Adaptive Adversarial Patch Attack on Face Recognition Models

Bei Yan[1,2] Jie Zhang[1,2] Zheng Yuan[1,2] Shiguang Shan[1,2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
yanbei19@mails.ucas.ac.cn zheng.yuan@vipl.ict.ac.cn {zhangjie,sgshan}@ict.ac.cn

2023 IEEE International Joint Conference on Biometrics (IJCB) | 979-8-3503-3726-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCB57857.2023.10449094

Abstract

Face recognition models have become widely used for identity authentication in scenarios such as cell phone unlocking and financial payment, but they are vulnerable to adversarial examples. Due to the realizability in the physical world, adversarial patch attack has emerged as a significant security threat. However, most existing adversarial patch attack methods focus on only one aspect of patch generation, such as patch location or shape. To overcome this limitation, we propose a novel unified Adaptive Adversarial Patch (AAP) attack framework for targeted attack on face recognition models. Our method comprehensively considers various factors during patch generation, including location, shape, and number. Our approach adaptively selects patch location and number based on saliency map and clustering, while simultaneously deforming patch shape and optimizing perturbations. Extensive experiments under both white-box and black-box settings demonstrate that our proposed method achieves higher attack success rates compared to SOTA methods.

1. Introduction

Deep neural networks (DNNs) have shown superior performance in various computer vision tasks, *e.g.*, image classification [17, 39], object detection [4, 32], face recognition [10, 33, 41, 46]. Due to the excellent performance, face recognition models based on DNNs have become widely used for identity authentication in scenarios such as cell phone unlocking and financial payment. However, Szegedy *et al.* [40] point out that neural network-based systems are vulnerable to adversarial examples, which raises significant concerns about the security of face recognition models.

Adversarial examples [14, 40] fool the neural network by adding small perturbations to every pixel of the input image, which can mislead the network to produce wrong output. L_p-norm constraints are usually used to limit the magnitude of the perturbations, making them imperceptible to humans. But such global perturbations are only practical in the digital world and are not feasible in the physical world. Thus, Brown *et al.* propose the concept of adversarial patch [5], which constrains the region of the perturbations rather than magnitude, only allowing the perturbations to be added in a small area. For example, Eykholt *et al.* [13] have successfully deceived real-world autonomous driving systems by sticking elaborately crafted patches to the stop signs.

Most previous works on adversarial patch attack concentrate only on one aspect of patch generation. For example, TAP [52] focuses on the pattern of patch, extending transfer-based attack to generate robust perturbations in a fixed eye area. Several existing works explore the effect of patch location. ROA [50] uses a gray rectangular patch to search for the most sensitive region of the input image. LO [37] generates patches by alternately optimizing location and perturbations. Additionally, DAP [9] investigates the influence of patch shape.

To overcome this limitation, we propose a novel unified Adaptive Adversarial Patch (AAP) attack framework for targeted attack on face recognition models. Unlike existing works, we comprehensively consider various factors of patch generation, including patch location, shape, and number. An overview of AAP is shown in Fig. 1. Our main contributions are concluded as follows:

1. Our approach adaptively selects patch location and number based on saliency map and K-Means clustering, while simultaneously deforming patch shape and optimizing perturbations.

2. Extensive experiments under white-box and black-box settings demonstrate that our proposed method achieves higher attack success rates compared to SOTA methods.

2. Related Work

2.1. Adversarial Attack

The concept of adversarial example was first introduced by Szegedy *et al.* [40], which could mislead the output of neural network by adding imperceptible L_p-norm con-

