Журавлев А.Д ББМО-02-23 Номер 7

Клонирование репозитория

```
!git clone https://github.com/ewatson2/EEL6812_DeepFool_Project.git
```

```
Cloning into 'EEL6812_DeepFool_Project'...
remote: Enumerating objects: 96, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 96 (delta 2), reused 1 (delta 1), pack-reused 93 (from 1)
Receiving objects: 100% (96/96), 33.99 MiB | 15.13 MiB/s, done.
Resolving deltas: 100% (27/27), done.
```

Смена директории и импорт библиотек

Смена директории и импорт библиотек

```
[16] import os
     os.chdir('EEL6812_DeepFool_Project')
```

```
[17] import numpy as np
     import json
     import torch
     from torch.utils.data import DataLoader, random_split
     from torchvision import datasets, models
     from torchvision.transforms import transforms
```

```
[18] from models.project_models import FC_500_150, LeNet_CIFAR, LeNet_MNIST, Net
     from utils.project_utils import get_clip_bounds, evaluate_attack, display_attack
```

Установка случайного значения – номер в списке группы «7»

Установка случайного значения - номер в списке группы "7"

```
[19] rand_seed = 7
```

```
np.random.seed(rand_seed)
torch.manual_seed(rand_seed)

use_cuda = torch.cuda.is_available()
device = torch.device('cuda' if use_cuda else 'cpu')
```

Загрузка датасета MNIST

+ Код + Текст

Загрузка датасета MNIST

15 сек.

```
[22] mnist_mean = 0.5
     mnist_std = 0.5
     mnist_dim = 28

     mnist_min, mnist_max = get_clip_bounds(mnist_mean,
                                            mnist_std,
                                            mnist_dim)
     mnist_min = mnist_min.to(device)
     mnist_max = mnist_max.to(device)

     mnist_tf = transforms.Compose([
         transforms.ToTensor(),
         transforms.Normalize(
             mean=mnist_mean,
             std=mnist_std)])

     mnist_tf_train = transforms.Compose([
         transforms.RandomHorizontalFlip(),
         transforms.ToTensor(),
         transforms.Normalize(
             mean=mnist_mean,
             std=mnist_std)])

     mnist_tf_inv = transforms.Compose([
         transforms.Normalize(
             mean=0.0,
             std=np.divide(1.0, mnist_std)),
         transforms.Normalize(
             mean=np.multiply(-1.0, mnist std),
```

Загрузка датасета CIFAR-10

Загрузка датасета CIFAR-10

✓ 3 мин.

```
[23] cifar_mean = [0.491, 0.482, 0.447]
     cifar_std = [0.202, 0.199, 0.201]
     cifar_dim = 32

     cifar_min, cifar_max = get_clip_bounds(cifar_mean,
                                            cifar_std,
                                            cifar_dim)
     cifar_min = cifar_min.to(device)
     cifar_max = cifar_max.to(device)

     cifar_tf = transforms.Compose([
         transforms.ToTensor(),
         transforms.Normalize(
             mean=cifar_mean,
             std=cifar_std)])

     cifar_tf_train = transforms.Compose([
         transforms.RandomCrop(
             size=cifar_dim,
             padding=4),
         transforms.RandomHorizontalFlip(),
         transforms.ToTensor(),
         transforms.Normalize(
             mean=cifar_mean,
             std=cifar_std)])

     cifar_tf_inv = transforms.Compose([
         transforms.Normalize(
             mean=[0.0, 0.0, 0.0]
```

Настройка DataLoader

Настройка DataLoader

```
batch_size = 64
workers = 4
mnist_loader_train = DataLoader(mnist_train, batch_size=batch_size, shuffle=True, num_workers=workers)
mnist_loader_val = DataLoader(mnist_val, batch_size=batch_size, shuffle=False, num_workers=workers)
mnist_loader_test = DataLoader(mnist_test, batch_size=batch_size, shuffle=False, num_workers=workers)
cifar_loader_train = DataLoader(cifar_train, batch_size=batch_size, shuffle=True, num_workers=workers)
cifar_loader_val = DataLoader(cifar_val, batch_size=batch_size, shuffle=False, num_workers=workers)
cifar_loader_test = DataLoader(cifar_test, batch_size=batch_size, shuffle=False, num_workers=workers)
```

```
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:557: UserWarning: This DataLoader will create 4 worker processes in t
  warnings.warn(_create_warning_msg(
```

FGSM атака

Стойкость к атаке моделей LeNet, FC на датасете MNIST и стойкость к атаке моделей Network-In-Network, LeNet на датасете CIFAR-10

LeNet MNIST

fgsm_eps = 0.001

```
model = LeNet_MNIST().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_lenet.pth'))

evaluate_clean(model, mnist_loader_test, device)


evaluate_attack(f'mnist_lenet_fgsm{fgsm_eps}.csv', 'results',
                device, model, mnist_loader_test,
                mnist_min, mnist_max, fgsm_eps, is_fgsm=True)


if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-63-2c6783ebe969>:2: FutureWarning: You are using `torch.loa
  model.load_state_dict(torch.load('weights/clean/mnist_lenet.pth'))
Точность до атаки: 98.34%
FGSM Test Error : 1.69%
FGSM Robustness : 8.06e-04
FGSM Time (All Images) : 1.24 s
FGSM Time (Per Image) : 123.91 us
```

fgsm_eps = 0.02

```
<ipython-input-73-2c6783ebe969>:2: FutureWarn
  model.load_state_dict(torch.load('weights/c
/usr/local/lib/python3.10/dist-packages/torch
  warnings.warn(_create_warning_msg(
Точность до атаки: 98.34%
FGSM Test Error : 2.56%
FGSM Robustness : 1.59e-02
FGSM Time (All Images) : 0.86 s
FGSM Time (Per Image) : 86.20 us
```

fgsm_eps = 0.5

```
    torch.cuda.empty_cache()
```

```
<ipython-input-81-2c6783ebe969>:2: FutureW
  model.load_state_dict(torch.load('weight
/usr/local/lib/python3.10/dist-packages/to
  warnings.warn(_create_warning_msg(
Точность до атаки: 98.34%
FGSM Test Error : 82.92%
FGSM Robustness : 3.83e-01
FGSM Time (All Images) : 0.85 s
FGSM Time (Per Image) : 85.18 us
```

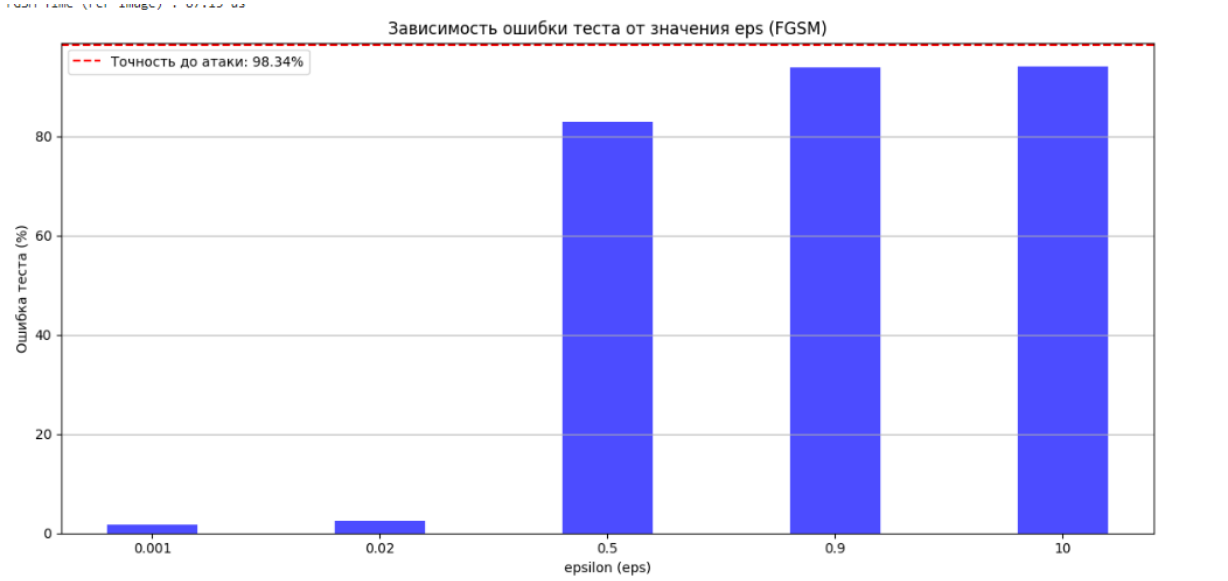
fgsm_eps = 0.9

```
  warnings.warn(_create_warning_msg(
Точность до атаки: 98.34%
/usr/local/lib/python3.10/dist-packages/torch,
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 93.80%
FGSM Robustness : 6.81e-01
FGSM Time (All Images) : 1.10 s
FGSM Time (Per Image) : 110.05 us
```

fgsm_eps = 10

```
<ipython-input-91-2c6783ebe969>:2: FutureWarnir
  model.load_state_dict(torch.load('weights/cle
/usr/local/lib/python3.10/dist-packages/torch/u
  warnings.warn(_create_warning_msg(
Точность до атаки: 98.34%
/usr/local/lib/python3.10/dist-packages/torch/u
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 94.15%
FGSM Robustness : 1.46e+00
FGSM Time (All Images) : 0.87 s
FGSM Time (Per Image) : 87.15 us
```

