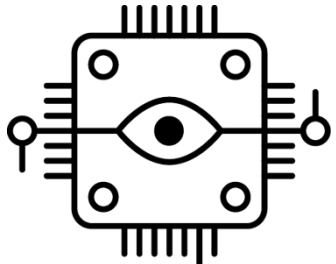
Adversarial Attacks on Black Box Face Recognition Challenge

May 14 – June 7

VisionLabs

MACHINES CAN SEE



MACHINES CAN SEE

Adversarial Attacks

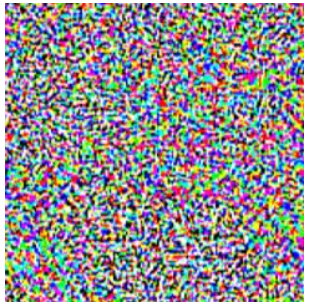
Neural Network

Cat → 97%

Dog → 1%

Other → 2%

+



=

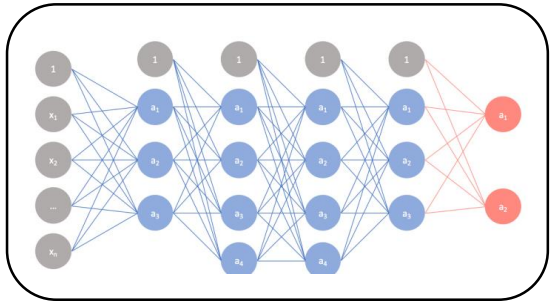
Neural Network

Cat → 4%

Dog → 96%

Other → 0%

Adversarial Attacks



Cat → 97%

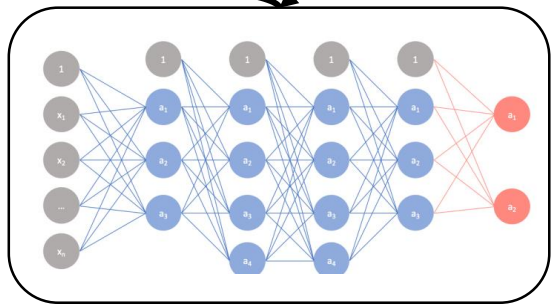
Dog → 1%

Other → 2%

+