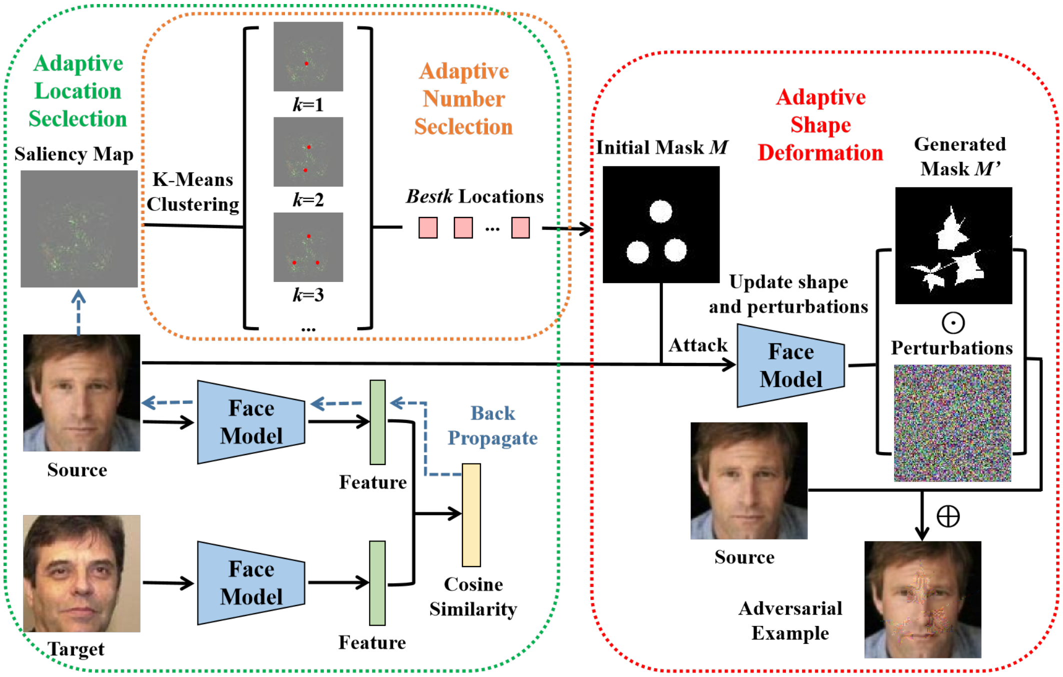

Figure 1. An overview of Adaptive Adversarial Patch (AAP) attack framework. When inputting the source and target face images, AAP uses saliency map and K-Means clustering to adaptively choose the optimal patch location and number. Based on the selected patch centers, iterative shape deformation and perturbation optimization are conducted to generate the adversarial example.

strained perturbations to the input image. According to the amount of information the attacker could access, adversarial attack can be divided into white-box and black-box attacks. Under white-box setting, the structure and parameters of the model are fully accessible. Classical methods include gradient-based attack, *e.g.*, I-FGSM [26], and optimization-based attack, *e.g.*, C&W [6]. Under black-box setting, the attacker cannot access any detail of the model, which could only use surrogate models to conduct transfer-based attack. This kind of attack can be mainly divided into two categories, *i.e.*, modifying gradient updates, *e.g.*, MI-FGSM [11], ABI-FGM [55], and conducting input transformations, *e.g.*, DI-FGSM [54], Admix [42], AITL [58]. In addition to transfer-based black-box attack, some task settings allow querying to obtain the predictions of target model during attack. Query-based black-box attack can be carried out by gradient estimation [7] and random search [1].

Apart from the attack methods mentioned above, which mainly aim at image classification, there exist various attacks on other vision tasks. For example, RAP [29] and UEA [45] are proposed for object detection. DAG [53] applies to both semantic segmentation and object detection. There are also recent works specifically designed for face recognition, such as A3GN [56] and DFAnet [59].

2.2. Adversarial Patch

The concept of adversarial patch has been proposed by Brown *et al.* [5]. Compared to imperceptible global perturbations, adversarial patch relaxes the L_p-norm constraints of perturbation magnitude but limits the perturbation region. Due to the physical realizability, adversarial patch turns out to be an important means to evaluate the robustness of real-world DNN-based systems. So far, it has been widely used in vision tasks such as image classification [5, 9, 23, 37, 50], object detection [21, 27, 34], traffic sign recognition [13].

GAP [5] first generates robust adversarial patch based on the EOT (Expectation Of Transformation) framework [2], which lays a solid foundation for the follow-up works. LaVAN [23] further generates local patch with smaller area. Both GAP and LaVAN focus on the pattern of patch during generation, ignoring the effect of patch location and shape. Some works, *e.g.*, ROA [50], LO [37], explore the effect of patch location. The former method uses a gray rectangular patch to locate the most sensitive region of the input image, while the latter alternately optimizes patch location and pattern. GDPA [28] trains an end-to-end dynamic patch generator, generating both patch pattern and patch location for each input image. Except for location, DAP [9] shows that patch shape also matters and conducts deformable patch attack. However, all these works mainly focus on a single factor of patch generation. Our work considers various aspects comprehensively.

2.3. Adversarial Patch on Face Recognition

As an important application of visual models in daily life, adversarial patch undoubtedly poses security risks to face recognition models. During the generation of adversarial patch, most approaches restrict the patch region to different areas based on prior knowledge. Adv-Glasses [38] limits perturbations within the glasses frame region to simulate a specific face by wearing the printed glasses frame. Adv-Hat [25] generates smooth, untargeted patches based on total variation loss in the hat region. Adv-Makeup [57] utilizes a generative adversarial network to generate imperceptible eye-shadow patches. TAP [52] extends existing transfer-based techniques to generate transferable adversarial patches in the fixed eye region. GenAP [52] optimizes latent vectors of a pretrained generative model and generates adversarial patches with human-like facial features.

Unlike these fixed-location methods, RHDE [43] proposes a region-based heuristic differential evolution algorithm to search for the optimal location of a given sticker, conducting query-based black-box attack. Under similar setting, Wei *et al*. [44] introduce reinforcement learning to the task of patch-based attack, optimizing location and perturbations for an adversarial patch at the same time.

2.4. Adversarial Defense

To eliminate the negative impact of adversarial examples on models, a variety of adversarial defense methods have emerged in recent years.

Adversarial training [14, 46] is one of the main empirical defense methods, which improves the robustness of models by involving adversarial examples in training data. But this approach incurs significant computational cost and may cause overfitting to the specific adversarial examples used in training. Several subsequent methods [48, 49] have been proposed to address these issues. There also exist other empirical defense methods, such as digital watermarking [16], local gradients smoothing [36], and denoising [30]. Compared to empirical defense, certified defense [22, 24, 47, 51] is a kind of proof-based defense strategy that guarantees the reliability of models within a specific range of input perturbations by mathematical proofs. Some works design defense strategies specifically for object detection [31] and face recognition [60]. All these defense methods contribute to improving the robustness of deep neural networks.