График



FC MNIST

fgsm_eps = 0.001

```
model = FC_500_150().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_fc.pth'))

evaluate_clean(model, mnist_loader_test, device)

evaluate_attack(f'mnist_fc_fgsm{fgsm_eps}.csv', 'results',
                device, model, mnist_loader_test,
                mnist_min, mnist_max, fgsm_eps, is_fgsm=True)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-64-56e1b1ab84b3>:2: FutureWarning: You are using `torch.l
  model.load_state_dict(torch.load('weights/clean/mnist_fc.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:5
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
FGSM Test Error : 3.07%
FGSM Robustness : 8.08e-04
FGSM Time (All Images) : 0.65 s
FGSM Time (Per Image) : 64.56 us
```

fgsm_eps = 0.02

```
torch.cuda.empty_cache()
```

```
<ipython-input-75-56e1b1ab84b3>:2: FutureW
  model.load_state_dict(torch.load('weight
/usr/local/lib/python3.10/dist-packages/to
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
FGSM Test Error : 5.54%
FGSM Robustness : 1.60e-02
FGSM Time (All Images) : 0.58 s
FGSM Time (Per Image) : 58.29 us
```

fgsm_eps = 0.5

```
<ipython-input-82-56e1b1ab84b3>:2: Future
  model.load_state_dict(torch.load('weigh
/usr/local/lib/python3.10/dist-packages/t
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
/usr/local/lib/python3.10/dist-packages/t
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 99.21%
FGSM Robustness : 3.86e-01
FGSM Time (All Images) : 0.59 s
FGSM Time (Per Image) : 58.66 us
```

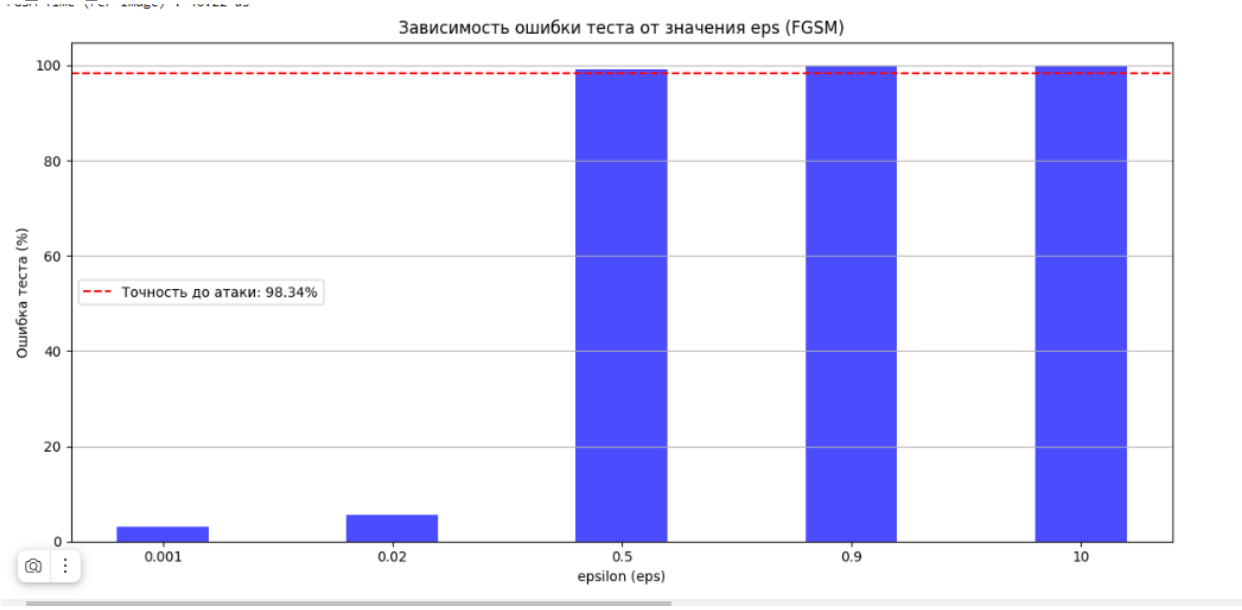
fgsm_eps = 0.9

```
<ipython-input-87-56e1b1ab84b3>:2: FutureWarn
  model.load_state_dict(torch.load('weights/c
/usr/local/lib/python3.10/dist-packages/torch
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
/usr/local/lib/python3.10/dist-packages/torch
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 99.87%
FGSM Robustness : 6.86e-01
FGSM Time (All Images) : 0.51 s
FGSM Time (Per Image) : 51.04 us
```

fgsm_eps = 10

```
<ipython-input-92-56e1b1ab84b3>:2: FutureWar
  model.load_state_dict(torch.load('weights/
/usr/local/lib/python3.10/dist-packages/torc
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
/usr/local/lib/python3.10/dist-packages/torc
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 99.87%
FGSM Robustness : 1.47e+00
FGSM Time (All Images) : 0.46 s
FGSM Time (Per Image) : 46.22 us
```

График



Network-In-Network CIFAR-10

fgsm_eps = 0.001

```
model = Net().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_nin.pth'))

evaluate_clean(model, cifar_loader_test, device)

evaluate_attack(f'cifar_nin_fgsm{fgsm_eps}.csv', 'results',
                device, model, cifar_loader_test,
                cifar_min, cifar_max, fgsm_eps, is_fgsm=True)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-65-91cbd74260da>:2: FutureWarning: You are using `torch.
  model.load_state_dict(torch.load('weights/clean/cifar_nin.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
FGSM Test Error : 10.12%
FGSM Robustness : 8.92e-04
FGSM Time (All Images) : 1.17 s
FGSM Time (Per Image) : 117.12 us
```

fgsm_eps = 0.02

```
<ipython-input-76-91cbd74260da>:2: FutureWarning:
  model.load_state_dict(torch.load('weights/clean
/usr/local/lib/python3.10/dist-packages/torch/uti
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
/usr/local/lib/python3.10/dist-packages/torch/uti
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 30.76%
FGSM Robustness : 1.78e-02
FGSM Time (All Images) : 1.45 s
FGSM Time (Per Image) : 144.64 us
```

fgsm_eps = 0.5

```
<ipython-input-83-91cbd74260da>:2: Future
  model.load_state_dict(torch.load('weigh
/usr/local/lib/python3.10/dist-packages/t
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
/usr/local/lib/python3.10/dist-packages/t
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 82.67%
FGSM Robustness : 4.40e-01
FGSM Time (All Images) : 1.06 s
FGSM Time (Per Image) : 106.36 us
```

fgsm_eps = 0.9

```
<ipython-input-88-91cbd74260da>:2: FutureWarning
  model.load_state_dict(torch.load('weights/clea
/usr/local/lib/python3.10/dist-packages/torch/ut
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
/usr/local/lib/python3.10/dist-packages/torch/ut
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 84.62%
FGSM Robustness : 7.79e-01
FGSM Time (All Images) : 1.01 s
FGSM Time (Per Image) : 101.45 us
```

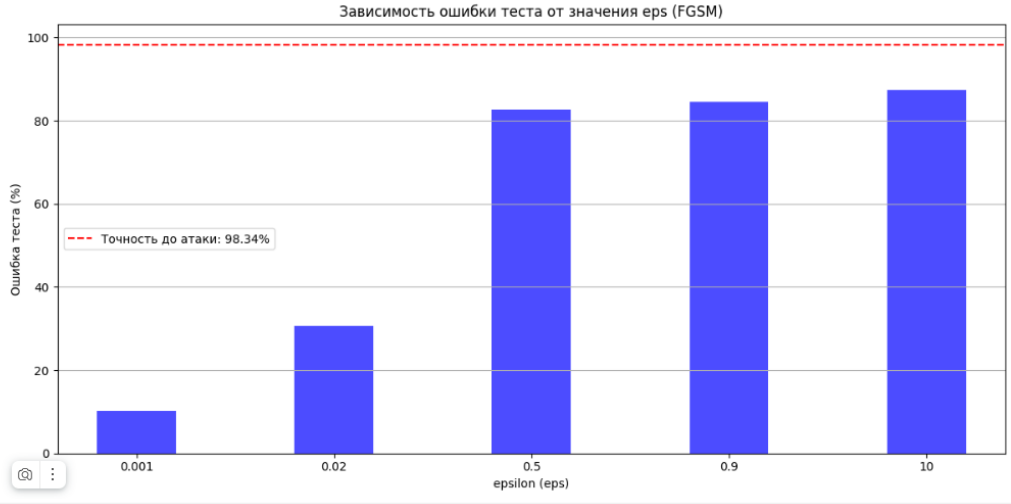
fgsm_eps = 10

```
    torch.cuda.empty_cache()
```

```
<ipython-input-94-91cbd74260da>:2: FutureWa
  model.load_state_dict(torch.load('weights,
/usr/local/lib/python3.10/dist-packages/tor
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
FGSM Test Error : 87.50%
FGSM Robustness : 2.46e+00
FGSM Time (All Images) : 1.05 s
FGSM Time (Per Image) : 105.28 us
```