$$\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)$$

=

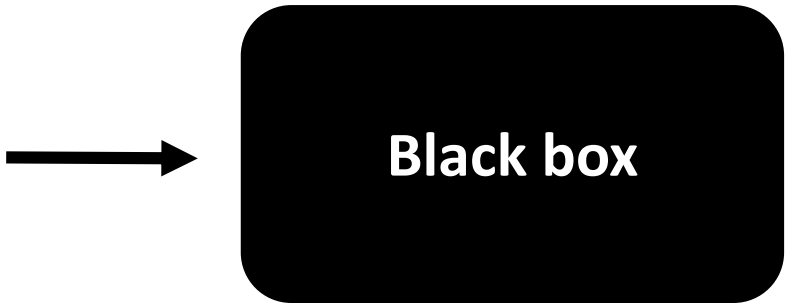


Cat → 4%

Dog → 96%

Other → 0%

Adversarial Attacks



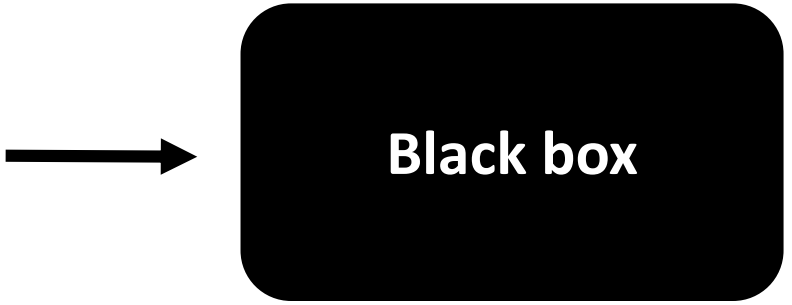
Cat → 97%

Dog → 1%

Other → 2%

+

=

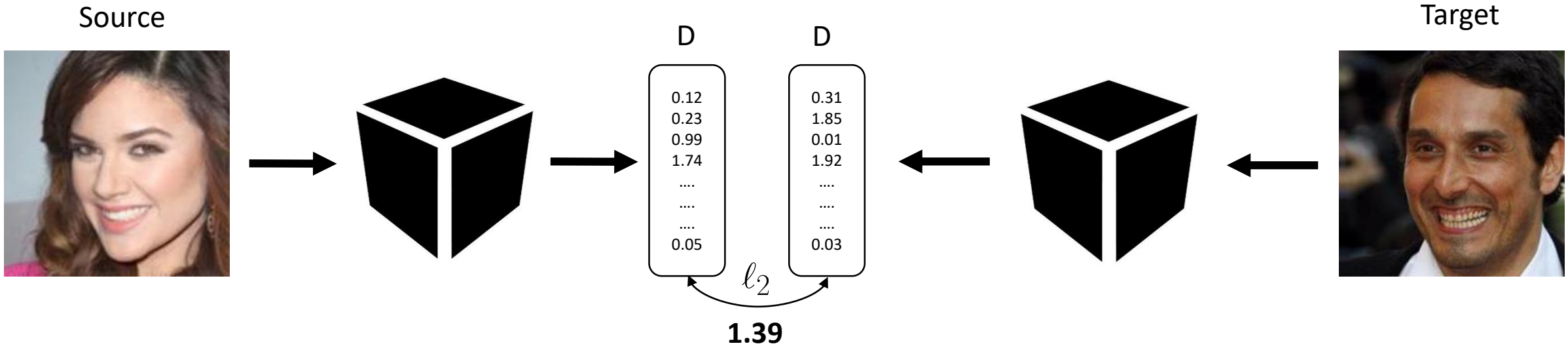


Cat → 4%

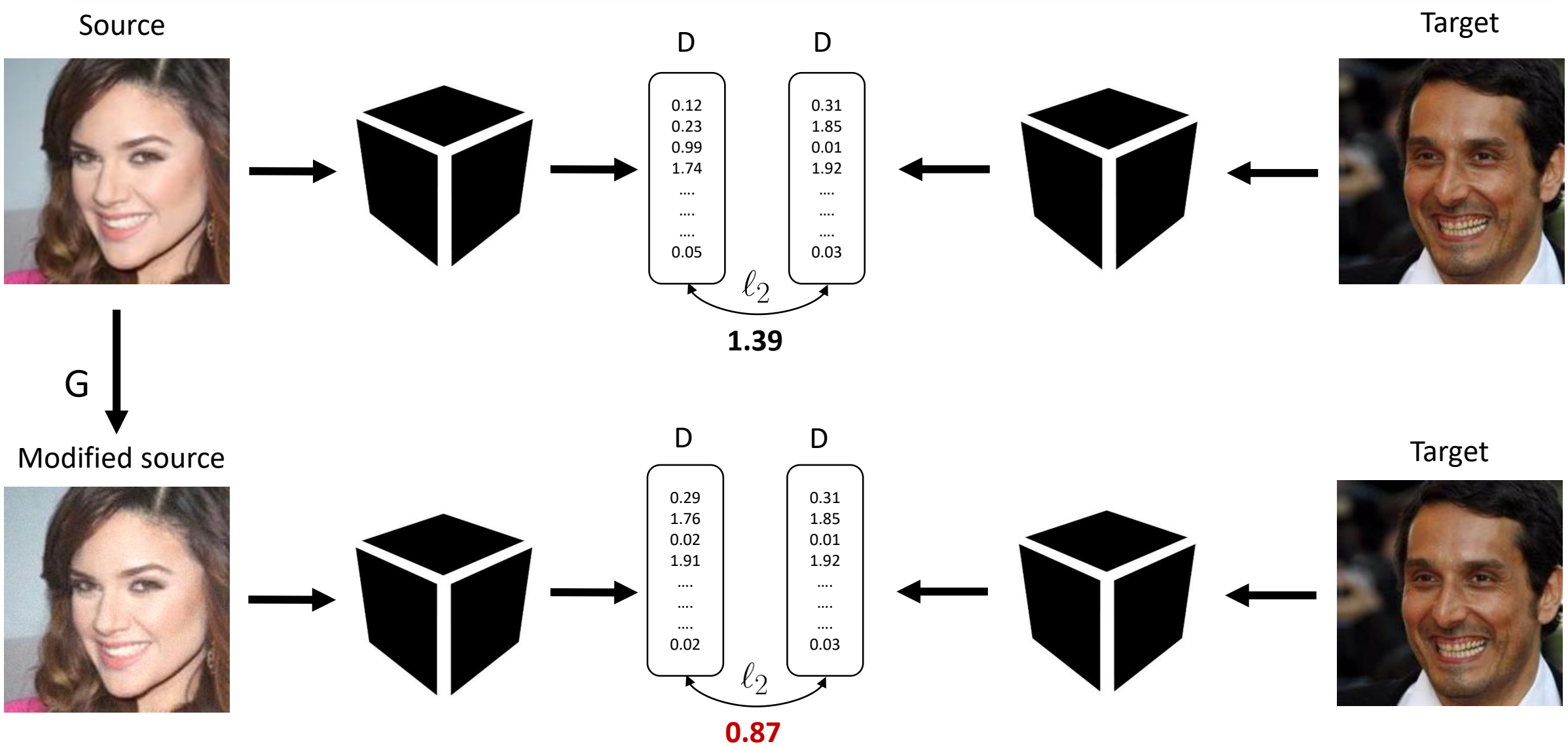
Dog → 96%

Other → 0%

Overview


Source
Target
D
D
0.12
0.23
0.99
1.74
....
....
....
0.05
0.31
1.85
0.01
1.92
....
....
....
0.03
ℓ_2
1.39

