid image coordinates only accept integers, this implies that the direct matching of pixel values at corresponding positions after transformation is not feasible. We employ bi-linear interpolation to compute pixel values at non-integer coordinates.

Given the *i*th pixel point $\mathrm{M_f}^{(i)}$ of the transformed image with coordinates $\left(\hat{\mathrm{a}}^{(i)},\hat{\mathrm{b}}^{(i)}\right)$. The coordinates of the four pixel points neighboring $\left(\mathrm{a}^{(i)},\mathrm{b}^{(i)}\right)$ are obtained using a rounding operation as $\left(\lfloor\mathrm{a}^{(i)}\rfloor,\lfloor\mathrm{b}^{(i)}\rfloor\right)$, $\left(\lfloor\mathrm{a}^{(i)}\rfloor+1,\lfloor\mathrm{b}^{(i)}\rfloor\right)$, $\left(\lfloor\mathrm{a}^{(i)}\rfloor,\lfloor\mathrm{b}^{(i)}\rfloor+1\right)$, and $\left(\lfloor\mathrm{a}^{(i)}\rfloor+1,\lfloor\mathrm{b}^{(i)}\rfloor+1\right)$, and the set of pixel points corresponding to the above coordinates is denoted by $N\left(\mathrm{a}^{(i)},\mathrm{b}^{(i)}\right)$. Based on the pixel values of each point in the set of domains, bi-linear interpolation is used to update all the pixel values in the image to obtain the changed key region mask. The interpolation formula is as follows:

$$\mathrm{M_f}^{(i)} = \sum_{j\in N\left(\mathrm{a}^{(i)},\mathrm{b}^{(i)}\right)} \mathrm{M_o}^{(j)}\left(1-\left|\mathrm{a}^{(i)}-\mathrm{a}^{(j)}\right|\right)\left(1-\left|\mathrm{b}^{(i)}-\mathrm{b}^{(j)}\right|\right). \tag{6}$$

Compared to the initially extracted mask positions, the final mask obtained considers more factors that fit the dataset and enhance stealthiness. This method exhibits a better performance for subsequently generated facial UAPs.

3.4. Generation of UAPs

We use the key regions mask $\mathrm{M_f}$ obtained in the previous section and learnable noise n to generate the final universal adversarial perturbation v. To enhance the stealthiness of the perturbation, we set the intensity of the superimposed perturbation in the non-key regions to be half of the intensity in the key regions as a way to ensure that the effective noise is concentrated in the key regions. The final generated adversarial sample $\mathrm{x_{adv}}$ is formulated as follows:

$$\mathrm{x_{adv}} = \mathrm{x} + \mathrm{v} = \mathrm{x} + \mathrm{Mask}\{\hat{\mathrm{n}},\mathrm{M_f}\}, \tag{7}$$

where v denotes the generated universal adversarial perturbation, and $\hat{\mathrm{n}}$ is the learned noise. $\mathrm{Mask}\{\cdot\}$ denotes the function that generates the perturbation using the mask, we normalize the mask to be within the interval $[1/2,1]$, thereby ensuring that the noise intensity in the non-key regions is half of that in the key regions. The whole optimization process of our scheme is shown in the following equation:

$$\left(\hat{\mathrm{n}},\hat{f}_{\mathrm{flow}}\right) = \arg\min_{\mathrm{n},f_{\mathrm{flow}}}\left[\mathcal{L}_{\mathrm{adv}}(\mathrm{x}+\mathrm{v},\mathrm{x}) + \lambda\mathcal{L}_{\mathrm{ste}}(\mathrm{x}+\mathrm{v},\mathrm{x})\right] \text{ s.t. } \|\mathrm{v}\|_{\mathrm{p}} \le \xi\,, \tag{8}$$

where n and f_{flow} denote the learnable noise and flow field. $\mathcal{L}_{\text{adv}}$ denotes the adversarial loss, $\mathcal{L}_{\text{ste}}$ denotes the stealthiness loss, and λ controls the balance between them, and ξ is a parameter controlling the size of the perturbation. Adversarial examples are generated using the final learned noise $\hat{\text{n}}$ and flow field $\hat{f}_{\text{flow}}$ to deceive the entire face recognition model. The fooling rate δ denotes the probability of a successful attack using universal adversarial perturbations. The pipeline of KRT-FUAP is provided in Algorithm 1.