3. Method

In this section, we introduce our proposed unified Adaptive Adversarial Patch (AAP) attack framework. Sec. 3.1 first presents the problem definition. Sec. 3.2 elaborates on our AAP algorithm.

3.1. Problem Definition

Face recognition is usually divided into two tasks, face verification and face identification. Face verification compares an input face image with a target face image to determine whether they belong to the same identity. Whereas face identification compares the input face image with multiple face images in a database to find out the most matching face image and the corresponding identity.

Let $f(x) : \mathbf{X} \rightarrow \mathbb{R}^d$ denotes a face recognition model that extracts a normalized feature vector for an input face image $x \in \mathbf{X}$. Given a pair of face images, the face recognition model extracts their features, respectively. If the cosine similarity of their feature vectors exceeds the predefined threshold of the model, the two images can be considered as belonging to the same identity.

Our goal for the targeted patch attack is to solve the following constrained optimization problem. Let x_s denote the source face image of the attacker, x_t denote the target face image of the victim, and x_{adv} denote the adversarial example with the adversarial patches added to x_s. The formal definition of our targeted patch attack is shown as follows:

$$\begin{aligned} &\underset{x_{adv}}{\operatorname{argmax}}\, L_{adv}(f(x_{adv}), f(x_t)), \\ \text{s.t. } &x_{adv} \odot (1 - M) = x_s \odot (1 - M), \\ &||x_{adv} \odot M - x_s \odot M||_\infty \leq \epsilon, \end{aligned} \tag{1}$$

where L_{adv} is a cosine similarity loss function, $\odot$ is the element-wise product, ϵ is the perturbation bound and M is a binary mask of patch. M controls the region of patch, including location, shape, and number. A pixel can only be perturbed when its corresponding value in M is 1.

3.2. Adaptive Adversarial Patch

In this section, we propose our AAP algorithm, which consists of adaptive patch location selection, patch shape deformation, and patch number selection.

3.2.1 Patch Location Selection

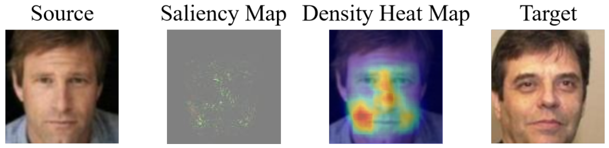

Figure 2. A visualization example of our generated saliency map and the corresponding density heat map.

To find the optimal location of patch, we introduce a patch location selection strategy based on saliency map. As defined in Sec 3.1, given the face pair $\{x_s, x_t\}$, our loss function for targeted attack is $L_{adv} = cos\,(f(x_s), f(x_t))$. We back propagate L_{adv} to obtain the gradients $\nabla_{x_s} L_{adv}$, then square and sum up the results of each channel to obtain a saliency map, *i.e.*, $SaliencyMap = \sum_{i=1}^{c} (\nabla_{x_s} L_{adv})^2$. It is assumed that the value of each pixel in the saliency map reflects its importance to recognition, which means the densest region can be regarded as the most susceptible region to attack. We calculate the density heat map of the saliency map to indicate the density of each region. The center of the densest region is selected as the center of our adversarial patch. An example of a saliency map and the corresponding density heat map is visualized in Fig. 2.

3.2.2 Patch Shape Deformation

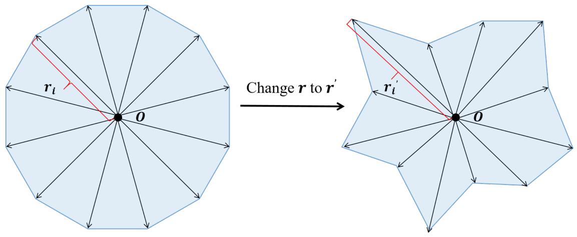

Figure 3. An example of patch shape representation. The shape of the polygon is determined by the center O and the lengths of rays $r = \{r_1, r_2, .., r_n\}$, which can be deformed by altering r.

Through adaptive patch location selection, the center of adversarial patch is determined. To include shape factor in the attack, we model the shape of patch. Inspired by DAP [9], we represent the shape of the patch as a polygon using polar coordinates with a central point O and a set of equidistant rays with lengths $r = \{r_1, r_2, .., r_n\}$. An example of shape representation is shown in Fig. 3. The shape can be deformed by altering the lengths of the rays r. The central point O and the lengths of rays r determine the shape of patch. Once the shape is determined, we can obtain the binary mask M of the patch. For each coordinate (i, j) in the image, we calculate whether it is inside the triangle formed by two adjacent rays by Gaussian Elimination. If inside, the corresponding mask value M_{ij} is set to 1; otherwise, it is set to 0. The details about computing the mask follow DAP [9].

