МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«МИРЭА – Российский технологический университет»**
**РТУ МИРЭА**

ИКБ направление «Кибербразведка и противодействие угрозам с применением технологий искусственного интеллекта» 10.04.01

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»

# Отчёт по лабораторной работе №2

по дисциплине: «Анализ защищенности систем искусственного интеллекта»

Группа:
ББМО-02-22
Выполнила:
Бардасова И.А.

Проверил:
Спирин А.А.

Москва, 2023

Содержание

## Введение

**Задачи:**

1. Реализовать атаки уклонения на основе белого ящика против классификационных моделей на основе глубокого обучения.

2. Получить практические навыки переноса атак уклонения на основе черного ящика против моделей машинного обучения.

## Ход выполнения работы

**Шаг 1. Набор данных:** Для этой части используем набор данных GTSRB (German Traffic Sign Recognition Benchmark). Набор данных состоит примерно из 51 000 изображений дорожных знаков (рис. 2). Загрузим набор данных по ссылке: https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign (рис. 1).

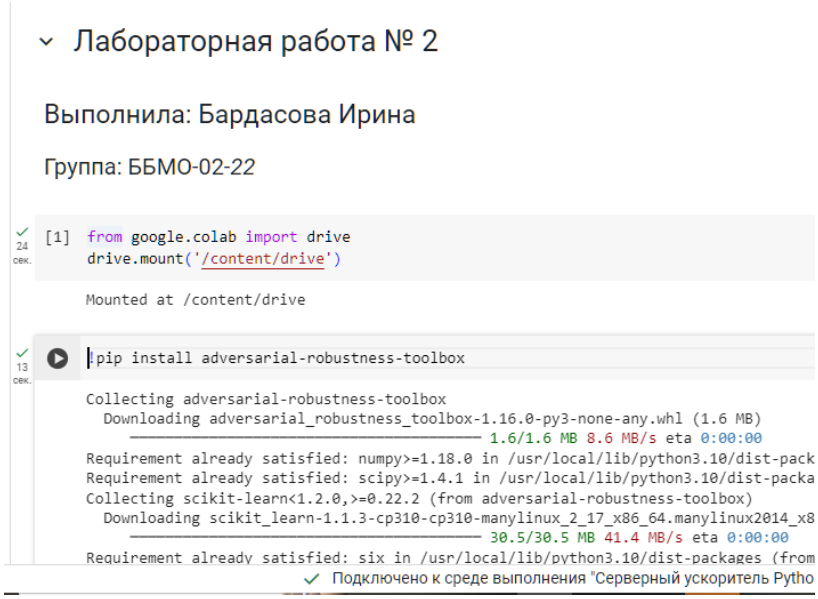



Рисунок 1 – Загрузка данных

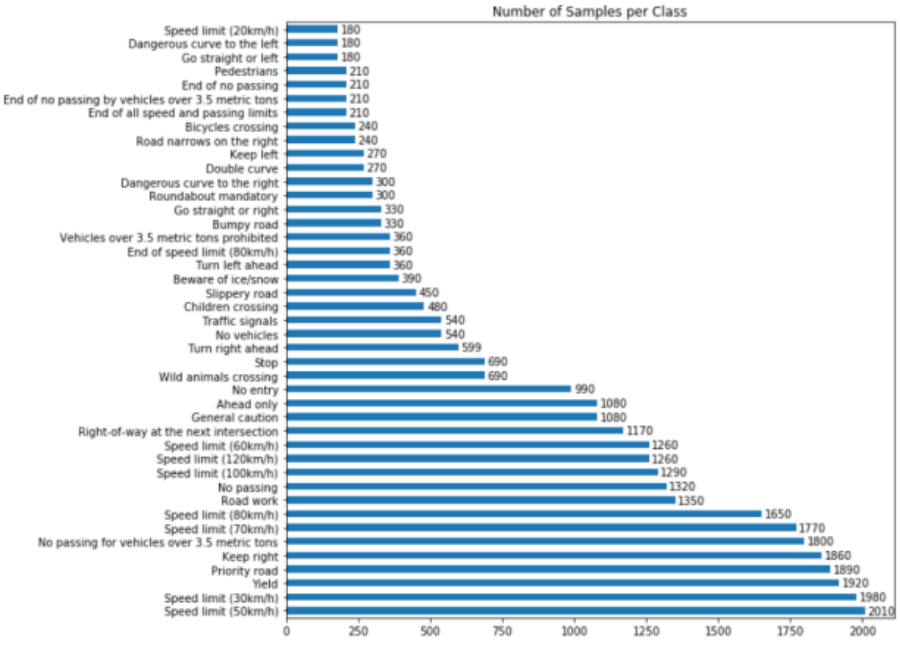



Рисунок 2 – Распределение изображений в GTRSB

## Задание 1

**Шаг 2. Обучить 2 классификатора на основе глубоких нейронных сетей на датасете GTSRB.** Ресурсы колаба не безграничны, поэтому используем только часть набора данных. Использовали следующие модели нейронных сетей: ResNet50 и VGG16. Будем использовать необходимые фреймворки. Поделим набор данных на обучающую и тестовую в соотношении 70/30.

Сначала извлечем изображения для создания тренировочной выборки.
На выходе, мы получим матричное представление изображения (рис. 3).

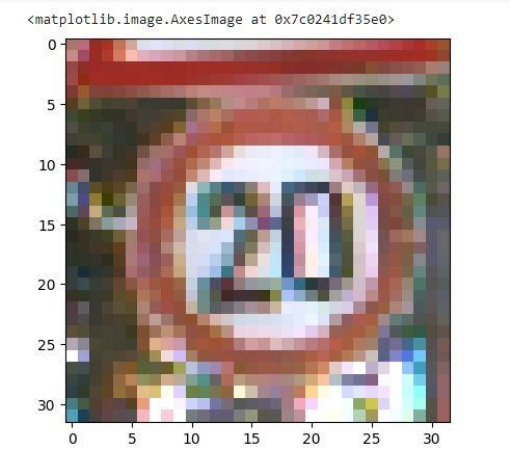



Рисунок 3 – Матричное представление изображения

**Шаг 3. Построение первой модели: ResNet50 (рис. 4).**

```
[10] x_train, x_val, y_train, y_val = train_test_split(data, labels, test_size=0.3, random_state=1)
```

```
[11] img_size = (224,224)
     model = Sequential()
     model.add(ResNet50(include_top = False, pooling = 'avg'))
     model.add(Dropout(0.1))
     model.add(Dense(256, activation="relu"))
     model.add(Dropout(0.1))
     model.add(Dense(43, activation = 'softmax'))
     model.layers[2].trainable = False
```

Рисунок 4 – Модель ResNet50

Определили оптимальное значение эпох обучения (5) и размера пакета (64). Для валидации будут выбраны 30 процентов тренировочного набора, сама валидация показана на рисунке 6. Графики процесса обучения представлены на рисунке 5.