Algorithm 1 The Algorithm of KRT-FUAP

```
Input: Preprocessed training set X, random noise n, face recognition network $F(\cdot)$, fooling
    rate $\delta$, $l_\infty$-norm $\xi$ of perturbation, and decision threshold t
Output: Universal adversarial perturbation v and learnable flow field $f_{\text{flow}}$.
Initialize $(\text{n}, f_{\text{flow}}) \leftarrow random$
Obtain the original mask $\text{M}_\text{o}$ from X
while fooling rate $< \delta$ do
    for $\text{x}_i$ in $X$ do
        Spatial transformation: $\text{M}_\text{f} \leftarrow \text{Flow}\{f_{\text{flow}}, \text{M}_\text{o}\}$
        Obtain $\text{v} \leftarrow \text{Mask}\{\text{n}, \text{M}_\text{f}\}$
        if Similarity$\{F(\text{x}_i + \text{v}), F(\text{x}_i)\} > \text{t}$ then
            $(\Delta\text{n}, \Delta f_{\text{flow}}) \leftarrow \arg\min_{(\text{n}, f_{\text{flow}})} \|(\text{n}, f_{\text{flow}})\|_2$
            s.t. Similarity$\{F(\text{x}_i + \text{v}), F(\text{x}_i)\} \leqslant \text{t}$
            Update the noise: $\text{n} \leftarrow \text{n} + \Delta\text{n}$
            Update the flow field: $f_{\text{flow}} \leftarrow f_{\text{flow}} + \Delta f_{\text{flow}}$
        end if
        Clip v to maintain the $l_\infty$ norm restriction
    end for
end while
return v
```

3.5. Loss Setting

3.5.1. Adversarial Loss

The purpose of the adversarial loss is to regulate the attack performance of adversarial perturbations. In image classification tasks, adversarial attacks are typically conducted using cross-entropy loss with respect to given labels. However, in face recognition tasks, adversarial loss is often computed based on cosine similarity or Euclidean distance to control the dissimilarity between clean samples and adversarial samples. Our approach circumvents intermediate values, directly optimizing perturbations by controlling the directions of feature vectors in the feature space between original and adversarial samples. In detail, we displace the direction of the adversarial sample's feature vector away from the direction of the original sample's feature vector in order to move the adversarial sample in the direction opposite to the clean image. The formula is as follows:

$$\mathcal{L}_{\text{adv}} = \sum_{\text{x}^{(i)} \in \text{D}} \left(\frac{\text{F}\left(\text{x}^{(\text{i})}\right)}{\left\|\text{F}\left(\text{x}^{(\text{i})}\right)\right\|_2} + \frac{\text{F}\left(\text{x}_{\text{adv}}^{(\text{i})}\right)}{\left\|\text{F}\left(\text{x}_{\text{adv}}^{(\text{i})}\right)\right\|_2} \right)^2, \tag{9}$$

where D refers to the training example set, i denotes the number of the image in the dataset, and $\text{F}\left(\text{x}^{(\text{i})}\right)$ and $\text{F}\left(\text{x}_{\text{adv}}^{(\text{i})}\right)$ denote the feature vectors extracted from the target model. In order to make the loss function become small, the feature direction of the adversarial samples will be limited to the opposite direction of the clean samples.

We selected the optimization direction for the adversarial sample's feature vector; this is equivalent to having a cosine similarity calculation value of -1 between this direction and the original direction, which represents the lowest score in a cosine similarity measurement. By constraining this direction, we effectively circumvent the calculation of the cosine similarity and directly displace the adversarial sample's feature vector in the opposite direction, thus achieving the purpose of adversarial attacks.

3.5.2. Stealthiness Loss

The purpose of stealthiness loss is to control the invisibility of the generated adversarial perturbations. The VGG [14] network is a well-known deep convolutional neural network architecture that is widely used in computer vision tasks such as image recognition. The network's core idea is to extract hierarchical image features by stacking multiple convolutional and pooling layers. It utilizes smaller convolutional kernels and deeper layers to increase the depth of the network, enabling it to better capture details and local features in images, thereby improving the accuracy of image classification. The network's shallow convolutional and pooling layers are mainly responsible for extracting low-level features, such as edges and textures. These features are highly sensitive to the local structure and details of the image, but relatively weak in representing overall semantics and high-level features. The deeper convolutional and pooling layers gradually increase the network's depth to extract higher-level semantic features. By stacking multiple convolutional and pooling layers, the network can gradually expand its receptive field and learn more abstract feature representations. These higher-level features can capture more global image information.

This study utilized the shallow outputs of the VGG network to capture low-level features in both clean images and adversarial samples, thereby computing the VGG loss to control the invisibility of UAPs. Specifically, the adversarial samples and clean images are fed separately into the VGG network, extracting only the shallow-level information to obtain low-level features. By comparing the differences in feature extraction between the two, we can derive the invisibility loss, which controls the visibility of the generated perturbations. The definition of stealthiness loss is shown in the following equation:

$$\mathcal{L}_{\text{ste}} = \sum_{\mathrm{x}^{(i)} \in \mathrm{D}} \left(\left\| \varphi_j\left(\mathrm{x}^{(i)}\right), \varphi_j\left(\mathrm{x}_{\text{adv}}^{(i)}\right) \right\|_2 \right), \tag{10}$$

where φ_j is the feature map of the jth layer of the VGG network. The imperceptibility of the adversarial samples is improved by reducing the difference between the VGG shallow features of both the adversarial samples and the clean images.

