Theoretic Fundamentals of Machine and Deep Learning

Adversarial Robustness and Attacks II: in Real World

Aleksandr Petiushko

Lomonosov MSU, Faculty of Mechanics and Mathematics
MIPT, RAIRI
Nuro, Autonomy Interaction Research

Winter-Spring, 2023

AP

Plan

1. Adversarial examples in real world
2. Adversarial attack on face detection
3. Adversarial attack on face ID
4. Defense from adversarial examples in real world
5. Black-box face restoration

AP

Major approach

- Usual training:

$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta^t} L(f_{\theta^t}(x), y)$$

- Adversarial perturbation:

$$x^{t+1} = x^t + \eta \nabla_{x^t} L(f_\theta(x^t), y)$$

AP

Robustness in Machine Learning

Robustness [informally]

Ability for a machine learning algorithm a to provide similar outputs on the similar data (i.e. having the same class or other invariant features)

Two types of **Robustness** in ML:

Generalization

Dataset issue: algorithm needs to be robust if the dataset to evaluate it differs (sometimes significantly: we can treat it is a distribution shift) from the training dataset

Adversarial Robustness

Noise issue: algorithm needs to provide the similar output w.r.t. both clean and noisy images (where the model of noise is the topic to consider itself)

For now we'll consider the **Adversarial Robustness**.

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations
- Certification (or verification): how to provide theoretical guarantees on the noise level not fooling the neural net

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations
- Certification (or verification): how to provide theoretical guarantees on the noise level not fooling the neural net
  - Certification will be considered the next time

AP

Taxonomy of generation methods

In general, the adversarial examples generation methods (in Computer Vision) can be divided into the following types:

- Using ℓ_2-based norm (including geometric ones): most convenient for classical optimization
- Using ℓ_∞-based norm: correspond to perception process by the human eye of any visual information
- Using ℓ_0-based norm: the area of perturbation is minimized, but not the delta value per pixel

Anyway it is not enough for generation of physically plausible adversarial examples.

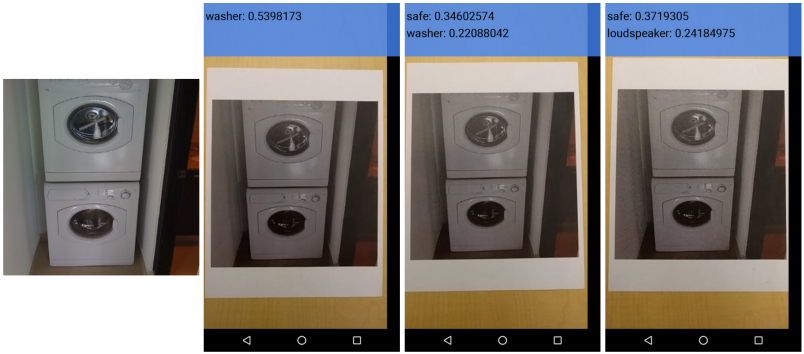
AP

Real-world adversarial examples (1)

- All adversarial examples until now were designed to work in digital domain: e.g. to change the image on a pixel level
- If there is no possibility to change the image before feeding it into NN, then the digital method is useless
- That's why **real-world** (or physical) adversarial examples are the most universal ones
- The first try of real-world adversarial examples[1] — generation of an image in the digital domain, then printing it out on the physical carrier (paper sheet), then photo by digital camera and finally NN recognition
- No any specific technology to generate the real-world adversarial examples was proposed: only its existence was shown

[1] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world." 2016

Real-world adversarial examples (2)



(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Adversarial examples in real world: EOT

- Don't have the control on the image pixels after the photo ⇒ the only option is to change the object appearance itself
- **E**xpectation **O**ver **T**ransformation (EOT)[2] to the rescue — takes into account the transformations of objects in the real world, e.g.:
  - Different scaling factors
  - Random translation and rotation
  - Luminosity / contrast variation, noise etc
- So for the object x in the real world the task is to find the adversarial perturbation r taking into account transformation $g \in T$:

