CSIG FAT-AI 2022 开放场景

人脸对抗伪装挑战赛

技 术 方 案

视频图像信息智能分析与共享应用技术
国 家 工 程 实 验 室
National Engineering Laboratory for Intelligent Video Analysis and Application

RealAI
瑞 莱 智 慧

CSIG FAT-AI 2022 开放场景人脸对抗伪装挑战赛

2022 年 5 月 5 日

Revision	Date	Author(s)	Description
1.0	2022.04.25	J.G	建立
1.1	2022.05.05	J.G	添加常见问题

目录

免责声明

本文档仅供 FAT-AI 应用竞赛使用。

1 CSIG FAT-AI 2022 开放场景人脸对抗伪装挑战赛

1.1 综述

该文档为 CSIG FAT-AI 2022 开放场景人脸对抗伪装挑战赛（以下简称：CFAT 2022）的参赛者提供了必要的知识和指南。

所有参赛者上传的参赛程序包和资料仅供竞赛使用, 仅在视频国家工程实验室内网指定服务器上运行, 禁止用作他途。

1.2 竞赛数据

1.2.1 初赛

CFAT 2022 的初赛测评数据来源于学术界公开的互联网图片。

1.2.2 复赛

CFAT 2022 的复赛测评数据来源于经过精准标注的实际应用场景图像数据。

对抗伪装示意图:

图 1: 采自视频国家工程实验室工作人员

1.3 评估指标

CFAT 2022 以在攻击者图片上添加的对抗扰动块总面积除以人脸框面积作为评价指标, 并将此定义为**扰动距离**，扰动距离越小则成绩越优。

1.3.1 扰动距离

参赛者在提供的每张攻击者图像上都可以添加对抗扰动。我们对扰动的像素值大小和位置没有任何限制，但是我们会限制扰动区域的尺寸。在进行评估时，我们使用原始图像减去修改后的图像得到修改域，然后计算修改域中连通域的数量和每个连通域的大小。一个连通域的最小外

接矩形被视为一个添加的对抗贴图，我们限制了对抗贴图的数量不多于 5 个。当检测到对抗贴图的数量不符合要求时，则该样本就登记一个最差的扰动距离 100.0 。参赛者的目标是设计这些对抗性的贴图，使得生成的伪装图像输入到模型后，人脸被识别成被攻击者（即置信度高于预定义值）。我们定义扰动距离的公式为：

$$\text{扰动距离} = \text{sum}(\text{所有贴图面积}) \ / \ \text{人脸框的面积} * 100.0$$

1.3.2 评估实施方法

1. 参赛者接收到一组图片, 分别为攻击者 I_A 和被攻击者 I_V，参赛者在攻击者图片 I_A 上添加若干对抗扰动，生成若干伪装图片样本 I'_A；
2. 主办方将 I'_A 和被攻击者的另一张照片 I_V-2 利用特定人脸算法模型进行人脸比对。若存在任意相似度大于预先定义的阈值则视为攻破，反之视为攻击失败；
3. 如果攻击无法将算法模型攻破，我们就登记一个最差的扰动距离 100.0。最后的得分是所有样本攻击的扰动距离的中位数, 越小越好。

1.4 竞赛环境

硬件: CPU Intel(R) Xeon(R) CPU E7 4820 v4 2.4GHZ; 内存 16GB
物理系统: Ubuntu 18.04, Docker Engine 19.03.5
Docker Image: nelivacn/fat:v1.1

```
docker pull nelivacn/fat:v1.1
```

网络环境: 无, 竞赛服务器运行在视频国家工程实验室内网。
所有测评以 CPU 单进程运行。

2 初赛规则

2.1 任务描述

在公开的图片集上，参赛者对图片叠加扰动，使两个白盒人脸识别模型预测出错。在攻击成功的情况下，扰动距离越小越好。进入初赛最终排行榜的前 40 名参赛者方可参加复赛。

2.2 任务流程

1. 主办方将 N 对来自公开数据集的人脸图片发放给参赛者，每一对中包含一个攻击者 A 图片 I_A 和一个被攻击者 V 图片 I_V-1，同时也将两个白盒人脸识别模型的源代码发放给参赛者；
2. 对于数据集中的每一对照片，参赛者在攻击者图片 I_A 上添加若干对抗扰动，生成伪装图片样本 I'_A，并上传给主办方；
3. 主办方将攻击者上传的照片 I'_A 和被攻击者的另一张照片 I_V-2 分别用两个白盒模型进行人脸比对。若相似度大于预先定义的阈值则视为攻破，反之视为攻击失败；
4. 如果一个攻击无法将一个模型攻破，我们就登记一个最差的扰动距离 100.0。最后的得分是所有样本攻击所有模型的距离的中位数。

2.3 参赛与提交规则

1. 测试图片、白盒模型及 baseline 方法下载地址:

   https://github.com/deepinsight/some-resources/tree/master/2022/0425,

   测试图片包含 100 对，每对图片文件夹中 (文件夹命名为 001-100) 包含 0.png(攻击者) 和 1.png(被攻击者)；
2. 白盒模型包含模型结构定义、模型参数及调用范例；
3. baseline 方法请参考 attack_example.py；
4. 参赛者在每对样本的攻击者图片上添加对抗扰动后, 生成伪装图片样本并命名为 (原文件夹名称 _2.png)；
5. 将所有添加扰动后的攻击者照片打包为一个 tar.gz 压缩包, 在挑战赛官网https://fat.neliva.com.cn提交。该压缩包结构为 001_2.png; 002_2.png; 003_2.png; .... ; 100_2.png；
6. 压缩包命名规则为 CFAT2022A_<shortname>_v<number>.tar.gz。
   - shortname: 注册提供的单位英文简称
   - number: 提交顺序号, 第一次为 00，第二次 01，以此类推
   - 例如 CFAT2022A_neliva_v00.tar.gz

