

## Fake Fingerprint Generation Using Edge Connect

CUAU 4기 토치맨과 보안요정들 팀

송유지(컴퓨터공학부), 유승태(소프트웨어학부), 윤서하(전자전기공학부), 이민정(전자전기공학부)

### [요약]

기술의 발달로 생체 인식은 다양한 분야에서 보안방식으로 채택되고 있고 그에 대한 사람들의 신뢰도가 높아지고 있다. 그와 동시에 카메라 기술이 발전하여 손쉽게 고해상도 사진을 찍을 수 있으며, 소셜미디어가 활성화됨에 따라 다른 사람의 사진을 쉽게 접할 수 있게 되었다. 따라서 이는 우연히 사진에 찍힌 생체 정보를 다른 사람이 쉽게 추출할 수 있는 시대가 도래하고, 보안을 위해 쓰인 생체 정보가 오히려 위협을 받을 수 있음을 암시한다. 우리는 보안에서 가장 흔하게 사용되는 생체 정보인 지문을 보호하고자, 사진에 찍힌 지문을 추출하여 가짜인 지문으로 자동으로 바꿔주는 novel end-to-end pipeline을 제안하고자 한다. 지문이 드러나게 찍힌 손 사진을 MediaPipe Hands[1]를 활용하여 손 끝을 추출하고, 이를 Image Inpainting[2] 기법을 활용하여, 결과적으로 높은 해상도를 유지하면서 가짜 지문을 생성하여 생체정보를 보호하는 알고리즘을 제안한다.

### 1. 서론

정보의 중요성이 대두됨에 따라 정보 보안의 방식은 ‘무엇을 가지고 있는가?(e.g. 주민등록증), ‘무엇을 알고 있는가?(e.g. 비밀번호)’에서 ‘누구인가?(e.g. 지문)’로 발전되어왔다. 생체 정보는 그 특성상 쉽게 변경 혹은 교체될 수 없는 정보이기 때문에 해당 정보를 활용한 보안 방식이 보안 시스템의 주류로 활용되고 있다. 하지만 이와 동시에 생체 정보가 위협에 노출되었다는 문제점 또한 존재한다. 촬영 기술의 발달로 카메라에 구애 받지 않고 고해상도의 사진을 찍을 수 있게 되었고, 소셜미디어에 사진을 공유하는 문화가 생활의 일부가 됨에 따라 다른 사람의 사진을 손쉽게 얻을 수 있게 되었다. 실제로 페이스북에 올라온 사진 속 손의 지문으로 범인을 잡는가[3] 하면, 연설을 하는 정치인의 사진에서 그의 지문을 추출하는 사건[4]도 벌어졌다. 이러한 사건들은 우리의 생체 정보가 더 이상 안전하지 않으며 심지어는 악용 가능성도 충분히 있다는 것을 보여준다[5].

이를 해결하기 위한 일 차원적인 방법으로는 사진에서 손가락 지문을 가리거나, 혹은 이 부분을 블러 처리하는 방법이 있다[6]. 하지만 이처럼 왜곡, 모자이크, 블러와 같은 방식으로 지문을 암호화 처리하는 것은 사진에 이질감을 주어 전체적인 품질을 떨어뜨리게 된다. 따라서 Image Inpainting 기법 중 Edge Connect 기법을 활용하면, 진짜 지문의 손가락 형태와 색깔 등을 그대로 유지한 채, 고화질의 가짜 지문 사진을 생성할 수 있다. 결과적으로 우리는 전체적인 사진의 품질을 훼손하지 않으며 생체 인식 보안 문제를 해결할 수 있는 알고리즘을 제안하고자 한다.

관련 연구 중 하나로는 Yoon et al.[6]가 지문 도용 방지 알고리즘을 제안하였다. 이 연구는 해커가 지문을 도용하지 못 할 정도로만 보안을 높이는 것에 초점을 맞추어, 손 끝 지문과 주변 영역을 블러 처리하는 방식을 채택하였다. 사진에서 지문을 추출할 수 없을 정도로 뭉개져 있기 때문에 좋은 지문 보안 방식이라 볼 수 있으나, 주변 영역까지 블러 처리되어 사진의 해상도가 낮아지는 문제점이 있다.