EOT

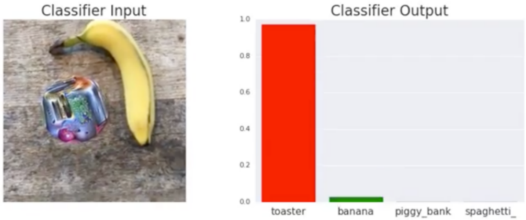
Find $\arg\min_r \mathbb{E}_{g\sim T}[P(y|g(x+r))]$ s.t.:

1. $\mathbb{E}_{g\sim T}[d(g(x+r), g(x))] < \epsilon$, where $d(a,b)$ – some distance function (e.g. $d(a,b) = ||a-b||_p$)
2. $x + r \in B$

[2] Athalye A. et al. "Synthesizing robust adversarial examples." 2017

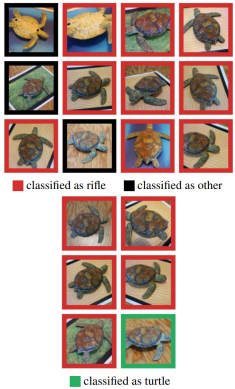
Examples of physical adversarial examples

Attack on ImageNet objects[3]:



Attack on road signs[4]:

3D adversarial objects:



[3]Brown T. et al. “Adversarial patch.” 2017

[4]Eykholt K. et al. “Robust physical-world attacks on deep learning models.” 2017

Physical adversarial examples: key ingredients

- ℓ_0-optimization (mask-based) + EOT: the must
- **T**otal **V**ariation (TV) loss — penalty for the perturbation to be non-smooth (in the real world there is no distinct pixel gradients):

$$TV(x) = \sum_{i,j} \sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2}$$

- **N**on-**P**rintability **S**core (NPS) — penalty for the perturbation colors that are out of the generator device (e.g., printer) limited gamut. E.g. if $G \subset [0,1]^3$ — limited device gamut, then the loss for using the pixel $q_0 \in [0,1]^3$:

$$NPS(q_0) = \Pi_{q \in G} ||q - q_0||_2$$

- Additional color adjustments (e.g. generator device provides not color c, but some its modification $m(c)$)

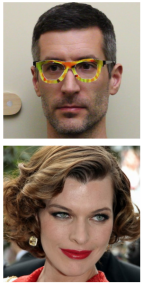
AP

Prior art: Face Det and ID attack

- Initially the **Camouflage Art**[5] was used to avoid the leading at that time Viola-Jones face detection system
- It was just the makeup crafted manually to fool the Haar detector
- Pioneering work[6] proposed to use printed adversarial glasses
- It uses ℓ_0-optimization + EOT + TV + NPS + color adjustments
- But it was used for closed-set recognition (a few predefined person ID for training) and for old generation FaceID NN