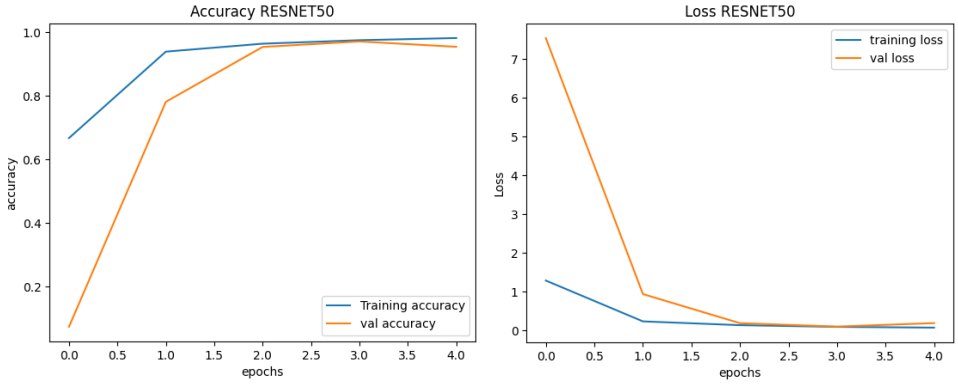



Рисунок 5 – Графики ResNet50

```
Epoch 5/5
429/429 [==============================] - 25s 57ms/step - loss: 0.0758 - accuracy: 0.9815 - val_loss: 0.1925 - val_accuracy: 0.9539
```

Рисунок 6 – Валидация ResNet50

Протестируем нашу модель на тестовом наборе. Результат валидации можно увидеть на рисунке 7. Итоговая точность составила – 90%.

```
11 сек.
print(f"Test loss: {loss}")
print(f"Test accuracy: {accuracy}")

395/395 [==============================] - 6s 13ms/step - loss: 0.5282 - accuracy: 0.8890
Test loss: 0.5281729698181152
Test accuracy: 0.8889944553375244
```

Рисунок 7 – Тестирование ResNet50

**Шаг 4. Построение второй модели: VGG16 (рис. 8).**

```
[17] del model
     del history
     img_size = (224,224)
     model = Sequential()
     model.add(VGG16(include_top=False, pooling = 'avg'))
     model.add(Dropout(0.1))
     model.add(Dense(256, activation="relu"))
     model.add(Dropout(0.1))
     model.add(Dense(43, activation = 'softmax'))
     model.layers[2].trainable = False

     Downloading data from https://storage.googleapis.com/tensorflow/kera
```

Рисунок 8 – Модель VGG16

Графики процесса обучения модели VGG16 показаны на рисунке 9. Валидационный результат представлен на рисунке 10.

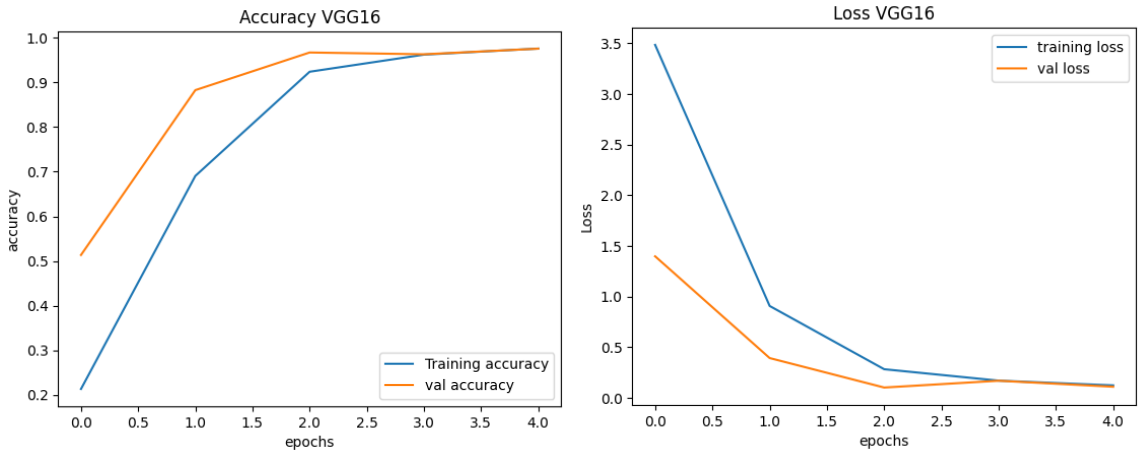



Рисунок 9 – Графики VGG16

```
Epoch 5/5
429/429 [==============================] - 17s 40ms/step - loss: 0.1215 - accuracy: 0.9760 - val_loss: 0.1080 - val_accuracy: 0.9758
```

Рисунок 10 – Валидация VGG16

Тестирование обученной модели на валидационном наборе (рис. 11):

```
395/395 [==============================] - 5s 10ms/step - loss: 0.2992 - accuracy: 0.9382
Test loss: 0.29915425181388855
Test accuracy: 0.9381631016731262
```

Рисунок 11 – Тестирование VGG16

**Шаг 5. Результаты:** Подведём результаты по первому заданию в таблице 1.1.

Таблица 1.1 – Результаты

| **Модель** | **Обучение** | | **Валидация** | | **Тест** | |
|---|---|---|---|---|---|---|
| | **loss** | **accuracy** | **loss** | **accuracy** | **loss** | **accuracy** |
| **ResNet50** | 0,0758 | 0,9815 | 0,1925 | 0,9539 | 0,5282 | 0,8890 |
| **VGG16** | 0,1215 | 0,9760 | 0,108 | 0,9758 | 0,2992 | 0,9382 |