4. Experimental Results

In this study, we conducted comprehensive experiments to validate the effectiveness of the proposed method. In this section, we provide an overview of the experimental setup. Then, we examine the impact of key regions on the face recognition accuracy and compare our proposed method with existing universal adversarial perturbation methods for natural image classification tasks. The results indicate that among different approaches, our method exhibits good stealthiness while possessing a certain level of attack effectiveness. Also, we conducted black-box testing, demonstrating a certain success rate of the method even in black-box scenarios. Finally, we performed various ablation experiments to assess the influence of different factors on the proposed method.

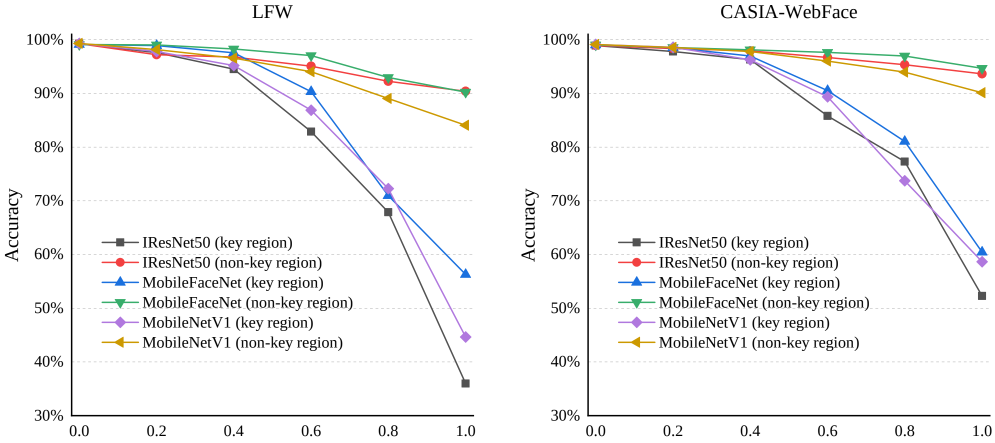
4.1. Experimental Setup

The experiments were accelerated using a single GTX TITAN XP GPU (12 GB) in the PyTorch 1.12.0 framework. We employed the LFW [37] and CASIA-WebFace [38] datasets for training facial UAPs. LFW and CASIA-WebFace are widely used datasets in the field of face recognition, encompassing a diverse range of facial poses, expressions, lighting conditions, and occlusions, thereby authentically reflecting the challenges of face recognition in various scenarios. LFW serves as a common benchmark dataset for face verification, extensively employed to evaluate the performances of facial feature extraction and matching algorithms. CASIA-WebFace contains a larger scale of data and is primarily used to aid models in learning facial feature representations. The research task of this study focused on generating universal adversarial perturbations for face recognition. By conducting experiments using these two datasets, we could obtain more objective and effective evaluation results. Both datasets contain paired images of the same individuals. The training set contains 6000 pairs of facial images, while the test set contains 3000 pairs of facial images. These facial images were resized to 112 × 112. We used Arcface to pre-train three backbone feature extraction networks: IResNet50 [39], MobileFaceNet [40], and MobileNetV1 [17]. In order to make the adversarial perturbation invisible to the human eye, the perturbation intensity ξ was set to 0.08. Considering the computational capabilities of our hardware resources and the training time for the task, we set a batch size of 10 pairs of images and a learning rate of 0.01 to achieve optimal perturbation generation results. By empirically fine-tuning the weight parameter of the loss function, we found that setting λ to 0.05 effectively balances the attack ability and stealth. For the evaluation on different datasets and backbone networks, we employed two objective stealthiness metrics, namely, the Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR).

4.2. Effect of Different Regions on Recognition Accuracy

We utilized the LFW and CASIA-WebFace datasets to examine the trend in the recognition accuracy of target models when occluding key and non-key regions on the IResNet50, MobileFaceNet, and MobileNetV1 backbone extraction networks. By controlling the level of random noise used for occlusion, we ascertained the extent of influence of different noise intensities on various regions. The results are depicted in Figure 4.

We randomly selected 2000 pairs of facial images from each dataset. Gaussian noise of varying levels ranging from 0 to 1 was overlaid on different regions of one of the paired images. In increasing the noise level, the results of testing on the three backbone networks using both datasets demonstrate that overlaying noise on facial key regions leads to a more substantial decrease in the final recognition accuracy. This suggests that features in facial key regions carry greater weight in face recognition discrimination.