cvdazzle.com

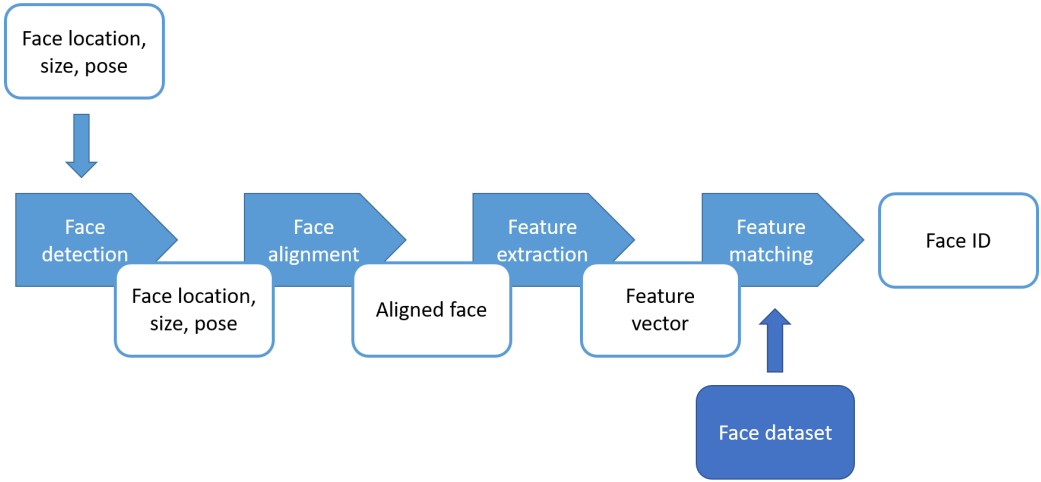
Adversarial glasses



[5] Feng R. et al. "Facilitating fashion camouflage art." 2013

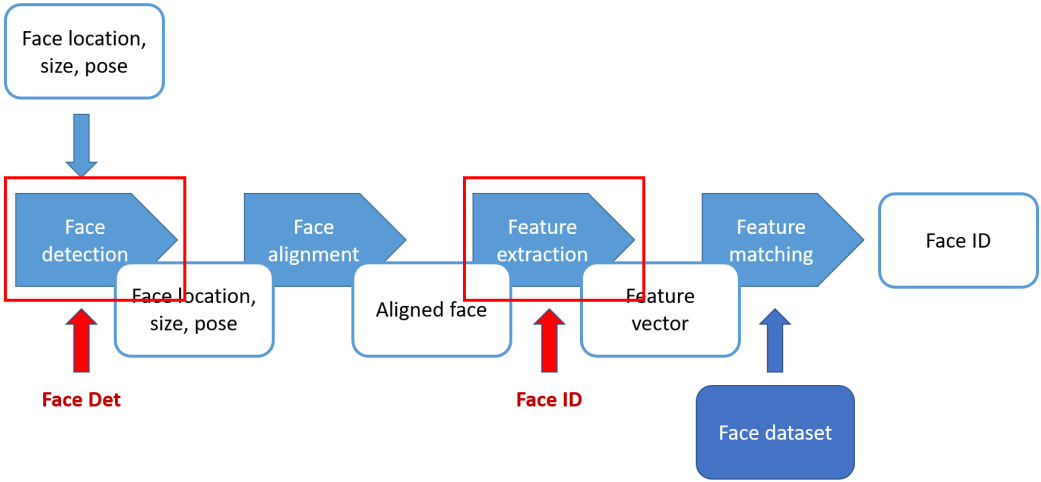
[6] Sharif M. et al. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." 2016

Face processing pipeline



AP

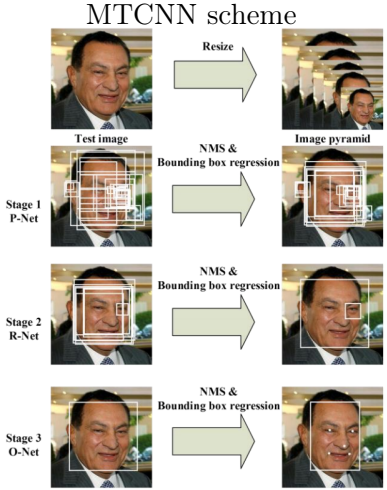
Face processing pipeline



AP

Face detection: MTCNN[7]

- Unlike modern and heavy detectors based on Faster RCNN and YOLO the MTCNN detector is quite shallow ⇒ smaller perception field, harder to change the detection conclusion
- MTCNN is cascade-based: in the beginning the rough approximation is provided (P-Net), then its tuning (R-Net and O-Net) is performed
- Based on our experiments, the most appropriate place for attack is the first P-Net and its classification loss (not bounding boxes or key points regression losses)

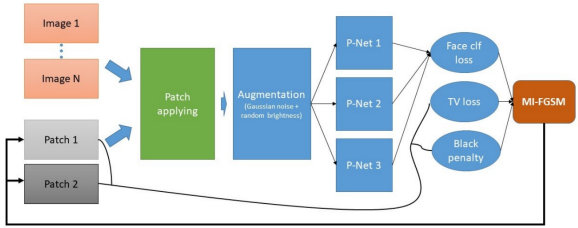


[7] Zhang K. et al. "Joint face detection and alignment using multitask cascaded convolutional networks." 2016

Adversarial attack on MTCNN face detector

- EOT: Gaussian noise, patch size, brightness, batch of different face images
- TV loss: used, NPS: not used
- Color adjustment: push the color to be the black one ($x_{i,j} = 1$) $\Rightarrow$ new additive loss part: $L_{BLK}(x) = \sum_{i,j}(1 - x_{i,j})$
- MI-FGSM as the optimizer

