FOOLING FACIAL RECOGNITION SYSTEMS AND MITIGATION

Eunice Koh Kexin[1], Fu Wentao (Claire)[1], Katelyn Kang Jia Xuan[1], Shen Bingquan[2]

[1]Raffles Girls' School, 2 Braddell Rise, Singapore 318871
[2]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract

Adversarial patch attacks challenge the robustness of deep learning models, particularly in applications like facial recognition. This research introduces a universal patch attack framework using simulated annealing to optimise patch design and placement. The method adapts patches to 3D facial contours with differentiable rendering and thin plate spline transformations, ensuring realistic integration. Differentiable rendering captures realistic lighting and viewing conditions, improving transferability. With our approach, we generate a universal patch – universal both in location and texture. We extend the concept of adversarial attacks to support multiple patches, allowing flexible attack strategies, and are trained against state-of-the-art models for enhanced robustness. As adversarial attacks against systems like these become increasingly advanced, we investigated how some common defence methods, such as exploiting higher patch frequencies and data augmentation, would impact the results of the adversarial attacks. We then explored potential methods to then counter these particular defence methods.

1. Introduction

Deep Neural Networks have been widely utilised in various domains, including computer vision applications like facial recognition (FR), object classification etc. One shortfall is that these are susceptible to adversarial attacks, and in real-world scenarios, often subjected to "cat-and-mouse" games where security systems are continuously updated to counter new attack strategies, while attackers develop increasingly sophisticated methods to bypass defences. Thus, to forestall such attacks, it is important to explore novel attack methods.

Adversarial attacks can be categorized into digital and physical attacks. Digital attacks perturb input images subtly in ways imperceptible to humans but capable of deceiving models. Physical attacks, such as creating adversarial patches, manipulate physical objects to achieve similar outcomes in real-world settings. The latter is particularly relevant to FR systems, which typically process physical inputs, and is the focus of this research.

Existing adversarial attacks have several shortcomings. First, whether physical or digital attacks, the optimisation process is conducted in a digital environment. Consequently, many current attacks fail to translate effectively into real-world environments, where factors, like lighting or texture, can significantly degrade the adversarial attack's performance. Thus, these factors must be taken into account to make the attack more robust. Existing attack methods are also often limited in both scope and practicality. Many are designed for highly specific applications, such as adversarial glasses [1] or makeup [2], restricting their versatility and applicability to broader contexts. Lastly, there are limitations to the universality of the adversarial patches. The patch must be universal in both texture and location. Currently, it is difficult to ascertain a universal location, as each individual's facial proportions are unique. During optimisation, patches are often placed in a particular (x, y) coordinate relative to an image of a certain size, or a relative location (e.g., "top", "centre") [3], assuming that it

corresponds to the same facial feature [4]. While such location specifics may not matter as much for object detection [5], this has several issues in FR – first, the variability in facial structure across individuals means that a fixed (x, y) coordinate might correspond to entirely different facial features for different people in the images, or the patch may not even lie within the face of the individual. Secondly, it cannot be used in real life, because it is unclear what that coordinates correspond to.

In light of these limitations, in this research we propose a *universal* single-and-multi adversarial patch framework for robust attacks on FR systems that can be implemented in real life by printing out the patch and placing it on the attacker's face.

2. Methodology

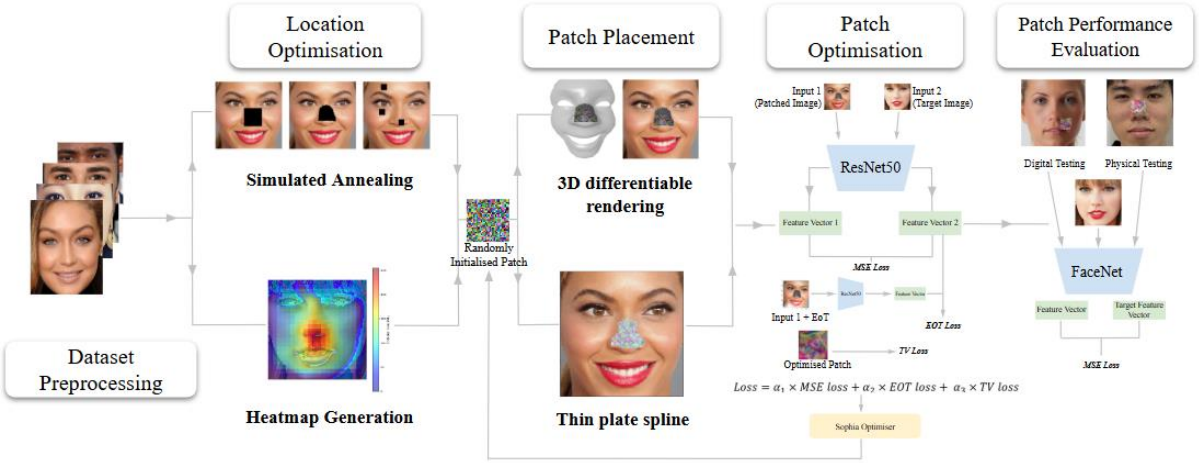


Fig. 2.1 Pipeline of adversarial attack framework. Best viewed zoomed in.

2.1 Dataset

We downloaded a face dataset, CelebA[1], a large dataset of more than 200, 000 images to train our adversarial patch. For data pre-processing, we cropped the photos based on boundary boxes enclosing the facial keypoints detected by MediaPipe[2], and removed photos which could not be detected, possibly because it is an extreme side profile, or contained sunglasses, detected using a glasses detector[3]. Due to time constraints as a large amount of time is required to optimise the patch, we used ~10, 000 images for training and ~3000 for testing.

2.2 Heatmap Generation and Simulated Annealing

To determine the most effective position for an adversarial patch, we explored two methods – using occlusion sensitivity to generate a heatmap and simulated annealing.

2.2.1 Heatmap generation

By covering parts of the target face image and finding the difference between the outputs of the model with the original and the masked image, we found which area, if occluded, would cause the greatest change in feature vector – in other words, which region is the most crucial to the model identifying a person as the target person. The mask was a 5 by 5 pixels black patch, shifted by a margin of 2 pixels each time. The difference was calculated using Mean Squared Error (MSE) loss between the feature vector of the original image, as detected by the ResNet50 model, and the masked image.

