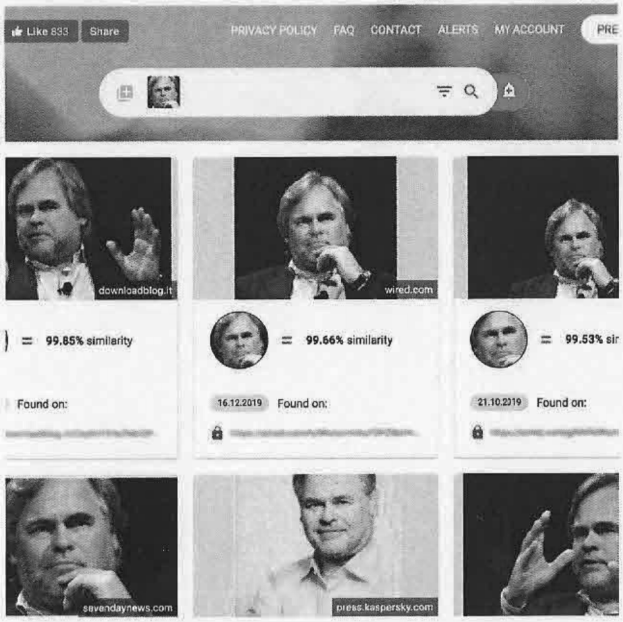
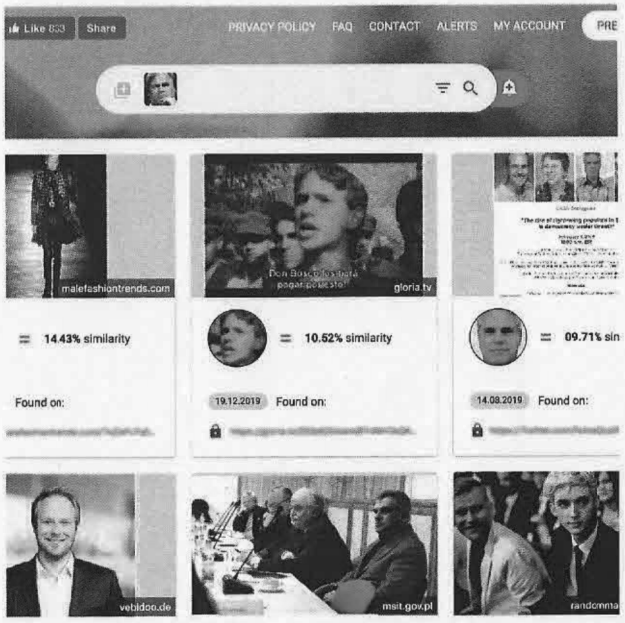
Постановка задачи

Задача: Исследовать и предложить алгоритм создания фильтров для изображений, «анонимизирующий» их для систем распознавания лиц на основе глубокого обучения.



До применения фильтра



После применения фильтра

Формальная постановка задачи

Пусть дана модель распознавания лиц $\boldsymbol{f(x): x \in X \rightarrow y \in R^d}$,где $\boldsymbol{X}$– множество лиц, а $\boldsymbol{R^d}$ – представление лица внутри нейронной сети (вложение), которая определяет вложение заданного лица. Требуется найти пример $\boldsymbol{x^* = x + \Delta x}$ в окрестности примера $\boldsymbol{x}$, такой что:

$$\boldsymbol{\Delta x = arg \max_{\Delta x} \|F(x + \Delta x) - F(x)\|_2 , \|\Delta x\|_\infty < \varepsilon.}$$

Опыт исследователей в данном направлении

- Объяснение и использование атакующих примеров. Goodfellow, 2014.
- Улучшение генерации атакующих примеров с использованием моментум. Dong, 2018.
- Улучшение переносимости атакующих примеров с использованием входного разнообразия. Xie, 2019
- Переносимость атакующих примеров для глубоких моделей распознавания лиц. Zhong, 2020.

Особенности и проблемы задачи

- У нас нет точного знания о системе, которая будет использоваться для распознавания;
- Качество изображения не должно заметно меняться для человеческого восприятия;
- Так как нам неважно, как изменится предсказание модели, то атака носит характер не таргетированной;
- Защита от атакующих примеров.