(a) Noise level (b) Noise level

Figure 4. (**a**,**b**), respectively, depict the variation trends of the face recognition accuracy tested on three backbone extraction networks using the LFW dataset and the CASIA-WebFace dataset. It can be observed from the figure that as the noise level increases, there is a certain decrease in recognition accuracy. Moreover, under the experimental condition of overlaying noise on key regions, the rate of decrease is greater.

4.3. Comparison Experiment

We trained our proposed KRT-FUAP using the LFW and CASIA-WebFace datasets on the two backbone extraction networks: IResNet50 and MobileFaceNet. Our proposed KRT-FUAP achieved approximately an 80% attack success rate on various test sets. To evaluate the perceptual quality of the adversarial perturbations, in addition to subjective human observation, we employed two objective stealthiness quality metrics: the SSIM and PSNR. Higher scores on both metrics indicate better image quality. Applying these perceptual metrics to the adversarial samples helps quantify the stealthiness performance of the generated facial UAPs. The results ultimately demonstrate that when overlaying universal perturbations onto facial images, they exhibit good imperceptibility. To underscore the efficacy of our meticulously generated perturbations, we conducted perturbation overlay experiments using randomly generated noise and measured the fooling rate as well as objective perceptual metrics.

Given the lack of UAP schemes specifically tailored for face recognition, in our comparative analysis, we compared our proposed method KRT-FUAP with existing solutions in other fields, including UAP [10], FG-UAP [41], and FTGAP [42], which are used for natural and texture images. Although these methods are initially designed for other tasks, we modified them to suit the face recognition task in this study. The relevant experimental results are shown in Table 1. We used the fooling rate (FR) to measure the attack performance of universal adversarial perturbation generated using different methods, while the SSIM and PSNR were used to assess their stealthiness. It can be observed that randomly generated noise performs much worse in terms of both attack effectiveness and imperceptibility compared to other carefully designed perturbations. Furthermore, UAP and FG-UAP are both spatial domain-based perturbation generation methods, and both use lp-norm constraints for imperceptibility. On the other hand, FTGAP considers the frequency domain and directly limits the strength of the perturbation in the frequency domain, hence yielding higher imperceptibility metrics in comparison.

Table 1. Comparison results of experimental fooling rate and objective evaluation parameters.

Dataset	Backbone	Method	FR ↑	SSIM ↑	PSNR ↑
LFW	IResNet50	Random	20.8%	0.4003	18.9271
		UAP [10]	76.6%	0.7607	27.5761
		FG-UAP [41]	80.4%	0.8678	30.6159
		FTGAP [42]	82.1%	0.8852	31.7033
		KRT-FUAP	81.9%	0.9304	34.0157
	MobileFaceNet	Random	23.2%	0.3614	18.0833
		UAP [10]	79.1%	0.8487	29.7213
		FG-UAP [41]	78.8%	0.8439	29.1161
		FTGAP [42]	79.4%	0.8507	30.5232
		KRT-FUAP	80.1%	0.9044	32.9812
CASIA-WebFace	IResNet50	Random	37.1%	0.4901	20.8767
		UAP [10]	71.2%	0.7759	28.0403
		FG-UAP [41]	74.7%	0.8126	29.0967
		FTGAP [42]	76.3%	0.8544	30.4638
		KRT-FUAP	78.4%	0.9172	32.6624
	MobileFaceNet	Random	30.3%	0.5099	21.4036
		UAP [10]	76.3%	0.7612	27.5446
		FG-UAP [41]	77.8%	0.7865	28.3971
		FTGAP [42]	78.4%	0.8691	30.6668
		KRT-FUAP	80.2%	0.8817	32.1412

Our proposed KRT-FUAP not only meticulously generates the overlaying of perturbations on facial key regions but also employs effective imperceptibility loss to control the final perturbation. This allows us to achieve a significant improvement in imperceptibility while maintaining comparable attack effectiveness. Corresponding visualizations are shown in Figure 5. Compared to several other methods, our adversarial samples exhibit superior imperceptibility, appearing more normal.