График

FGSM Time (Per Image) : 105.28 us



LeNet CIFAR-10

fgsm_eps = 0.001

```
model = LeNet_CIFAR().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth'))

evaluate_clean(model, cifar_loader_test, device)

evaluate_attack(f'cifar_lenet_fgsm{fgsm_eps}.csv', 'results',
                device, model, cifar_loader_test,
                cifar_min, cifar_max, fgsm_eps, is_fgsm=True)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-50-dffa1e4e0d26>:2: FutureWarning: You are using `torch.l
  model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:5
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
FGSM Test Error : 22.72%
FGSM Robustness : 8.92e-04
FGSM Time (All Images) : 1.36 s
FGSM Time (Per Image) : 136.39 us
```

fgsm_eps = 0.02

```
<ipython-input-77-dffa1e4e0d26>:2: FutureWarning:
  model.load_state_dict(torch.load('weights/clean/
/usr/local/lib/python3.10/dist-packages/torch/util
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
/usr/local/lib/python3.10/dist-packages/torch/util
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 47.76%
FGSM Robustness : 1.78e-02
FGSM Time (All Images) : 1.30 s
FGSM Time (Per Image) : 129.97 us
```

fgsm_eps = 0.5

```
<ipython-input-84-dffa1e4e0d26>:2: FutureWarning:
  model.load_state_dict(torch.load('weights/clean,
/usr/local/lib/python3.10/dist-packages/torch/uti
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
/usr/local/lib/python3.10/dist-packages/torch/uti
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 95.17%
FGSM Robustness : 4.40e-01
FGSM Time (All Images) : 1.11 s
FGSM Time (Per Image) : 111.30 us
```

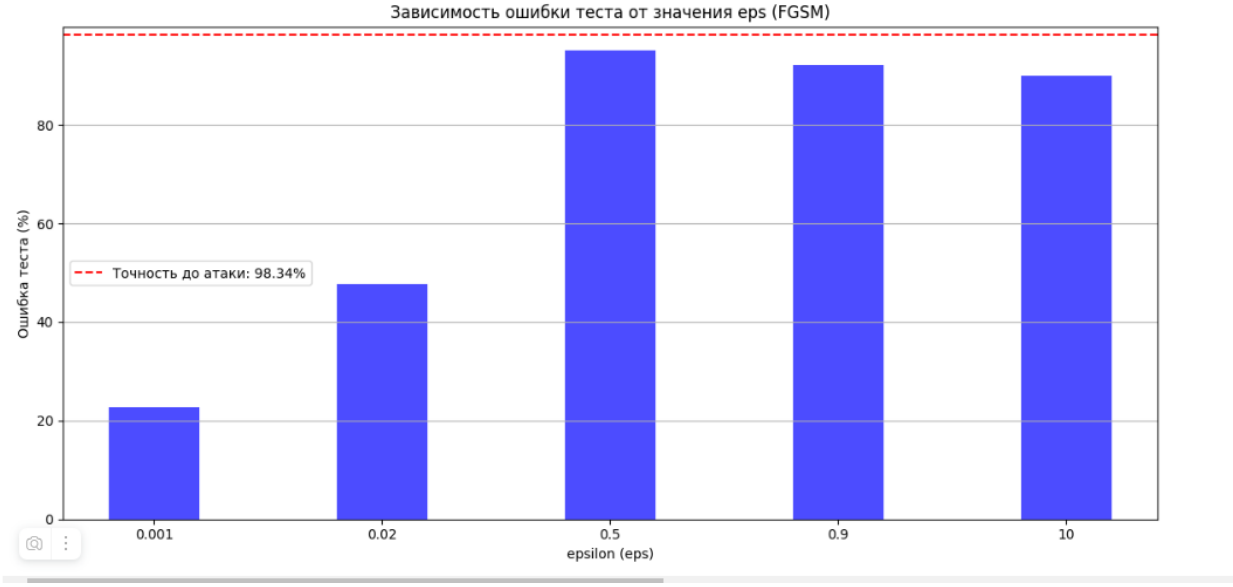
fgsm_eps = 0.9

```
<ipython-input-89-dffa1e4e0d26>:2: FutureWar
  model.load_state_dict(torch.load('weights/
/usr/local/lib/python3.10/dist-packages/torc
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
/usr/local/lib/python3.10/dist-packages/torc
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 92.04%
FGSM Robustness : 7.80e-01
FGSM Time (All Images) : 1.13 s
FGSM Time (Per Image) : 113.44 us
```

fgsm_eps = 10

```
/usr/local/lib/python3.10/dist-packages/torch/
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
/usr/local/lib/python3.10/dist-packages/torch/
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 89.90%
FGSM Robustness : 2.47e+00
FGSM Time (All Images) : 1.15 s
FGSM Time (Per Image) : 115.27 us
```

График


Зависимость ошибки теста от значения eps (FGSM)
Точность до атаки: 98.34%
Ошибка теста (%)
0
20
40
60
80
0.001
0.02
0.5
0.9
10
epsilon (eps)

DeepFool атака

Стойкость к атаке моделей LeNet, FC на датасете MNIST и стойкость к атакае моделей Network-In-Network, LeNet на датасете CIFAR-10

LeNet MNIST

```
model = LeNet_MNIST().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_lenet.pth'))

evaluate_clean(model, mnist_loader_test, device)

evaluate_attack('mnist_lenet_deepfool.csv', 'results',
                device, model, mnist_loader_test,
                mnist_min, mnist_max, deep_args, is_fgsm=False)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-67-9a4fabdb4dc1>:2: FutureWarning: You are using `to
  model.load_state_dict(torch.load('weights/clean/mnist_lenet.pth')
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader
  warnings.warn(_create_warning_msg(
Точность до атаки: 98.34%
DeepFool Test Error : 98.74%
DeepFool Robustness : 9.64e-02
DeepFool Time (All Images) : 193.32 s
DeepFool Time (Per Image) : 19.33 ms
```

FC MNIST

```
model = FC_500_150().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_fc.pth'))

evaluate_clean(model, mnist_loader_test, device)

evaluate_attack('mnist_fc_deepfool.csv', 'results',
                device, model, mnist_loader_test,
                mnist_min, mnist_max, deep_args, is_fgsm=False)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-69-f4287413aeee>:2: FutureWarning: You are using `torch
  model.load_state_dict(torch.load('weights/clean/mnist_fc.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py
  warnings.warn(_create_warning_msg(
Точность до атаки: 97.03%
DeepFool Test Error : 97.92%
DeepFool Robustness : 6.78e-02
DeepFool Time (All Images) : 141.81 s
DeepFool Time (Per Image) : 14.18 ms
```

Network-In-Network CIFAR-10

```
model = Net().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_nin.pth'))

evaluate_clean(model, cifar_loader_test, device)

evaluate_attack('cifar_nin_deepfool.csv', 'results',
                device, model, cifar_loader_test,
                cifar_min, cifar_max, deep_args, is_fgsm=False)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-70-d39c82e071ac>:2: FutureWarning: You are using `tor
  model.load_state_dict(torch.load('weights/clean/cifar_nin.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.
  warnings.warn(_create_warning_msg(
Точность до атаки: 90.72%
DeepFool Test Error : 93.76%
DeepFool Robustness : 2.12e-02
DeepFool Time (All Images) : 185.12 s
DeepFool Time (Per Image) : 18.51 ms
```

LeNet CIFAR-10

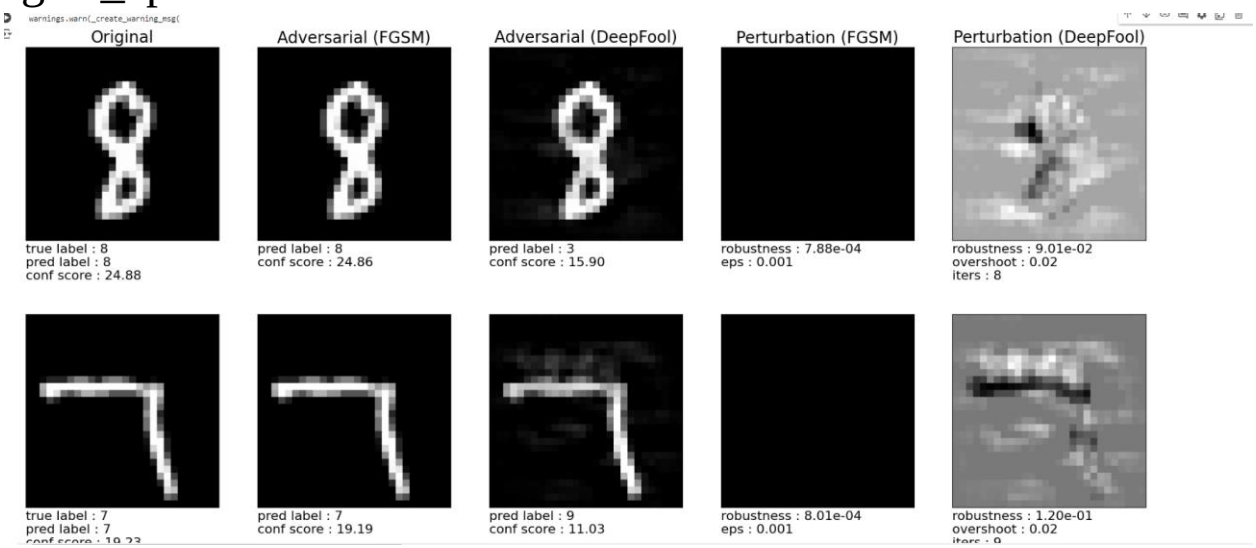
```
model = LeNet_CIFAR().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth'))

evaluate_clean(model, cifar_loader_test, device)

evaluate_attack('cifar_lenet_deepfool.csv', 'results',
                device, model, cifar_loader_test,
                cifar_min, cifar_max, deep_args, is_fgsm=False)

if device.type == 'cuda':
    torch.cuda.empty_cache()
```

```
<ipython-input-71-71a3964ca979>:2: FutureWarning: You are using `torch.
  model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth'))
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:
  warnings.warn(_create_warning_msg(
Точность до атаки: 78.66%
DeepFool Test Error : 87.81%
DeepFool Robustness : 1.78e-02
DeepFool Time (All Images) : 73.27 s
DeepFool Time (Per Image) : 7.33 ms
```