As the entire computing process is differentiable, we can optimize the lengths of the rays r and the perturbations δ simultaneously by gradient ascent. Suppose x^i_{adv} is the adversarial image generated at the i-th iteration, the mask M^i is determined by r^i and the patch center location O obtained in location selection,

$$x^i_{adv} = x_s \odot (1 - M^i) + \delta^i \odot M^i. \tag{2}$$

In summary, we treat the perturbation δ^i and the ray lengths r^i as optimizable variables. We back propagate the loss function L and obtain the gradient ∇L to update r^i and δ^i. The updating process can be expressed as follows:

$$\begin{gathered} r^{i+1} = \text{Clip}_{[1,+\infty]}(r^i + \gamma \cdot sign(\nabla_{r^i} L)), \\ \delta^{i+1} = \text{Clip}_{[x_s-\epsilon, x_s+\epsilon]}(\delta^i + \alpha \cdot sign(\nabla_{x^i_{adv}} L)), \\ r^0 = \{r'\}^n, \;\; \delta^0 = x_s, \end{gathered} \tag{3}$$

where γ is the step size of r, α is the step size of δ, ϵ is the perturbation bound, r' is the initial value of r.

To restrict the size of patch area, we add a penalty term L_{shape} to the loss function:

$$L_{shape} = Mean(M^i). \tag{4}$$

The modified loss function is as

$$L = \begin{cases} L_{adv}, & ,p \leq s \\ L_{adv} - \beta \cdot L_{shape}, & ,p > s \end{cases}, \tag{5}$$

where p is the current patch area, s is the predefined limit of patch area, β is the hyper parameter.

3.2.3 Patch Number Selection

As mentioned in Sec. 3.2.1, we aim to select the densest region based on the saliency map. However, it can be observed in Fig. 2 that there also exist several relatively dense regions in the saliency map. We spontaneously think of splitting the patch into multiple pieces to cover as many of these regions as possible. To achieve this, we propose the patch number selection strategy, which further enhances the attack performance.

We utilize K-Means clustering to adaptively select the number of patches. Given the value of k, with the saliency map as a weight matrix, K-Means clustering method can partition all pixels in the image into k parts. SSE, which refers to within-cluster sum of squared errors, is used to assess the quality of clustering. To obtain the most appropriate number of clusters, we follow the Elbow Method [3]. Specifically, we calculate the SSE under different values of k and plot the curve between SSE and k. An example is shown in Fig. 4. The infection point of the curve, where the slope changes dramatically, is identified as the elbow, *i.e.*, the optimal value of k. With the $bestk$ determined, the corresponding $bestk$ cluster centers are automatically selected as the centers of adversarial patches, carrying out the attack in combination with adaptive shape deformation.

The entire process of AAP is illustrated in Alg. 1.

4. Experiments

In this section, we experimentally evaluate the attack performance of our proposed AAP algorithm. Sec. 4.1 illustrates our experimental settings. Sec. 4.2 presents our ablation study on AAP. Sec. 4.3 shows the comparison results between AAP and SOTA methods. Sec. 4.4 investigates the effect of some hyper parameters. Sec. 4.5 combines AAP with existing perturbation generation methods.

4.1. Experimental Settings

Datasets. In our experiments, LFW dataset [20] and IJB-C dataset [35] are used for evaluation. We randomly select 1000 face pairs with different identities from each dataset for targeted attack.

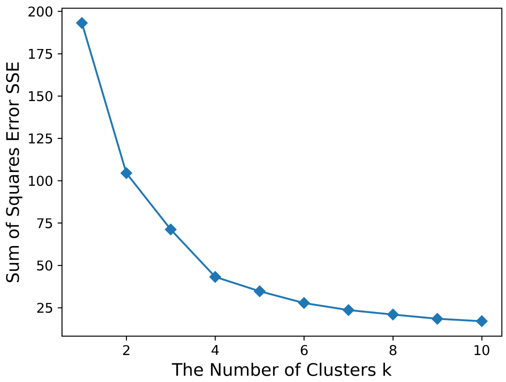

Figure 4. An example of the curve between SSE and the number of clusters k. When $k = 4$, the curve exhibits a significant inflection, $i.e.$, the elbow. Thus, the optimal number of clusters is 4.

Models. We choose models with different network architectures, $i.e.$, IR100 [17], IR50 [17], IR152 [17], IRSE50 [19], MobileFace [8], as the target face recognition models. The details of the models are shown in Table 1. Under black-box setting, we select part of these models as surrogate models to generate adversarial examples, while the others remain unseen for evaluation.

Compared Methods. We use several SOTA methods, $i.e.$, TAP [52], DAP [9], ROA [50], LO [37], for comparison. To ensure the fairness of comparison, all methods in our experiments generate perturbation in the same way, $i.e.$, I-FGSM [26], and are subject to the same limit of patch area.

In detail, the number of iterations T is set to 100. The perturbation bound ϵ is set to 32. The perturbation step size α is set to 1. The limit of patch area s is set to 1600, which is 20×80 for TAP, 40×40 for ROA and LO. For DAP and our proposed AAP, which involve shape deformation, the number of rays n is set to 16, the hyper parameter β is set to 200. For efficiency, shape deformation is only conducted in the first half of iterations. The shape step size γ is set to 1 and the perturbation step size α is set to 8 during the joint shape-perturbation optimization. Due to the characteristics of polygons, it is not possible to control the final generated patch area precisely equal to s, but we control the total patch area strictly less than s. For AAP, the maximum number of patches is set to 5. All other experimental settings follow the official settings in original papers.