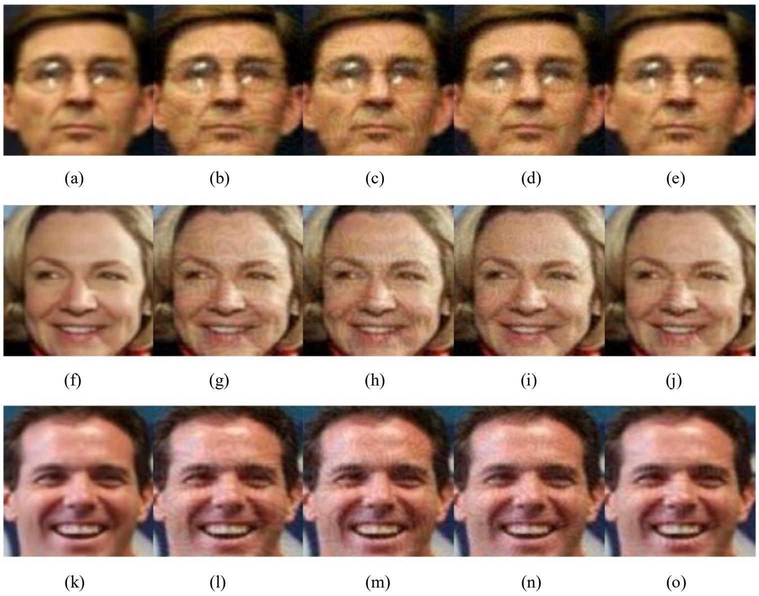


Figure 5. Some examples for the adversarial image: (**a**,**f**,**k**) are clean images; (**b**,**g**,**l**) were generated using UAP; (**c**,**h**,**m**) were generated using FG-UAP; (**d**,**i**,**n**) were generated using FTGAP; and (**e**,**j**,**o**) were generated using KRT-FUAP.

We compared the KRT-FUAP method with several existing approaches in terms of the time required to generate perturbations. Using the LFW dataset, we tested the time taken by different methods to achieve various fooling rates on the two backbone networks, IResNet50 and MobileFaceNet, under conditions with which the devices do not perform any other computational tasks. The fooling rates were set to several fixed values above 50%, and the corresponding comparison results are illustrated in Figure 6.

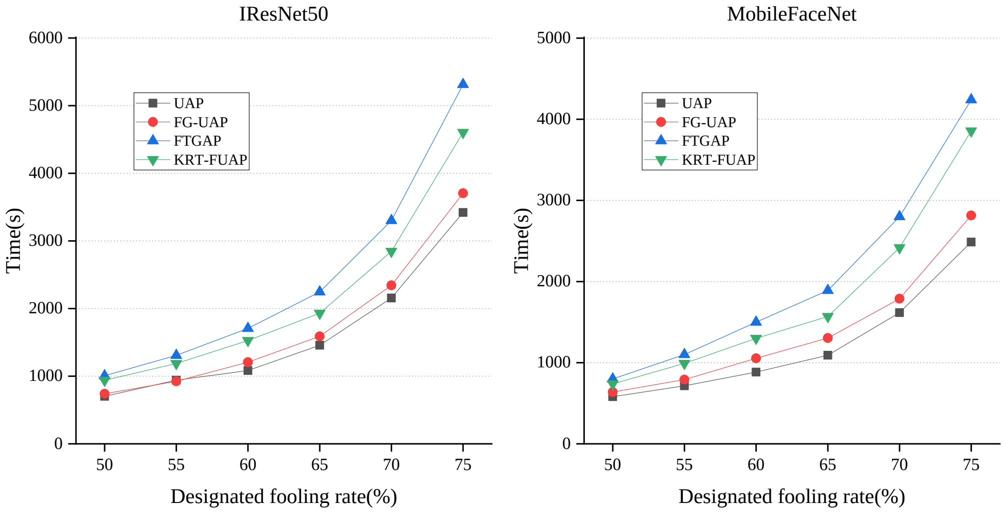


Figure 6. The time required to generate universal adversarial perturbations to achieve the designated fooling rates on IResNet50 and MobileFaceNet using four different approaches.

The UAP and FG-UAP methods both conduct adversarial perturbation training directly in the spatial domain, relying solely on norm constraints for stealthiness. Their algorithmic complexity is relatively low, resulting in shorter training times and lower hardware resource usage. FTGAP introduces a frequency domain perspective by transforming spatial domain images to the frequency domain for perturbation training and intensity restriction. Subsequently, it converts them back to the spatial domain to obtain the final universal adversarial perturbation. This method involves transformations between the spatial and frequency domains, resulting in higher algorithmic complexity. Consequently, it requires more time to achieve different fooling rates and utilizes the most hardware resources. In comparison, our proposed KRT-FUAP method operates solely in the spatial domain by generating adversarial perturbations in key semantic regions of the face and using a learnable flow field to fine-tune these regions. This approach produces universal adversarial perturbations that are better suited for facial images. Additionally, our loss function includes both stealthiness and adversarial loss components, providing dual control over the stealthiness and effectiveness of the generated perturbations. Consequently, our algorithm has higher computational complexity compared to UAP and FG-UAP, resulting in longer training times and greater hardware resource usage. This corroborates the results of our experiments.