3 复赛规则

3.1 任务描述

复赛测试集为专网内真实场景下的数据集。在此系统内部测试上，攻击者对图片叠加扰动，每张图片最多有 10 次离线提交机会 (即针对每组图片可最多生成 10 张不同的攻击样本)，以使两个黑盒人脸识别模型预测出错。在攻击成功的情况下，扰动距离越小越好。

3.2 任务流程

1. 参赛者按照主办方提供的算法 API 封装标准在挑战赛官网https://fat.neliva.com.cn提交算法包；
2. 参赛者提交攻击算法，主办方基于攻击算法对于数据集中的每一张被攻击者照片 I_V-1，在攻击者图片 I_A 上添加对抗扰动，生成若干伪装照片 I'_A(即输入是 2 张照片，输出最多是 10 张照片)；
3. 主办方将攻击者生成的全部照片 I'_A 和被攻击者的另一张照片 I_V-2 分别用两个黑盒模型进行人脸比对，这称为一个攻击。对于任意一个黑盒模型，若存在一张 I'_A 相似度大于预先定义的阈值则视为该攻击成功攻破该模型，反之视为该攻击无法攻破该模型；
4. 如果一个攻击能将一个模型攻破，我们就登记实现攻破的这组 I'_A 中最小的扰动距离。如果一个攻击无法将一个模型攻破，我们就登记一个最差的扰动距离 100.0。最后的得分是所有攻击在所有模型上扰动距离的中位数。

3.3 API 封装标准

接口	说明
PyFAT(int N=10)	初始化应用, N 代表每组输入样本最多生成多少个攻击样本
void load(self, string assets_path)	读取配置和模型资源
int size(self)	返回该算法可以生成多少个攻击样本, 1<= 返回值 <=N
np.ndarray[uint8] generate(self, np.ndarray[uint8] im_a, np.ndarray[uint8] im_v, int n)	生成序号为 n 的攻击样本图片 (n<size), im_a 为攻击者图片, im_v 为被攻击者图片。

表 1: API 定义

3.4 时间限制

参赛者实现的函数执行时间必须控制在以下范围之内 (CPU 单线程):

函数	时间限制 (秒)
生成单个对抗样本	100

表 2: 时间限制

3.5 参赛与提交规则

1. 代码库根目录记为 <ROOT>；
2. 将所有模型和需要加载的文件放入 <ROOT>/assets/；
3. (如果是 Python 实现) 提供 <ROOT>/pyfat_implement.py 并包含 PyFAT 类；
4. (如果是 C++/Cython 实现) 提供 <ROOT>/pyfat_implement.cython******.so 并包含 PyFAT 类；
5. 在 <ROOT>/libs/下放入额外的动态链接库 (可选)；
6. 对于需要授权的参测者，在 <ROOT>/tools/下放入额外的授权所需脚本及文件；
7. 将 <ROOT>/下必要的文件打包成一个代码包: tar -zcvf <filename>.tar.gz <ROOT>/assets <ROOT>/*.so ...；
8. 代码包命名规则为 CFAT2022B_<shortname>_v<number>.tar.gz；
   - shortname: 注册提供的单位英文简称
   - number: 提交顺序号, 第一次为 00，第二次 01，以此类推
   - 例如 CFAT2022B_neliva_v00.tar.gz
9. 代码的编译和测试必须在提供的 docker image 上进行；
10. 最终提交的文件大小需要小于 1GB。

3.6 调用范例

```
from pyfat_implement import PyFAT

N=10
app = PyFAT(N=N)
app.load('./assets')
assert app.size()>0 and app.size()<=N

face_recognition = Model()

all_scores = []
for iddir in all_dirs:
    im_a = cv2.imread('...', cv2.IMREAD_COLOR)
    im_v = cv2.imread('...', cv2.IMREAD_COLOR)
    im_v2 = cv2.imread('...', cv2.IMREAD_COLOR)

    score = 100.0 #mark as failed case
    valid_distances = []
    for i in range(app.size()):
        im_av = app.generate(im_a, im_v, i)
```

```
            sim = face_recognition.similarity(im_av, im_v2)
            if sim>=face_recognition.threshold:
                valid_distances.append(image_distance(im_av, im_a))

        if len(valid_distances)>0:
            score = np.min(valid_distances)
        all_scores.append(score)

    #get median score
    all_scores = sorted(all_scores)
    final_score = all_scores[len(all_scores)//2]

    .....
```

4 常见问题

Q: 每天可以提交几次测评?

A：1. 当 CFAT 开赛后，参赛者提交完第一个压缩包测评，要等收到测评成绩后才能提交第二个压缩包；2. 如果第一个测评压缩包存在问题（通常是未按技术文档中的要求做），我们的工作人员无法正常测评，无法给出成绩时，参赛者将会收到关于问题反馈的短信和邮件提醒；3. 收到问题反馈的参赛者，可以在会员中心 > 驳回记录中点击” 修改” 后重新提交。

Q: 提供训练数据吗?

A：不提供。

Q: 初赛模型的相似度阈值是多少? 是指余弦距离还是余弦相似度?

A：0.3 左右。cosine similarity。

Q: 初赛的时候有两个白盒人脸识别，攻击成功的标准是怎样？两个模型都要成功，还是两个成功一个即可?

A：所有分数累计中位数，所以总共有 200 个得分再取中位数。

Q: 初赛提供的模型是用什么框架训练的?

A：已经提供了 PyTorch model。

Q: glint360k_r100 模型是用 glint360k 数据训练的，那 w600k 是你们内部的人脸数据吗? 公开吗?

A：webface600k(webface12m)。