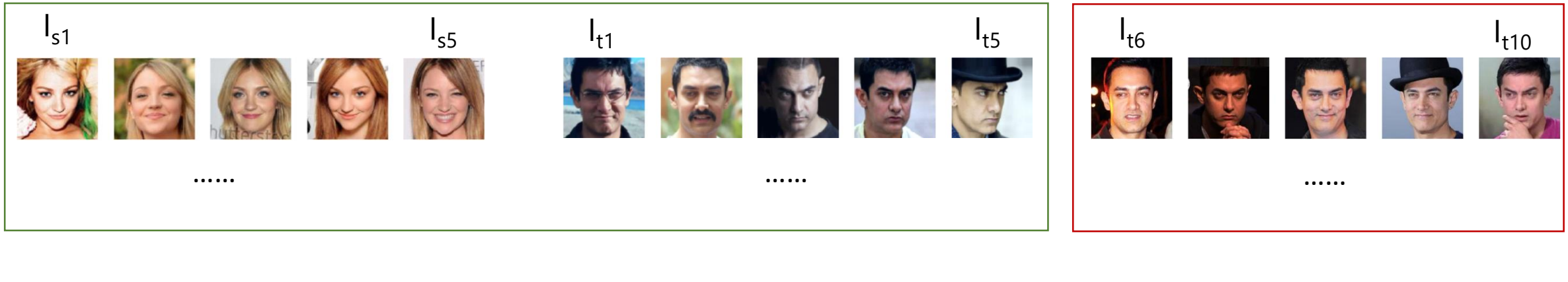
Overview


Source
Target
D
D
0.12
0.23
0.99
1.74
....
....
....
0.05
0.31
1.85
0.01
1.92
....
....
....
0.03
ℓ_2
1.39
G
Modified source
Target
D
D
0.29
1.76
0.02
1.91
....
....
....
0.02
0.31
1.85
0.01
1.92
....
....
....
0.03
ℓ_2
0.87

Pipeline

Source

Target



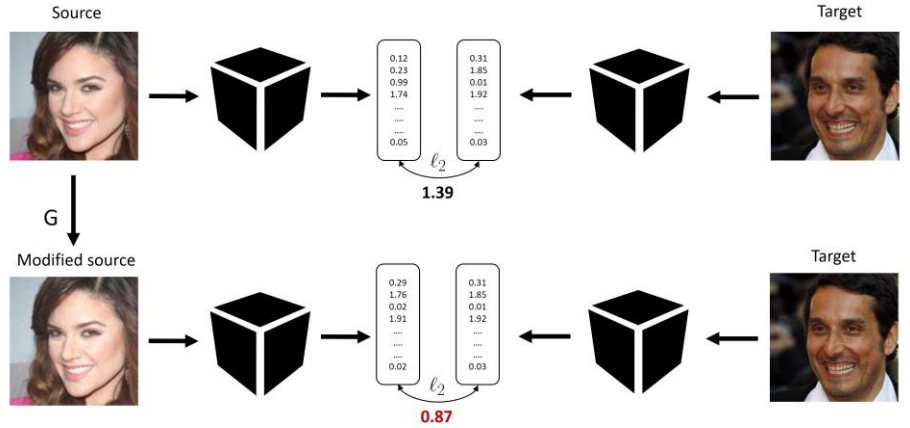
1000 pairs

25% public LB

75% private LB

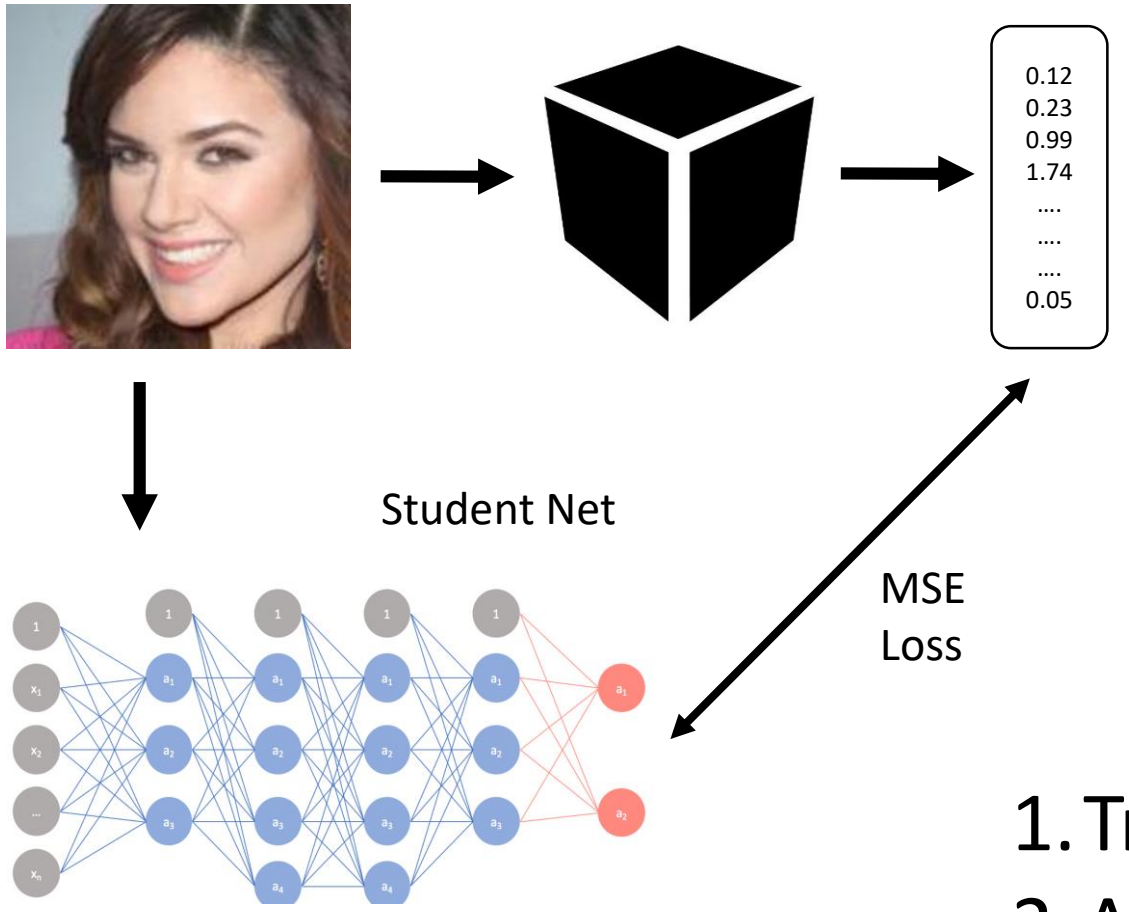
$$||D(G(I_s))-D(I_t)||_2 \rightarrow \min,$$

$$SSIM(G(I_s), I_s) \geq 0.95,$$



$$Score = 1/N_{pairs} * 1/25 * \Sigma_{k=1..Npairs} \Sigma_{i=1..5} \Sigma_{j=6..10} ||D(G(I_{s(k,i)}))-D(I_{t(k,j)})||_2$$

