# Adversarial purification for IQA models

## Название темы

Защита от атак на метрики качества изображений с помощью предобработки входных данных.

## Постановка задачи

Вы --- начинающий специалист в области безопасности искусственного интеллекта. К Вам обращается компания, использующая кодеки в продакшене. Для выбора лучшего кодека они используют нейросетевые метрики качества изображений и видео. Однако за последнее время компания столкнулась с рядом инцидентов, где метрика показывала необоснованное завышение показателей за счёт небольших изменений во входных данных. Компания просит Вас как можно быстрее решить эту проблему.

Компания поделилась с Вами исходным кодом и весами нейросетевой метрики на PyTorch. Это модуль, в котором содержатся класс самой нейронной сети ( `nn.Module` ).

Ваша цель --- разработать метод защиты от состязательных атак на метрики качества путём очищения входных данных.

В Вашем распоряжении:

- код нейросетевой метрики качества изображения Linearity;
- датасет с разными атакованными изображениями;

## Формат решения

Решение сдаётся в виде файла `defense.py` , который должен содержать класс `Defense` с методом apply_defense:

```
def apply_defense(self, images: torch.Tensor) -> torch.Tensor:
```

- на вход принимает батч атакованных изображений **images**;
- возвращает очищенное от состязательного возмущения изображение;

Внутри класса можно реализовать, какую угодно защиту: составную из разных простых, даже нейросетевую, но важно не выйти за лимиты.

# Правила оценивания

Для оценивания используются следующие метрики, усредненные по типу атаки:

- Gain Score = $\frac{1}{n}\sum_{i=1}^{n}\frac{\left|f(x_i')-f(x_i)\right|}{f(x_i)} * 100$ (меньше-лучше),
  где $x_i$ -- исходное изображение $x_i'$ -- атакованное изображение, а f(.) -- метрика качества изображения
- Quality Score = $\frac{SSIM+PSNR/80}{2}$ (больше - лучше) --- значения SSIM и PSNR между исходным и восстановленным изображением
- SRCC (больше - лучше) между Linearity на исходных изображениях и на защищенных изображениях

**Результаты оценки в системе - разность каждой из метрик между вашей защитой и бейзлайном.**

# Файлы

- `dataset.zip` - архив с атакованными и исходными изображениями (public test) ссылка на скачивание
- `test.py` - файл для прогона решения и получения результирующих метрик, используйте `python test.py --help` для описания входных параметров
- `defense.py` - файл с макетом класса Defense (ваше решение)
- `model.py` - код метрики Linearity
- `p1q2.pth` - чекпоинт метрики Linearity, должен лежать рядом с `model.py` ссылка на скачивание