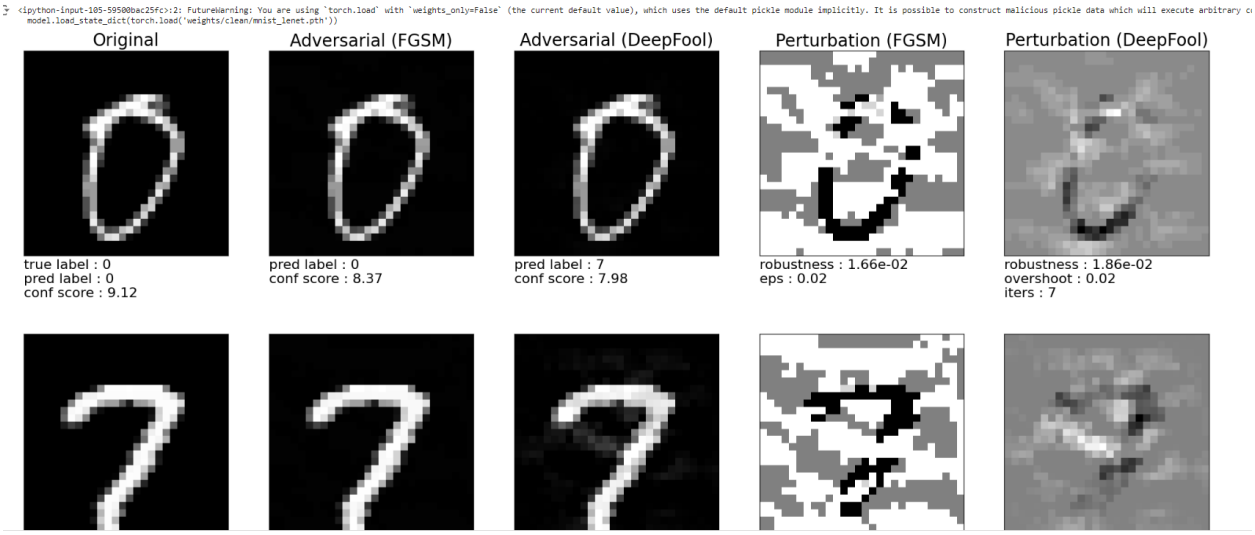
Визуальное представление

LeNet MNIST

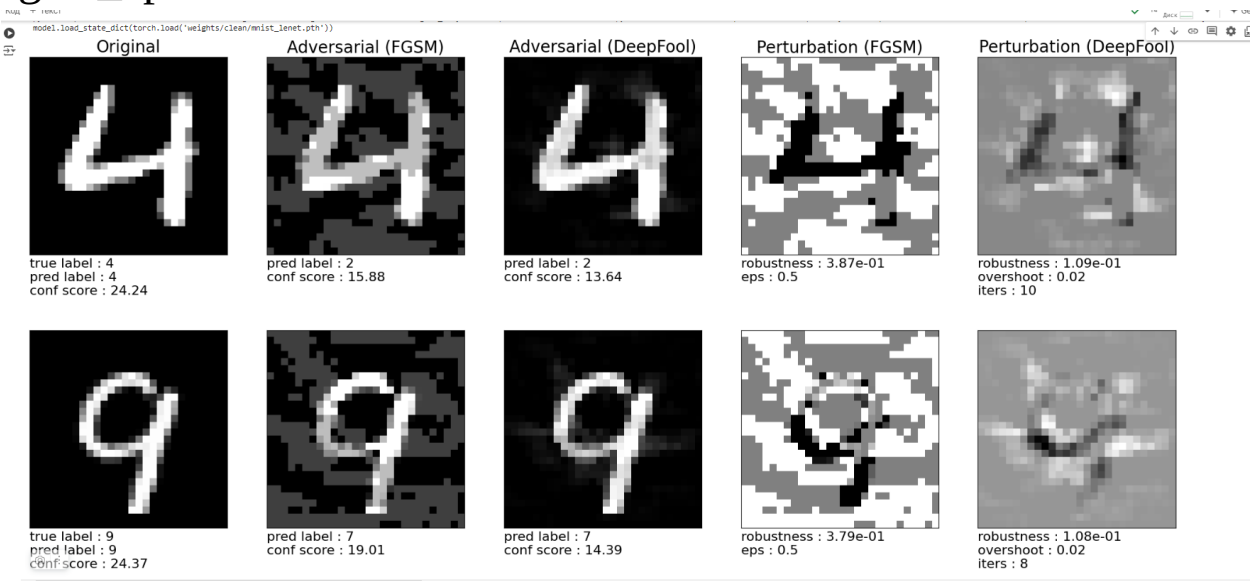
fgsm_eps = 0.001



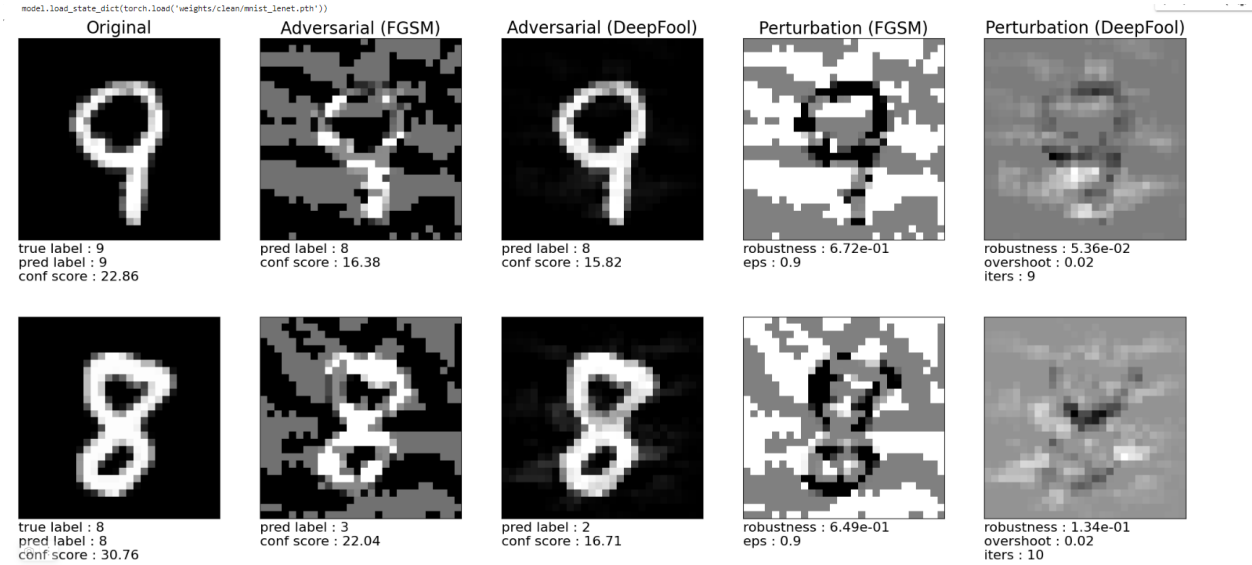
fgsm_eps = 0.02



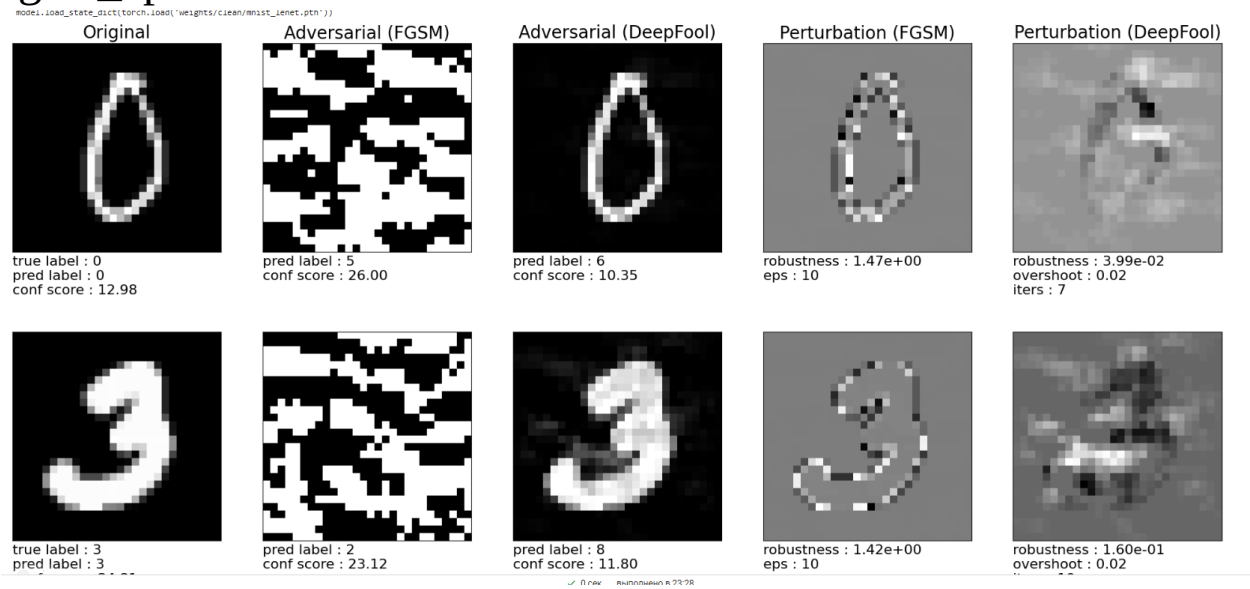
fgsm_eps = 0.5



fgsm_eps = 0.9