Baseline



1M face dataset

1. Train Student Net
2. Apply white-box attack on Student Net
3. Hope it will also affect Teacher Net

Challenge Results

Results					
#	User	Entries	Date of Last Entry	Team Name	Score ▲
1	alexey.grankov	1	06/06/18		0.952 (1)
2	mortido	1	06/06/18	Homeless Nonames	0.962 (2)
3	rl_agent	1	06/06/18	Boyara Power	0.985 (3)
4	kulikovpavel	1	06/06/18		0.989 (4)
5	grip	1	06/06/18		1.039 (5)
6	bearstrikesback	1	06/06/18		1.087 (6)
7	lmSasha	1	06/06/18		1.162 (7)
8	che	1	06/06/18		1.267 (8)
9	denilv	1	06/06/18		1.330 (9)
10	kaquff	1	06/06/18		1.342 (10)
11	michaelswan	1	06/06/18		1.399 (11)
12	lucidyan	1	06/06/18		1.407 (12)

130 registered participants → 15 above baseline

Challenge Results

1	150 000 ₽ **+ NVIDIA GeForce GTX 1080 Ti**		Aleksey Grankov
2	75 000 ₽ **+ NVIDIA GeForce GTX 1080 Ti**		Igor Kotenkov Tatiana Gaintseva Alexander Veysov Alexander Kiselev
3	36 000 ₽ **+ NVIDIA GeForce GTX 1080 Ti**		Oleksii Hrinchuk Valentin Khrulkov Elena Orlova
4	24 000 ₽		Pavel Kulikov
5	15 000 ₽		Annalisa Verdolova

Alexey Grankov

1st place

MCS 2018. Adversarial Attacks on Black Box Face Recognition

Method overview

- Iterative Fast Gradient Sign Method (I-FGSM)

$$\mathbf{I}_{\boldsymbol{\rho}}^{i+1} = \mathrm{Clip}_{\epsilon}\left\{\mathbf{I}_{\boldsymbol{\rho}}^{i} + \alpha\ \mathrm{sign}(\nabla \mathcal{J}(\boldsymbol{\theta}, \mathbf{I}_{\boldsymbol{\rho}}^{i}, \ell)\right\}$$

- White box NN which imitates Black Box outputs

White box Neural Network

- Finetuning FaceNet model
- Using model pretrained on ImageNet dataset
- Replacing classification layer wth embeddings with L2 norm
- Using larger images size than black box input
- Other training techniques such as: classification of each person, center loss, triplet loss.

Different research results

Model	MSE loss	Epochs	LB result
Facenet (from the box)	-	-	1.275
Facenet (Finetune)	6.398 * 1e-4	400+400	~1.205
Resnet18	4.860 * 1e-4	400(391)	-
Resnet50	17.968 * 1e-4	400(35)	-
Resnext101	7.464 * 1e-4	400(93)	-
Resnet34	4.399 * 1e-4	400(374)	1.024

I-FGSM tweaks

- Adaptive epsilon parameter
- Minimize to center of target images cluster
- Mean of noise from stack of different models
- Selecting result with minimum metric distance

Leaderboard results

Model	Leaderboard result
Fine tuned facenet	1.275
Resnet34	1.024
2xResnet34	0.980
4xResnet34	0.950
6xResnet34	0.943

MCS 2018

Adversarial Attacks on Black Box Face Recognition

Igor Kotenkov,Tatiana Gaintseva,Alexander Veysov,Alexander Kiselev

June 7, 2018

Team Homeless Nonames

Student Model Results

Table 1: Student Model distillation results

Architecture	Pre-trained (*)	LR regime	Best MSE	Epochs
ResNet18	Yes	(1)	4.51 * 1e-4	25
ResNet34	Yes	(1)	3.36 * 1e-4	25
DenseNet161	Yes	(2)	3.08 * 1e-4	25
XCeption	No	(4)	4.57 * 1e-4	23
ResNet50	No	(4)	4.07 * 1e-4	11

(1) 1e-3 + pre-trained on ImageNet + LR decay + adam
(2) 1e-4 + pre-trained on ImageNet + LR decay + adam
(3) 1e-3 + pre-trained on ImageNet + LR decay + adam
(4) manual adjustments each epoch

Attack results

Attack	Hack	Student CNNs	BB score	LB pub	LB priv
FGSM	-	DenseNet161	1.25	-	-
FGSM	(1)	DenseNet161	1.16	-	-
FGVM	(3)	2 CNNs	0.97	1.05	-
FGVM	(4)	5 CNNs	0.91	-	-
FGVM + 1 pixel	(4)	5 CNNs	0.90	0.99	-
FGVM + 6 pixel	(4)	5 CNNs	0.87	-	-
FGVM + 16 pixel	(5)	5 CNNs	0.87	-	0.96

