Attacking On Disrupting-Deepfakes: Methodology Transferability And Perturbation Removal

Group3: Ting-Chih Chen and Xiao Guo

Agenda

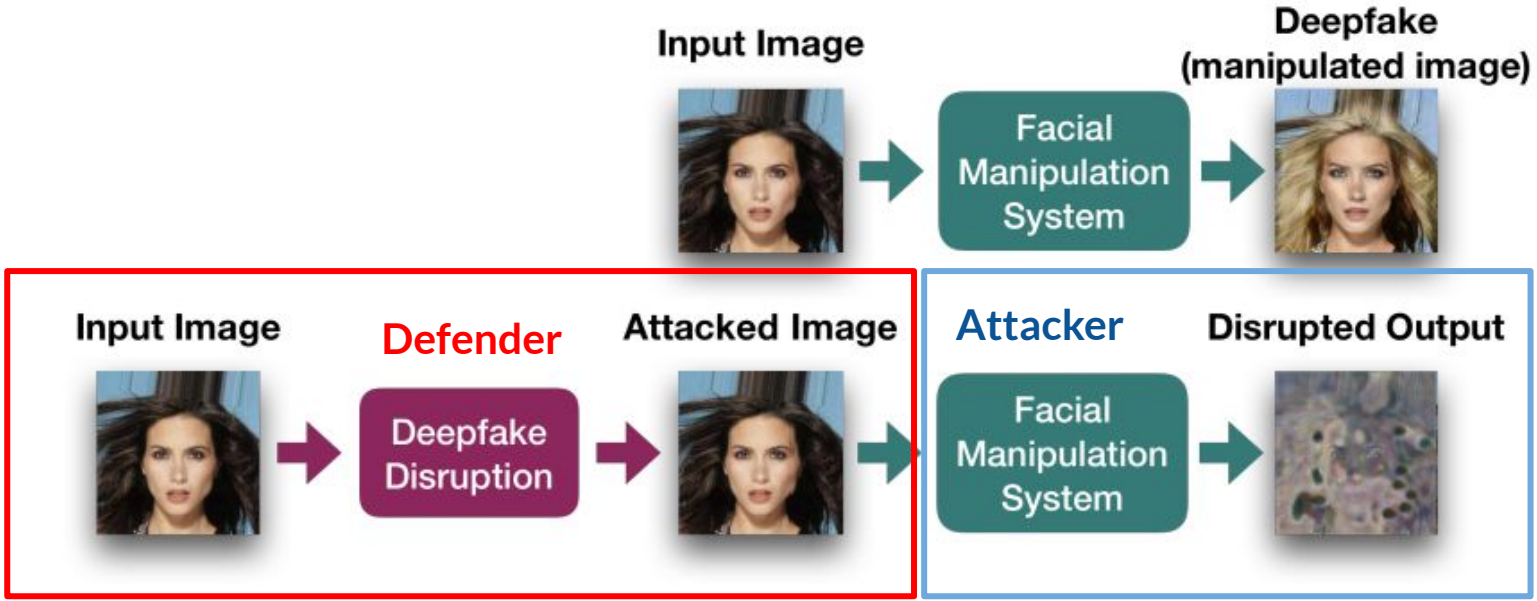
- Introduction
- Recap and motivation
- Transferability
  - Assumption
  - Model#1 : Few-shot face translation
  - Model#2 : GHOST
  - Model#3 : StyleGAN-NADA
- Perturbation Removal
  - Motivation
  - Auto-encoder
  - Restormer
- Discussion

Introduction & motivation

- GAN-Based Deepfake models has caused many issues impacting individuals privacy, safety and properties.
- Defense strategy has been developed: *Detection* and *Defense*
- Defense by adding invisible perturbation: ***Disrupting Deepfake***
- **Our Target**: as the **attacker**(who wants to achieve perfect Deepfake), we need to destroy the perturbation based-defense by the **defender**(who wants to prevent Deepfake)

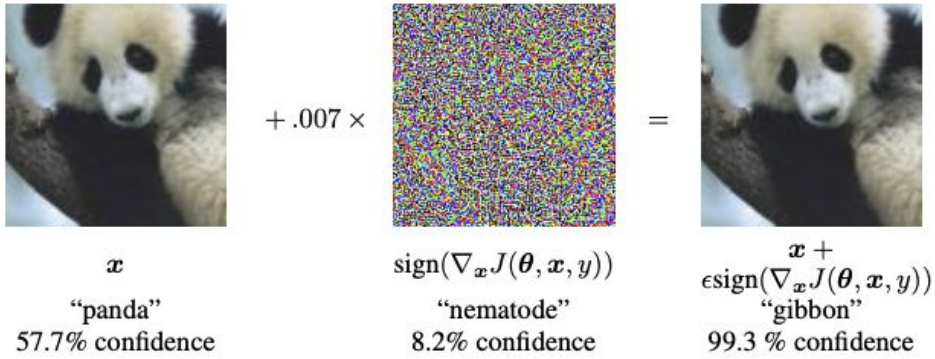
Recap

- Target paper: Disrupting Deepfakes: Adversarial Attacks Against Conditional Image Translation Networks and Facial Manipulation Systems