[1] https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[2] https://github.com/google-ai-edge/mediapipe
[3] https://github.com/mantasu/glasses-detector

Our target person was chosen to be Taylor Swift. The heatmap indicates that the nose region, when masked, greatly changes the feature vector as output of the model – the nose has high occlusion sensitivity. The regions of warmer color suggest that the corresponding region, if masked, results in greater change in feature vector. This suggests that placing an adversarial patch on the nose could be effective, as it likely contains features central to the model's identification process.

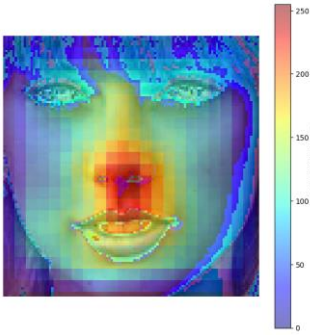


Fig. 2.3.1.1 Heatmap of Target Person's face

2.2.2 Simulated annealing

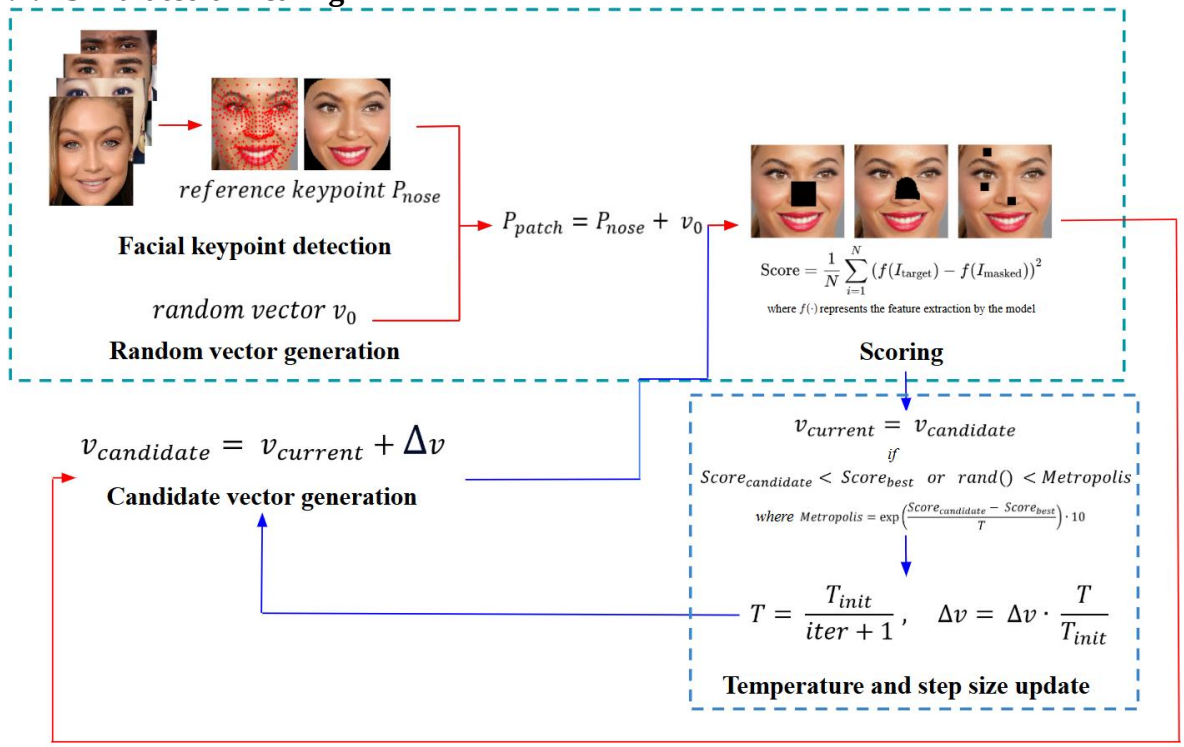


Fig. 2.3.2.1 Simulated Annealing Pipeline: the red arrows represent the first iteration, and the blue subsequent iterations until the final iteration, where the best vector is yielded. Best viewed zoomed in.

Similar to [6], we employed simulated annealing to find the optimal location, but applied it to facial images, introducing several crucial distinctions. Simulated annealing is computationally efficient compared to an exhaustive search. It is a better option because it reduces computational costs by strategically narrowing down the search space instead of exhaustively evaluating every possible location, doing so while balancing *exploitation* (focusing on promising regions) and *exploration* (testing worse areas to avoid local minima).

We began by detecting facial keypoints using MediaPipe, generating random vectors for patch placement relative to the nose. These coordinates were constrained to lie within the face mesh framing the face, and iterated across the simulated annealing dataset of ~5000 images. The patch position is then a relative position such that it is not bound by constraints of different facial proportions. For each candidate location, masked regions were generated for single-patch, and multi-patch cases. Losses included: (1) MSE loss between the masked and target images, and (2) MSE loss between the masked and original images. This ensures the patch's

effectiveness arises from its adversarial properties, not simple feature obstruction. The rationale is that the black patch can serve as a proxy for estimating actual patch location.

Simulated annealing takes into account a temperature variable. At a higher temperature, it is more likely to accept a wrong or worse solution, meaning that the loss is higher than the previous, to prevent being stuck at a local minimum. It starts out from a high temperature, which gradually decreases. This method adjusts the probability of accepting less optimal solutions over time, gradually focusing on regions most likely to yield the global minimum.

The candidate vector, determined by a step size (dependent on temperature), is accepted based on a probabilistic criterion or as the new "best" vector if it reduces loss. As temperature decreases, the search converges to an optimal patch location, yielding the final relative position.

2.3 3D Differentiable Rendering and Thin Plate Spline (TPS) Transformation

When the printed patch is placed on an attacker's face, the patch would fold, bend or curve based on the contours of the attacker's face. Hence, to create a patch that works in real life, we have to consider this 3D aspect of placing the patch to ensure robustness in different physical environments.

2.3.1 3D Differentiable Rendering

Fig. 2.4.1.1 3D Face Reconstruction Pipeline. Best viewed zoomed in.