Unlike perturbations for specific images, which require retraining for each new image, universal adversarial perturbations only need to be trained once and can then be applied to the entire dataset. This is a significant advantage of universal adversarial perturbations. Our proposed KRT-FUAP method, while incurring a slight increase in training cost, significantly enhances the aggressiveness and stealthiness of the perturbation, thereby demonstrating the effectiveness of our approach.

4.4. Black-Box Performance

We also conducted black-box testing on KRT-FUAP, as shown in Table 2. This experiment evaluated the attack performance of three different backbone networks on the LFW dataset. The data on the diagonal represent the success rate of white-box attacks, while the other values indicate the success rate of black-box attacks. From the data in the table, we observe that under black-box conditions, the success rates of universal adversarial perturbations vary to some extent. However, since this method extracts facial key regions from the dataset, it still demonstrates some effectiveness in black-box scenarios. Nevertheless, achieving black-box attacks for universal adversarial perturbations in the field of face recognition remains highly challenging.

Table 2. White-box and black-box performances on the LFW dataset.

FR	IResNet50	MobileFaceNet	MobileNetV1
IResNet50	81.9%	26.8%	25.4%
MobileFaceNet	54.6%	80.1%	41.3%
MobileNetV1	45.2%	33.5%	79.7%

4.5. Ablation Study

4.5.1. Impact of the Facial Key Regions Mask

The initial facial key regions masks in this experiment were extracted from the training dataset. Based on our previous experimental results, features contained within the facial key regions have a greater impact on the accuracy of face recognition models. Therefore, we randomly selected 1000 facial images from the dataset, extracted the key point coordinates for each facial image, applied the convex hull algorithm to compute the positions of key regions, and finally obtained the intersection to derive the key region masks specific to the dataset. To demonstrate the effectiveness of this approach, we directly optimized the global universal adversarial perturbations and evaluated the fooling rates and stealthiness objective metrics obtained using the IResNet50 and MobileFaceNet backbone extraction networks under the LFW dataset. As shown in Table 3, the corresponding results indicate that in controlling the perturbation weights using key regions, the intensity of perturbations can be better distributed, allowing noise to focus more on attacking key regions, thereby reducing noise intensity in non-key regions and promoting the enhancement of perturbation concealment.

Table 3. Fooling rate and objective stealthiness metrics for key regions and global regions.

Dataset	Backbone	Method	FR ↑	SSIM ↑	PSNR ↑
LFW	IResNet50	KRT-FUAP	81.9%	0.9304	34.0157
		Global regions	77.8%	0.8926	31.6674
	MobileFaceNet	KRT-FUAP	80.1%	0.9044	32.9812
		Global regions	77.7%	0.8639	30.5034

4.5.2. Impact of the Learnable Flow Field

The facial key region mask ultimately utilized in this experiment was a learnable region mask controlled by a learnable flow field. Through a set loss, we continuously modified the parameters within the learnable flow field, thus continuously adapting the fixed key region mask extracted from the dataset. This approach allows our mask to not be limited to facial key regions extracted from the entire dataset, enhancing the generalization performance of the generated perturbations through iterative learning. To demonstrate the role of the learnable flow field, we conducted experiments using fixed key regions, testing the fooling rates and objective stealthiness metrics under IResNet50 and MobileFaceNet. As shown in Table 4, the final experimental results indicate that employing a learnable flow field enables the consideration of more relevant information, because it not only considers the unified characteristics of the dataset but also incorporates individual features from different training data, resulting in more effective attack and the concealment of universal adversarial perturbations.

Table 4. Fooling rate and objective stealthiness metrics with and without learnable flow field.

Dataset	Backbone	Method	FR ↑	SSIM ↑	PSNR ↑
LFW	IResNet50	KRT-FUAP	81.9%	0.9304	34.0157
		Fixed key regions	80.2%	0.9147	33.1845
	MobileFaceNet	KRT-FUAP	80.1%	0.9044	32.9812
		Fixed key regions	78.9%	0.8916	32.1108

4.5.3. Impact of Adversarial and Stealthiness Loss

The loss function in this experiment consisted of two components: adversarial loss and stealthiness loss. The adversarial loss is achieved by directly selecting the optimization direction of the adversarial samples, set to be opposite for better attack effectiveness. Simultaneously, the VGG network was utilized to control stealthiness metrics, enhancing the imperceptibility of images. To demonstrate the effectiveness of our loss function settings, we compared them with commonly used loss functions in face recognition, such as cosine similarity loss and Euclidean distance loss. The other settings remain unchanged; the cosine similarity loss reduces the similarity between features, while the Euclidean distance loss increases the distance between features. We tested the fooling rates and stealthiness metrics on IResNet50 and MobileFaceNet. As shown in Table 5, the corresponding experimental results show that in simultaneously controlling adversarial effectiveness and stealthiness in the loss function, the two metrics of the generated universal adversarial perturbations exhibit superior performances.