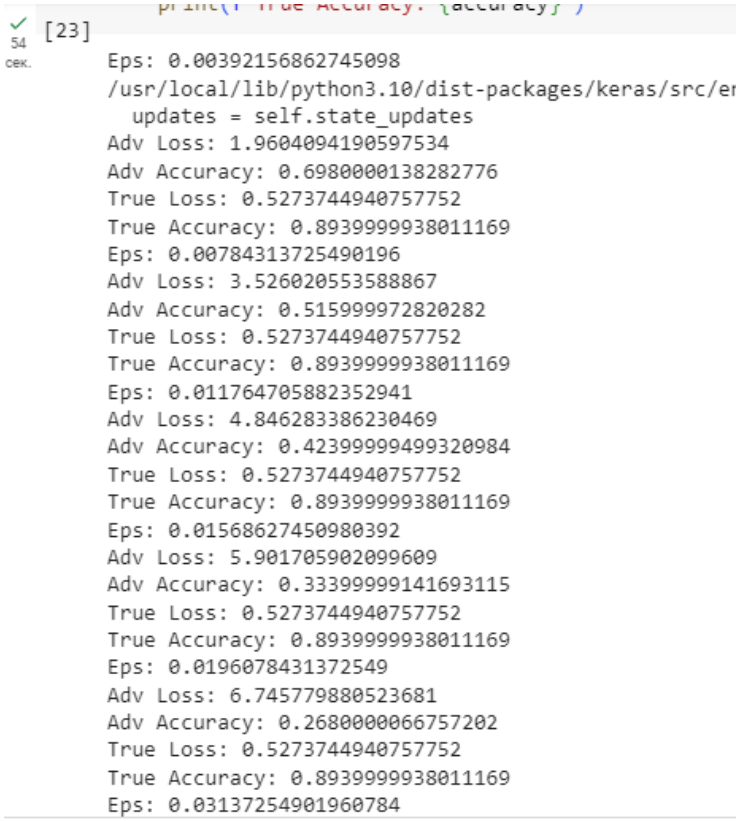
## Задание 2

**Шаг 6. Применим нецелевую атаку уклонения на основе белого ящика против моделей глубокого обучения.** Реализуем атаку Fast Gradient Sign Method (FGSM) и Projected Gradient Descent (PGD).

Для создания нецелевых атакующих примеров используем первые 1,000 изображений из тестового множества. Также используем следующие значения параметра искажения для атак на изображения: $\epsilon$ = [1/255, 2/255, 3/255, 4/255, 5/255, 8/255, 10/255, 20/255, 50/255, 80/255].

**Шаг 7. Атака FGSM на ResNet50.** Создаем модель атаки, которая будет основываться на обученном классификаторе для внесения шума в изображение.

Результаты каждого параметра на рисунке 12.



```
[23] Eps: 0.0196078431372549
     Adv Loss: 6.745779880523681
     Adv Accuracy: 0.2680000066757202
     True Loss: 0.5273744940757752
     True Accuracy: 0.8939999938011169
     Eps: 0.03137254901960784
     Adv Loss: 8.335723251342774
     Adv Accuracy: 0.1679999977350235
     True Loss: 0.5273744940757752
     True Accuracy: 0.8939999938011169
     Eps: 0.0392156862745098
     Adv Loss: 8.916819450378417
     Adv Accuracy: 0.11999999731779099
     True Loss: 0.5273744940757752
     True Accuracy: 0.8939999938011169
     Eps: 0.0784313725490196
     Adv Loss: 9.802408462524413
     Adv Accuracy: 0.03200000151991844
     True Loss: 0.5273744940757752
     True Accuracy: 0.8939999938011169
     Eps: 0.19607843137254902
     Adv Loss: 9.217128112792969
     Adv Accuracy: 0.007000000216066837
     True Loss: 0.5273744940757752
     True Accuracy: 0.8939999938011169
     Eps: 0.3137254901960784
     Adv Loss: 8.476739044189452
     Adv Accuracy: 0.004000000189989805
     True Loss: 0.5273744940757752
```

Рисунок 12 – Параметры искажения для атак на изображения

График зависимости точности предсказания модели на атакованных изображениях от параметра искажения представлена на рисунке 13.

Отобразим исходное изображение из датасета и атакующие изображения с указанием величины параметра $\epsilon$ = [1/255, 5/255, 10/255, 50/255, 80/255] (рис. 14).

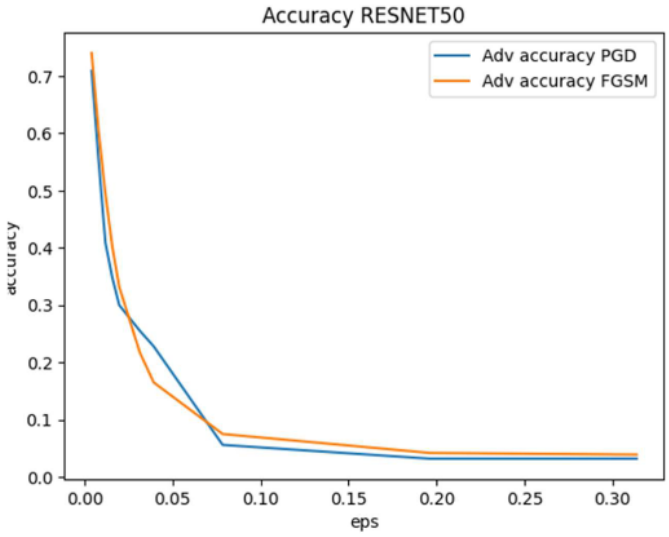
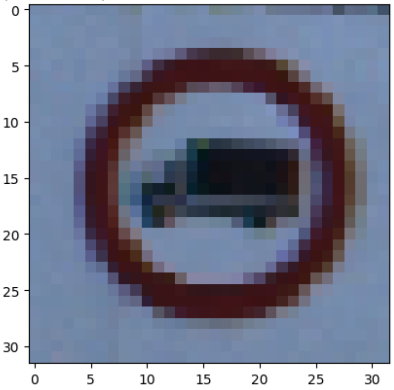
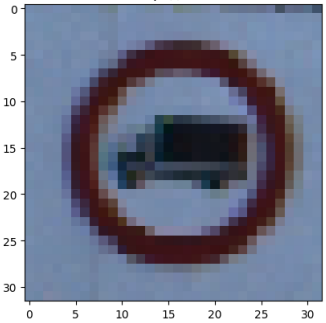



Рисунок 13 – График зависимости

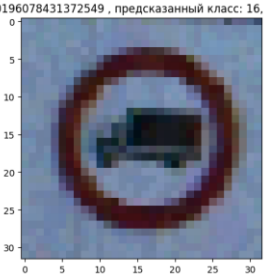
Исходное изображение, предсказанный класс: 16, действительный класс 16



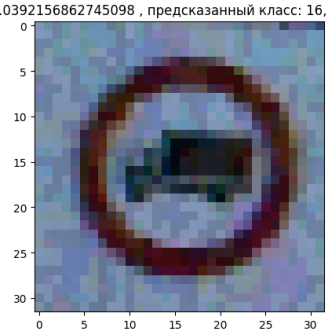
Изображение с eps: 0.00392156862745098 , предсказанный класс: 16, действительный класс 16