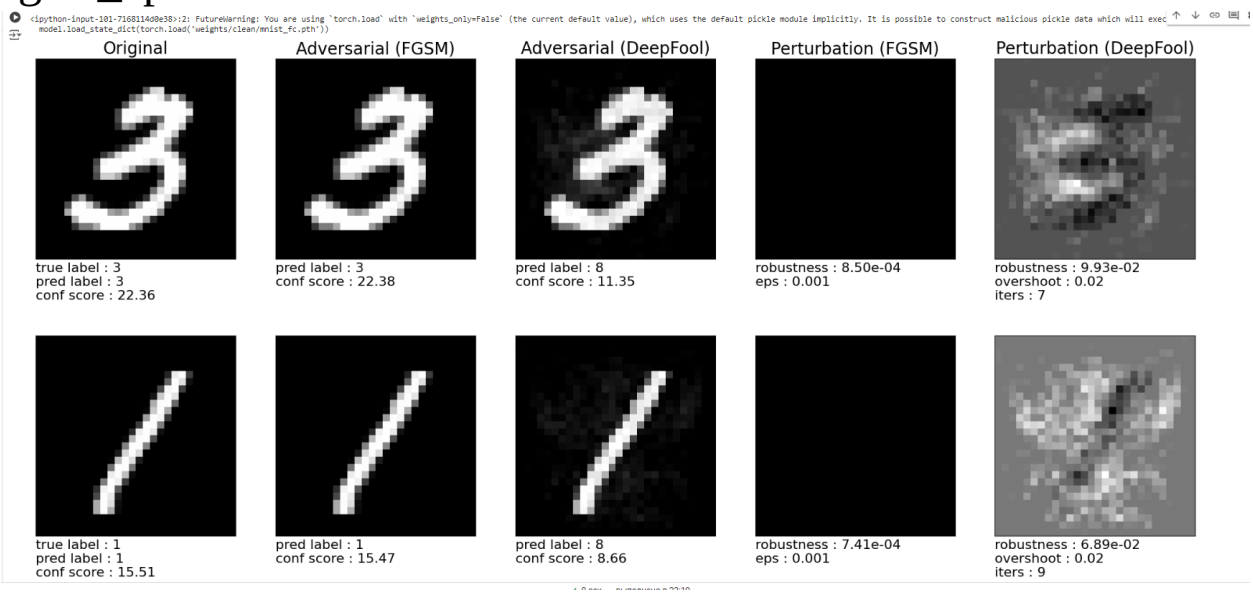


fgsm_eps = 10

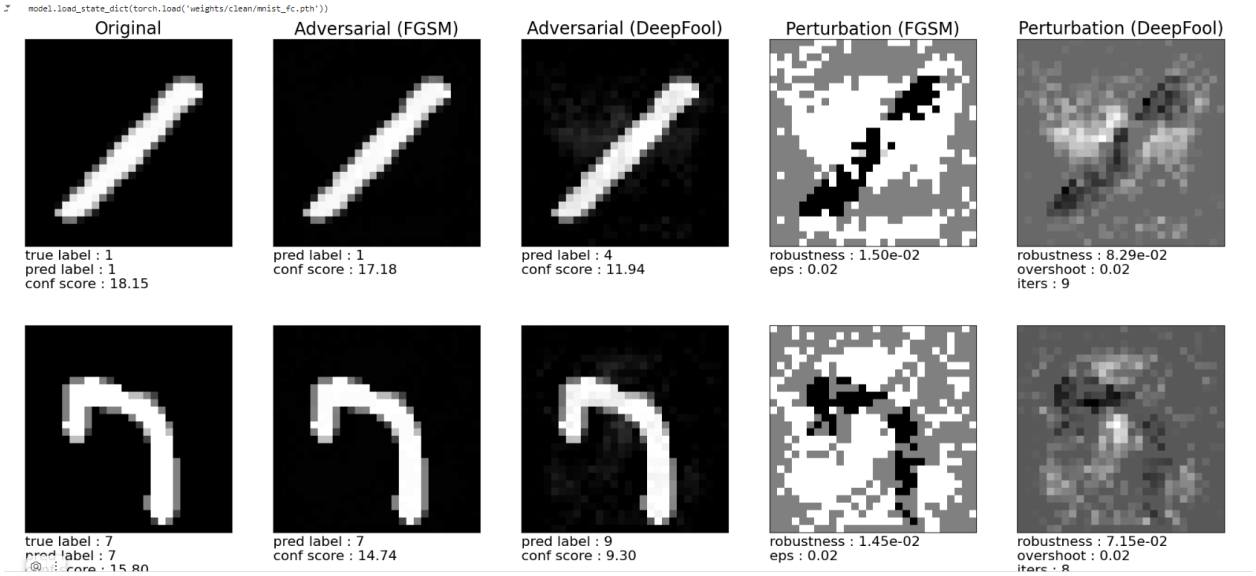


FC MNIST

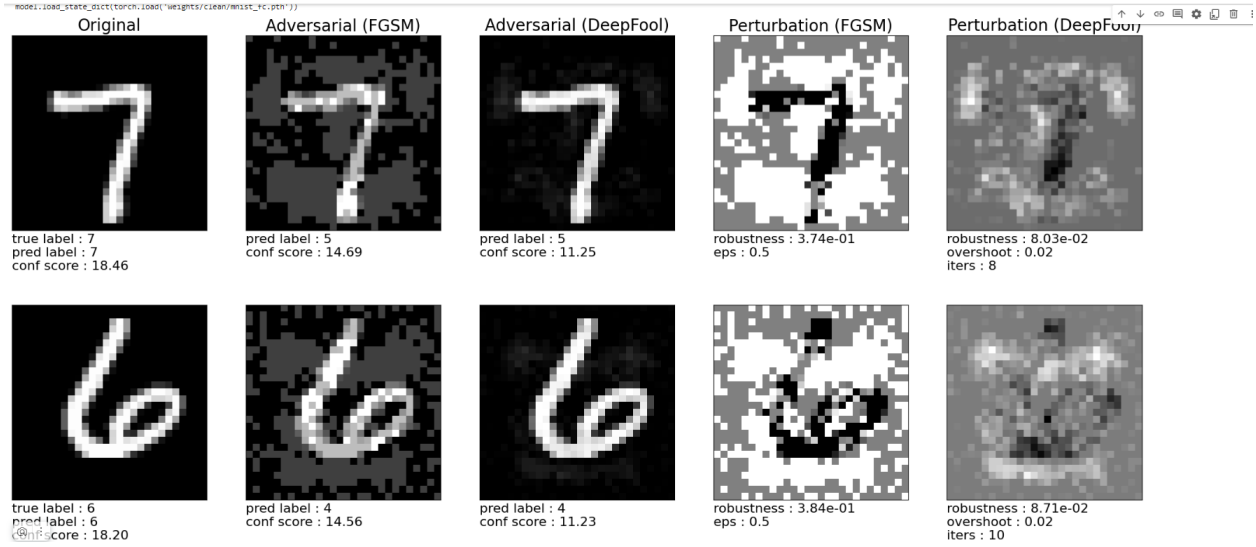
fgsm_eps = 0.001



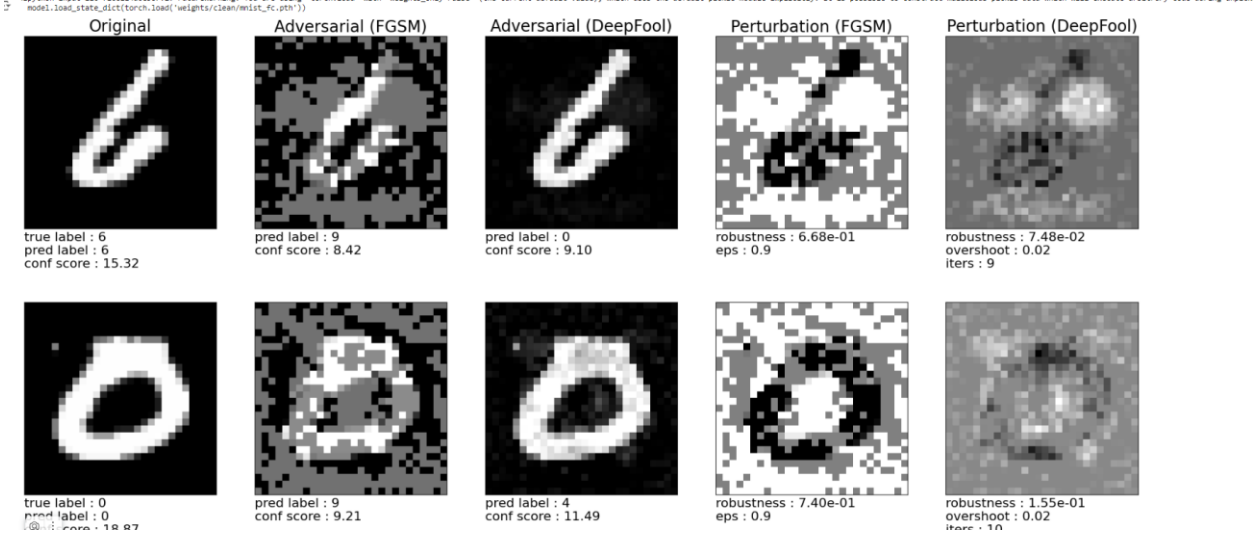
fgsm_eps = 0.02



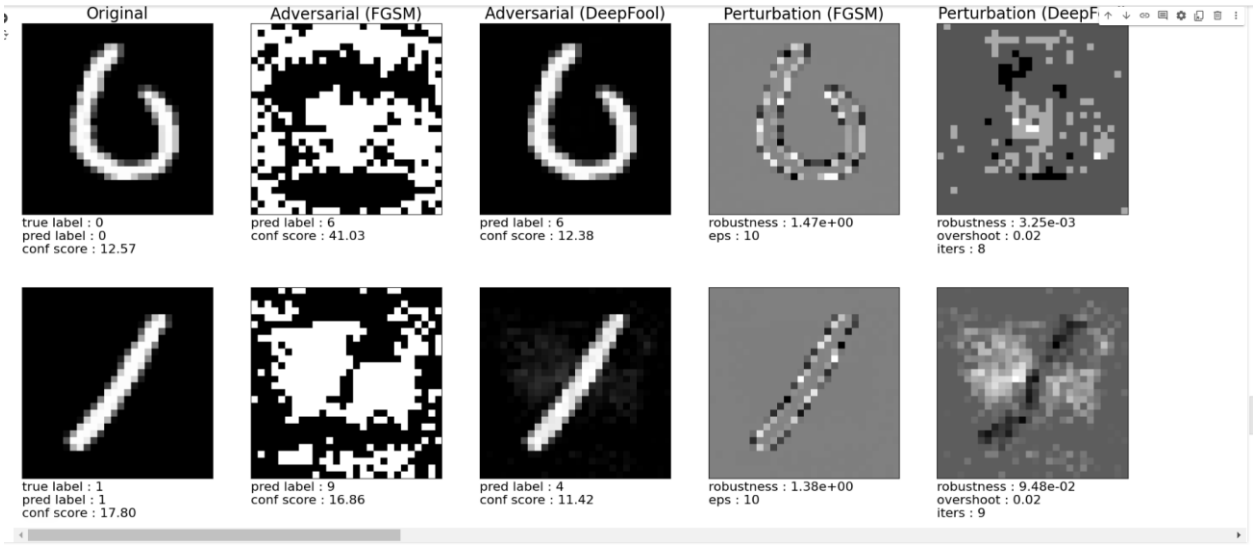