또한 인공지능(Deep Learning, DL) 기법을 이용한 연구도 존재한다. DL은 비선형의 문제를 인공지능으로 해결할 수 있어서 요즘 각광받고 있는 분야이다. 그 중에서도 Generative Adversarial Network(GAN) 기법은 Computer Vision 분야에서 가장 많이 쓰이는 기법이다. Minaee et al.[7]은 GAN을 사용하여 가짜 지문 사진을 생성하는 Finger-GAN을 제안하였다. 하지만 이 모델은 흑백인 지문 사진을 input으로 이용하여 흑백인 가짜 지문 사진을 생성하는 모델이기 때문에 color image에 적용할 수 없다는 문제점이 존재한다.

본 연구는 Image Inpainting 기법 중 Edge Connect를 이용하여 지문 추출 위험성이 높고, 색깔이 있는 이미지의 지문 영역을 고해상도의 가짜 지문으로 대체하는 알고리즘을 제안한다.

### 2. 본론

모델의 파이프라인은 Figure 1과 같이

구성되어있다.

## 2.1 Data Pre-Processing

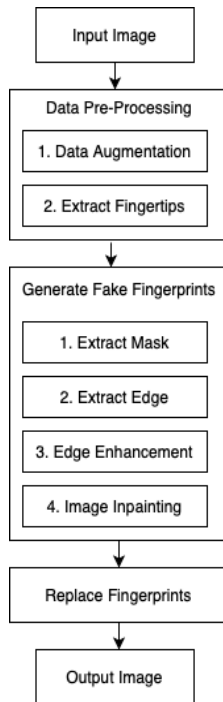


Figure 1 : 전체 파이프라인

### 2.1.1 Data Augmentation

200 여 개의 지문 데이터를 수집하였지만 학습을 하기에는 그 수가 크게 부족하였다. 그렇기 때문에 다양한 augmentation 기법 들을 활용하여 데이터의 숫자를 늘리는 것이 필수적인 과정이다. 그리고 지문의 texture 와 detail 을 보존하는 것이 중요하다 판단하였기에 최대한 정보를 보존할 수 있는 positional, 그리고 color augmentation 방식들을 사용하였다.

Positional augmentation 방법으로는 random crop, flipping, padding, rotation, translation 기법들을 활용하였다. 또한 color augmentation 방법으로는 brightness, contrast, 그리고 saturation 을 수정하여 지문의 고유성을 유지하면서 모델이 일반화될 수 있도록 수정하였다.

### 2.1.2 손가락 끝 위치 추출

사진에서 지문의 디테일을 변경하기 위해서는 지문의 위치를 찾는 것이 필수적이다. Google 의 MediaPipe[1]를 활용하여 손가락 끝의 마디 위치를

파악하고 해당 부분을 잘라냈다. 이때 사진 속 손의 크기가 모두 다르고, 손가락이 겹쳐 있어 지문이 보이지 않는 경우를 제외하기 위해 다음과 같은 방법들을 활용하였다.

다양한 크기의 이미지에 맞춰 올바른 크기의 손 끝부분을 추출하기 위해 MediaPipe 에서 제공하는 landmark(Figure 2) 간의 거리를 활용하였다. 손가락 마지막 마디의 landmark 의 위치와 중간 마디의 landmark 사이의 거리를 계산하여 올바른 손 끝 위치를 파악하였다.

손가락이 겹쳐 있는 경우에는 지문이 아닌 이미지가 추출될 수 있다. 따라서 총 2 가지 방법을 사용하여 이러한 문제가 발생하는 것을 막았다.

첫 번째 방법: 다음 공식을 활용하여 landmark 사이의 각도를 구한다:

$$\theta = \arccos\left(a \cdot \frac{b}{|a||b|}\right) \quad (\text{Eq 1})$$

Eq 1 을 이용하여 손가락의 landmarks 가 일직선 상에 위치하였는지 판단한다.

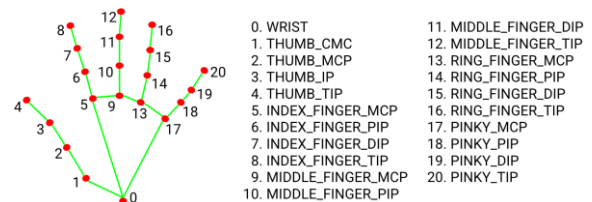


Figure 2 : MediaPipe 21 hand landmarks

두 번째 방법: 손이 정면을 보고 있을 경우에는 첫 번째 방법을 적용할 수 없다. 따라서 landmarks(Figure 2) 의 손가락 끝(4, 8, 12, 16, 20 번)과 기준점(0 번)과의 거리를 손가락 중간 지점(2, 6, 10, 14, 18 번) 과 비교하여 손이 겹쳐 있는지 파악한다.