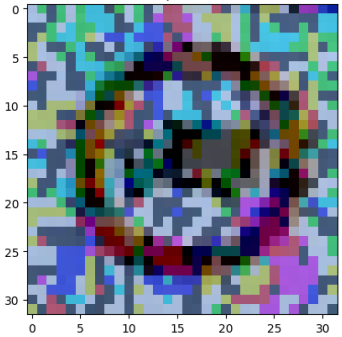
Изображение с eps: 0.0196078431372549 , предсказанный класс: 16, действительный класс 16



Изображение с eps: 0.0392156862745098 , предсказанный класс: 16, действительный класс 16



Изображение с eps: 0.19607843137254902 , предсказанный класс: 5, действительный класс 16



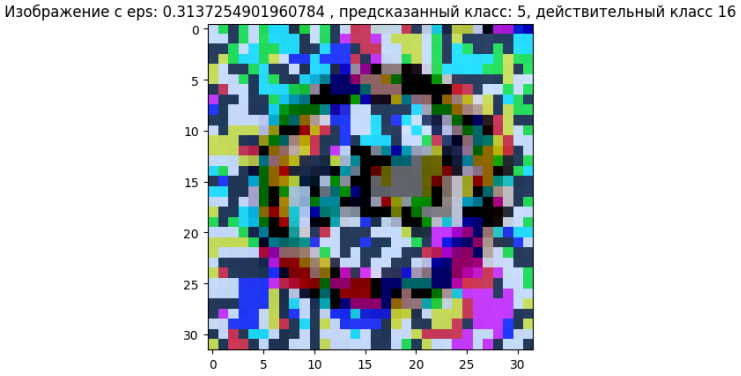



Рисунок 14 – Исходные изображения и искажённые изображения FGSM

**Шаг 8.** Теперь сделаем то же самое с ResNet50 через PGD. Подобно FGSM реализуем атаку PGD для различных значений eps (рис. 16). Результаты каждого параметра на рисунке 15.

```
[27] print(f"True Accuracy: {accuracy}")

Eps: 0.00392156862745098
Adv Loss: 2.2387321014404296
Adv Accuracy: 0.6700000166893005
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.00784313725490196
Adv Loss: 4.190534460067749
Adv Accuracy: 0.4690000116825104
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.011764705882352941
Adv Loss: 6.045370964050293
Adv Accuracy: 0.3580000102519989
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.01568627450980392
Adv Loss: 7.296794589996338
Adv Accuracy: 0.29499998688697815
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.0196078431372549
Adv Loss: 8.150178527832031
Adv Accuracy: 0.24300000071525574
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.03137254901960784
Adv Loss: 9.762843841552735
```

```
Adv Loss: 8.150178527832031
Adv Accuracy: 0.24300000071525574
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.03137254901960784
Adv Loss: 9.762843841552735
Adv Accuracy: 0.19200000166893005
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.0392156862745098
Adv Loss: 10.258387588500977
Adv Accuracy: 0.17399999499320984
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.0784313725490196
Adv Loss: 23.217008361816408
Adv Accuracy: 0.013000000268220901
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.19607843137254902
Adv Loss: 38.92885894775391
Adv Accuracy: 0.0010000000474974513
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
Eps: 0.3137254901960784
Adv Loss: 43.524760925292966
Adv Accuracy: 0.0010000000474974513
True Loss: 0.5273744940757752
True Accuracy: 0.8939999938011169
```

Рисунок 15 – Параметры искажения для атак на изображения

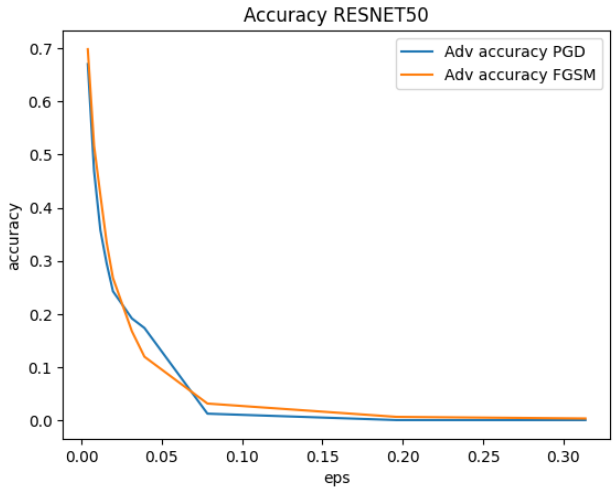


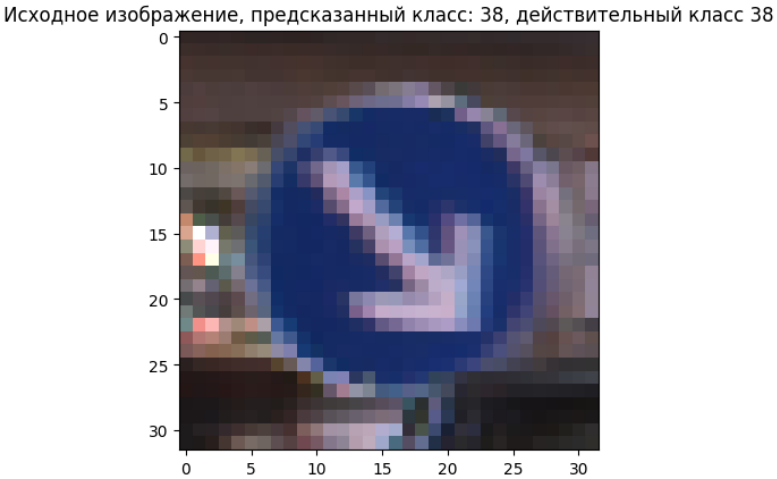

Рисунок 16 – График зависимости PGD

**Шаг 9. Сделаем то же самое для VGG16 – начнём с атаки FGSM (рис. 18).** Результаты каждого параметра на рисунке 17.

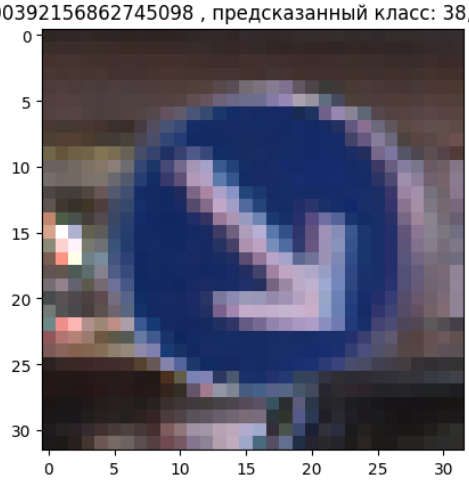
```
Eps: 0.00392156862745098
Adv Loss: 0.9649940905570984
Adv Accuracy: 0.8389999866485596
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.00784313725490196
Adv Loss: 1.6718001012802124
Adv Accuracy: 0.7289999723434448
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.011764705882352941
Adv Loss: 2.314442026138306
Adv Accuracy: 0.6349999904632568
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.01568627450980392
Adv Loss: 2.9762243146896363
Adv Accuracy: 0.5370000004768372
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0196078431372549
Adv Loss: 3.571226887702942
Adv Accuracy: 0.4569999873638153
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.03137254901960784
Adv Loss: 4.840155690193177

Adv Accuracy: 0.28700000047683716
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0392156862745098
Adv Loss: 5.448998310089111
Adv Accuracy: 0.20200000703334808
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0784313725490196
Adv Loss: 6.630562896728516
Adv Accuracy: 0.0560000017285347
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.19607843137254902
Adv Loss: 6.108267921447754
Adv Accuracy: 0.03099999949336052
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.3137254901960784
Adv Loss: 5.541807586669922
Adv Accuracy: 0.029999999329447746
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
```