FGSM - Fast Gradient Sign Method

FGVM - Fast Gradient Value Method

Final heuristics

FGVM

- Noise **eps * clamp(grad / grad.std(), -2, 2)**
- Ensemble of several CNNs via weighting their gradients
- Save changes only if it reduces mean loss
- Use target combinations for more robust targeting

Genetic One Pixel

- popsize = 30
- max_iter = 5

Student CNN distillation

What worked

- Transfer learning
- ADAM + clever LR regime to avoid under-fitting
- Best architectures are reasonably heavy **ResNet34** and **DenseNet161**

What did not

- Inception-based architectures (not-suitable due to high down-sampling)
- VGG based architectures (overfitting)
- "Light" architectures (SqueezeNet / MobileNet - underfitting)
- Image augmentations (w/o modifying descriptors)
- Working with 224x224 images

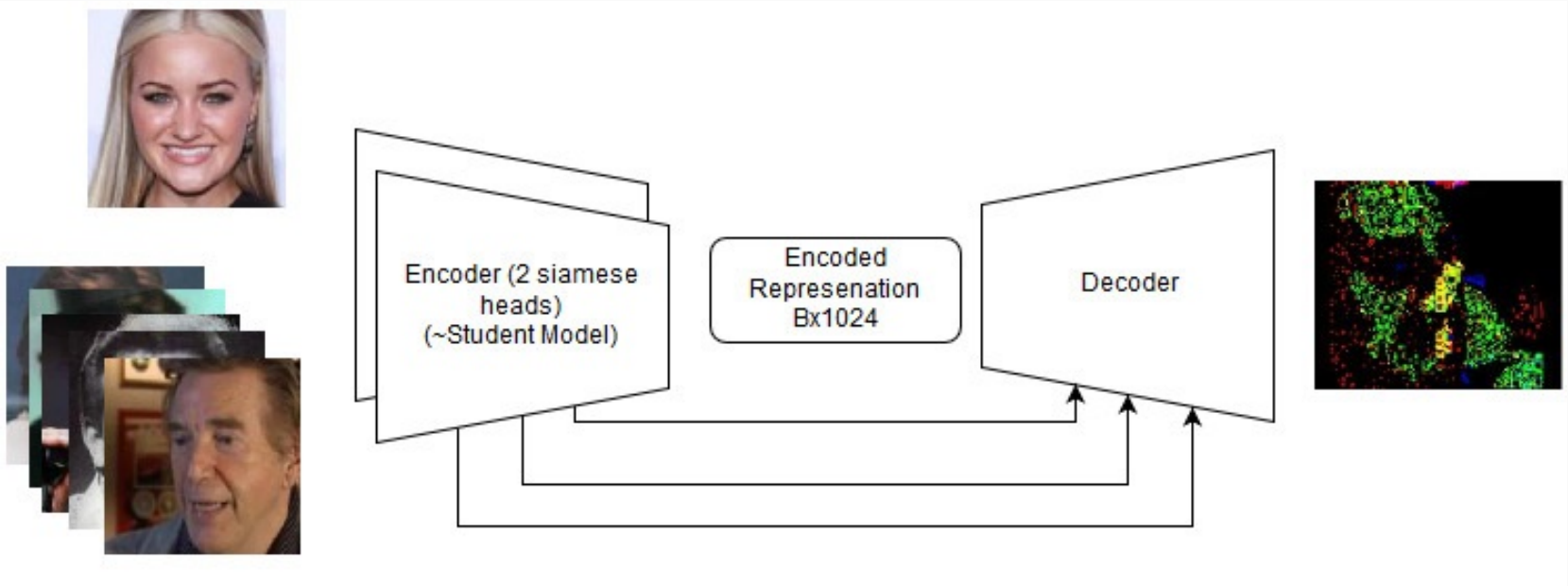
Other Attack Approaches

We also tried:

- **FGVM with momentum**
  https://arxiv.org/abs/1710.06081v3
- **CW** – good for white-box attacks
  https://arxiv.org/abs/1608.04644

End-to-end architectures (1)