위와 같은 방식을 통하여 Figure 3 처럼 손이 나온 사진에서 손 끝 부분을 추출할 수 있다.

## 2.2 Create Mask file and Edge file for Model

### 2.2.1 Mask 파일 생성

Edge Connect 모델을 적용할 때, 기존에 사용되는 random mask 를 활용하는 것은 손가락의 지문 뿐 아니라 배경의 무의미한 데이터까지 함께 학습하고 생성하기에 본 연구 방향과 맞지 않았다. 따라서 배경이 아닌 오직 지문이 존재하는 범위만을

추출하는 masking 기법을 고안했다. 손가락 색상의 범위를 먼저 파악하고 그 범위를 넘어가는 부분은 background로 판단하는 방식으로 mask를 만들었다. 이때 색상을 나타내는 기준으로 RGB(빨강, 초록, 파랑), YCbCr(휘도, 색차), HSV(색조, 채도, 명도) 세 가지의 색 표현 방식을 사용하였다.



Figure 3 : MediaPipe 을 이용한 지문 추출 (a) 원본. (b) 추출된 5 개의 손 끝 부분

우선 지문이 가지고 있는 색상의 범위를 파악하기 위하여 사진의 중심으로부터 30 x 30 의 공간을 추출하고 각 channel 의 최대값과 최소값을 찾는다. 최대값과 최소값의 범위 내에 존재하는 범위를 추출하여 마스크로 사용한다. 이때 RGB, YCrCb, HSV 세 가지 모두를 활용하여 필터링 하였을 때에 가장 배경을 잘 구분하는 mask 를 생성할 수 있었다.

## 2.2.2 Edge 파일 생성

### 1) Fingerprint Edge Extraction

지문을 추출하는 과정은 총 5 단계에 걸쳐있다. (Figure 4)

1. 원본 사진에서 밝기와 대비를 조절한다. 밝기=25, 대비=0.8 로 설정하였다.
2. Gray scale 로 변환한다
3. Histogram Equalization 전처리 기법을 적용한다.
4. Normalization 을 적용한다
5. Adaptive Gaussian C Threshold 를 적용한다. 이때 block size=15, c=2 로 설정하였다.

여기서 설정한 수치는 모두 실험을 통해 결정된 수치이다. Figure 4 (f) 가 최종적으로 손가락

이미지에서 지문 edge 를 추출한 결과이다.

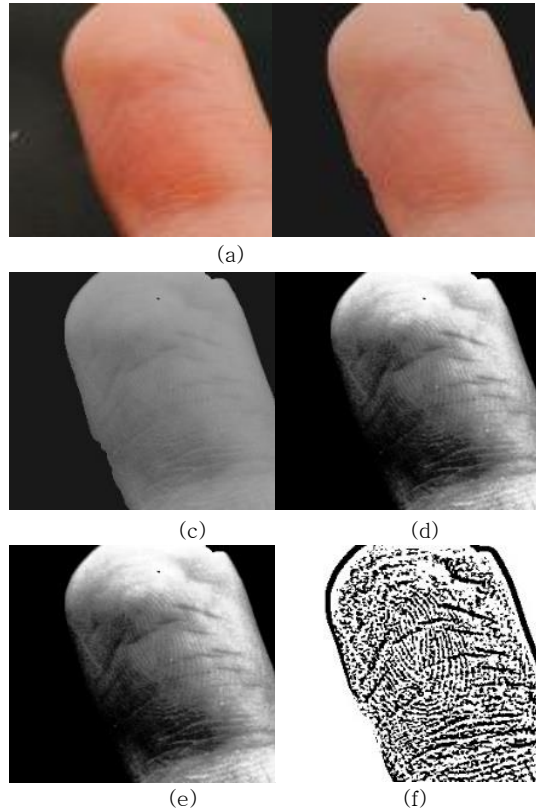


Figure 4 : 지문 edge 추출 순서. (a) 원본. (b) 밝기=25, 대비=0.8로 조절. (c) RGB 이미지를 Gray scale로 변형 (d) Histogram Equalization 전처리 기법을 적용 (e) Normalization. (f) Adaptive Gaussian C threshold 적용 (block size=15, c=2로 설정)