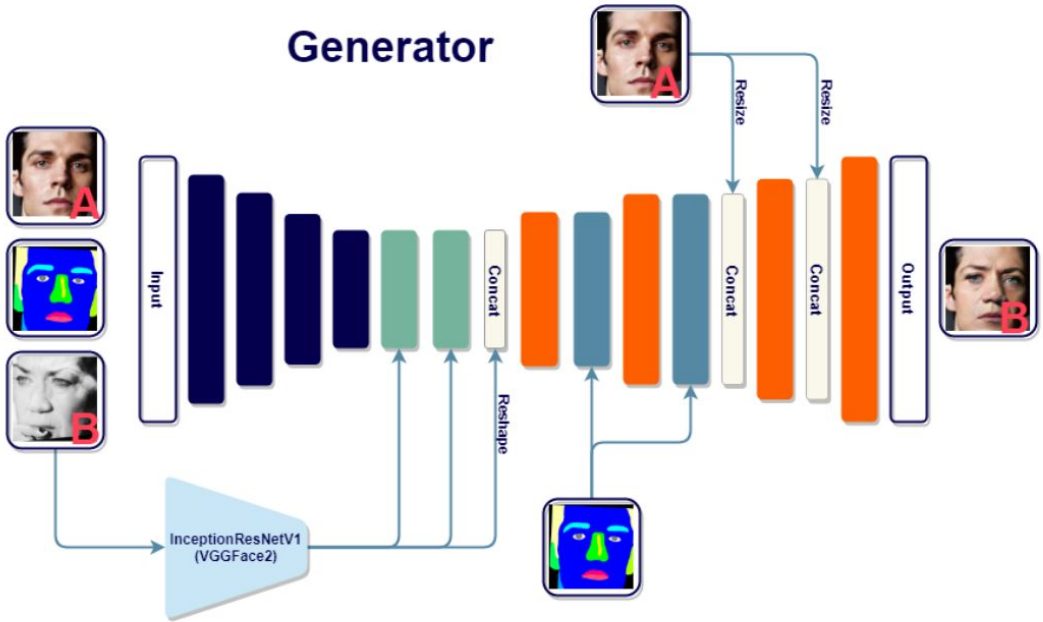
Task#1: Transferability

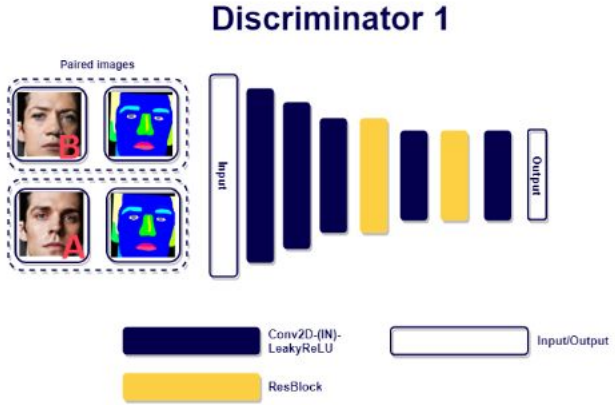
- Author's defense scheme needs to utilize the image translation model that the attacker choose to conduct either PGD/FGSM attack
- However, which model to use is not decided by the defenders(authors)
- Deepfakes Models:
  - Few-shot face translation
  - GHOST
  - StyleGAN-NADA



Target images

Few-shot face translation

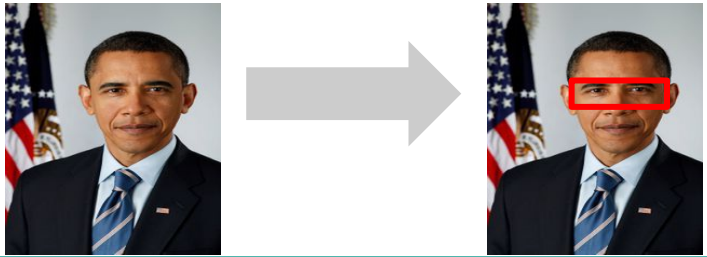




- FC Adaptive instance normalization
- Spatially-adaptive normalization

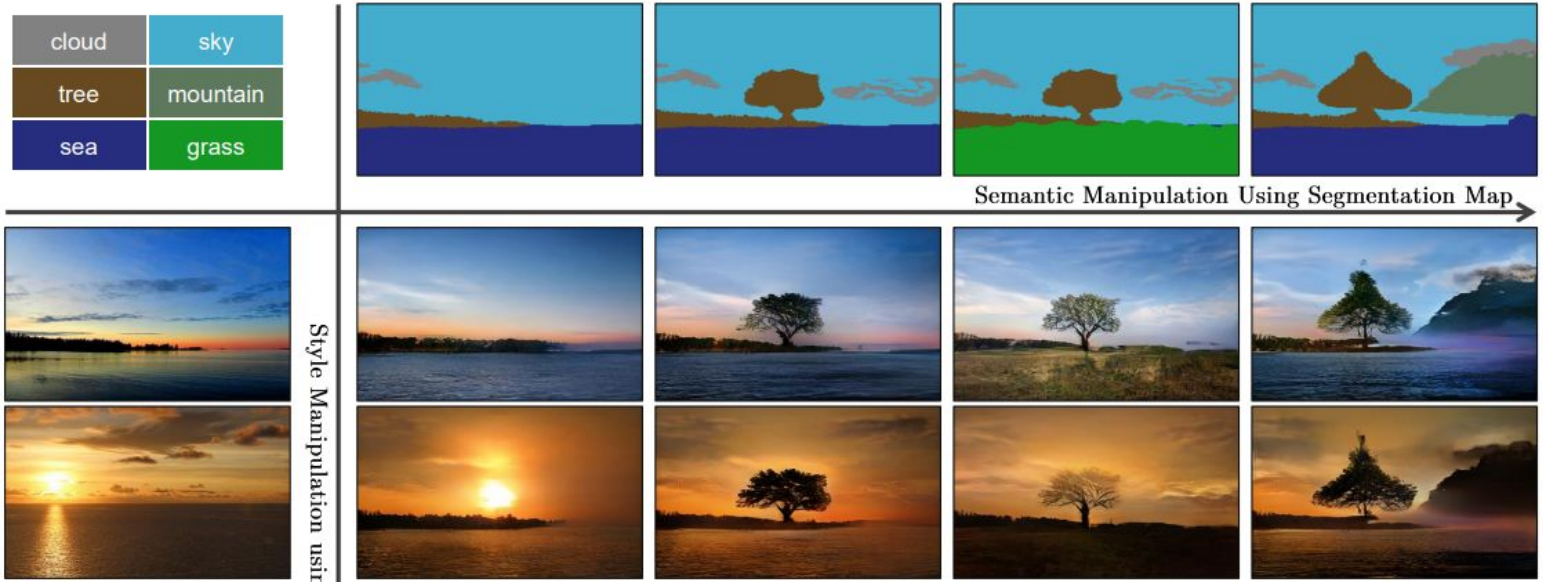
Adaptive instance normalization(FC-AdaIN)