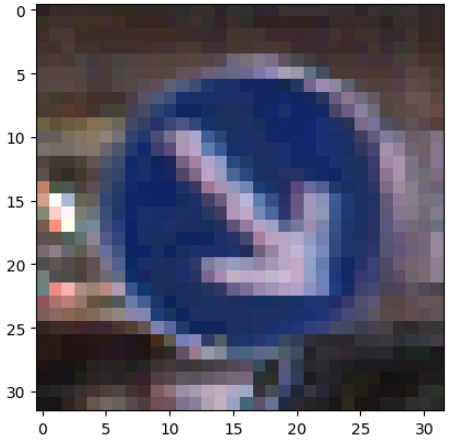
Рисунок 17 – Параметры искажения для атак на изображения



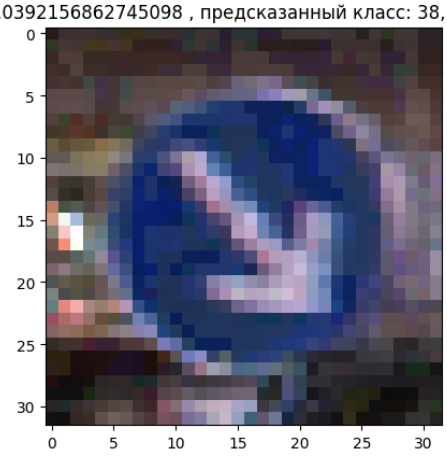
Изображение с eps: 0.00392156862745098 , предсказанный класс: 38, действительный класс 38



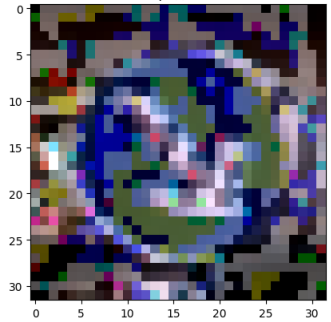
Изображение с eps: 0.0196078431372549 , предсказанный класс: 38, действительный класс 38



Изображение с eps: 0.0392156862745098 , предсказанный класс: 38, действительный класс 38



Изображение с eps: 0.19607843137254902 , предсказанный класс: 38, действительный класс 38



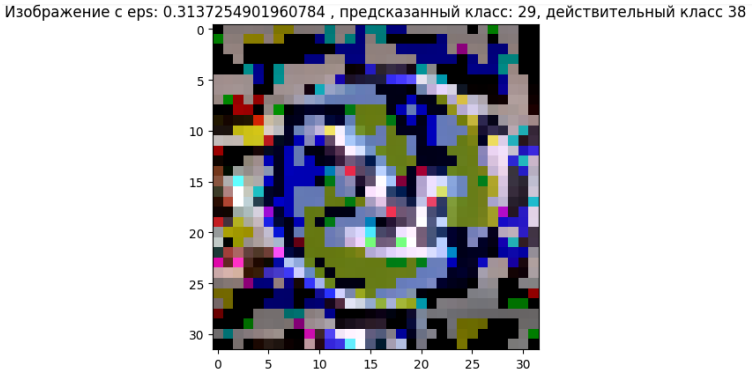



Рисунок 18 – Исходные изображения и искажённые изображения FGSM

**Шаг 10. VGG16 PGD (рис. 20).** Подобно FGSM реализуем атаку PGD для различных значений eps. Результаты каждого параметра на рисунке 19.

```
Eps: 0.00392156862745098
Adv Loss: 1.153845339655876
Adv Accuracy: 0.824999988079071
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.00784313725490196
Adv Loss: 2.1025033988952635
Adv Accuracy: 0.722000002861023
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.011764705882352941
Adv Loss: 2.9856135816574096
Adv Accuracy: 0.6359999775886536
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.01568627450980392
Adv Loss: 4.066827007293702
Adv Accuracy: 0.5460000038146973
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0196078431372549
Adv Loss: 4.867372756958008
Adv Accuracy: 0.48100000619888306
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.03137254901960784
Adv Loss: 6.533005279541015
Adv Accuracy: 0.3700000047683716
True Loss: 0.3210929125417024
```

```
Adv Accuracy: 0.48100000619888306
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.03137254901960784
Adv Loss: 6.533005279541015
Adv Accuracy: 0.3700000047683716
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0392156862745098
Adv Loss: 7.113522884368897
Adv Accuracy: 0.3149999976158142
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.0784313725490196
Adv Loss: 18.210905303955077
Adv Accuracy: 0.07199999690055847
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.19607843137254902
Adv Loss: 46.494306640625
Adv Accuracy: 0.0030000000026077032
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
Eps: 0.3137254901960784
Adv Loss: 57.72623907470703
Adv Accuracy: 0.0020000000949949026
True Loss: 0.3210929125417024
True Accuracy: 0.9350000023841858
```