fgsm_eps = 0.5



fgsm_eps = 0.9

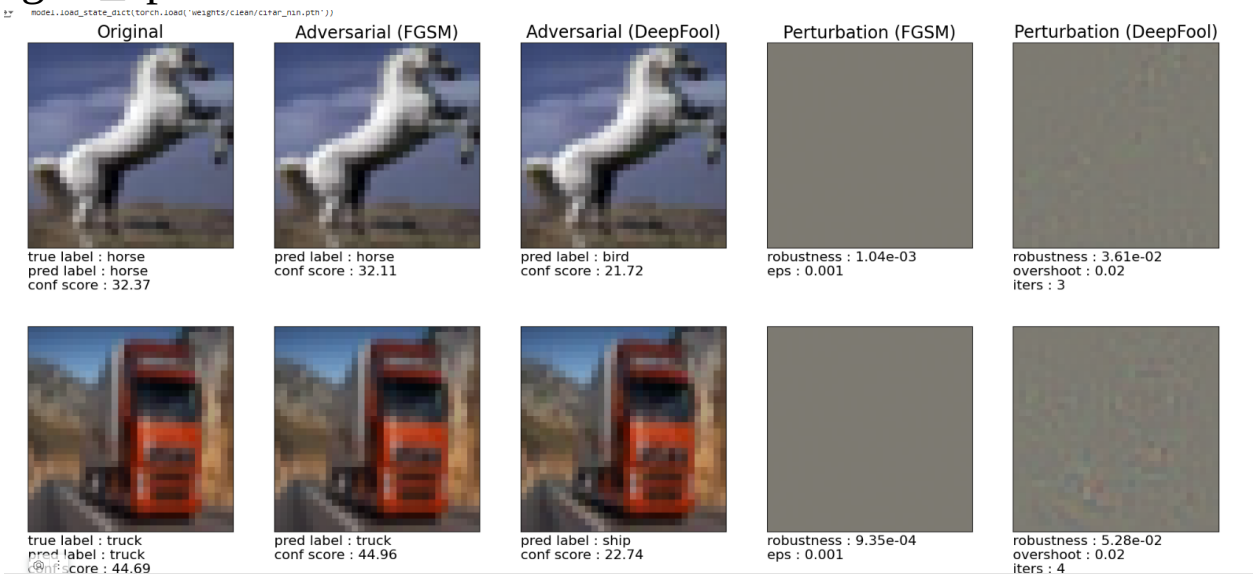


fgsm_eps = 10

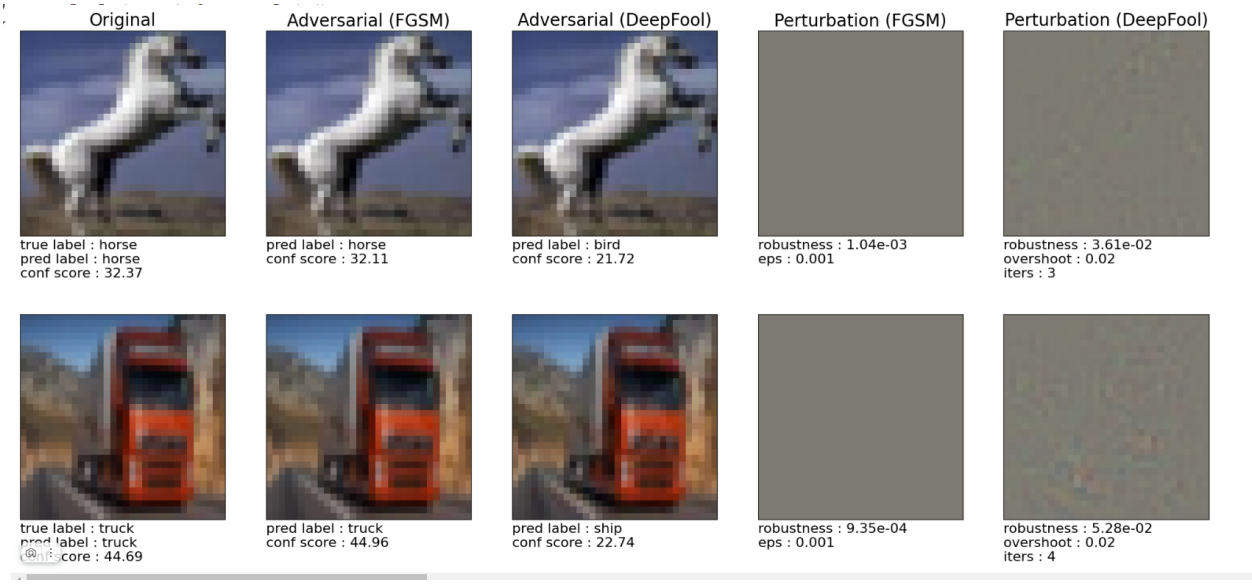


Network-In-Network CIFAR-10

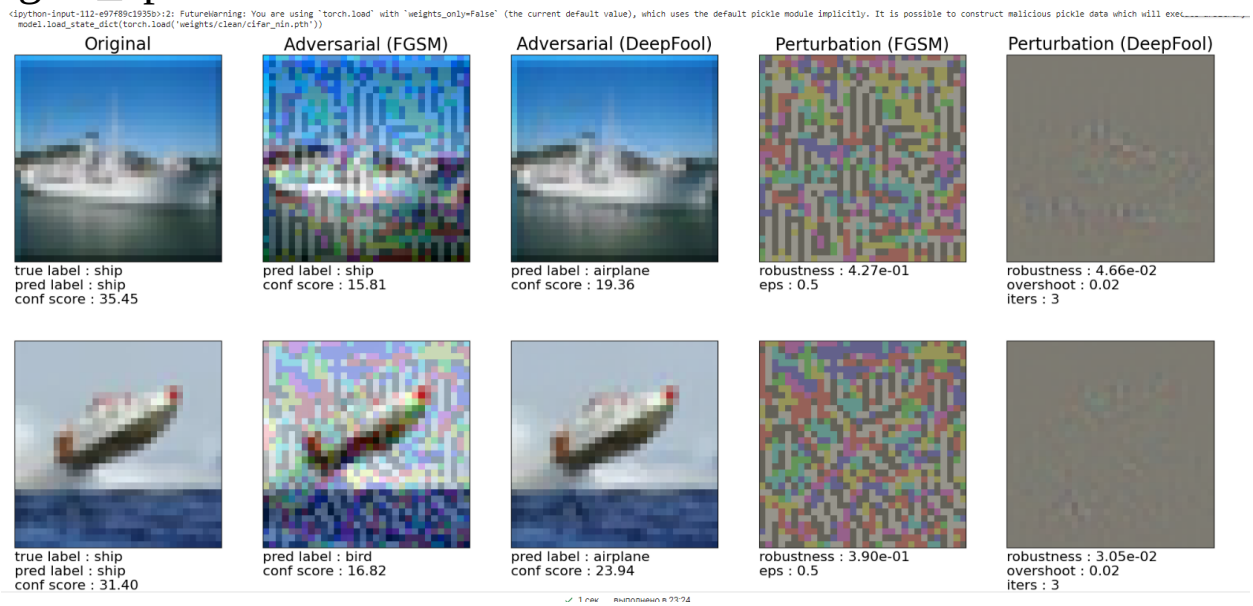
fgsm_eps = 0.001



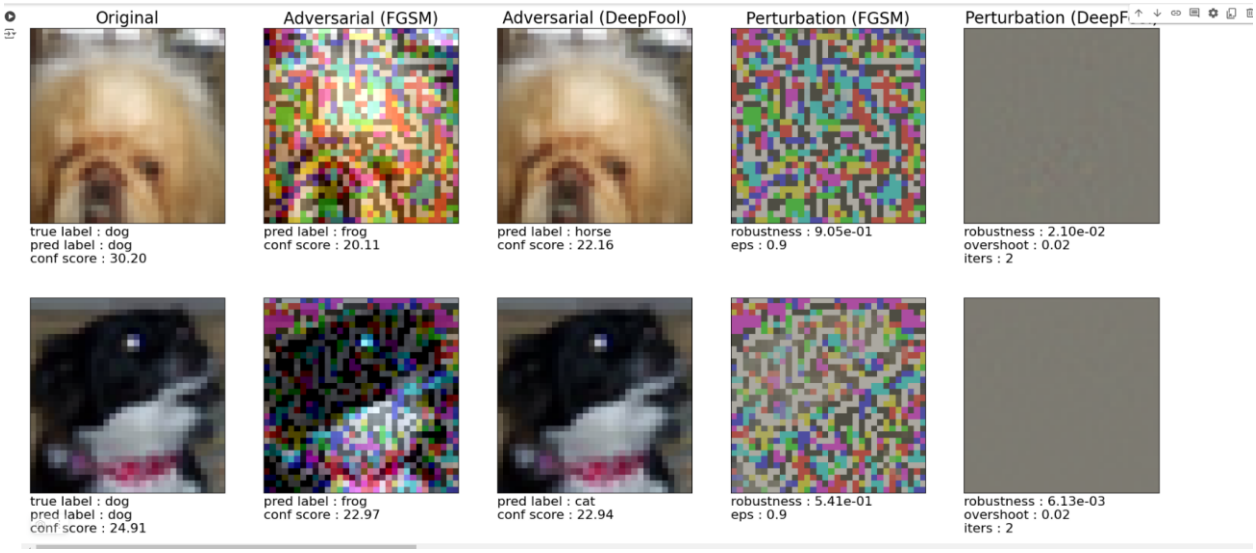
fgsm_eps = 0.02