To simulate placing the patch on an actual 3D face, a 3D face mesh of each individual in our dataset was generated using the MediaPipe face mesh function, where the predefined face mesh vertices[4], which serve as a standardized 3D representation of the face, was used alongside the detected facial landmarks of each individual. Using PyTorch3D[5], the patch was applied as a texture mapped to the face mesh, and rendered using its differentiable rendering function. The function allowed for the patch to be applied in a manner that accounted for 3D transformations, such as rotation, scaling, and depth perspective. Lighting settings were randomised during the optimisation process to simulate varying lighting conditions. Once the patch was rendered on the 3D face mesh with these realistic effects, it was extracted from the rendered image using a pixel intensity threshold, isolating the patch from the rest of the face mesh. The extracted patch, with its 3D effects and shading, was then overlaid onto the original 2D image at the corresponding location.

Fig. 2.4.1.2 Random patch without 3D differentiable rendering

Fig. 2.4.1.3 Random patch with 3D differentiable rendering

2.3.2 TPS Transformation[6]

[4] https://github.com/google/mediapipe/blob/master/mediapipe/python/solutions/face_mesh_connections.py

[5] https://github.com/facebookresearch/pytorch3d

[6] https://gist.github.com/catalys1/3eef0b6ee5749b5d8851755a7ee0e6e1

In TPS, the patch is like a thin metal plate constrained at certain points, causing it to bend. TPS warps the patch based on fixed points by minimising the bending energy. We warped the patch based on points on the attacker's face, detected using MediaPipe. This is similar to how in a real-world scenario, the patch would bend based on the attacker's face and have a shape similar to that of the nose, when placed on the nose.

Fig. 2.4.2.1 Patch with TPS transformation

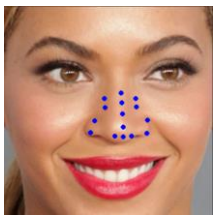
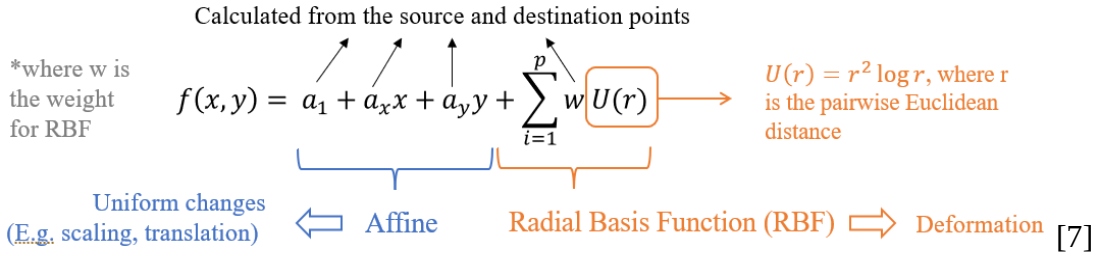


Fig. 2.4.2.2 Points used for TPS



[7]

2.4 Optimisation

For optimisation, we used Sophia[7] [8], a second order optimiser that performs better than other optimisers like Adam. It uses curvature approximation for adaptive gradient updates.

2.4.1 Loss Calculation

For the calculation of the loss, we included various losses to make our patch more robust:

1. MSE Loss between the outputs for the perturbed image and the target image: This causes the patch to optimise such that the difference between feature vectors of the patched face and of the target image's face would decrease, ensuring the patch is effective in causing the attacker to be misclassified as the target individual.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

2. EOT (Expectation Over Transformation) Loss: EOT Loss [9] is the loss after making some changes to the perturbed image, such as rotating, adding lighting or noise etc. By including these changes during training, the patch would be much more robust towards similar changes, improving its transferability to real-world settings. Additionally, this preempts defensive methods that neutralize attacks by introducing transformations like resizing or rotation.

3. TV (Total Variation) Loss of the patch: TV Loss is the difference between neighbouring pixels. It increases the smoothness of the patch and makes it less pixelated. This also takes into account certain defence methods that identify perturbations by extracting the image in terms of frequency. Many adversarial patches have high frequencies, and can be mitigated by extracting and omitting high-frequency regions of the image.

The below equation shows the loss calculation.

$$Loss = \alpha_1 \times MSE\ loss + \alpha_2 \times EOT\ loss + \alpha_3 \times TV\ loss$$

where the coefficients of each hyperparameter were fine-tuned to $\alpha_1 = 1$, $\alpha_2 = 0.3$, $\alpha_3 = 0.01$

2.5 Models

We used two separate models for training and testing of the adversarial patch to investigate the robustness of the patch across different models. For training, we used ResNet50[8] with its built-

[7] https://github.com/Liuhong99/Sophia

[8] https://github.com/deepinsight/insightface

in weights. For testing, we used PyTorch's FaceNet[9] [10]. This is to promote robustness of our universal patch such that our attack is generally successful even across models, which mimics real-world black-box scenarios where different FR systems use different models, and the attacker is unable to predict which one will be used.

3. Results and Discussion

3.1 Preliminaries

3.1.1 Patch Design Parameters

For single patches, the patch size was determined to be 30 by 30 pixels. From Fig. A in the Appendix, a 30-by-30 patch is the most likely to balance between effectiveness and practicality. For multiple patches, we used a 10-by-10 patch, though it is more versatile in that using this method, the number of patches and size can be otherwise specified for any particular context.

3.1.2 Testing

Our success criteria was MSE loss between feature vectors of the patched image and the target image. We tested the patches both digitally and physically for the 2D-optimised patches, and physically only for 3D. In physical testing, the printed patches were of size 3cm by 3cm, deemed most appropriate when transferring digitally to physically, based on average facial length. Following that, for multiple patches, the patches were of size 1cm by 1cm.

Generally, our patch is effective in causing the model to misclassify the attacker as the target person Taylor Swift. We considered a threshold of 0.95 for the MSE loss between the outputs for the perturbed image and Taylor Swift's image, determined from the average MSE loss between images of Taylor Swift and the target – which was approximately 0.9. A loss lower than the threshold is a successful misclassification.