Рисунок 19 – Параметры искажения для атак на изображения

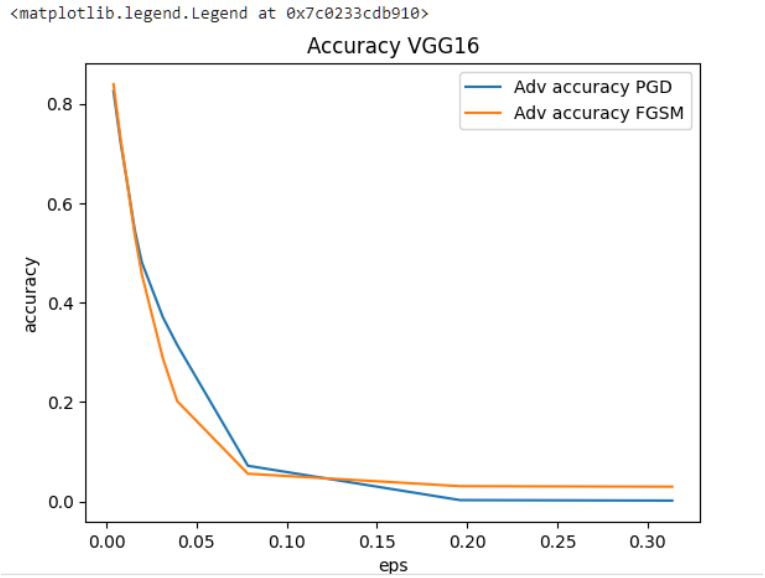



Рисунок 20 – График зависимостей PGD

**Шаг 11. Результаты (таблица 2.1).**

Таблица 2.1 - Результаты

| Модель | Исходные изображения | Adversarial images $\epsilon\epsilon$=1/255 | Adversarial images $\epsilon\epsilon$=5/255 | Adversarial images $\epsilon\epsilon$=10/255 |
|---|---|---|---|---|
| ResNet50 - FGSM | loss: 0.5275<br>accuracy: 0.894 | loss: 1.9604<br>accuracy: 0.698 | loss: 6.7457<br>accuracy: 0.268 | loss: 8.4767<br>accuracy: 0.004 |
| ResNet50 - PGD | loss: 0.5275<br>accuracy: 0.894 | loss: 2.2387<br>accuracy: 0.67 | loss: 8.1501<br>accuracy: 0.243 | loss: 43.5247<br>accuracy: 0.001 |
| VGG16 - FGSM | loss: 0.3210<br>accuracy: 0.935 | loss: 0.9649<br>accuracy: 0.839 | loss: 3.5712<br>accuracy: 0.457 | loss: 5.5418<br>accuracy: 0.0299 |
| VGG16 - PGD | loss: 0.3210<br>accuracy: 0.935 | loss: 1.1538<br>accuracy: 0.825 | loss: 4.8673<br>accuracy: 0.4810 | loss: 57.7262<br>accuracy: 0.3210 |

## Задание 3

Применение целевой атаки уклонения методом белого против моделей глубокого обучения.

**Шаг 12.** Используем изображения знака «Стоп» (label class 14) из тестового набора данных. Применим атаку Projected Gradient Descent (PGD) на знак «Стоп» с целью классификации его как знака «Ограничение скорости 30» (target label class = 1). Будем изменять значения искажений $\epsilon\epsilon$= [1/255, 3/255, 5/255, 10/255, 20/255, 50/255, 80/255].

Повторим атаку методом FGSM и заполним таблицу 3.1.

Таблица 3.1 – Результаты

| $\epsilon$ | **FGSM - Stop** | **PGD – Stop** |
|---|---|---|
| $\epsilon$ = 1/255 | loss: 0.0290<br>accuracy: 0.9888 | loss: 0.0308<br>accuracy: 0.9888 |
| $\epsilon$ = 3/255 | loss: 0.9453<br>accuracy: 0.8185 | loss: 0.5882<br>accuracy: 0.8555 |
| $\epsilon$ = 5/255 | loss: 2.6736<br>accuracy: 0.5407 | loss: 1.399<br>accuracy: 0.6296 |
| $\epsilon$ = 10/255 | loss: 7.3347<br>accuracy: 0.088 | loss: 2.5267<br>accuracy: 0.47037 |
| $\epsilon$ = 20/255 | loss: 10.321<br>accuracy: 0.0 | loss: 8.7211<br>accuracy: 0.1296 |
| $\epsilon$ = 50/255 | loss: 10.1705<br>accuracy: 0.0 | loss: 21.8093<br>accuracy: 0.0 |
| $\epsilon$ = 80/255 | loss: 8.8265<br>accuracy: 0.0 | loss: 23.3313<br>accuracy: 0.0 |

Подробнее ниже. Результаты каждого параметра **FGSM** представлены на рисунке 21.

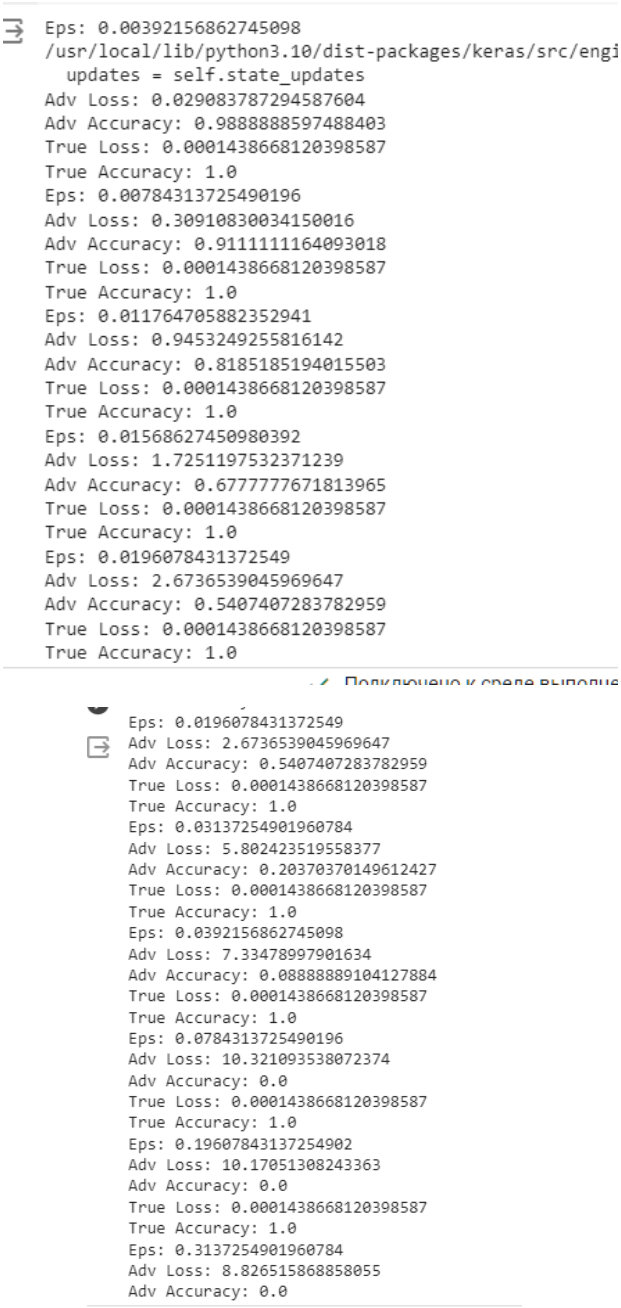
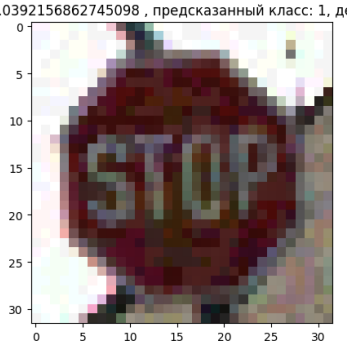