## 2) Fingerprint Enhancement

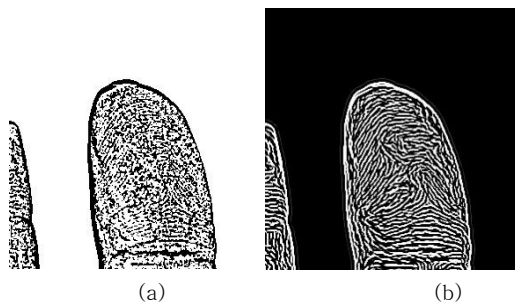


Figure 5 : Fingerprint Enhancement. (a)지문 edge 추출 (b) fingerprint enhancement 적용

지문 edge 추출만으로는 지문이 명확하게 그려지지 않아서 enhancement를 진행하였다. [8] (Figure 5) Fingerprint enhancement 과정은 다음과 같다.

1. Fingerprint Segmentation
2. Estimation of local ridge orientation
3. Estimation of local ridge frequency
4. Fingerprint enhancement

Sobel filter 와 gabor kernel 을 활용해서 enhancement 를 진행하였다. Figure 5 (a)의 지문 추출 이미지에 반해 Figure 5 (b)에서 지문이 더 선명해진 것을 확인할 수 있다.

## 2.3 Model

### 2.3.1 Edge Connect

Image Inpainting 을 진행하여 가짜 지문을 생성한다. EdgeConnect[9] Model 코드를 활용하여 진행하였다.

EdgeModel 과 InpaintModel 은 각각 CelebA[10] 데이터를 활용해서 2,000,000 iteration 만큼 pretrain 을 진행한 모델을 활용하였다.

#### 1) 기존 모델 학습(Edge + Inpaint)

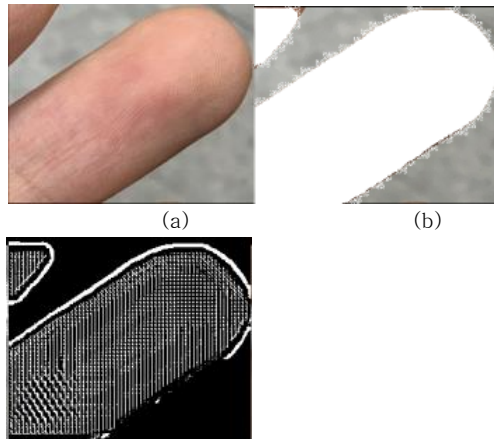


Figure 6 : 기존 Edge Connect 모델 학습. (a) 원본. (b) Masking image. (c) Edge Map

기존 Edge-Connect 모델의 진행 순서는 다음과 같다.

1. Mask 파일, Mask 가 적용된 Edge 파일, GrayScale 된 이미지에 Mask 가 적용된 파일을 활용해서 EdgeModel 을 학습시킨다. 결과물로 Masking 된 부분의 Edge 를 Model 이 직접 그린 Edge Map 이 생성된다. (Figure 6 (c))
2. 생성된 Edge Map 을 활용해서 Inpainting Model 을 학습시킨다. 그 결과 원본 이미지의 Masking 된 부분을 Edge Map 을 활용해서 Inpainting 한 결과물이 생성된다.

EdgeModel 과 InpaintModel 은 1 epoch 이상을 할 경우 overfitting 이 일어나서 1 epoch 만

진행하였다. 하지만 지문 edge 를 EdgeModel 에 학습시킬 경우 생성된 Edge Map 에서 지문이 지워지는 문제점이 발생하여 학습을 중단하였다.

#### 2) 최종 사용 모델 학습(Inpaint)

따라서 EdgeModel 을 사용하지 않고

InpaintModel 만 사용해서 학습을 진행하였다. 기존 지문 Edge file 또한 2.2.2 의 과정을 거쳐 생성된 Edge 이기 때문에 EdgeModel 을 사용하지 않고도 가짜 지문을 생성하는 효과를 낼 수 있었다 (Figure 7). InpaintModel 도 마찬가지로 1 epoch 이상 진행하면 overfitting 현상이 보여서 1 epoch 만큼 학습을 진행했다. 최종적으로 원본 이미지와 생성된 Edge 파일, Mask 파일을 활용해서 가짜 지문을 생성하였다.