3.2 Optimisation Process

During earlier optimisation, the patches were trained using only MSE loss between feature vectors of patched image and target image. One salient detail was that much more semantically meaningful patches were obtained, for example that of eyes, noses, or a partial face, versus after other losses were added (See Appendix). A possible reason is that the simpler optimisation pipeline directly focused on minimizing differences in feature vectors, which might have led to patches exploiting the most distinguishable facial features. These early patches may have aligned with high-salience regions due to their role in FR models, such as eyes and noses, which are heavily weighted in the embeddings generated.

However, with the addition of more sophisticated loss components like EOT Loss, the optimisation gained additional constraints to ensure robustness, universality, and physical realism. These constraints could diffuse the optimisation focus across multiple objectives, leading to patches that prioritize generalizability and robustness over semantic specificity. Though this may render the patch less effective, it is nonetheless an important trade-off, when taking into account real-world constraints like conspicuousness. For example, materials like paper or fabric cannot perfectly reproduce fine-grained pixel-level details. TV Loss helps generate smoother patches that are easier to replicate on real-world objects while preserving effectiveness. Besides, patches with sharp edges tend to contain high-frequency signals. When captured by a camera or processed, these signals can be distorted, smoothed out, or discarded [11] which can make the patch less effective or detectable as an anomaly.

[9] https://github.com/timesler/facenet-pytorch

3.3 Evaluation of Different Strategies for Adversarial Patch

3.3.1 Heatmap and Simulated Annealing

From the simulated annealing process, the best vector obtained, relative to the nose coordinates in a 112-pixel by 112-pixel image was (12, 1). To most front-facing faces, this corresponded to a region between the nose and mouth, slightly towards the individual's left cheek. As for multiple patches, the best vectors with the same reference point were (-18, 17), (18, 6) and (-3, 27). This corresponded also to regions near the nose and mouth.

Overall, we found that simulated annealing, with a success rate increase of ~34%, is more effective in finding an effective location, while that of the heatmap was ~32% (See also, Appendix, Fig. *H*). Our results corroborated with our hypothesis that simulated annealing would be the most effective compared to occlusion sensitivity; while both methods should work better than placing the patch on a random non-relative coordinate.

This is likely because the heatmap generation was done in relation to the target image's face, whereas the simulated annealing process was in relation to many faces. One obvious issue is that, though not without basis, it may be spurious to assume that particular feature which is the most salient to Taylor Swift (the nose, in this case) would be the optimal location to place the patch universally. At most, if solely based on the heatmap, the nose can be concluded to be the "optimal" location solely because it is the *only* possible drawn conclusion – all other options are illogical, because if the target image's nose is her most salient feature, then naturally the universal patch ought to be placed on the nose. In essence, this result cannot be assumed to generalize universally. On the other hand, simulated annealing outputs the most apparent location through global search across varying facial proportions and features, which returns the most direct and apparent optimal, universal location.

3.3.2 2D Patches, 3D Differentiable Rendering and TPS Transformation

When tested in practice, despite an overall decrease in loss with reference to the clean data, most were unable to meet the threshold for a misclassification. This could be due to the camera capturing process — the pixels may also have been picked up differently when captured. Cameras process images using algorithms that may apply compression, sharpening, or noise reduction, all of which can subtly change the pixel values. This is as opposed to the optimisation process, where the patch was in a simulated environment with pixel values precisely defined and consistent. During the conversion from HEIC format to JPEG, and subsequent resizing, slight distortions or artifacts were introduced, further altering the pixel-level representation of the patch (See Appendix, Fig. *M*).Other factors like color aberrations in printing can also cause deviations from the original digital patch. Additionally, during training, the patch was placed with reference to a 112-by-112 image. The patch may have been optimised only relative to such a pixel space – for example, a specific value difference between a specific pixel bordering the patch and face. In practice, however, faces don't have "resolutions". Thus, when the patch is applied to faces outside the 112-by-112 resolution context, the relative spatial relationships and pixel-level interactions assumed during training may no longer hold. This is all while the actual printed patch remains 30-by-30 pixels.

Contrary to our expectations, the 3D-optimised patches did not perform significantly better than the 2D patches. Because paper is non-elastic, it does not conform entirely to the contours of a person's face, as was assumed during differentiable rendering – folds, wrinkles, or air gaps when applied to the face were not accounted for. In fact, when the patch was placed, because of its stiffness, it did not bend much at all.

3.3.3 Single Patch and Multiple Patches

Unsurprisingly, performance was generally poorer for multiple patches. This is likely because the size of each patch in multiple patches is much smaller, the total area covered is also much smaller, as it is only 3 patches of 10 by 10 pixels compared to 30-by-30. Every pixel can be thought of as an "attack", and with smaller patches (in terms of pixel size), it means that there is less "attack potential". Additionally, the possible combinations of pixels decrease exponentially, leaving less area for exploration. This effect may further be exacerbated by the use of TV loss to ensure smoothness. As for the testing on real faces, especially with the resizing and conversion, the patch texture, especially for one which is much smaller, is likely to have been severely distorted.

4. Mitigation and Defence

In defence against such adversarial attacks, we trained a Convolutional Neural Network (CNN) that could differentiate between the target person's face (i.e. Taylor Swift), non-targets' faces and faces containing adversarial patches. This was achieved through freezing the base VGG16 model and creating a custom top layer for the specific classification of the target person's face. Then, the model was trained with a dataset of faces containing randomly generated adversarial patches, a dataset of other random faces, and a dataset of Taylor Swift's face only, assigned classes 0, 1 and 2 respectively. Random cropping and random rotations were used to enhance the CNN's generalisation [12] helping the model focus on invariant features rather than overfit to specific individual pixels in the dataset. Augmented data also reduced the effectiveness of attacks including EOT loss by exposing the model to similar perturbations during training, helping the model improve its ability to recognise and classify adversarial inputs.

We also used the Fast Fourier Transform (FFT). Adversarial patches generated often have high frequencies[10] [13] as patch generation processes rely on some form of iterative noising in the patch region [14], resulting in differences between frequency domains of the patched region and the rest of the image. We extracted the frequencies of the image using FFT, covered high-frequency regions, and reconstructed the modified image. The threshold for the low-frequency region (0.4) was determined by a histogram (see Appendix) of extracted image frequencies. As the model has been trained with random cropping, it can make an accurate prediction about the person's identity, even if some facial features are covered. While FFT has been explored in literature [14], it has rarely been applied to FR systems.