Рисунок 21 – Параметры FGSM

Выведем 5 примеров классификации класса 14 как класс 1 при помощи целевой FGSM атаки (рис. 22):

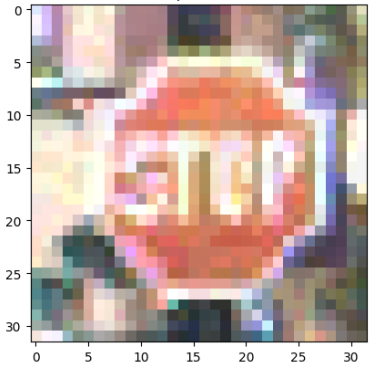
Исходное изображение, предсказанный класс: 14, действительный класс 14


Изображение с eps: 0.0392156862745098 , предсказанный класс: 1, действительный класс 14

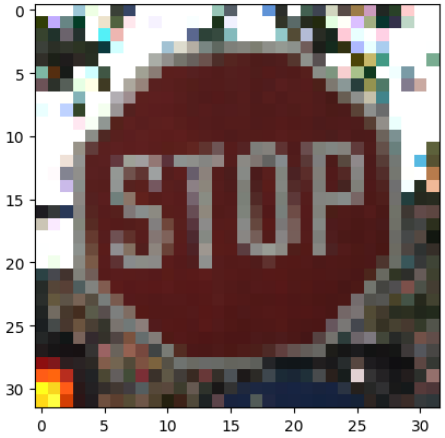



Исходное изображение, предсказанный класс: 14, действительный класс 14

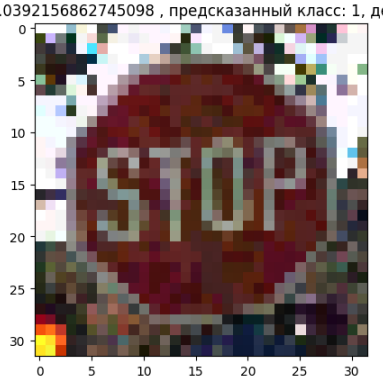
Изображение с eps: 0.0392156862745098 , предсказанный класс: 1, действительный класс 14



Исходное изображение, предсказанный класс: 14, действительный класс 14

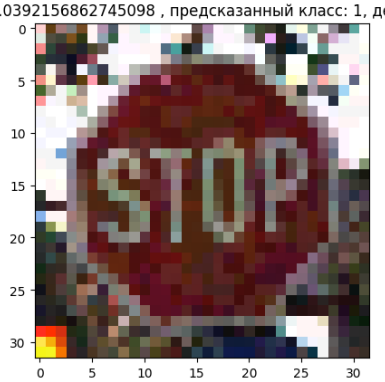



Изображение с eps: 0.0392156862745098 , предсказанный класс: 1, действительный класс 14




Исходное изображение, предсказанный класс: 14, действительный класс 14


Изображение с eps: 0.0392156862745098 , предсказанный класс: 1, действительный класс 14



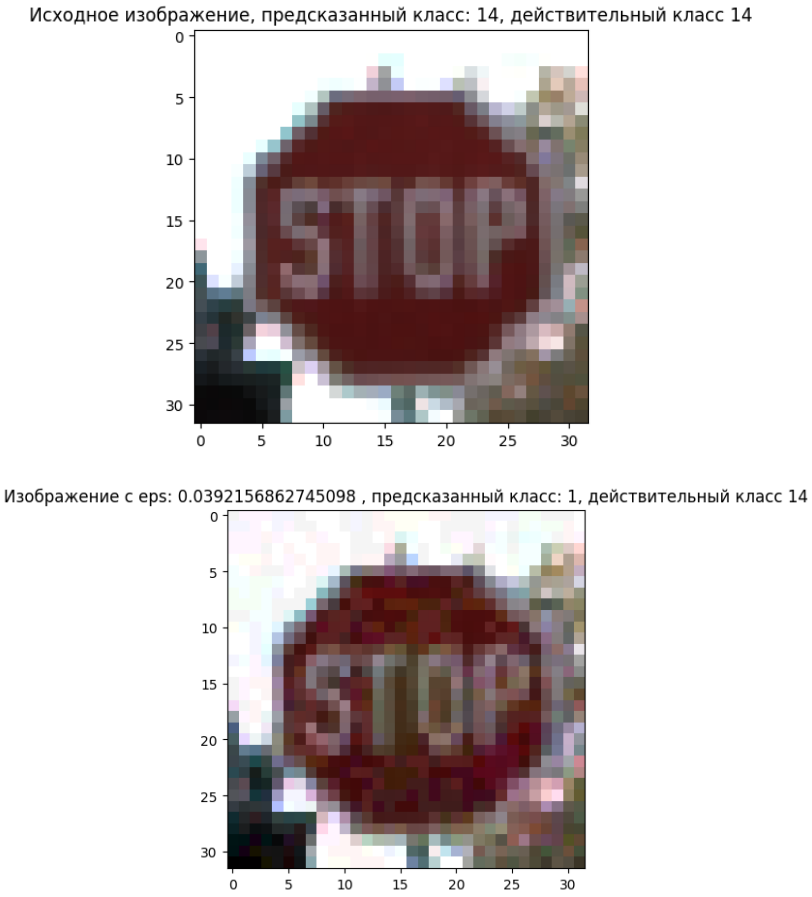



Рисунок 22 – FGSM искажение

Результаты каждого параметра **PGD** представлены на рисунке 23.

```
Eps: 0.00392156862745098
Adv Loss: 0.030893028648225247
Adv Accuracy: 0.9888888597488403
True Loss: 0.0001438668120398587
True Accuracy: 1.0
Eps: 0.00784313725490196
Adv Loss: 0.2406274570359124
Adv Accuracy: 0.9222221970558167
True Loss: 0.0001438668120398587
True Accuracy: 1.0
Eps: 0.011764705882352941
Adv Loss: 0.5882822204519201
Adv Accuracy: 0.855555534362793
True Loss: 0.0001438668120398587
True Accuracy: 1.0
Eps: 0.01568627450980392
Adv Loss: 1.0544115953975253
Adv Accuracy: 0.7666666507720947
True Loss: 0.0001438668120398587
True Accuracy: 1.0
Eps: 0.0196078431372549
Adv Loss: 1.3990310554151182
Adv Accuracy: 0.6296296119689941
True Loss: 0.0001438668120398587
```