- The AdaIN residual block is a residual block using the AdaIN as the normalization layer
- AdaIN normalizes the activations of a sample in each channel to have a zero mean and unit variance
- Then, it scales the activations using a learned affine transformation consisting of a set of scalars and biases
- The affine transformation can be used to **obtain global appearance info**
- Ex:
    - Latent representation(object appearance) -> Decoder with AdaIN -> obtain the content image(locations of eyes)



Spatially-adaptive normalization(SPADE)

- SPADE is a layer for synthesizing photorealistic images given an input semantic layout



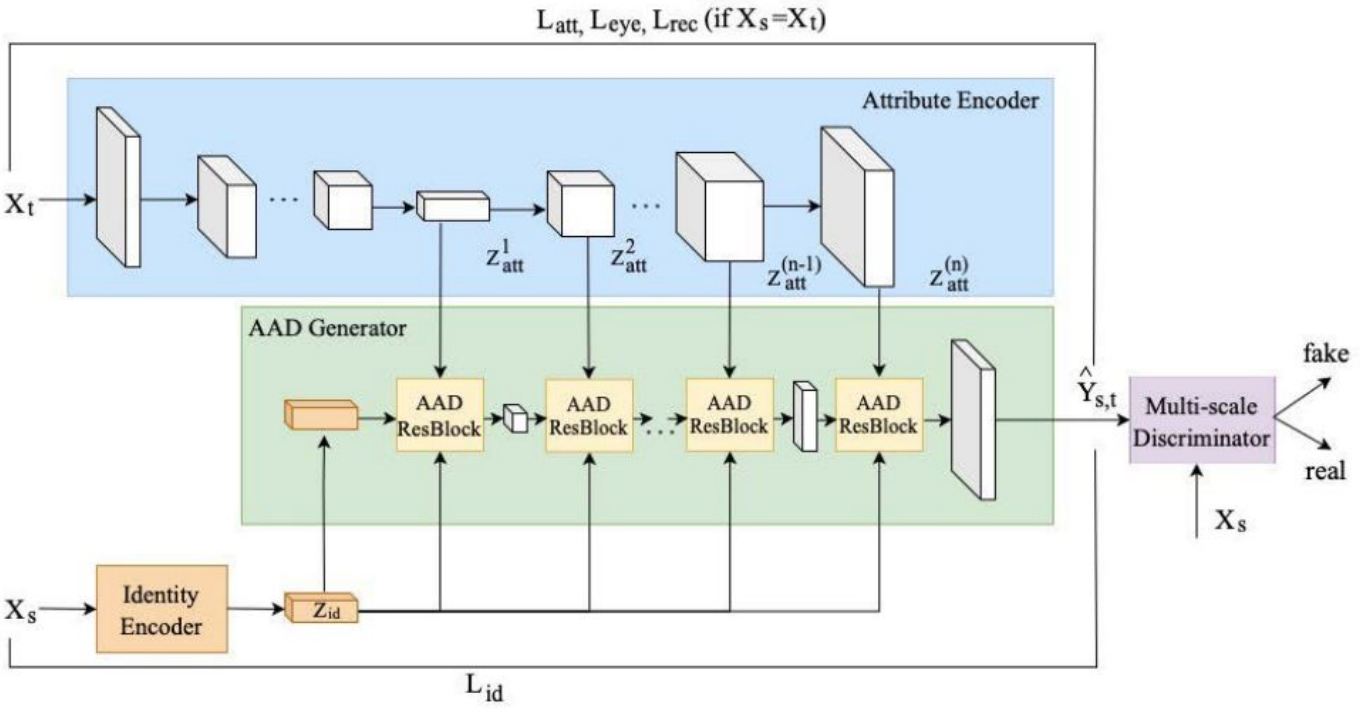
Few-shot face translation Results

First row is the source images from CelebA. The second row images are DeepFake outputs with source images. The last row images are DeepFake outputs with perturbation images.

GHOST

- **Identity encoder** is a pre-trained ArcFace model that extracts the features from the source images
- **Attribute encoder** is a model with a U-Net architecture that extracts attributes from the target images
    - U-Net is fully convolutional network. Authors think this can yield more precise segmentations
- **AAD generator** is a model to mix attributes and the identities. Then, it will generate new face with source identity and target attribute features

GHOST Architecture



GHOST Results

First row is the source images from CelebA. The second row images are DeepFake outputs with source images. The last row images are DeepFake outputs with perturbation images.

StyleGAN-NADA

- CLIP-Guided Domain Adaptation of Image Generators
- StyleGAN-NADA converts a pre-trained generator to new domains using only a textual prompt and no training data
- **Face Swapping** by setting target images

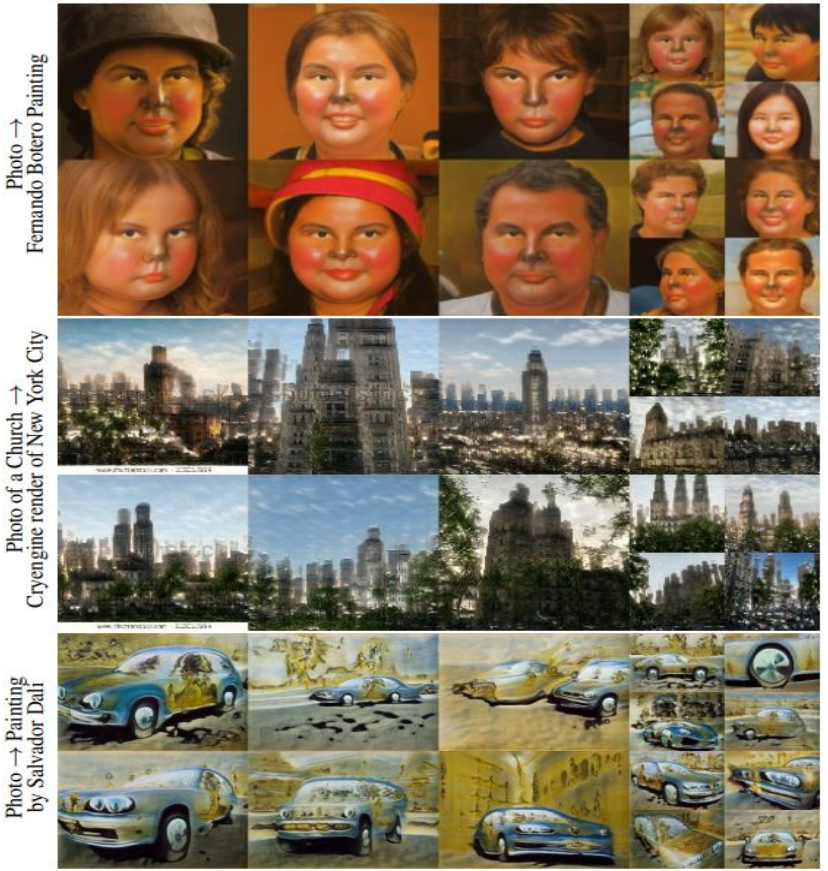


Fig 1. Examples of text-driven, out-of-domain generator adaptations induced by our method[1]