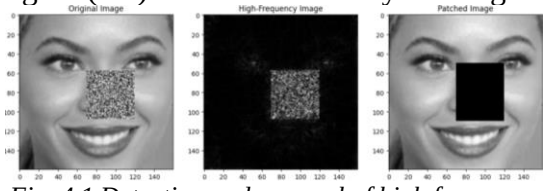


Fig. 4.1 Detection and removal of high frequency patch from the perturbed image

4.1 Effectiveness of Mitigation and Defence Strategies on the Patch

In this section, *clean accuracy* refers to the percentage of images correctly categorized within the non-adversarial datasets, while *attack success rate* is the proportion of adversarial images misclassified as Taylor Swift. The adversarial class consists of images generated as described in previous sections (see Fig. O.1-O.8, Appendix for results).

For single patches, the combination of data augmentation and FFT proves most effective, as anticipated. This is due to the model's ability to discard most patches and generalize invariant features. However, for multiple patches, the best performance is achieved with either no augmentation or FFT alone. This suggests the model may have overfitted to augmented data

[10] Higher image frequencies are at the parts of images that are rapidly changing, such as sharp edges – or in this case, the adversarial patch.

patterns and struggled with new augmentations. Additionally, as the model was trained primarily on larger patches (30x30), it may have been more sensitive to global pattern changes, failing to detect adversarial attacks when only local patterns were altered. The use of multiple patches exploited this vulnerability.

Although the model performs well, it requires significant training time (>1 hour) and the assembly of large datasets (at least 300 images of the target). However, its ease of use compensates for these drawbacks, as users only need to add a layer to a pretrained model and apply FFT to process the input.

4.2 Potential Strategies to Counter Defensive Measures

However, if we can take these defence strategies into account during the training process, we can train the adversarial patch to be robust against defence strategies. One salient feature of patches that are exploited in defence systems is its high frequency. Therefore, it is worthwhile to explore methods which result in a low-frequency patch that is indistinguishable from the rest of the image in this front. This is precisely the reason that TV Loss was used, for it can reduce the high frequencies of the adversarial patches generated, thereby allowing the patch to go undetected by defence systems.

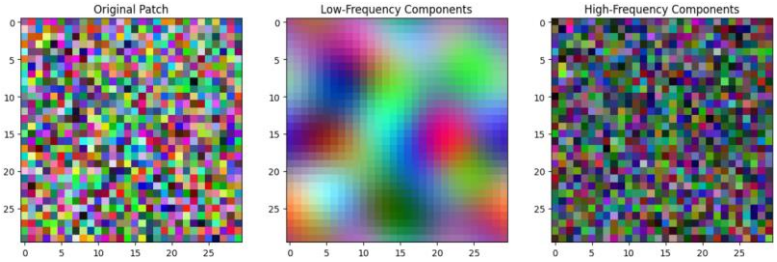


Fig. 4.2.1 Frequency components of a randomly generated patch

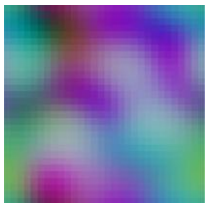


Fig. 4.2.2 patch during optimisation

Integrating FFT can further smoothen the patch. We do this by optimising the low-frequency components of a randomly generated patch as usual (Fig. 4.2.1). To retain the low frequency, the low-frequency components were again extracted and further optimised after 20 iterations. (Fig. 4.2.2) When input into the model we trained, it was unable to detect the semi-optimised patch and was thus unable to enact any further defence strategies (see Fig. M, Appendix).

5. Conclusion and Discussion

In conclusion, we presented a novel method to generate universal adversarial patches against FR systems, leveraging on simulated annealing, 3D differentiable rendering and thin plate spline transformation to ensure universality and effectiveness even when transferred to real-world settings. Results show the effectiveness of the patches produced.

We also evaluated various defence strategies against our patches. Though the defence strategies showed high effectiveness in preventing the adversarial patches from working, when these defence strategies were taken into account during the training process, the adversarial patch was still able to bypass the defence methods and fool the FR system. Defence and attack systems often pose a “chicken-and-egg” problem – if adversarial attacks are successful in bypassing defence systems, researchers must develop even more robust defences that take into account such attacks. However, attackers, in turn, are constantly innovating, creating a continuous cycle of escalation. Researchers are often one step behind, as they cannot predict the next attack strategy with certainty.

In a real-world context, this highlights the critical need for highly robust and adaptable defence systems capable of withstanding both known and unforeseen adversarial techniques. Defence strategies must not only address current vulnerabilities but also anticipate potential avenues for attack. This calls for the development of methods that generalize well across a wide range of adversarial scenarios, including those yet to be conceived.

5.1 Potential Future Work

First, accounting for material properties during 3D differentiable rendering—such as how paper might fold – could improve real-world transferability.

Second, our approach to multiple patches could be expanded, particularly in exploring patch interaction – sizes, locations, and combinations. Research could investigate whether smaller, distributed patches provide equivalent or superior effectiveness compared to a single larger patch by disrupting recognition across multiple feature points. Additionally, overlapping patches or patterns of placement, such as symmetric positioning, could introduce unique vulnerabilities in FR algorithms. Another avenue could be optimising the interaction between patches under physical constraints, such as occlusion or distortion when patches overlap or are viewed from different angles. Using multiple patches can also potentially address the issue of conspicuousness.

Finally, low-frequency patch attacks remain an avenue for future work. Integrating more sophisticated frequency domain extraction methods, like multi-scale frequency analysis during training, and introducing other parameters to ensure robustness. Incorporating techniques like Gaussian splatting could produce more robust and inconspicuous patches.

6. Acknowledgements

We would like to thank our mentor, Dr Shen Bingquan, for his invaluable guidance and insight throughout the research. Additionally, our heartfelt thanks goes out to Mr Sta Maria Jansen Jarret for his counsel. Many thanks, as well, to Mr Triston Chan Sheen, who has generously provided his knowledge, expertise and advice. Our gratitude goes out to Ms Sim Yu Jin Emeline from the Human Resource department, for her assistance in the administrative portion of the project. We thank the members of the DSO intern room for their generous help in our patch testing process.