Table 5. Fooling rate and objective stealthiness metrics for different loss setting.

Dataset	Backbone	Method	FR ↑	SSIM ↑	PSNR ↑
LFW	IResNet50	KRT-FUAP	81.9%	0.9304	34.0157
		Euclidean distance	80.2%	0.8875	31.7418
		Cosine similarity	80.4%	0.8963	31.9427
	MobileFaceNet	KRT-FUAP	80.1%	0.9044	32.9812
		Euclidean distance	78.9%	0.8622	30.8143
		Cosine similarity	79.1%	0.8657	30.8807

5. Discussion

Compared to adversarial perturbations targeting specific faces, universal adversarial perturbations offer the advantage of being generated once and applied multiple times, yet they pose greater challenges. Adversarial perturbations targeting specific faces only necessitate identifying features of individual images and modifying these features through perturbations to deceive the model. In contrast, universal adversarial perturbations require investigating the underlying patterns across an entire dataset. In leveraging these patterns, perturbations are applied to modify relevant features across the entire facial dataset, resulting in high rates of model deception in this dataset for face recognition.

Our search for the key regions of the face is in line with the property that the universal adversarial perturbation targets the entire dataset, considering that the semantic regions of the face after alignment are at some fixed locations, and these semantic regions have a decisive impact on the accuracy of face recognition. UAP and FG-UAP generate universal adversarial perturbations globally. Although FTGAP considers frequency domain information to enhance stealthiness, the perturbation generated ultimately remains within the spatial domain globally. We focused on the perturbation within these key regions, overlaying less perturbation on other regions. The results show that there is a better effect after considering the local information.

The fixed mask region extracted from the dataset constitutes a unified characteristic of the dataset, representing the distribution of facial key regions across the entire dataset while disregarding certain individual features. By incorporating a variable flow field, we fine-tune this fixed region, continuously updating it during training to accommodate individual features. This approach aims to integrate individual features from the training data onto the unified features, thereby broadening the scope of the considered information. As the perturbations primarily affect key regions of aligned faces, there may be some efficacy in black-box attacks; however, the true black-box scenario remains unpredictable, limiting its effectiveness across all scenarios. Our research can serve as both an attack method for face recognition systems and a means of protecting facial privacy.

6. Conclusions

In this paper, we propose KRT-FUAP, a facial universal adversarial perturbation generation approach that utilizes a learnable flow field to fine-tune key regions. This study examined the vulnerability of face recognition system to universal adversarial perturbations and evaluated the influence of key regions on the accuracy of the system. A key region mask is extracted from the dataset and fine-tuned using a learnable flow field as a dimension to modify universal adversarial perturbations in the spatial domain, resulting in the generetion of adversarial samples for faces. Additionally, we propose a scheme that balances the adversarial effectiveness and stealthiness of perturbation by incorporating adversarial loss and stealthiness loss. Experimental results indicate that our proposed method achieves a fooling rate of approximately 80% across different datasets and backbone networks. The stealthiness of our method shows a significant advantage both in the visualization of adversarial examples and in objective stealth evaluation metrics. Specifically, on the IResNet50 backbone network, the perturbations we generated achieved a SSIM value of 0.9304 and a PSNR value of 34.0157. Although our method has a relatively higher complexity and requires a longer training time compared to existing adversarial perturbation generation algorithms, it is acceptable to obtain better aggressiveness and stealthiness by occupying slightly more hardware resources.

Our proposed approach also has several limitations. Although it demonstrates certain black-box attack capabilities, its attack efficacy and stealth metrics may be affected in more complex black-box scenarios. The KRT-FUAP approach proposed in this paper involves additive perturbations, generating adversarial examples by superimposing universal perturbations at selected effective locations in the spatial domain of facial images. Consequently, the effectiveness of the perturbation depends on this superimposition operation. This method has certain limitations when applied to side facial images, as the key regions in side facial images are significantly different from those in front facial images. Research on perturbations targeting profile images presents substantial challenges.

Adversarial attack is the opposite of adversarial defense. Designing universal adversarial attack strategies targeting face recognition technology holds significant guiding importance for enhancing model robustness and developing effective defense mechanisms. Additionally, because adversarial perturbations can disrupt the feature representations extracted using face recognition models, this provides a potential approach for protecting facial privacy. Future research can explore other perspectives, and we hope our work will inspire further innovative studies.