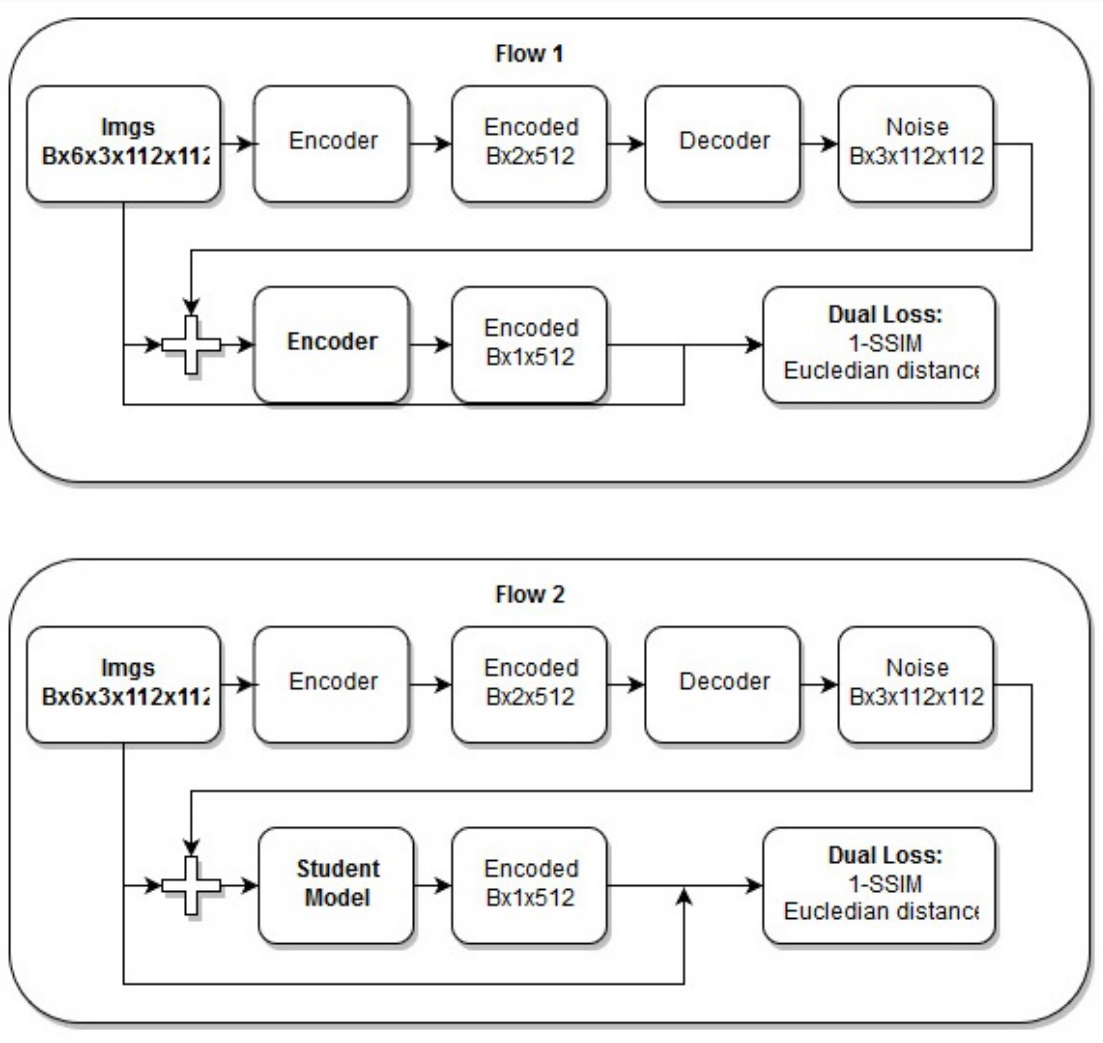
- Key ideas - use a mixture of VAE / Siamese LinkNet
- 2 part loss - PyTorch SSIM + Eucledian distance



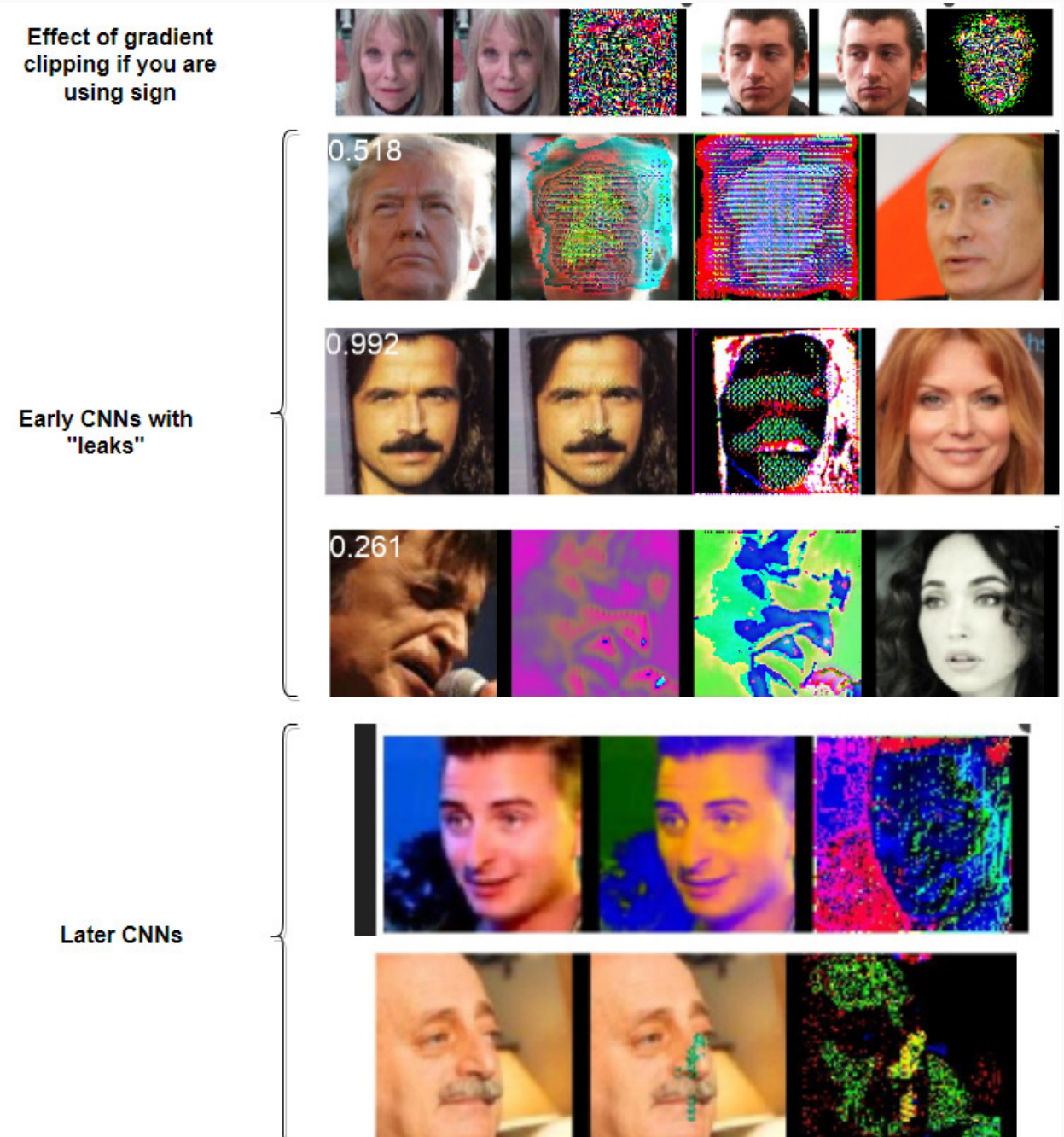
End-to-end architectures (2)