[1]Gal, Rinon, et al. "Stylegan-nada: Clip-guided domain adaptation of image generators." ACM Transactions on Graphics (TOG) 41.4 (2022): 1-13.

StyleGAN-NADA-Architecture

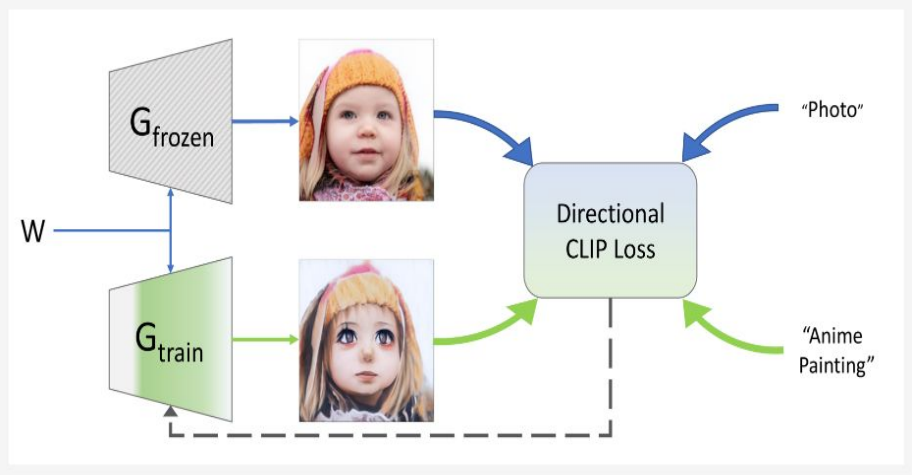


Fig 1. Model Concept[1]

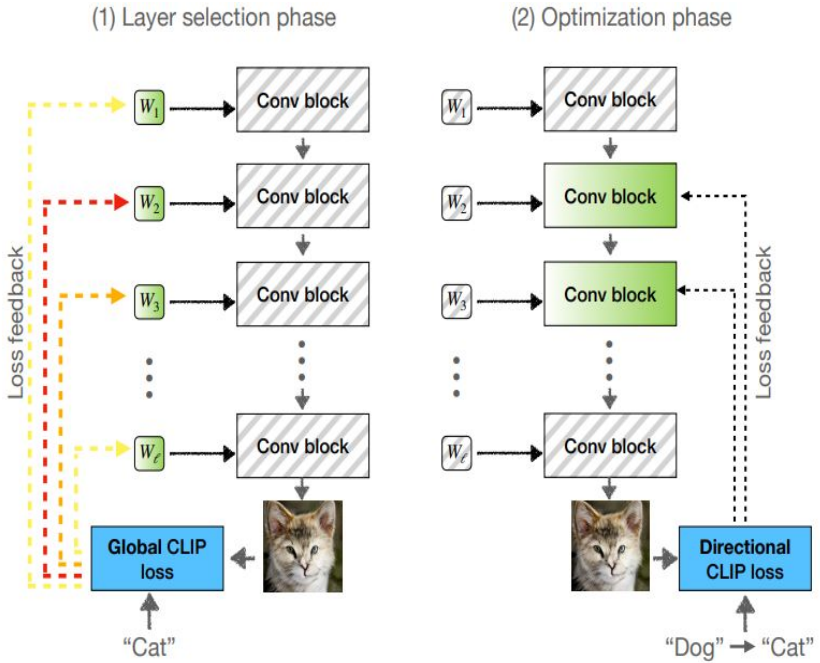


Fig 2. The adaptive layer-freezing mechanism has two phases[1]

[1]Gal, Rinon, et al. "Stylegan-nada: Clip-guided domain adaptation of image generators." ACM Transactions on Graphics (TOG) 41.4 (2022): 1-13.