Appendix

Graph of Loss Against Length of Patch/ pixels

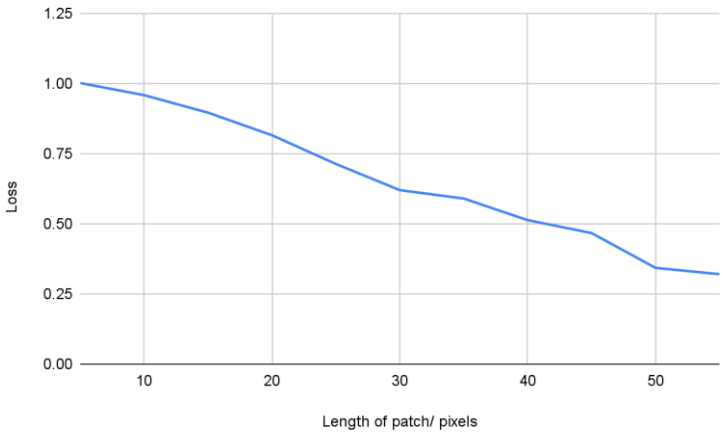


Fig. A

Example Patches During Optimisation

MSE Loss Only:

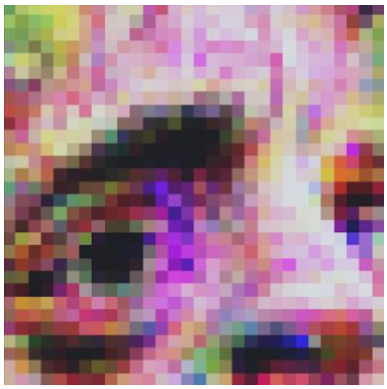


Fig. B.1

Fig. B.2

Fig. B.3

Fig. B

Multi-parameter Loss:

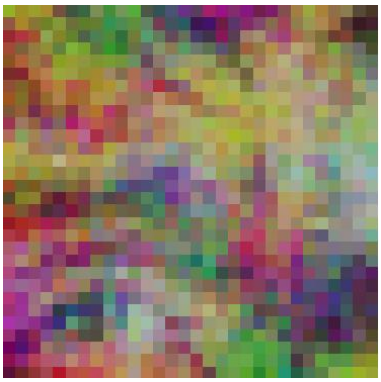


Fig. C.1

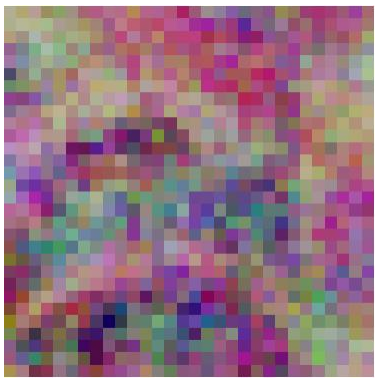


Fig. C.2

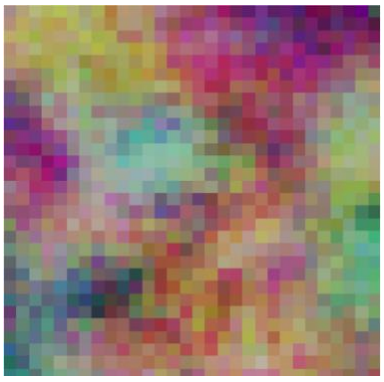


Fig. C.3

Fig. C

Image Frequency Histogram

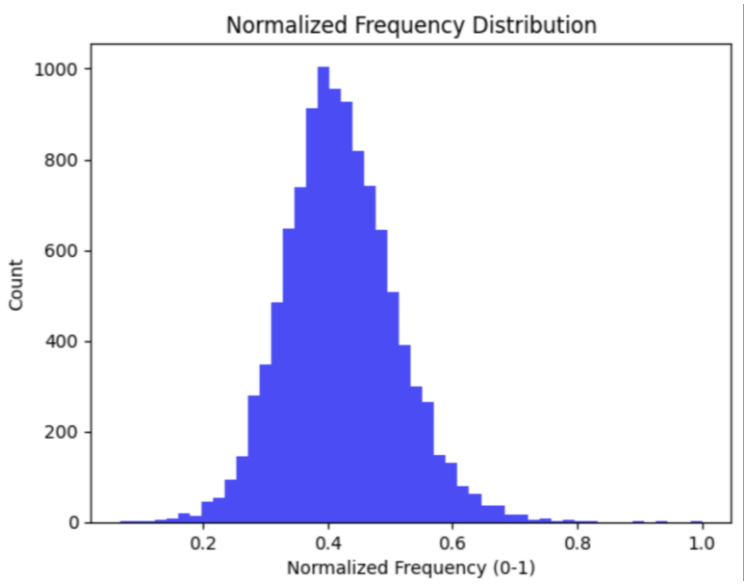


Fig. D

Patches After Optimisation

Single Patches:

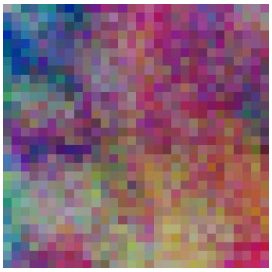
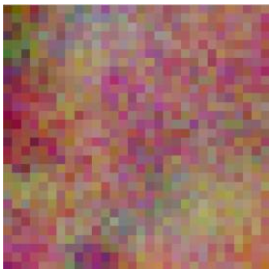
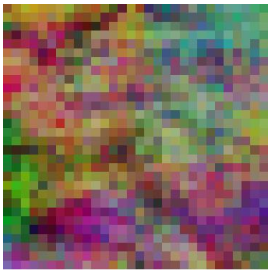
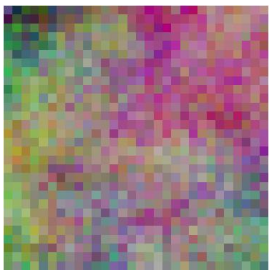
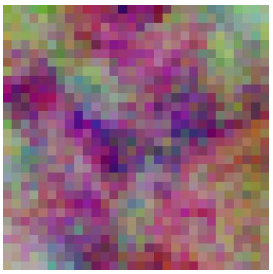
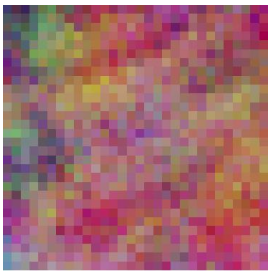
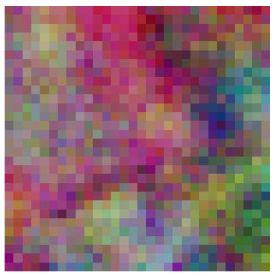
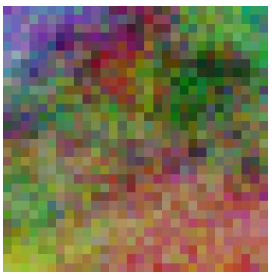
	Random Coordinates	Nose Coordinates	Simulated Annealing Coordinates
2D			
3D			
Thin Plate Spline			

