MCS 2018

Adversarial Attacks on Black Box Face Recognition

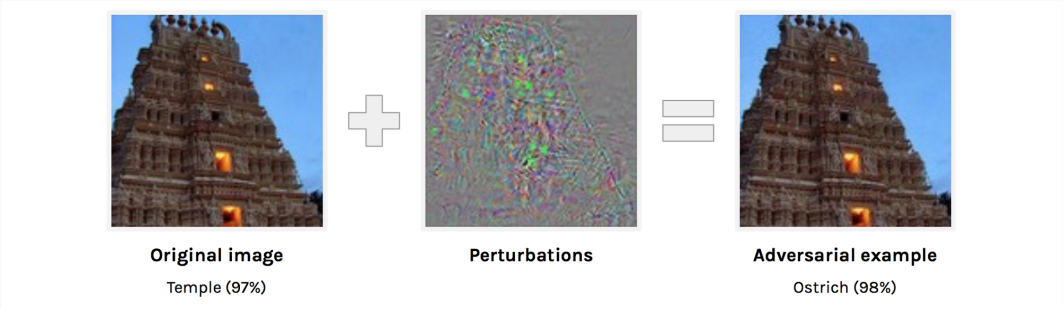
Igor Kotenkov,Tatiana Gaintseva,Alexander Veysov,Alexander Kiselev

7 июля 2018 г.

Team Homeless Nonames

Problem statement

Adversarial Attack Problem Statement



Goal: Change image so that model won't recognize it

Variations:

- Change image slightly (in what term?)
- Black Box (BB) or White Box (WB) attack
- Targeted or non-targeted attack

MCS2018 Problem Statement

Our task:

- Targeted BB task on face images 250x250
- Change image slightly (SSIM $\geqslant 0.95$)
- BB: image $\rightarrow$ 512d vector
- BB accepts one image as input
- Loss: euclidean distance

MCS2018 Problem Statement

Data:

- Pool of 1M faces with corresponding vectors
- 1K pairs of source and target identities (Test data)
- 5 images per identity

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

We are making BB be mistaken on people

Solution

Baseline

Main idea:

1. Train student distilled network SN (for example, ResNet18)
2. apply FGSM-attack on student network

Score for non-changed images = 1.4

Baseline code was of good quality, essential scripts for data processing and solution submission were provided

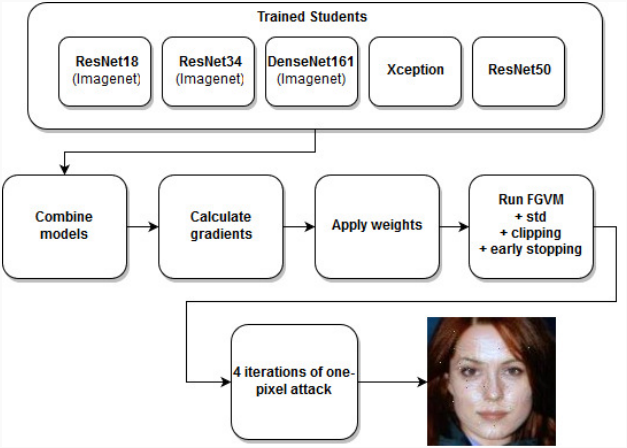
Baseline: FGSM attack

FGSM-attack:

1. For every source identity ID_s do:
2. adv_noise = 0
3. For every target identity ID_t do:
4. pass ID_s through student network: $V_s = \mathrm{SN}(ID_s)$
5. compute loss: $\mathrm{LOSS}(V_s, V_t)$, V_t – vector for ID_t
6. backpropagate loss through SN
7. adv_noise = adv_noise + eps $\cdot$ sign(gradient(ID_s))
8. $ID_s = ID_s$ - adv_noise
9. Repeat 1-8 while SSIM $\geqslant 0.95$

Our Solution: Overview

- Ensemble of several CNNs (that always works, hah)
- FGSM $\rightarrow$ FGVM (use grad value instead of sign)
- One Pixel genetic attacks after FGVM



Code is available at:
https://github.com/Atmyre/MCS2018_Solution

Our Solution: Student Model Results

Таблица 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

Our Solution: Attack 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

- scipy.optimize.differential_evolution
- popsize = 30
- max_iter = 5

Our Solution: 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

Other Ideas And Observations

Other observations

Why didn't we get 1st place? Because we are not so boring!

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

Alexsey Grankov (alexey.grankov) presentation slide

Other observations

Some interesting points:

- BB gives normalized vectors
- Baseline code had a mistake that brought the attack to nothing:

```
#SSIM checking
ssim = compare_ssim(np.array(original_img, dtype=np.float32),
                    np.array(changed_img, dtype=np.float32),
                    multichannel=True)
```

- Training set of 1K faces had more than 5 faces per one person

Other approaches: CW

Survey paper proposed by contest authors:

Method	Black/White box	Targeted/Non-targeted	Strength
L-BFGS [22]	White box	Targeted	* * *
FGSM [23]	White box	Targeted	* * *
BIM & ILCM [35]	White box	Non targeted	****
JSMA [60]	White box	Targeted	* * *
One-pixel [68]	Black box	Non Targeted	**
C&W attacks [36]	White box	Targeted	* * * * *
DeepFool [72]	White box	Non targeted	****
Uni. perturbations [16]	White box	Non targeted	* * * * *
UPSET [146]	Black box	Targeted	****
ANGRI [146]	Black box	Targeted	****
Houdini [131]	Black box	Targeted	****
ATNs [42]	White box	Targeted	****

https://arxiv.org/pdf/1801.00553.pdf

Other approaches: CW

Key Idea:

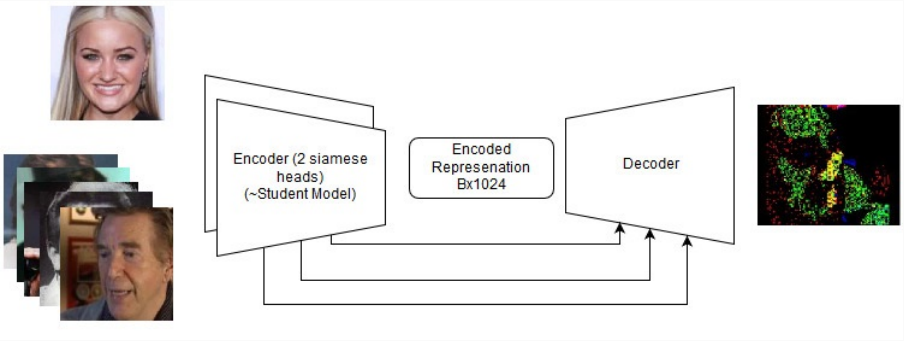
- Create trainable modifier variable to add to image
- use LOSS = scale_const * loss between vectors - SSIM
- binary search scale_const

Performs GREAT at WB, but poor at BB:

https://arxiv.org/pdf/1608.04644.pdf

Other approaches: End2end

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



Other approaches: Ideas

What we didn't manage to try:

- Training student model with attacked pictures
- Use different training parameters (e.g. different augmentation)
- Iterative momentum: https://arxiv.org/abs/1710.06081v3

Conclusion

Conclusion: Impressions

Our competition impressions:

- Good baseline code quality (though had mistakes)
- Good and fast support from organizers
- Little competition time
- Friendly atmosphere and knowledge sharing (though there were not so many participants)

Conclusion: What We Learned

What we have learned:

- Baseline code may contain deadly mistakes
- Look wider: model architecture guess played key role
- Look deeper: it's worth noticing small details (e.g. BB vectors norm)
- Get GPU! (Meh..)