Evaluation Metric. For targeted attack on face recognition, we use ASR (Attack Success Rate) as the evaluation metric, which is expressed as follow:

$$ASR = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\tau}(cos(f(x_{adv}), f(x_t)) > \tau), \quad (6)$$

where $\mathbb{1}_{\tau}$ denotes the indicator function, N is the size of our test set, x_{adv} is the generated adversarial example, and x_t is the target face image. τ is the threshold at given FAR (False Acceptance Rate), $i.e.$, 10^{-2} under LFW and 10^{-4} under IJB-C, for each target model f.

```
Algorithm 1: Adaptive Adversarial Patch (AAP)
Input: A target face model $f$, a source face image
       $x_s$, a target face image $x_t$
Input: Number of iterations $T$, perturbation bound
       $\epsilon$, number of rays $n$, perturbation step size $\alpha$,
       shape step size $\gamma$, limit of patch area $s$,
       maximum number of patches $K$
Output: $x_{adv}$
Initialize $\delta^0 = x_s$
Calculate the gradient $\nabla_{x_s} L_{adv}$
$SaliencyMap = \sum_{i=1}^{c} (\nabla_{x_s} L_{adv})^2$
for $k$ = 1 to $K$ do
    Use K-Means to divide all pixels into $k$ clusters
    Calculate the corresponding SSE of $k$
end for
Plot the curve between SSE and $k$
Calculate the change ratio of slope at each $k$
$bestk$ = $k$ with the highest absolute value of change
  ratio
$Locations$ = center coordinates of $bestk$ clusters
Initialize $r^0 = \{\frac{1}{2}$ sqrt$(\frac{s}{bestk})\}^{bestk \times n}$
for $i$ = 1 to $T$ do
    if $i < \frac{T}{2}$ then
        Calculate $M^i$ based on $Locations$ and $r^i$
        $x_{adv}^i = x_s \odot (1 - M^i) + \delta^i \odot M^i$
        $r^{i+1} = \text{Clip}_{[1,+\infty]}(r^i + \gamma \cdot sign(\nabla_{r^i} L))$
        $\delta^{i+1} = \text{Clip}_{[x_s-\epsilon, x_s+\epsilon]}(\delta^i + \alpha \cdot sign(\nabla_{x_{adv}^i} L))$
    else if $i == \frac{T}{2}$ then
        Calculate $M^i$ and fix $M$ as $M^i$
    else
        $x_{adv}^i = x_s \odot (1 - M) + \delta^i \odot M$
        $\delta^{i+1} = \text{Clip}_{[x_s-\epsilon, x_s+\epsilon]}(\delta^i + \alpha \cdot sign(\nabla_{x_{adv}^i} L))$
    end if
end for
return $x_{adv}$
```

4.2. Ablation Study

We investigate the effect of various factors of patch on attack performance through the ablation study on AAP.

Table 2 elaborates the influence of each factor on attack

Table 1. The details of the target face recognition models used in our experiments.

Models		
Backbone	Head	Traning Data
IR50 [17]	CosFace [41]	WebFace600K [61]
IR100 [17]	3D-BERL [18]	MS1MV2 [10]
IR152 [17]	ArcFace [10]	MS-Celeb-1M [15]
IRSE50 [19]	ArcFace [10]	MS-Celeb-1M [15]
MobileFace [8]	ArcFace [10]	MS-Celeb-1M [15]

Table 2. The results of the ablation study on AAP under LFW when using single surrogate model IR100. * indicates white-box attack.

Factors of Patch			Attack Success Rate				
Location	Shape	Number	IR100*	IR50	IR152	IRSE50	MobileFace
×	×	×	0.968*	0.397	0.348	0.149	0.085
✓	×	×	0.999*	0.819	0.715	0.300	0.171
✓	✓	×	1*	0.811	0.722	0.317	0.173
✓	✓	✓	**1***	**0.831**	**0.781**	**0.370**	**0.205**

success rate under LFW. In the baseline, which does not take any factor into account, we select a random location and generate an adversarial patch with fixed shape.

Adding the location selection part, we adaptively select the densest region of the saliency map as the center of patch, instead of choosing a random location in baseline. As shown in Table 2, ASR rises by 42.2% compared to baseline in black-box attack against IR50, which verifies the importance of patch location for attack performance.

Further, we include the shape factor, which adaptively deforms the patch shape after selecting the location. Since the generated polygon patch area is strictly less than the set patch area limit, the loss in patch area makes the improvement of ASR less significant. But it still demonstrates that patch shape matters in patch attack.

Finally, we consider the location, shape, and number factors comprehensively, $i.e.$, conducting the complete AAP attack. We adaptively choose the optimal number of patches with corresponding locations, and then launch joint shape-perturbation optimization. The complete AAP achieves the highest ASR. Specifically, AAP is 3.2% higher than baseline on ASR in white-box attack, while 43.4%, 43.3%, 22.1%, 12% higher in black-box attack against IR50, IR152, IRSE50 and MobileFace, respectively. The enhancement of ASR validates the effect of patch number.