Fig. E

Multiple Patches:

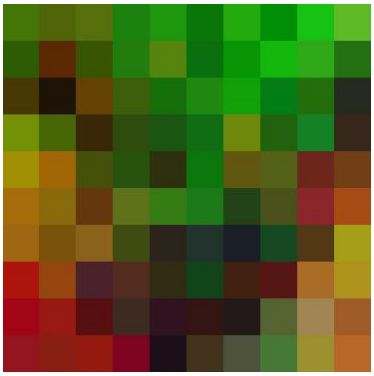
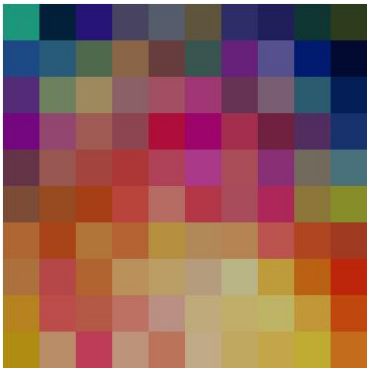
Patch 1 *Vector (-18, 17) from nose*	Patch 2 *Vector (18, 6) from nose*	Patch 3 *Vector (-3, 27) from nose*
		

Fig. F

Reference: Test on Raw Dataset – Unpatched

Average loss: 1.1684080978827861
Percentage misclassified (%): 40.27932960893855

Fig. G

Test on Testing Dataset – Patched (2D)

Single Patches:

	Random Coordinates	Nose Coordinates	Simulated Annealing Coordinates
2D	Average loss: 0.9057503263285752 Percentage misclassified (%): 64.03390483529185	Average loss: 0.8728682492147221 Percentage misclassified (%): 72.52937776921596	Average loss: 0.8801842340562321 Percentage misclassified (%): 74.77981121171257

Fig. H

Multiple Patches:
Average loss (%): 0.9130112315919031
Percentage misclassified: 62.646888846079754

Fig. I

Reference: Test on Real Faces – Unpatched

Average loss: 1.2843632551339955
Percentage misclassified (%): 15.384615384615385

Fig. J

Test on Real Faces – Patched

Single Patches:

	Random Coordinates	Nose Coordinates	Simulated Annealing Coordinates
2D	Average loss: 1.1468529471984277 Percentage misclassified (%): 15.38461538461585	Average loss: 1.1058614758344798 Percentage misclassified (%): 15.384615384615385	Average loss: 1.0090523270460277 Percentage misclassified (%): 38.46153846153847
3D	Average loss: 1.199671376835216 Percentage misclassified (%): 0.0	Average loss: 1.0610311123041005 Percentage misclassified (%): 7.6923076923076925	Average loss: 1.0487112402915955 Percentage misclassified (%): 23.076923076923077
Thin Plate Spline		Average loss: 1.1430250497964711 Percentage misclassified (%): 7.6923076923076925	Average loss: 1.0380303722161512 Percentage misclassified (%): 38.46153846153847

Fig. K

Multiple Patches:
Average loss: 1.1402382117051344
Percentage misclassified: 15.384615384615385