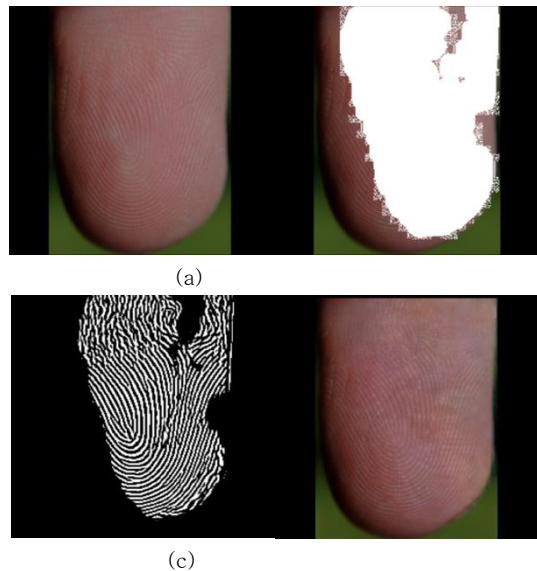


Figure 7 : 최종 모델 학습. (a) 원본. (b) Masking image (c) Edge (d) Inpainting 결과

### 2.3.2 Other works

EdgeConnect Model 이외에 다른 Inpainting 모델도 시도해보았다.

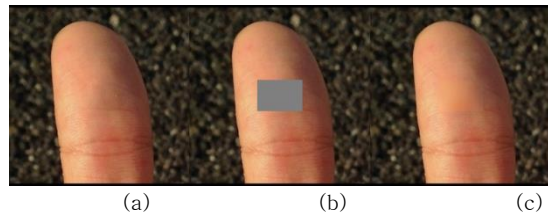


Figure 8 : Generative Inpainting 을 활용한 결과 (a) 원본. (b) Masking image (c) result



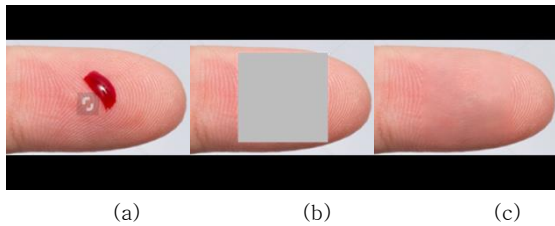


Figure 9 : Shift-Net 을 활용한 결과 (a) 원본. (b) Masking image (c) result

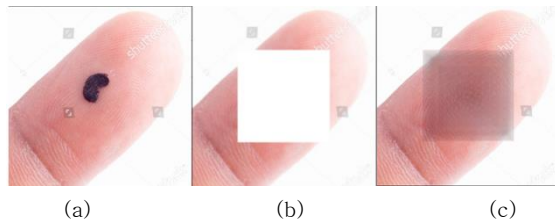


Figure 10 : RFR-Inpainting 을 활용한 결과 (a) 원본, (b) Masking image (c) result

Figure 8, 9, 10 은 각각 Generative Inpainting[11], Shift-Net[12], 그리고 RFR-Inpainting[13]을 적용한 결과이다. Generative Inpainting 은 결과가 좋은 이미지도 분명 존재하였지만, 결과가 좋지 않은 경우 마스크를 씌운 부분을 불러 처리하는 것에 지나지 않았다 (Figure 8 (c)). Shift-Net 또한 마스크를 씌운 부분을 불러 처리하는 정도의 결과만 보여주었다 (Figure 9 (c)). RFR-Inpainting 모델은 손의 색깔을 정확히 잡지 못하는 결과를 냈다 (Figure 10 (c)). 여러 모델에서 나온 결과를 수치적으로 환산해서 무엇이 더 본 연구에 적합한 모델인지 판단할 방법이 없었기 때문에 최종적으로 눈으로 보기에 가장 괜찮은 모델인 Edge Connect 를 적용시키기로 하였다. 추가적으로 Edge 를 사용하여 inpainting 을 진행하기 때문에 본 연구의 방향과 더 맞을 것이라 판단하였다.