Key take-aways

- Performs well on **WB** and poorly on **BB**
- Difficult to balance Loss - use running mean scaling
- Problems with scaling images back - use some eps
- Model parametrization - open question
- Pass image as skip connection



Some fun illustrations



Team Boyara Power

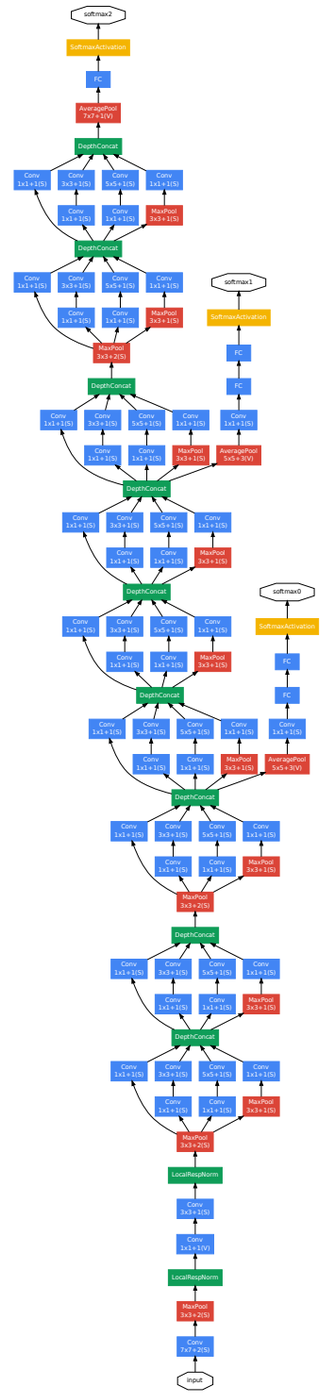
Oleksii Hrinchuk, Valentin Khrulkov, Elena Orlova

Copycat Network

- Finetune the pretrained Facenet based on **Inception v1.**

- Replace last 3 layers with the FC layer with 512 outputs, followed by **BatchNorm** and L_2 **normalization.**

- Train for 10 epochs, with learning rate decay after every 3 epochs.



Data augmentation

- Augment the data with 4 possible corner crops, and by performing horizontal flip, zoom and shift.

- Include previously computed submissions into the training set to better approximate the network in the proximity of given data.

- Generate synthetic inputs based on identifying directions in which the models' output is varying.

$$\boldsymbol{x} + \lambda \cdot \nabla_{\boldsymbol{x}} J(\boldsymbol{x}, \mathcal{D}(\boldsymbol{x}))$$

From "Practical Black-Box Attacks against Machine Learning", Nicolas Papernot et al. ASIA CSS (2017)

Attacker

- Targeted FGM with momentum

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \alpha \cdot \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t, y)$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{g}_{t+1}$$

From “Boosting Adversarial Attacks with Momentum”, Yinpeng Dong et al. arxiv preprint (2017)

Attacker

- Targeted FGM with momentum

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \alpha \cdot \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t, y)$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{g}_{t+1}$$

- Targeted FGM with **Nesterov momentum**

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \alpha \cdot \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t - \mu \cdot \boldsymbol{g}_t, y)$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{g}_{t+1}$$

From “Boosting Adversarial Attacks with Momentum”, Yinpeng Dong et al. arxiv preprint (2017)

Other tricks

- 998 out of 1000 target people were present in images of source people (can be found by looking at L_2 distances between target and all sources).