Fig. L

Physical Patch vs Digital Patch

Physical	Digital

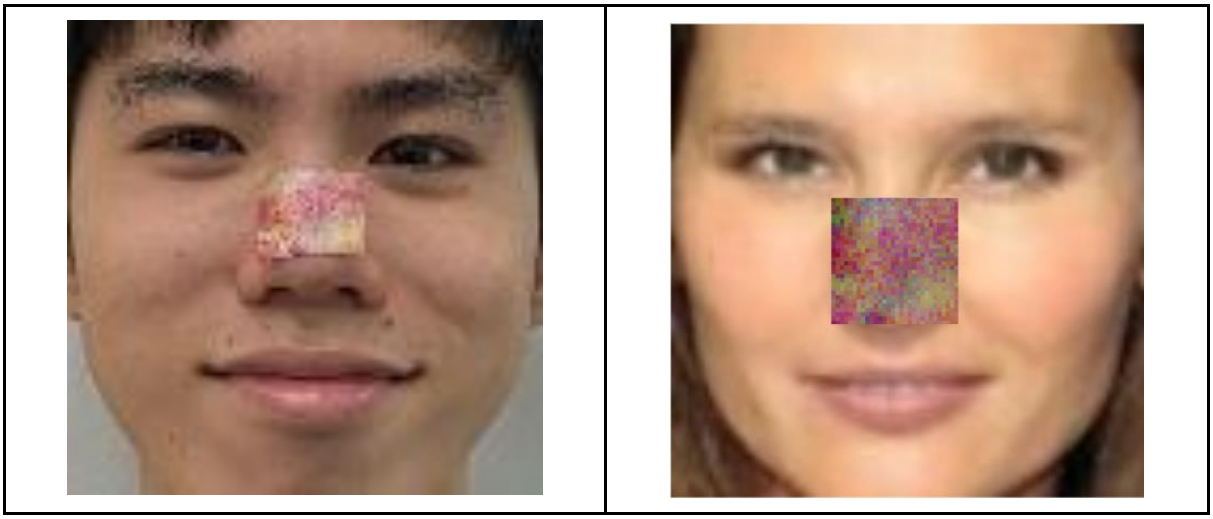


Fig. M

Patch Detection Using Frequency, With Low-Frequency Attack

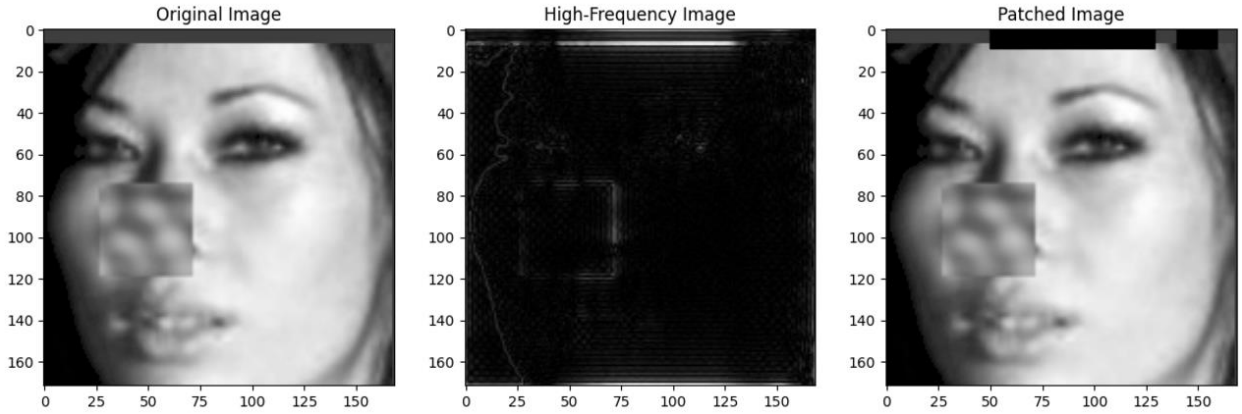


Fig. N

Results of Evaluation of Defence Model

Singular Patch

Simulated Annealing 2D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	2.02
With Augmentation	93.50	0.00
With FFT	92.50	3.03
With Augmentation, With FFT	94.00	2.02

Fig. O.1

Simulated Annealing 3D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	3.03
With Augmentation	92.50	1.01
With FFT	92.50	3.03
With Augmentation, With FFT	91.50	2.02

Fig. O.2

Nose Coordinates 2D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	5.05
With Augmentation	92.50	3.03
With FFT	92.50	5.05
With Augmentation, With FFT	92.50	3.03

Fig. O.3

Nose Coordinates 3D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	3.03
With Augmentation	93.00	4.04
With FFT	92.50	3.03
With Augmentation, With FFT	93.00	4.04

Fig. O.4

Random Coordinates 2D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	2.00
With Augmentation	92.00	4.00
With FFT	92.50	2.00
With Augmentation, With FFT	92.50	5.00

Fig. O.5

Random Coordinates 3D

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	5.00
With Augmentation	93.00	7.00
With FFT	92.50	5.00
With Augmentation, With FFT	92.50	3.00

Fig. O.6

TPS Nose

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	2.02
With Augmentation	94.00	2.02
With FFT	92.50	2.02
With Augmentation, With FFT	92.00	1.01

Fig. O.7

In General

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	3.16
With Augmentation	92.93	3.01
With FFT	92.50	3.31
With Augmentation, With FFT	92.57	2.87

Fig. O.8

Multiple Patches

	Clean Accuracy/ %	Attack Success Rate/%
No Augmentation, No FFT	92.50	11.11
With Augmentation	92.00	15.15
With FFT	92.50	11.11

With Augmentation, With FFT	93.50	17.17

Fig. O.9

References

[1] Mahmood Sharif et al. (2019) A General Framework for Adversarial Examples with Objectives. https://doi.org/10.1145/3317611

[2] Bangjie Yin et al. (2021) Adv-Makeup: A New Imperceptible and Transferable Attack on Face Recognition https://arxiv.org/pdf/2105.03162

[3] Xiao Yang et al. (2020) Design and Interpretation of Universal Adversarial Patches in Face Detection https://arxiv.org/pdf/1912.05021

[4] Zihao Xiao et al. (2021) Improving Transferability of Adversarial Patches on Face Recognition with Generative Models https://openaccess.thecvf.com/content/CVPR2021/papers/Xiao_Improving_Transferability_of_Adversarial_Patches_on_Face_Recognition_With_Generative_CVPR_2021_paper.pdf

[5] Tom B. Brown et al. (2018) Adversarial patch https://arxiv.org/pdf/1712.09665

[6] Phoenix Neale Williams and Ke Li. (2023) CamoPatch: An Evolutionary Strategy for Generating Camouflaged Adversarial Patches https://papers.nips.cc/paper_files/paper/2023/file/d482f1362bd6a8448d7c35e717c7063a-Paper-Conference.pdf

[7] Gianluca Donato and Serge Belongie (n.d.) Approximate Thin Plate Spline Mappings https://cseweb.ucsd.edu/~sjb/eccv_tps.pdf

[8] Hong Liu et al. (2024) Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training https://arxiv.org/pdf/2305.14342

[9] Anish Athalye et al. (2018) Synthesizing Robust Adversarial Examples https://arxiv.org/pdf/1707.07397

[10] Florian Schroff et al. (2015) FaceNet: A Unified Embedding for Face Recognition and Clustering https://arxiv.org/pdf/1503.03832

[11] Wan-Yi Lin et al. (2021) Certified robustness against physically-realizable patch attacks via randomized cropping https://openreview.net/pdf?id=g4NNK4RH715

[12] Ryo Takahashi et al. (2015) Data Augmentation using Random Image Cropping and Patching for Deep CNNs https://arxiv.org/pdf/1811.09030

[13] X. Lei et al. (2023) Using Frequency Attention to Make Adversarial Patch Powerful Against Person Detector https://ieeexplore.ieee.org/document/9923929

[14] Zhun Zhang et al. (2024) Towards a Novel Perspective on Adversarial Examples Driven by Frequency https://arxiv.org/pdf/2404.10202v1

[15] Dominic Simon et al. (2023) Detecting and Removing Adversarial Patches using Frequency Signatures https://openreview.net/forum?id=fsc9GXX6Z6

[16] Chaoqun Li et al. (2024) Prompt-Guided Environmentally Consistent Adversarial Patch https://arxiv.org/pdf/2411.10498

[17] JiaHao Xie et al. (2020) A Random-patch based Defense Strategy Against Physical Attacks for Face Recognition Systems https://arxiv.org/pdf/2304.07822

We acknowledge the use of ChatGPT in some parts of code generation and report writing.