StyleGAN-NADA-Results

Base Image Unprotected Perturbed

Transferability Evaluation

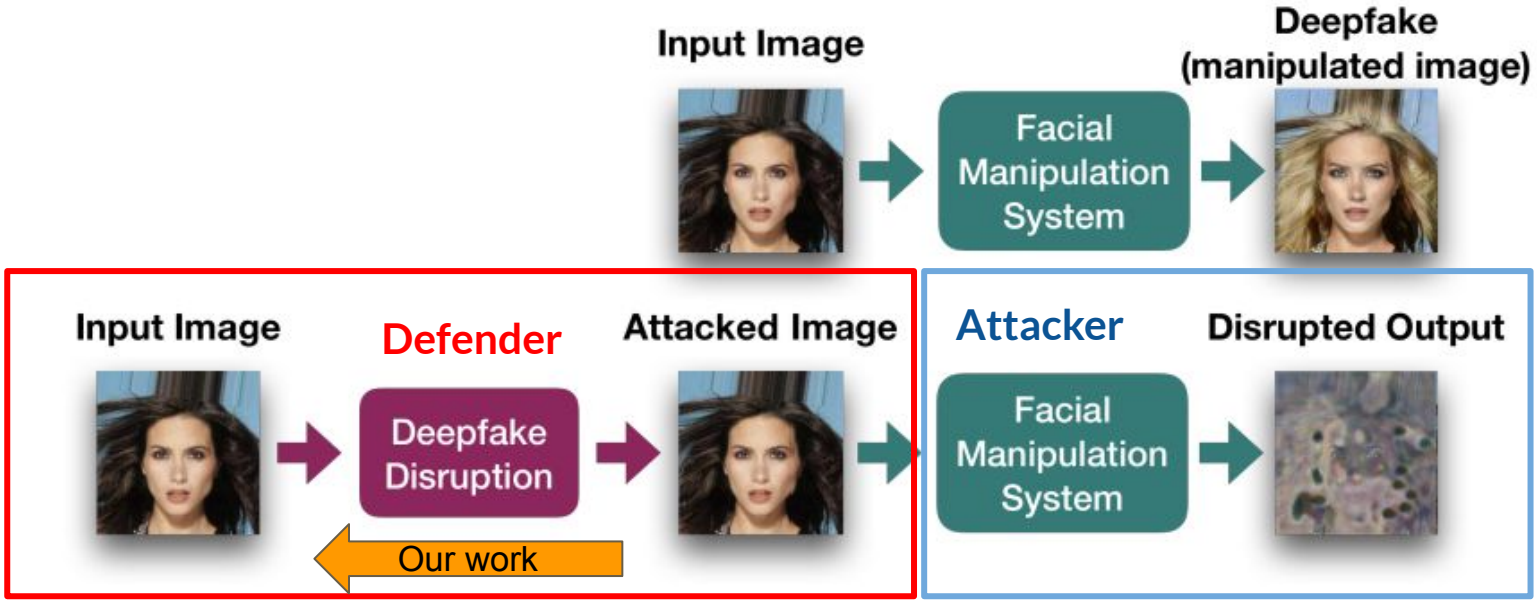
- MSE
  - the most widely used and also the simplest full reference metric which is calculated by the squared intensity differences of distorted and reference image pixels
- SSIM(structural similarity index measure)
  - gives normalized mean value of structural similarity between the two images, considers luminance, contrast, structure
- PSNR(peak signal-to-noise ratio)
  - the ratio between the maximum possible signal power and the power of the distorting noise which affects the quality of its representation

Transferability Evaluation

	MSE	SSIM	PSNR
GHOST	**0.737**	**0.998**	**50.396**
Few-shot	57.684	0.951	32.295
StyleGAN-NADA	247.152	0.777	25.259
StyleGAN-Baseline	226.91	0.799	26.611

Task#2: Perturbation Removal

- Restore the perturbed image to unprotected version->(Re)enables Deepfake



Auto-encoder

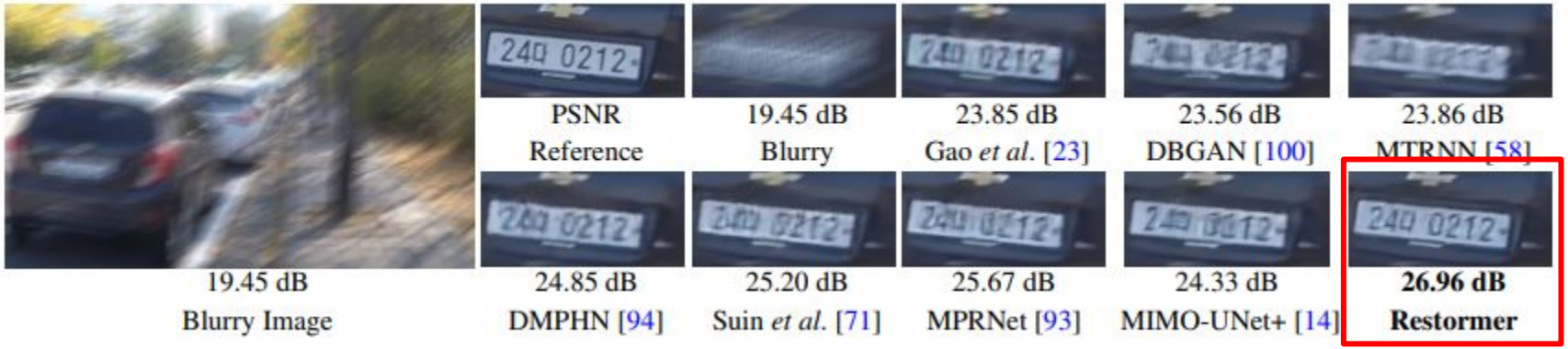
- Naive Idea of **Denoising autoencoder** to remove the perturbation
- Trained on 1000 images from the original paper's data pipeline(CelebA)
- 8 conv layers, Average pooling, 720k parameters
- Blurred Output: **Super resolution autoencoder**
- 10 conv layers, max pooling & up-samling, 1.1m parameters

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0

 conv2d (Conv2D)             (None, 256, 256, 64)      4864

 average_pooling2d (AveragePooling2D)  (None, 128, 128, 64)      0

 conv2d_1 (Conv2D)           (None, 128, 128, 64)      102464

 average_pooling2d_1 (AveragePooling2D)  (None, 64, 64, 64)       0

 conv2d_2 (Conv2D)           (None, 64, 64, 64)        102464

 average_pooling2d_2 (AveragePooling2D)  (None, 32, 32, 64)       0

 conv2d_3 (Conv2D)           (None, 32, 32, 64)        102464

 average_pooling2d_3 (AveragePooling2D)  (None, 16, 16, 64)       0

 conv2d_transpose (Conv2DTranspose)  (None, 32, 32, 64)       102464

 conv2d_transpose_1 (Conv2DTranspose)  (None, 64, 64, 64)       102464

 conv2d_transpose_2 (Conv2DTranspose)  (None, 128, 128, 64)     102464

 conv2d_transpose_3 (Conv2DTranspose)  (None, 256, 256, 64)     102464

 conv2d_4 (Conv2D)           (None, 256, 256, 3)       4803

=================================================================
Total params: 726,915
Trainable params: 726,915
Non-trainable params: 0
_________________________________________________________________
```