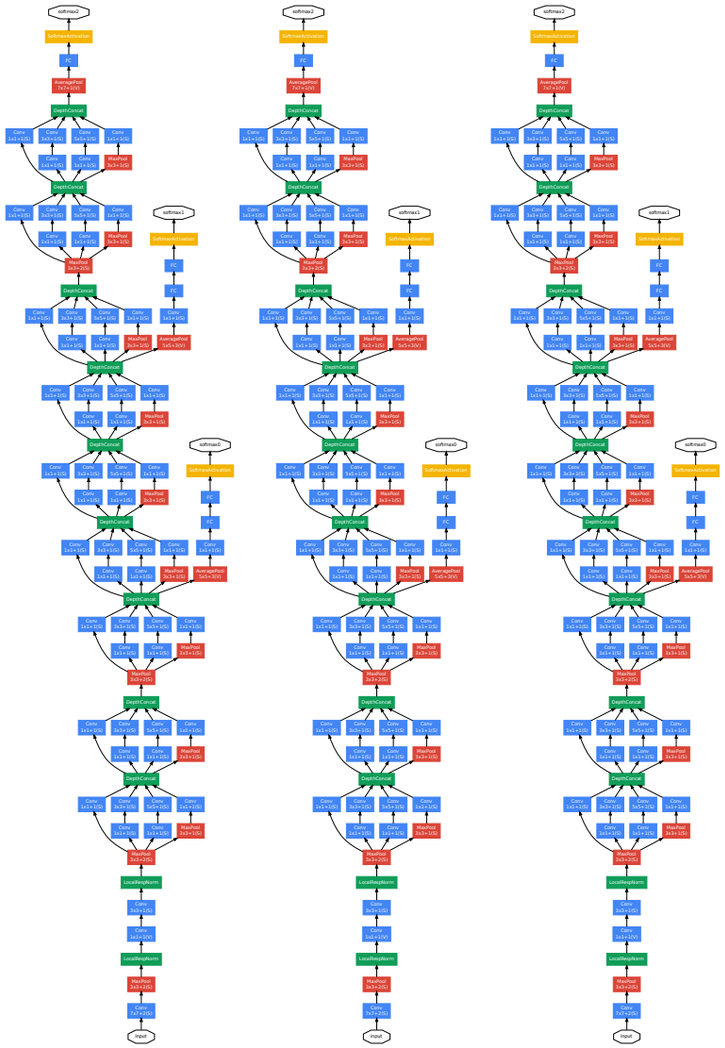
- Instead of attacking 5 targets we were attacking 20 (original targets, corresponding sources, mirror reflections of both).

- The difference between local score and public score was ~ 0.03.

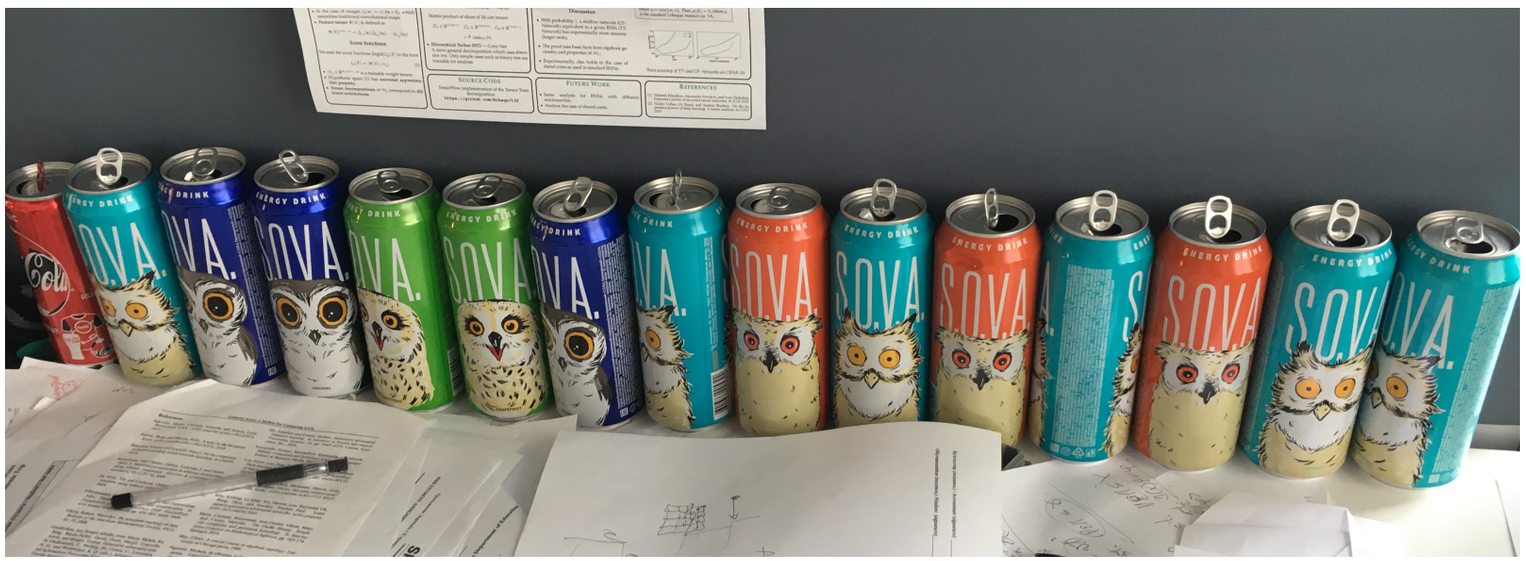
Road to < 1.0

Model	Public Score
Fine tuned facenet	1.256
+ train augmentation	1.114
+ ensemble & Nesterov	1.007
+ batch normalization	0.981

Final setup



$$\alpha = \{0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$$




```
RuntimeError                              Traceback (most recent call last)
<ipython-input-34-ddf967410fe7> in <module>()
      4 import MCS2018
      5 gpu_id = 1
----> 6 net = MCS2018.Predictor(gpu_id)

RuntimeError: [enforce fail at common_gpu.cc:129] error == cudaSuccess. 10 vs 0. Error at: /opt/VisionLabs/MCS2018/py
torch/caffe2/core/common_gpu.cc:129: invalid device ordinal
```