Методы решения

- Создавать атакующие примеры будем для передовых моделей распознавания лиц в открытом доступе;
- Для создания «анонимизированных» изображений будем использовать атакующие примеры (Adversarial attack);
- Экстраполяция атакующих примеров будет осуществляться с помощью улучшения атак методом итеративной вероятностной аугментации изображений.

Алгоритм

- Вход: классификатор f с функцией потерь J без последнего слоя, пример x;
- Гипер-параметры: ε – размер возмущений, M – количество итераций, μ – сила импульса;
- Выход: Атакующий пример x^*.

1 $\alpha = \varepsilon/M;\ g_0 = 0;\ x_0^* = x.$

2 for t = 0 to M – 1 do

3 x_t^* = T(x_t^*)

4 Пропускаем пример x_t^* через сеть и получаем градиент $\nabla_x J(x_t^*)$

5 $g_{t+1} = \mu g_t + \frac{\nabla_x J(x_t^*)}{\|\nabla_x J(x_t^*)\|}$

6 $x_{t+1}^* = x_t^* + \prod_{x+S}(\alpha \cdot sign(g_{t+1}))$

7 end for

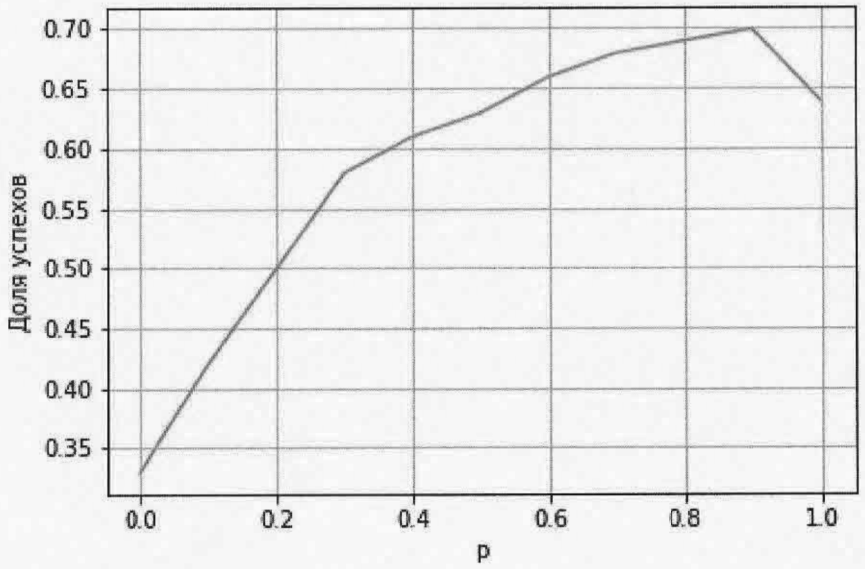
8 return x_M^*

$$T(x_t^*; \mathrm{p}; \mathrm{r}) = \begin{cases} Aug(x_t^*; r; \alpha), \text{с вероятностью } p \\ x_t^*, \text{с вероятностью } 1-p \end{cases}$$

$r \sim U(135, 160)$

$\alpha \sim U(-15, 15).$

Гипер параметр p



Процент успешных атакующих примеров

	Веб-сервис	Другая модель
Базовый алгоритм	34.5	53.5
Алгоритм со случайной аугментацией	55.5	66.1
Алгоритм со случайной аугментацией и поворотом	59.4	70.0

Перспективы развития

- рассмотрение других аугментаций изображения;
- применение других методов генерации атакующих примеров;
- изучение влияния набора данных на переносимость атакующих примеров;
- реализация решения в виде продукта (например мобильное приложение).

Результаты

- исследована генерация атакующих примеров для глубоких нейронных сетей в задаче распознавания лиц;
- исследована переносимость атакующих примеров между различными глубокими нейронными сетями в задаче распознавания лиц;
- разработан алгоритм генерации атакующих примеров для глубоких нейронных сетей в задаче распознавания лиц;
- реализован предложенный алгоритм;
- улучшена переносимость атакующих примеров.

Спасибо за внимание