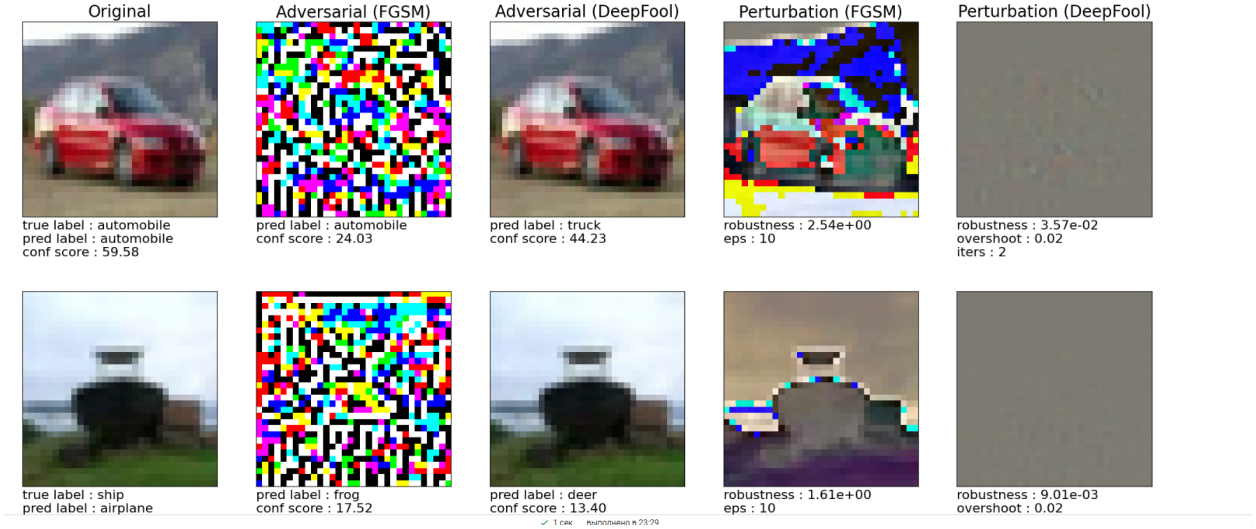
fgsm_eps = 0.5



fgsm_eps = 0.9

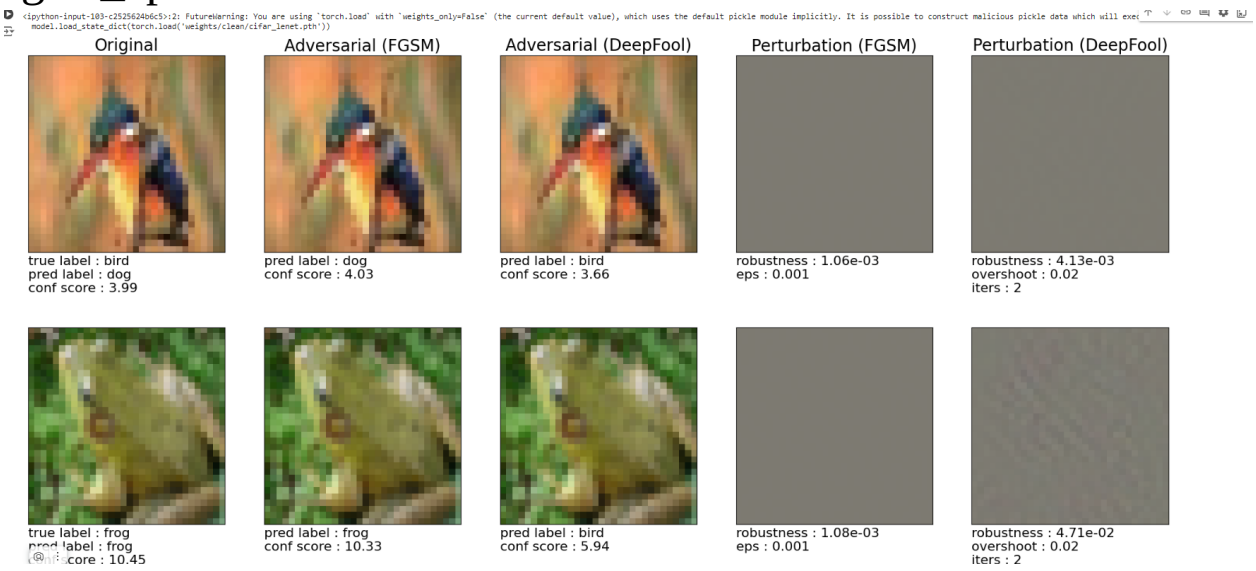


fgsm_eps = 10

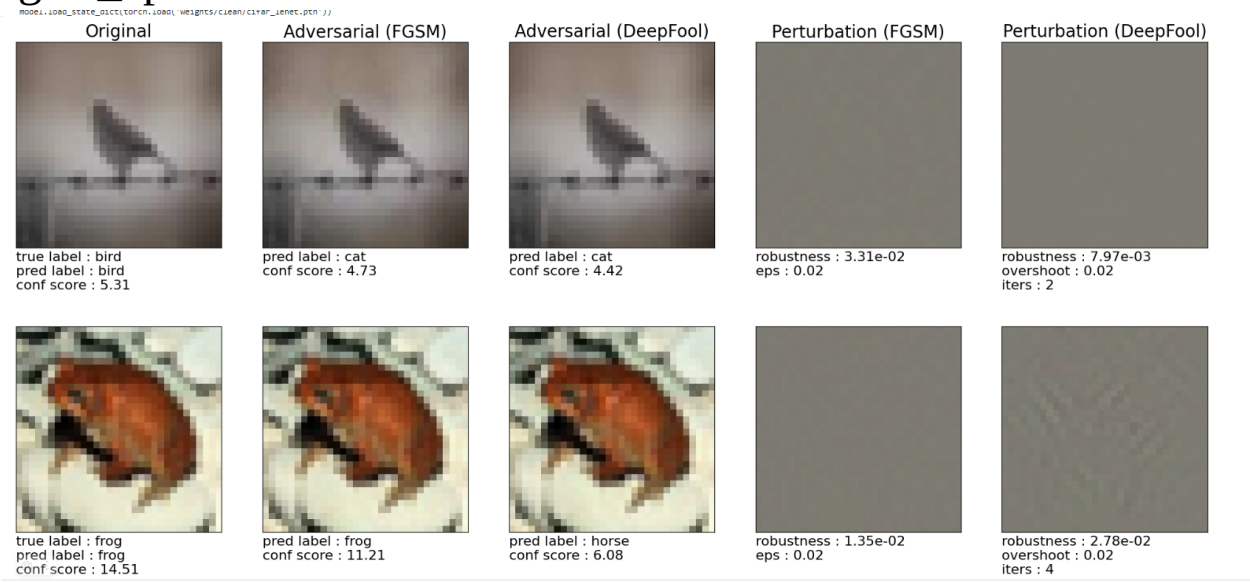


LeNet CIFAR-10

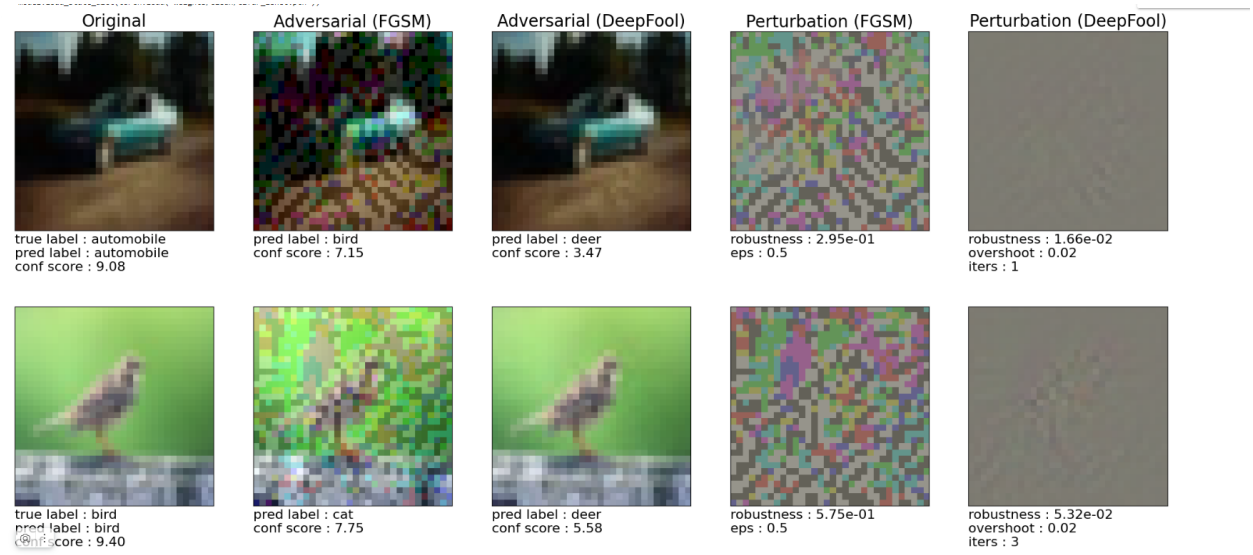
fgsm_eps = 0.001