MTCNN adversarial attack

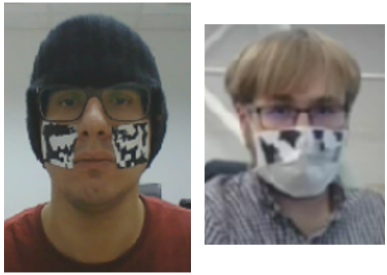


AP

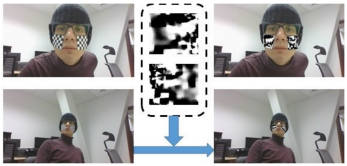
Adversarial attack on MTCNN face detector

- ℓ_0-based optimization: two versions of adversarial patches
  1. two distinct patches on cheeks
  2. the whole medicine mask
- MTCNN has small perceptive field $\Rightarrow$ patches are not semantical (unlike for FaceID, see next)
- Need to estimate the local affine projections parameters based on the prepared special grid

Patches



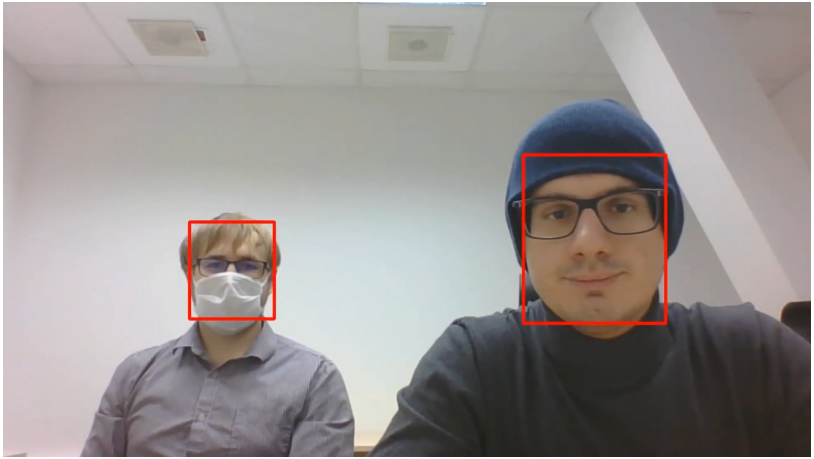
Projection



AP

Adversarial attack on MTCNN face detector: outcome

Details: paper[8] (IEEE-2019) and video[9].



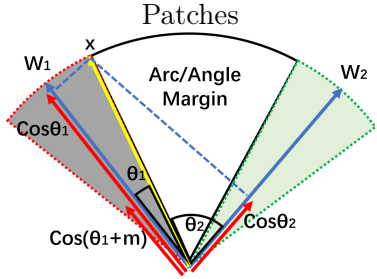
[8] Kaziakhmedov E. et al. “Real-world attack on MTCNN face detection system.” 2019

[9] `https://www.youtube.com/watch?v=OY70OIS8bxs`

AP

FaceID: ArcFace[10]

- For face ID adversarial attack the best public face ID system was chosen: ArcFace
- Main idea of ArcFace — to use angle margin (aligned with cosine similarity)
- Huge training dataset (MS1M) and deep CNN (ResNet-100) are used

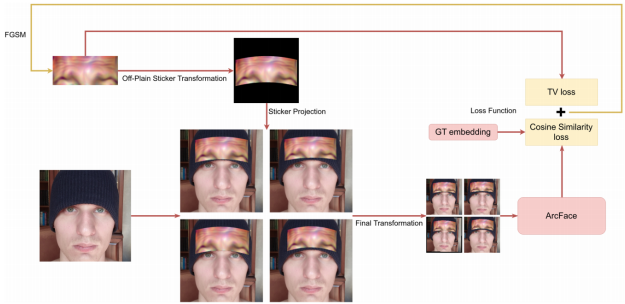


[10] Deng J. et al. "Arcface: Additive angular margin loss for deep face recognition." 2018