Through analysis of Table 2, we conclude the addition of adaptive location selection makes the greatest improvement on attack performance, which can be considered as the most important factor of patch generation to some degree.

4.3. Comparison With SOTA

We conduct extensive experiments under both white-box and black-box settings to demonstrate the superior performance of our proposed method over SOTA methods.

We compare AAP and SOTA methods under LFW dataset. Table 3 displays the comparison results on attack success rate under LFW when using single surrogate models. It is demonstrated that our proposed AAP achieves higher ASR than current SOTA methods. When the surrogate model is IR100, under white-box setting, the ASR of AAP reaches 100%. Under black-box setting, compared to ROA, it increases by 11.8% against IR152 and 9.8% against IRSE50. When using IR152 as the surrogate model, AAP obtains the highest ASR under both settings as well. In detail, its ASR is 15.8% higher than ROA against IR50, 20.6% higher against IR100.

Table 3. The comparison results between AAP and SOTA methods on attack success rate under LFW when using single surrogate model IR100 and IR152. * indicates white-box attack.

Method	Attack Success Rate				
	IR100*	IR50	IR152	IRSE50	MobileFace
TAP [52]	1*	0.607	0.574	0.223	0.125
LO [37]	0.880*	0.277	0.262	0.110	0.063
ROA [50]	1*	0.748	0.663	0.272	0.158
DAP [9]	0.984*	0.436	0.405	0.183	0.097
AAP (Ours)	**1***	**0.831**	**0.781**	**0.370**	**0.205**

Method	Attack Success Rate				
	IR152*	IR50	IR100	IRSE50	MobileFace
TAP [52]	1*	0.347	0.431	0.158	0.098
LO [37]	0.886*	0.140	0.210	0.084	0.054
ROA [50]	1*	0.444	0.534	0.190	0.115
DAP [9]	0.992*	0.249	0.379	0.144	0.075
AAP (Ours)	**1***	**0.602**	**0.740**	**0.283**	**0.142**

Further experiments are conducted using multiple surrogate model ensembles. As shown in Table 4, four models are used as the surrogate model ensemble and the remaining one is used for black-box evaluation. With model ensemble, the ASRs of all methods have been enhanced, among which AAP still achieves the highest ASR.

Additionally, Table 5 illustrates the comparison results on average generation time of adversarial examples. Compared to ROA, which achieves the highest ASR among current methods, our speed of generating adversarial examples is approximately 5 times faster under single surrogate model setting and about 11 times faster under ensemble setting. Compared to other methods, our speed is not much slower but our ASR is significantly higher than theirs.

Apart from LFW, we evaluate the performance of AAP on a more challenging dataset, IJB-C. To simulate a realistic

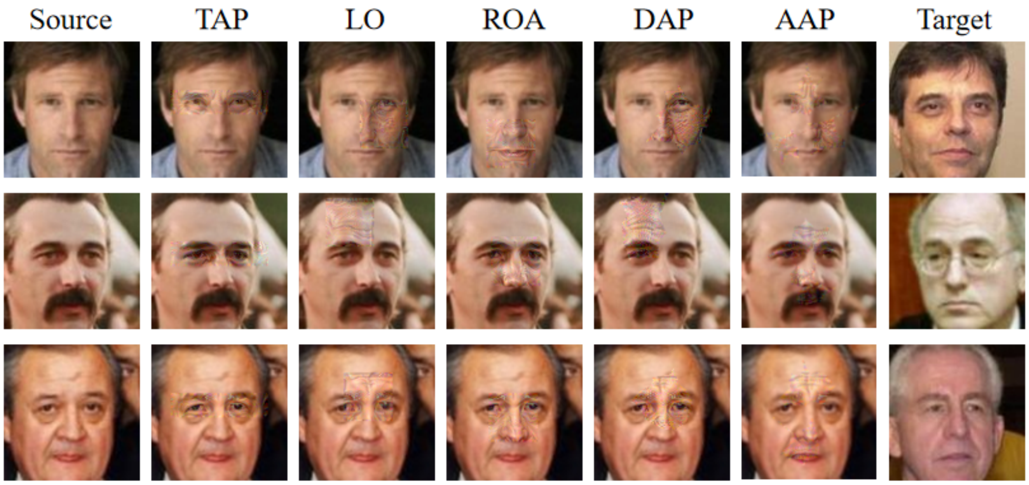

Figure 5. Visualizations of adversarial examples generated by different methods.

Table 4. The comparison results between AAP and SOTA methods on attack success rate under LFW when using multiple surrogate model ensembles. For each column, the written model is regarded as the target black-box model while the rest four models are used as the ensemble of surrogate models.

Method	Attack Success Rate				
	IR50	IR100	IR152	IRSE50	MobileFace
TAP [52]	0.677	0.746	0.703	0.575	0.497
LO [37]	0.369	0.414	0.36	0.326	0.269
ROA [50]	0.821	0.875	0.825	0.692	0.629
DAP [9]	0.503	0.589	0.516	0.436	0.342
AAP (Ours)	**0.881**	**0.933**	**0.870**	**0.813**	**0.756**

Table 5. The comparison results between AAP and SOTA methods on average generation time of adversarial examples under LFW when using different surrogate settings. Single indicates using single surrogate model IR100. Ensemble indicates using ensemble of surrogate models, *i.e.*, IR50, IR152, IRSE50, MobileFace. The experiments are conducted on TITAN RTX GPU.