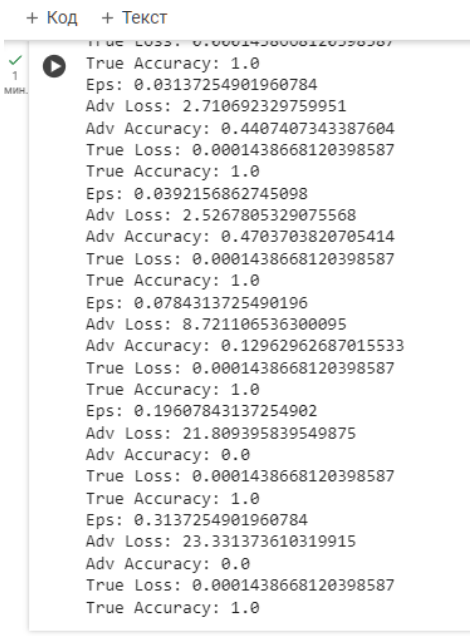
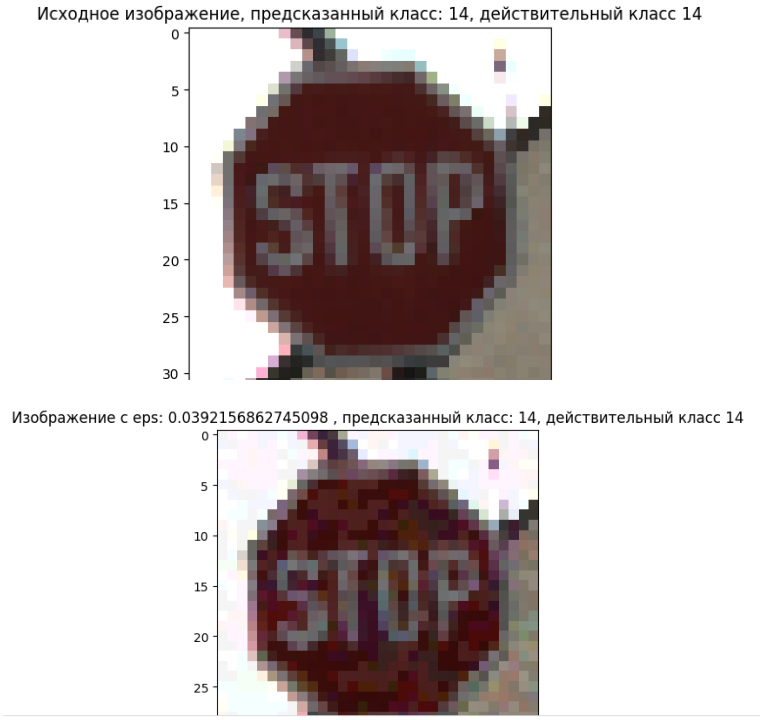



Рисунок 23 – Результаты для PGD

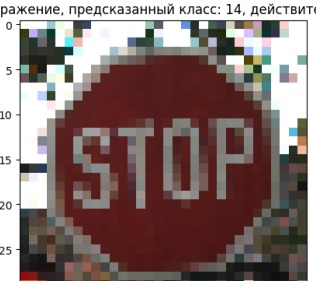
Выведем 5 примеров классификации класса 14 как класс 1 при помощи целевой PGD атаки (рис. 24):



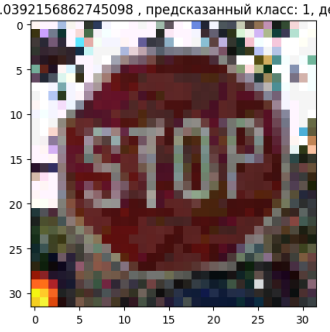
Исходное изображение, предсказанный класс: 14, действительный класс 14


Изображение с eps: 0.0392156862745098 , предсказанный класс: 14, действительный класс 14


Исходное изображение, предсказанный класс: 14, действительный класс 14




Изображение с eps: 0.0392156862745098 , предсказанный класс: 1, действительный класс 14




Исходное изображение, предсказанный класс: 14, действительный класс 14

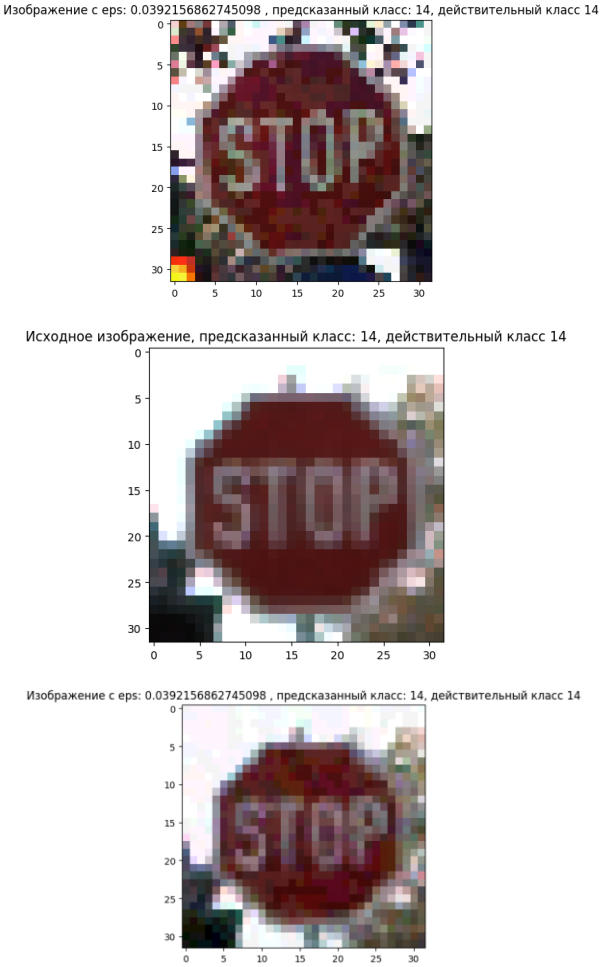



Рисунок 24 – PGD искажение

## Заключение

В результате выполнения работы были также проведены эксперименты по атаке на модели машинного обучения методом черного и белого ящика, а также целевые и нецелевые.

Также были рассмотрены модели VGG16 и ResNet50, VGG16 показала несколько большую устойчивость к атакам, хоть и не значительную, но уже не в рамках погрешности. Было отмечено сильно ухудшение качества и точности моделей по достижении отметки искажения в 20/255. Метод FGSM плохо подходит для целевых атак. С ростом искажения классификация начинает давать сбои. PGD больше подходит для целевых атак. При больших искажениях, модель будет определять нужный класс, но картинка сильно испортится шумом.