### 3. 결 론

개인 정보 유출 위험성이 증가함에 따라 생체 정보 유출에 대한 우려가 높아졌고, 따라서 이를 방지하기 위한 프로그램이 필요해졌다. 그리하여 우리는 지문이 노출되어 있는 이미지를 이용하여 사진의 해상도는 유지하며 지문을 다른 지문으로 대체하는 Fake Fingerprint Generator 를 제안한다.

Figure 11 와 12 에서 보이는 것 같이 높은 해상도를 유지하며 가짜 지문으로 잘 대체된 것을 확인할 수 있다. 하지만 몇몇 아쉬운 실험결과 또한



Figure 11: (왼쪽) 원본. (오른쪽) 가짜 지문으로 대체된 이미지

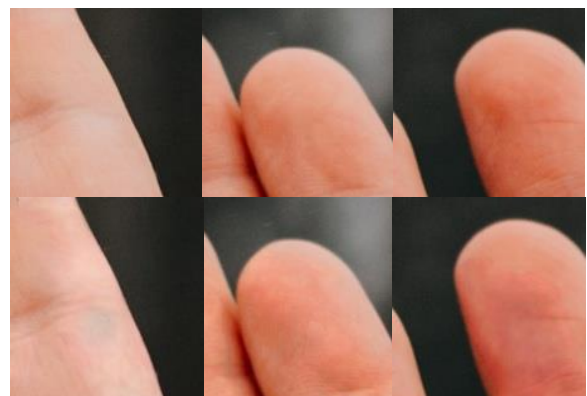


Figure 12: (윗 줄) Figure 11 (a) 에서 추출한 5 개의 손 끝 부분. (아랫줄) 생성된 가짜 지문

존재하였는데 우리가 분석한 failure case 의 원인은 다음과 같다. 첫 번째로 지문의 위치를 못 잡는 경우, 두 번째로는 지문과 배경을 구분하지 못 하는 경우에 failure case 가 간혹 발생하는 것을 확인하였다. 이러한 문제점을 해결하기 위해서는 보다 정교한 생성자 모델의 필요성을 느꼈고, 그렇기 때문에 이미지의 context 를 잘 이해할 수 있는 모델을 연구해볼 예정이다.

한편, 해당 모델의 예상치 못한 성과가 있었는데 지문 이외에도 손가락 마디 주름 등의 디테일 또한 잘 생성하는 결과를 확인했다. 따라서 지문 이외의 도메인에 대한 연구를 통하여 지문 뿐 아니라 다양한 생체 정보 보안 분야에 활용될 수 있을 것이라 기대한다.

## 참고 문헌

- [1] MediaPipe,  
<https://google.github.io/mediapipe/>
- [2] Pathak, Deepak et al. "Context Encoders: Feature Learning by Inpainting." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 2536–2544.
- [3] 페이스북에 올린 사진으로 잡힌 마약상,  
<https://www.google.co.kr/amp/s/www.bbc.com/news/uk-england-merseyside-57226165.amp>
- [4] 정치인의 지문이 추출된 사건,  
<https://www.bbc.com/news/technology-30623611>
- [5] Pathak, Deepak et al. "Context Encoders: Feature Learning by Inpainting." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 2536–2544.
- [6] W.-S. Yoon and S.-G. Kim, "고해상도로 찍은 이미지에서의 손가락 지문 채취 방식에 관한 연구," *Journal of Convergence for Information Technology*, vol. 10, no. 6, pp. 19–27, Jun. 2020.
- [7] Shervin Minaee and Amirali Abdolrashidi. 2018. Finger-GAN: Generating Realistic Fingerprint Images Using Connectivity Imposed GAN. (2018). arXiv:cs.CV/1812.10482
- [8] Hands on Fingerprint Recognition with OpenCV and Python,  
[https://www.comp.hkbu.edu.hk/wsb2021/lecturer\\_details.php?lect\\_id=2](https://www.comp.hkbu.edu.hk/wsb2021/lecturer_details.php?lect_id=2)
- [9] Nazeri, Kamyar, et al. "Edgeconnect: Generative image inpainting with adversarial edge learning." arXiv preprint arXiv:1901.00212 (2019).
- [10] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [11] Yu, Jiahui, et al. "Generative image inpainting with contextual attention." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [12] Yan, Zhaoyi, et al. "Shift-net: Image inpainting via deep feature rearrangement." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
- [13] Li, Jingyuan, et al. "Recurrent feature reasoning for image inpainting." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.