Method	Average Generation Time (s)	
	Single	Ensemble
TAP [52]	6.337	14.348
LO [37]	11.770	35.119
ROA [50]	60.650	236.944
DAP [9]	7.994	20.091
AAP (Ours)	11.793	20.065

scenario, we consider multiple gallery images for each target. Table 6 shows the comparison results on attack success rate under IJB-C. As shown, our AAP significantly outperforms SOTA methods. Interestingly, all methods achieve relatively low ASRs when attacking IRSE50, possibly due to the limited model capacity of IRSE50 for extremely challenging evaluation. Since MobileFace is a weaker model than IRSE50, it is not evaluated in our experiments.

Table 6. The comparison results between AAP and SOTA methods on attack success rate under IJB-C when using single surrogate model IR100. * indicates white-box attack.

Method	Attack Success Rate			
	IR100*	IR50	IR152	IRSE50
TAP [52]	0.829*	0.044	0.071	0.014
LO [37]	0.389*	0.019	0.019	0.006
ROA [50]	0.946*	0.133	0.133	0.026
DAP [9]	0.722*	0.058	0.050	0.014
AAP (Ours)	**0.987***	**0.232**	**0.205**	**0.038**

The visualizations of some adversarial examples generated by our proposed AAP and SOTA methods are presented in Fig. 5. With the same size of patch area, AAP obtains the best attack performance.

4.4. The Effect of Hyper Parameters

In this section, we evaluate the influence of some hyper parameters on the attack performance, *i.e.*, the number of rays, the size of patch area.

The Number of Rays. Fig. 6 illustrates the effect of the number of rays. It can be observed that the ASRs first increase with the number of rays, then fluctuate and decrease. Given a small size of patch area, the increase of the number of rays brings more complexity to shape modeling, which makes it more difficult to optimize. Considering the effectiveness, we set the number of rays to 16 in our experiments.

The Size of Patch Area. Fig. 7 shows the effect of the

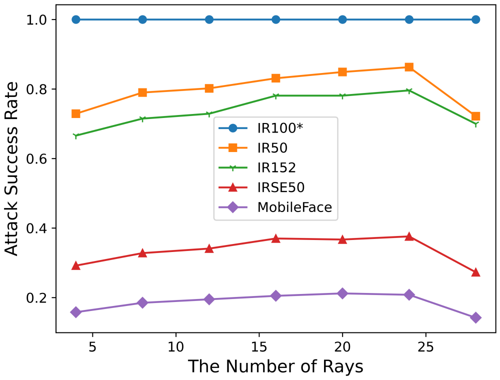

Figure 6. The curve between the ASR of AAP and the number of rays under LFW when using single surrogate model IR100. As the number of rays increases, the ASRs first increase, then fluctuate and decrease.

size of patch area on the attack performance of AAP. We can clearly observe that with the increase of patch area, the ASRs of AAP are significantly improved. Considering that a large size of patch area would make the adversarial patch too conspicuous and easily noticeable, we choose a compromised size of patch area, *i.e.*, we set the patch area to 1600 in our experiments, which occupies only about 11% of the total image pixels.

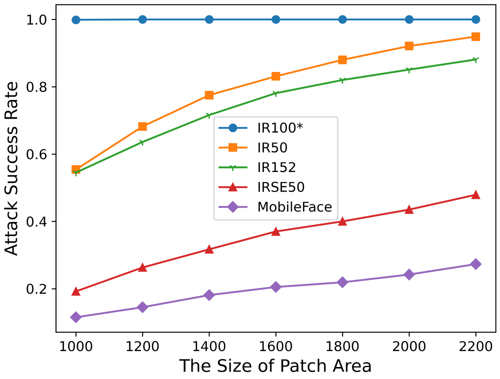

Figure 7. The curve between the ASR of AAP and the size of patch area under LFW when using single surrogate model IR100. As the size of patch area increases, the ASRs increase.

4.5. Combination with Different Perturbation Generation Methods

AAP can be easily combined with different perturbation generation methods, *e.g.* DI-FGSM [54], TI-FGSM [12], to further enhance the attack performance.

Table 7 shows that the attack success rates of AAP are boosted when integrated with various perturbation generation methods under LFW. Among them, AAP-DI achieves the highest ASR. In detail, compared to the baseline AAP, the ASR of AAP-DI increases by 7.2%, 8%, 6%, 5.7% in the black-box attack against IR50, IR152, IRSE50, MobileFace respectively.

Table 7. The attack success rates of AAP combined with different perturbation generation methods under LFW when using single surrogate model IR100. * indicates white-box attack.

Method	Attack Success Rate				
	IR100*	IR50	IR152	IRSE50	MobileFace
AAP	1*	0.831	0.781	0.370	0.205
AAP-TI	1*	0.851	0.792	0.384	0.232
AAP-DI	**1***	**0.903**	**0.861**	**0.430**	**0.262**

5. Limitation and Social Impact

5.1. Limitation

There exist some limitations in our current work. Our experiments are conducted only in digital domain without verifying the physical feasibility of our proposed method, which needs further investigation. Besides, our generated adversarial patches are composed of irregular adversarial perturbations, which still needs to be improved in terms of the concealment of patch.