Adversarial attack on ArcFace face ID

- EOT: Different patch projection parameters, single face image
- TV loss: used, NPS: not used, Color adjustment: not used
- Additive similarity loss to work in open-set setting:
  $L_{sim}(x, x_{gt}) = cos(emb(x), emb(x_{gt}))$, where x_{gt} — template image for the person, $emb(x)$ — feature vector of x
- MI-FGSM as the optimizer

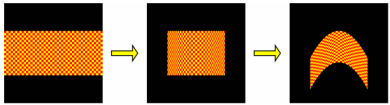
ArcFace adversarial attack



Adversarial attack on ArcFace face ID

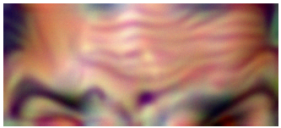
- ℓ_0-based optimization: color patch on the forehead
- Deep NN $\Rightarrow$ large perception field $\Rightarrow$ patch is semantical
- Nonlinear off-plane projection: $(x, y, 0) \to (x', y, z'), z' = a \cdot x'^2$
- All image transformation done by differentiable Spatial Transformer Layer[11]

Off-plane projection



$$x = a \cdot \left(|x'| \cdot \sqrt{(x')^2 + \frac{1}{4 \cdot a^2}} + \frac{1}{4 \cdot a^2} \cdot \ln \left(|x'| + \sqrt{(x')^2 + \frac{1}{4 \cdot a^2}} \right) - \frac{1}{4 \cdot a^2} \cdot \ln \left(\frac{1}{2 \cdot a} \right) \right)$$

Semantic patch examples



[11] Jaderberg M. et al. "Spatial transformer networks." 2015

AdvHat — invisibility hat

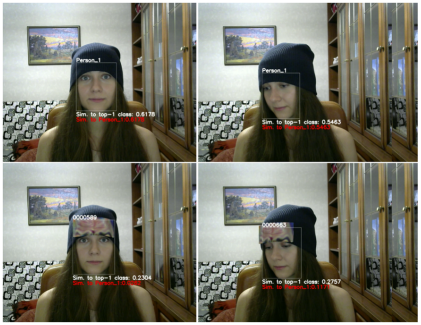
Due to the better projection procedure and richer color information, the attack is robust to rotations and brightness variation

Frontal face
(advhat: no)
Similarity to origin: **0.61**

|

|

Frontal face
(advhat: yes)
Similarity to origin: **0.02**
Similarity to other: **0.23**



Rotated face
(advhat: no)
Similarity to origin: **0.54**

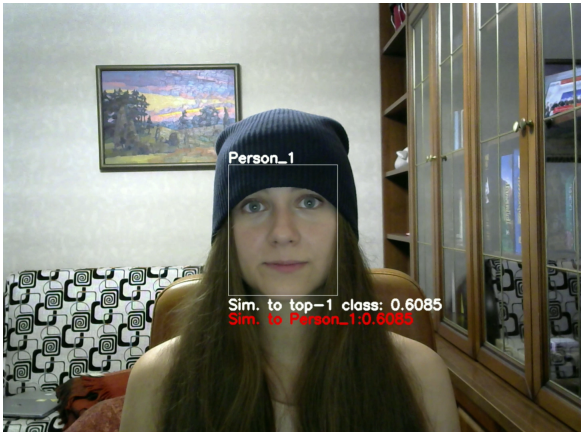
|

|

Rotated face
(advhat: yes)
Similarity to origin: **0.11**
Similarity to other: **0.27**

Adversarial attack on ArcFace face ID: outcome

Details: paper[12] (ICPR-2020) and video[13].



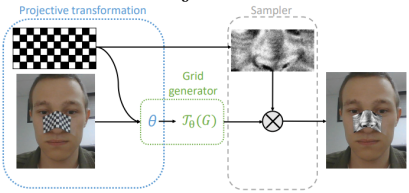
[12] Komkov S. et al. "Advhat: Real-world adversarial attack on arcface face id system." 2019

[13] `https://www.youtube.com/watch?v=a4iNg0wWBsQ`

Adversarial attack on ArcFace face ID: grayscale patch[14] (IEEE-2019)

- Combination of two previous approaches:
  - Grayscale color loss adjustment
  - Local affine grid projection
- Patch is also semantical