```
Model: "model_4"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 input_5 (InputLayer)        [(None, 256, 256, 3)]     0         []

 conv2d_35 (Conv2D)          (None, 256, 256, 64)      1792      ['input_5[0][0]']

 conv2d_36 (Conv2D)          (None, 256, 256, 64)      36928     ['conv2d_35[0][0]']

 max_pooling2d_6 (MaxPooling2D)  (None, 128, 128, 64)      0         ['conv2d_36[0][0]']

 conv2d_37 (Conv2D)          (None, 128, 128, 128)     73856     ['max_pooling2d_6[0][0]']

 conv2d_38 (Conv2D)          (None, 128, 128, 128)     147584    ['conv2d_37[0][0]']

 max_pooling2d_7 (MaxPooling2D)  (None, 64, 64, 128)       0         ['conv2d_38[0][0]']

 conv2d_39 (Conv2D)          (None, 64, 64, 256)       295168    ['max_pooling2d_7[0][0]']

 up_sampling2d_6 (UpSampling2D)  (None, 128, 128, 256)     0         ['conv2d_39[0][0]']

 conv2d_40 (Conv2D)          (None, 128, 128, 128)     295040    ['up_sampling2d_6[0][0]']

 conv2d_41 (Conv2D)          (None, 128, 128, 128)     147584    ['conv2d_40[0][0]']

 add_6 (Add)                 (None, 128, 128, 128)     0         ['conv2d_41[0][0]',
                                                                  'conv2d_38[0][0]']

 up_sampling2d_7 (UpSampling2D)  (None, 256, 256, 128)     0         ['add_6[0][0]']

 conv2d_42 (Conv2D)          (None, 256, 256, 64)      73792     ['up_sampling2d_7[0][0]']

 conv2d_43 (Conv2D)          (None, 256, 256, 64)      36928     ['conv2d_42[0][0]']

 add_7 (Add)                 (None, 256, 256, 64)      0         ['conv2d_43[0][0]',
                                                                  'conv2d_36[0][0]']

 conv2d_44 (Conv2D)          (None, 256, 256, 3)       1731      ['add_7[0][0]']