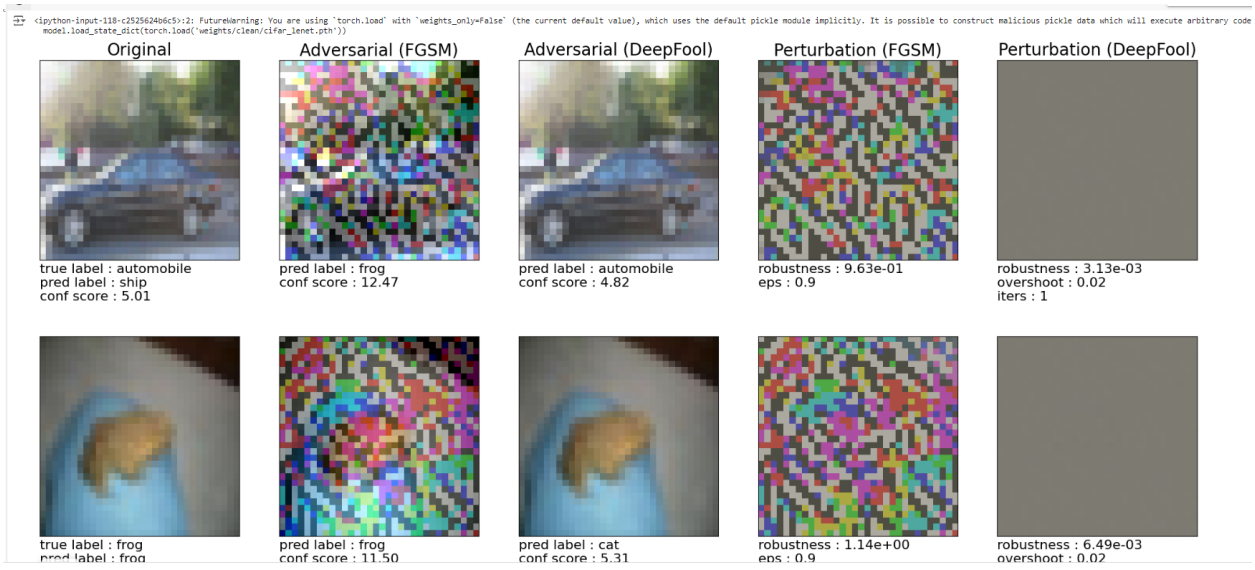
fgsm_eps = 0.02



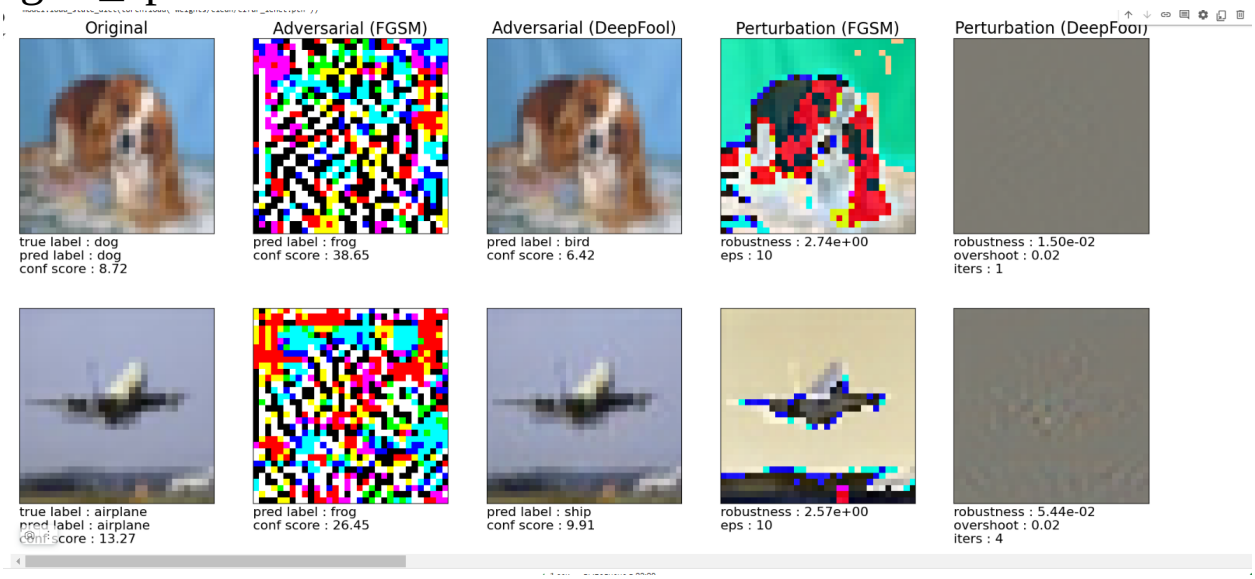
fgsm_eps = 0.5



fgsm_eps = 0.9



fgsm_eps = 10



Заключение

Когда fgsm_eps увеличивается, сети становятся уязвимее к атакам. Значительно уязвимее они становятся со значения fgsm_eps = 0.5