Projection



Patches

[14] Pautov M. et al. "On adversarial patches: real-world attack on ArcFace-100 face recognition system." 2019

FaceID adversarial defense in real world[16]

- Almost all of the real world attacks are patch-based
  - Proposal: **A**dversarial **T**raining (AT)[15] in the pixel space with patch-based augmentation
- AT decoupling:
  1. Best (=max loss) location of gray patch by:
     - ★ Exhaustive search
     - ★ Max gradient locations w.r.t. input
  2. PGD inside this patch

- Common training procedure:

$$\min_{\theta} \mathbb{E}_{x,y}[L(\theta, x, y)]$$

- Adversarial Training:

$$\min_{\theta} \mathbb{E}_{x,y}[\max_{r \in \Delta} L(\theta, x + r, y)]$$

[15] Goodfellow I. et al. "Explaining and harnessing adversarial examples." 2014

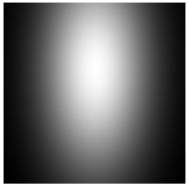
[16] Wu T. et al. "Defending Against Physically Realizable Attacks on Image Classification." 2019

Black-box face restoration

- Black-box model M: $M(x) = y$, where
  - x — input image of the face
  - y — its feature representation (embedding)
- **Task**: to recover x' to preserve the person identity
- **Prior art**: using reconstruction MSE and perceptual metrics / GANs (NBNet[17])
- **Our approach**: use zero-order optimization to find such x' so as FaceID $M(x') \approx M(x)$
  - Use as M the public SotA in FaceID: ArcFace
  - Test by using independent critic: FaceNET[18]
  - Similarity loss: $1 - cos(M(x'), M(x))$
  - Additional term: $(||M(x')||_2 - ||M(x)||_2)^2$

- **Main difficulty**: huge search space in pixel domain
- **Solution**: to use prior knowledge about face — 2D Gaussians

$$G(x, y) = A \cdot e^{\frac{(x-x_0)^2}{2\sigma_1^2} + \frac{(y-y_0)^2}{2\sigma_2^2}}$$



$$(x_0, y_0, \sigma_1, \sigma_2, A) = (56, 72, 22, 42, 1)$$

[17] Mai G. et al. "On the reconstruction of face images from deep face templates." 2018

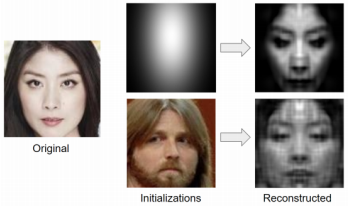
[18] Schroff F. et al. "Facenet: A unified embedding for face recognition and clustering." 2015.

Black-box face restoration: successful tricks

- Even prior info about face is not enough
- **Trick1**: Use vertical face symmetry ⇒ use only half of the face to search
- **Trick2**: For identity preservation usually no need in color ⇒ use only a single color channel instead of 3

Symmetrical, non-symmetrical, color restoration

- **Initialization**: What to use as the starting point?
- **Common approach**: to use other face (can be biased)
- **Our approach**: optimal Gaussian blob (additional loss term is needed)



AP

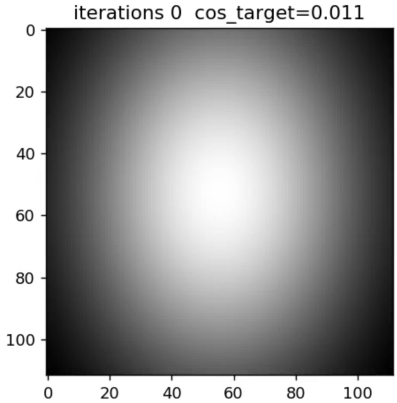
Black-box face restoration: results

Our method:						
ArcFace:	0.97	0.97	0.94	0.97	0.85	0.73
FaceNet:	0.70	0.75	0.72	0.78	0.38	-0.09
NBNet (WB):						
ArcFace:	0.17	0.21	0.12	0.26	0.06	0.09
FaceNet:	0.02	0.32	0.25	0.46	-0.01	0.35
NBNet (RGB):						
ArcFace:	0.28	0.46	0.34	0.54	0.12	0.21
FaceNet:	0.59	0.53	0.44	0.74	0.18	0.41
Original:						

AP

Black-box face restoration: outcome

Details: paper[19] (ECCV-2020) and video presentation[20].