5.2. Possible Negative Social Impact

Though our work may pose a possible security threat to face recognition models, our main purpose is to provide a novel robustness evaluation method for real-world face recognition systems. In the future, we plan to explore the potential application of our method in adversarial defense area, expecting to construct more robust face recognition models.

6. Conclusions

In this paper, we propose a novel unified Adaptive Adversarial Patch Attack framework for targeted attack on face recognition models, AAP. Unlike existing patch attack works that focus only on one aspect of patch generation, we comprehensively consider multiple factors including patch location, patch shape, and patch number. Our approach adaptively selects patch location and number based on saliency map and K-Means clustering, and then jointly optimizes the shape and perturbations of each patch. Extensive experiments are conducted on face models with different architectures. The experimental results demonstrate the superiority of our proposed AAP method over SOTA methods.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pages 484–501. Springer, 2020.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[3] P. Bholowalia and A. Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. In *Proceedings of the NeurIPS Workshop*, 2017.

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[7] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[8] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13*, pages 428–438. Springer, 2018.

[9] Z. Chen, B. Li, S. Wu, J. Xu, S. Ding, and W. Zhang. Shape matters: deformable patch attack. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 529–548. Springer, 2022.

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

[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[12] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[13] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[15] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 87–102. Springer, 2016.

[16] J. Hayes. On visible adversarial perturbations & digital watermarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1597–1604, 2018.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] M. He, J. Zhang, S. Shan, and X. Chen. Enhancing face recognition with self-supervised 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4062–4071, 2022.

[19] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[20] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[21] H. Huang, Y. Wang, Z. Chen, Z. Tang, W. Zhang, and K.-K. Ma. Rpattack: Refined patch attack on general object detectors. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[22] J. Jia, X. Cao, B. Wang, and N. Z. Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. *arXiv preprint arXiv:1912.09899*, 2019.

[23] D. Karmon, D. Zoran, and Y. Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515. PMLR, 2018.

[24] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

[25] S. Komkov and A. Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 819–826. IEEE, 2021.

[26] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[27] M. Lee and Z. Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019.

[28] X. Li and S. Ji. Generative dynamic patch attack. *arXiv preprint arXiv:2111.04266*, 2021.

[29] Y. Li, D. Tian, M.-C. Chang, X. Bian, and S. Lyu. Robust adversarial perturbation on deep proposal-based models. *arXiv preprint arXiv:1809.05962*, 2018.

[30] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1778–1787, 2018.

[31] J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982, 2022.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[34] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

[35] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 international conference on biometrics (ICB)*, pages 158–165. IEEE, 2018.

[36] M. Naseer, S. Khan, and F. Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307. IEEE, 2019.

[37] S. Rao, D. Stutz, and B. Schiele. Adversarial training against location-optimized adversarial patches. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 429–448. Springer, 2020.

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

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[40] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[41] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[42] X. Wang, X. He, J. Wang, and K. He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16158–16167, 2021.

[43] X. Wei, Y. Guo, and J. Yu. Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[44] X. Wei, Y. Guo, J. Yu, and B. Zhang. Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[45] X. Wei, S. Liang, N. Chen, and X. Cao. Transferable adversarial attacks for image and video object detection. *arXiv preprint arXiv:1811.12641*, 2018.

[46] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 499–515. Springer, 2016.

[47] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

[48] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[49] D. Wu, S.-T. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

[50] T. Wu, L. Tong, and Y. Vorobeychik. Defending against physically realizable attacks on image classification. In *International Conference on Learning Representations*, 2020.

[51] K. Y. Xiao, V. Tjeng, N. M. Shafiullah, and A. Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.

[52] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. Gao, X. Zhang, J. Zhou, and J. Zhu. Improving transferability of adversarial patches on face recognition with generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11845–11854, 2021.

[53] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

[54] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[55] B. Yang, H. Zhang, Z. Li, Y. Zhang, K. Xu, and J. Wang. Adversarial example generation with adabelief optimizer and crop invariance. *Applied Intelligence*, 53(2):2332–2347, 2023.

[56] L. Yang, Q. Song, and Y. Wu. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia tools and applications*, 80:855–875, 2021.

[57] B. Yin, W. Wang, T. Yao, J. Guo, Z. Kong, S. Ding, J. Li, and C. Liu. Adv-makeup: A new imperceptible and transferable attack on face recognition. *arXiv preprint arXiv:2105.03162*, 2021.

[58] Z. Yuan, J. Zhang, and S. Shan. Adaptive image transformations for transfer-based adversarial attack. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 1–17. Springer, 2022.

[59] Y. Zhong and W. Deng. Towards transferable adversarial attack against deep face recognition. *IEEE Transactions on Information Forensics and Security*, 16:1452–1466, 2020.

[60] J. Zhou, C. Liang, and J. Chen. Manifold projection for adversarial defense on face recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 288–305. Springer, 2020.

[61] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021.