Author Contributions: Conceptualization, X.J., Y.L. and H.W.; methodology, X.J., Y.L. and G.S.; software, X.J.; validation, G.S., Y.C., Z.D. and H.W.; writing—original draft preparation, X.J. and Y.L.; writing—review and editing, G.S., Y.C., Z.D. and H.W.; supervision, H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China (NSFC) under grant U23B2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The datasets can be found at http://vis-www.cs.umass.edu/lfw (accessed on 1 March 2024) and https://paperswithcode.com/dataset/casia-webface (accessed on 1 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.R. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
2. Malik, M.; Malik, M.K.; Mehmood, K.; Makhdoom, I. Automatic speech recognition: A survey. *Multimed. Tools Appl.* **2021**, *80*, 9411–9457. [CrossRef]
3. Lauriola, I.; Lavelli, A.; Aiolli, F. An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing* **2022**, *470*, 443–456. [CrossRef]
4. Maurício, J.; Domingues, I.; Bernardino, J. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Appl. Sci.* **2023**, *13*, 5521. [CrossRef]
5. Taskiran, M.; Kahraman, N.; Erdem, C.E. Face recognition: Past, present and future (a review). *Digit. Signal Process.* **2020**, *106*, 102809. [CrossRef]
6. Yerlikaya, F.A.; Bahtiyar, Ş. Data poisoning attacks against machine learning algorithms. *Expert Syst. Appl.* **2022**, *208*, 118101. [CrossRef]
7. Li, Y.; Jiang, Y.; Li, Z.; Xia, S.T. Backdoor learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 5–22. [CrossRef]
8. Akhtar, N.; Mian, A.; Kardan, N.; Shah, M. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access* **2021**, *9*, 155161–155196. [CrossRef]
9. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
10. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1765–1773.

11. Parmar, R.; Kuribayashi, M.; Takiwaki, H.; Raval, M.S. On fooling facial recognition systems using adversarial patches. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
12. Hu, S.; Liu, X.; Zhang, Y.; Li, M.; Zhang, L.Y.; Jin, H.; Wu, L. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 15014–15023.
13. Mery, D. True black-box explanation in facial analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1596–1605.
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
18. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
19. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VII 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 499–515.
20. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
21. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
22. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 99–112.
23. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2574–2582.
24. Mopuri, K.R.; Ojha, U.; Garg, U.; Babu, R.V. Nag: Network for adversary generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 742–751.
25. Poursaeed, O.; Katsman, I.; Gao, B.; Belongie, S. Generative adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4422–4431.
26. Mopuri, K.R.; Garg, U.; Babu, R.V. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv* **2017**, arXiv:1707.05572.
27. Mopuri, K.R.; Ganeshan, A.; Babu, R.V. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2452–2465. [CrossRef] [PubMed]
28. Zhang, C.; Benz, P.; Imtiaz, T.; Kweon, I.S. Understanding adversarial examples from the mutual influence of images and perturbations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 14521–14530.
29. Dai, J.; Shu, L. Fast-uap: An algorithm for expediting universal adversarial perturbation generation using the orientations of perturbation vectors. *Neurocomputing* **2021**, *422*, 109–117. [CrossRef]
30. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 Acm Sigsac Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1528–1540.
31. Komkov, S.; Petiushko, A. Advhat: Real-world adversarial attack on arcface face id system. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 819–826.
32. Ibsen, M.; Rathgeb, C.; Brechtel, F.; Klepp, R.; Pöppelmann, K.; George, A.; Marcel, S.; Busch, C. Attacking Face Recognition with T-shirts: Database, Vulnerability Assessment and Detection. *IEEE Access* **2023**, *11*, 57867–57879. [CrossRef]
33. Rozsa, A.; Günther, M.; Boult, T.E. LOTS about attacking deep features. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; pp. 168–176.
34. Dabouei, A.; Soleymani, S.; Dawson, J.; Nasrabadi, N. Fast geometrically-perturbed adversarial faces. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1979–1988.
35. Dong, Y.; Su, H.; Wu, B.; Li, Z.; Liu, W.; Zhang, T.; Zhu, J. Efficient decision-based black-box adversarial attacks on face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7714–7722.
36. Xiao, C.; Zhu, J.Y.; Li, B.; He, W.; Liu, M.; Song, D. Spatially transformed adversarial examples. *arXiv* **2018**, arXiv:1801.02612.

37. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, 17–20 October 2008 .
38. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Learning face representation from scratch. *arXiv* **2014**, arXiv:1411.7923.
39. Duta, I.C.; Liu, L.; Zhu, F.; Shao, L. Improved residual networks for image and video recognition. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 9415–9422.
40. Chen, S.; Liu, Y.; Gao, X.; Han, Z. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In Proceedings of the Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, 11–12 August 2018; Proceedings 13; Springer: Berlin/Heidelberg, Germany, 2018; pp. 428–438.
41. Ye, Z.; Cheng, X.; Huang, X. Fg-uap: Feature-gathering universal adversarial perturbation. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–8.
42. Deng, Y.; Karam, L.J. Frequency-tuned universal adversarial perturbations. In Proceedings of the Computer Vision–ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; Proceedings, Part V 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 494–510.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.