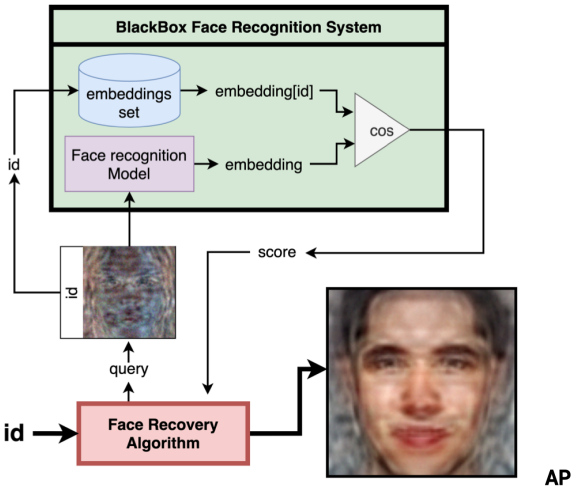


[19] Razzhigaev A. et al. "Black-Box Face Recovery from Identity Features." 2020

[20] `https://www.youtube.com/watch?v=sOrTcqRTw2A`

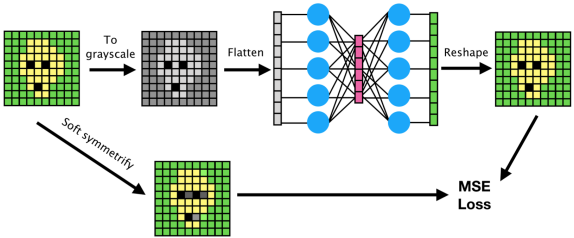
Black-box face restoration: more constraints

- **Task**: to recover x' from the Black-box model M, where we can **use only the similarity**:
  $S(x, x') = 1 - cos(M(x'), M(x))$
- Now the reconstruction is done fully **colorful**
- For this case the usage of so-called "**eigenfaces**" (actually, columns of the autoencoder decoder weights matrix) is added



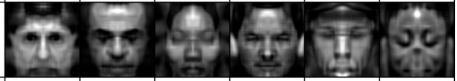
Darker than a Black-box face restoration: losses

- Autoencoder: $y = W_2(W_1 x)$, trained[21] on the public faces dataset
- W_1 — encoder, W_2 — decoder, z — latent representation ($z \in \mathbb{R}^{1024}$)
  - In our case, z is providing the projection weights to "eigenfaces space"
- Loss: $L(x) = \left(\frac{x+2R(x)}{3} - W_2(W_1 x)\right)^2 + (W_2 z - x)^2$
  - $z \sim N(0, I)$ and $R(\cdot)$ is a vertical reflection operator



[21] Razzhigaev, Anton, et al. "Darker than black-box: Face reconstruction from similarity queries." 2021

Darker than a Black-box face restoration: results

Ours (RGB)						
ArcFace	0.99	0.99	0.99	0.99	0.99	0.99
FaceNet	0.77	0.82	0.77	0.61	0.62	0.72
Ours (gray-scale)						
ArcFace	0.98	0.99	0.99	0.98	0.99	0.99
FaceNet	0.71	0.78	0.60	0.51	0.59	0.74
Gaussian sampling (gray-scale)						
ArcFace	0.97	0.97	0.94	0.97	0.85	0.73
FaceNet	0.70	0.75	0.72	0.78	0.38	-0.09
NBNet (RGB)						
ArcFace	0.28	0.46	0.34	0.54	0.12	0.21
FaceNet	0.59	0.53	0.44	0.74	0.18	0.41
Original						

AP

Takeway notes

- Digital $\rightarrow$ physical domain attack translation is hard
- But even the most successful face ID systems can be fooled by a simple grayscale patch from common printer
- ℓ_0-based local attack + TV loss + EOT are the must
- Need to use projection schemes allowing gradient backpropagation
- Adversarial training in practice (or certified robustness in theory) can help to defense
- Face image can be restored even in black-box setting using only similarities

AP

Thank you!