==================================================================================================
Total params: 1,110,403
Trainable params: 1,110,403
Non-trainable params: 0
__________________________________________________________________________________________________
```

Restormer

- Restormer is a network based on transformer to do image restoration
  - Deblurring, Deraining, Gaussian Denoising and Real Denoising
- We assume we can use Restormer to remove invisible perturbation



Restormer Architecture

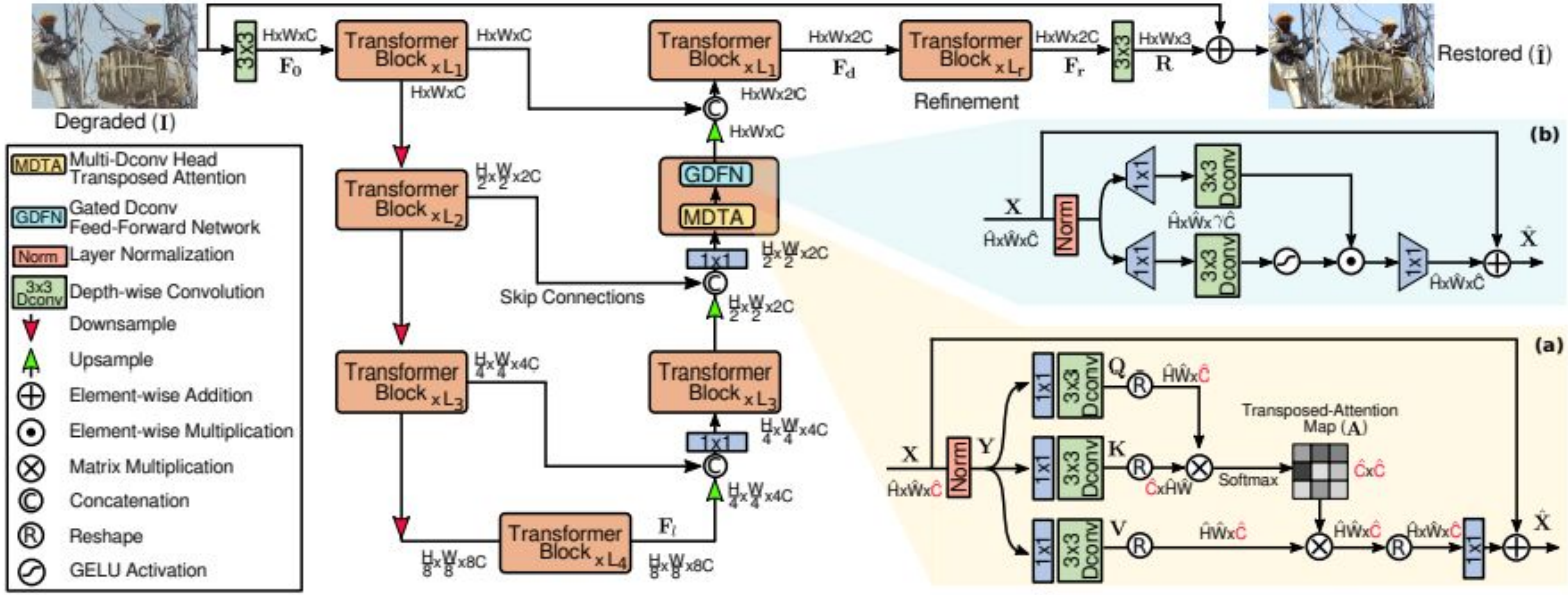
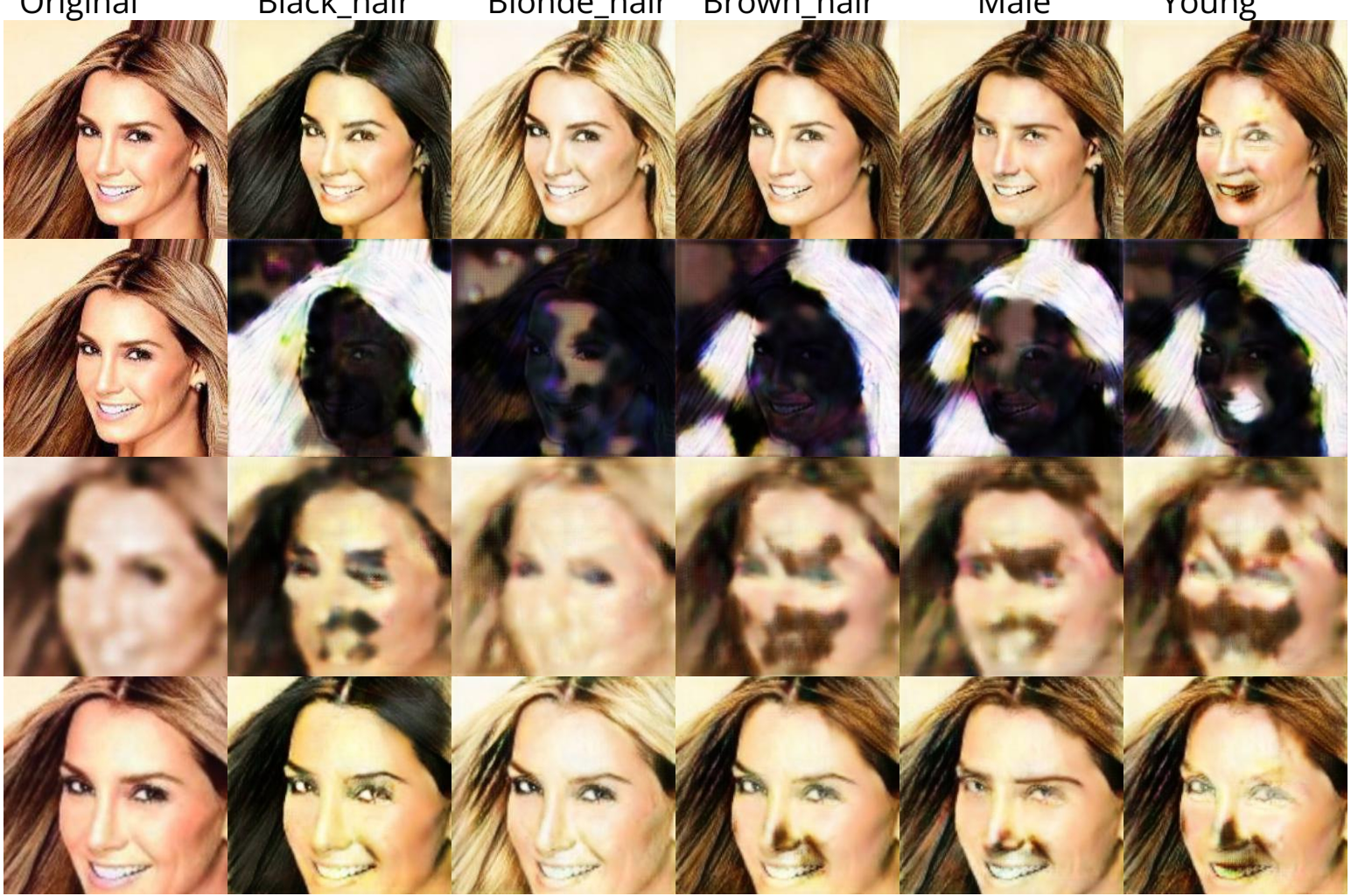
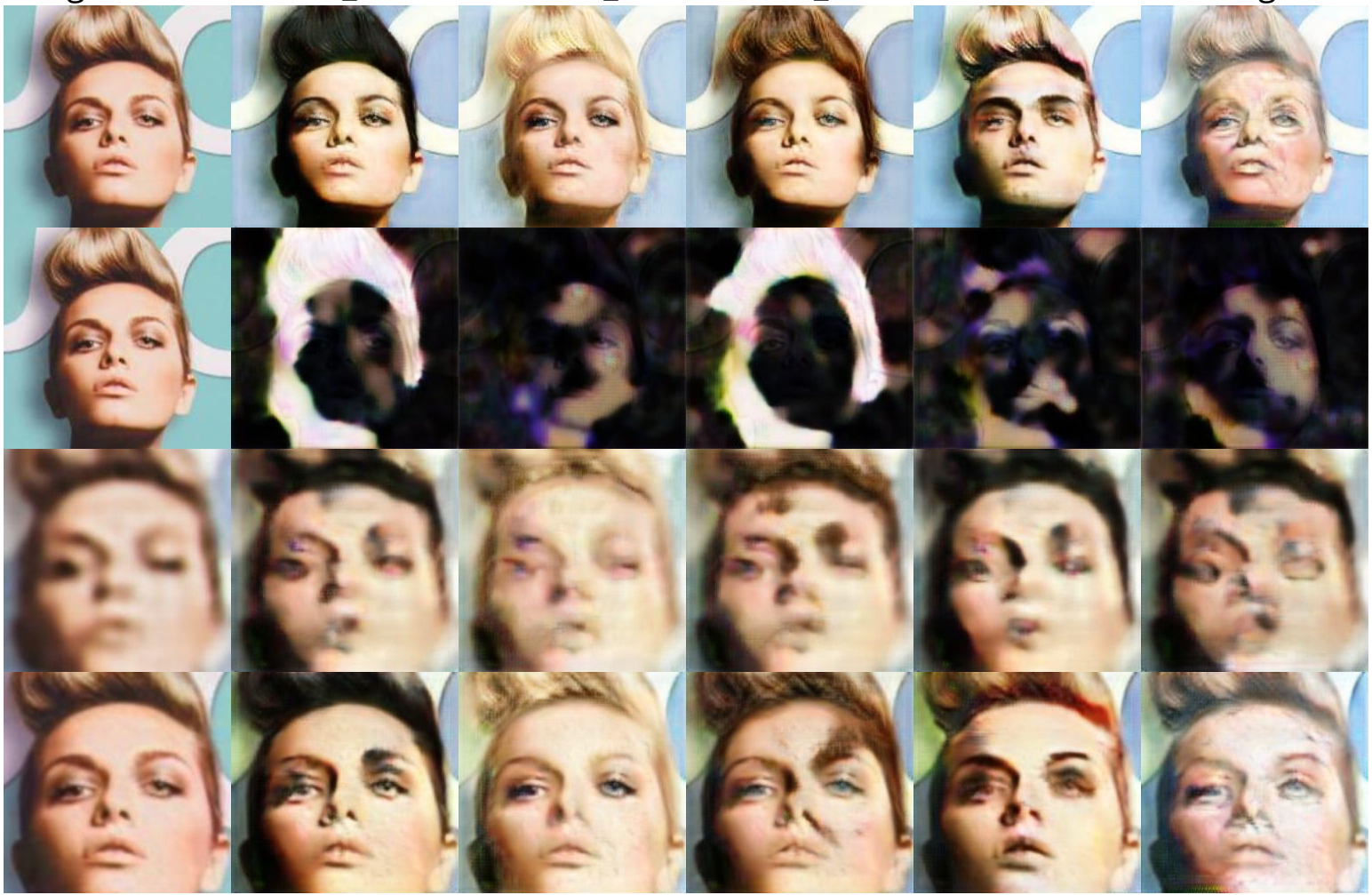
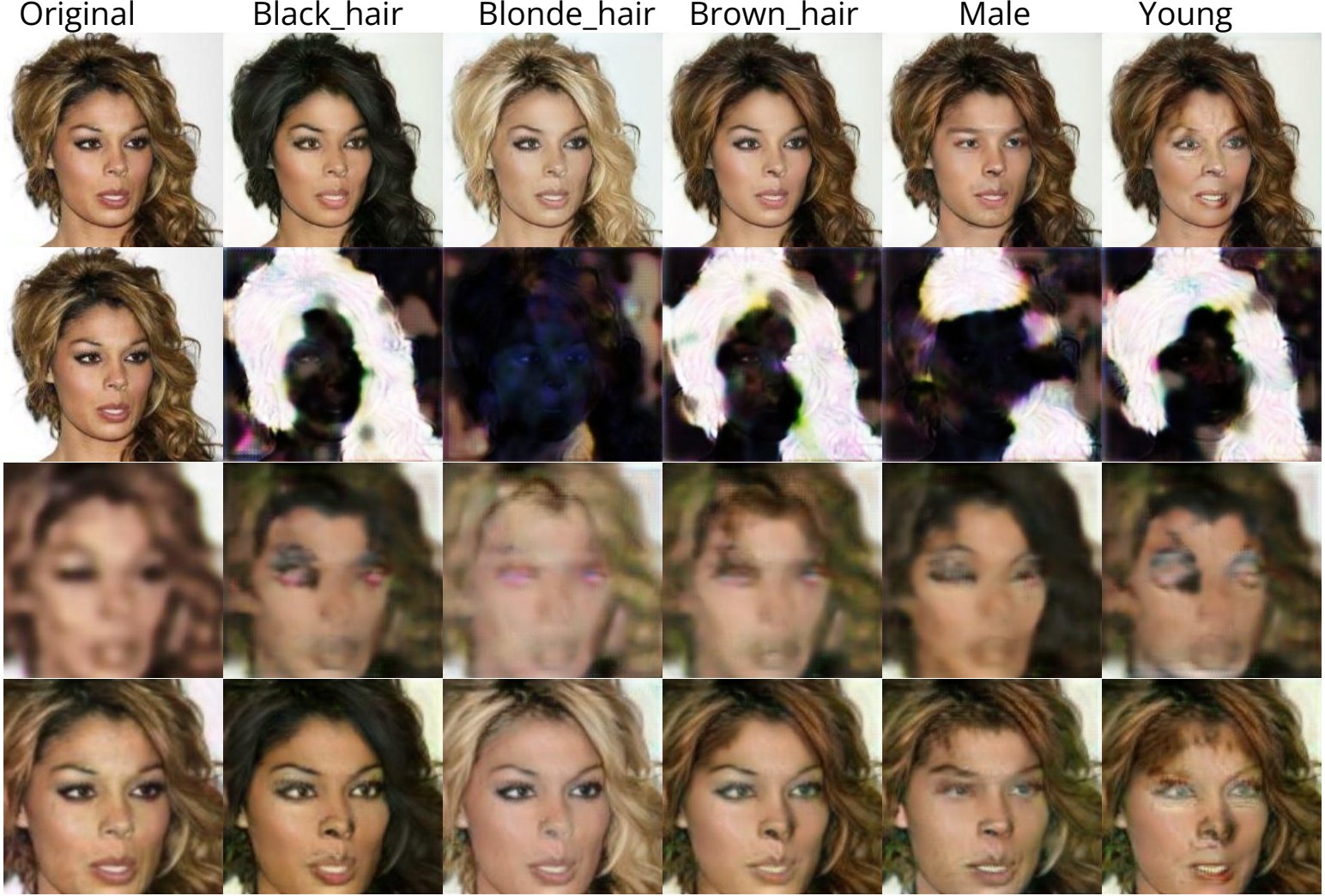


Figure 2. Architecture of Restormer for high-resolution image restoration. Our Restormer consists of multiscale hierarchical design incorporating efficient Transformer blocks. The core modules of Transformer block are: **(a)** multi-Dconv head transposed attention (MDTA) that performs (spatially enriched) *query-key* feature interaction across channels rather the spatial dimension, and **(b)** Gated-Dconv feed-forward network (GDFN) that performs controlled feature transformation, *i.e.*, to allow useful information to propagate further.


Original
Black_hair
Blonde_hair
Brown_hair
Male
Young
Source image
Perturbed image
Auto Encoder
ReFormer


Original
Black_hair
Blonde_hair
Brown_hair
Male
Young
Source image
Perturbed image
Auto Encoder
ReFormer


Original
Black_hair
Blonde_hair
Brown_hair
Male
Young
Source image
Perturbed image
Auto Encoder
ReFormer

Perturbation Removal Evaluation

Restoration Quality	MSE	SSIM	PSNR
AE-Perturbed	267.63	0.68	24.16
Re-Perturbed	**8.46**	**0.97**	**39.16**

Removal effectiveness	MSE	SSIM	PSNR
AEDF-PerfectDF	5661.78	0.33	11.17
ReDF-PerfectDF	5830.58	0.32	11.04

Discussion

- Limitation:
  - Auto encoder poor performance
    - Blur images
    - Perturbation are not fully removed
    - Hardware limitation and implementation
- Finding:
  - Perturbation can not transfer to other deepfakes models
  - Image restoration techniques can eliminate the perturbation
- Future Work:
  - Increase sample size
  - More complex network pipeline(detect-based discriminator from MagDR)
- Thanks! Questions?