МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

Институт кибербезопасности и цифровых технологий

ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 2

по дисциплине

«Анализ защищенности систем искусственного интеллекта»

Выполнил:

ББМО–01–22

Чадов В. Т.

Проверил:

Спирин А. А.

«Зачтено» «__»_____2023 г. _____

Москва 2023

Задание

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.

Набор данных: Для этой части используйте набор данных GTSRB (German Traffic Sign Recognition Benchmark). Набор данных состоит примерно из 51 000 изображений дорожных знаков. Существует 43 класса дорожных знаков, а размер изображений составляет 32×32 пикселя. Распределение изображений по классам показано на рис. 1. Набор данных: https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

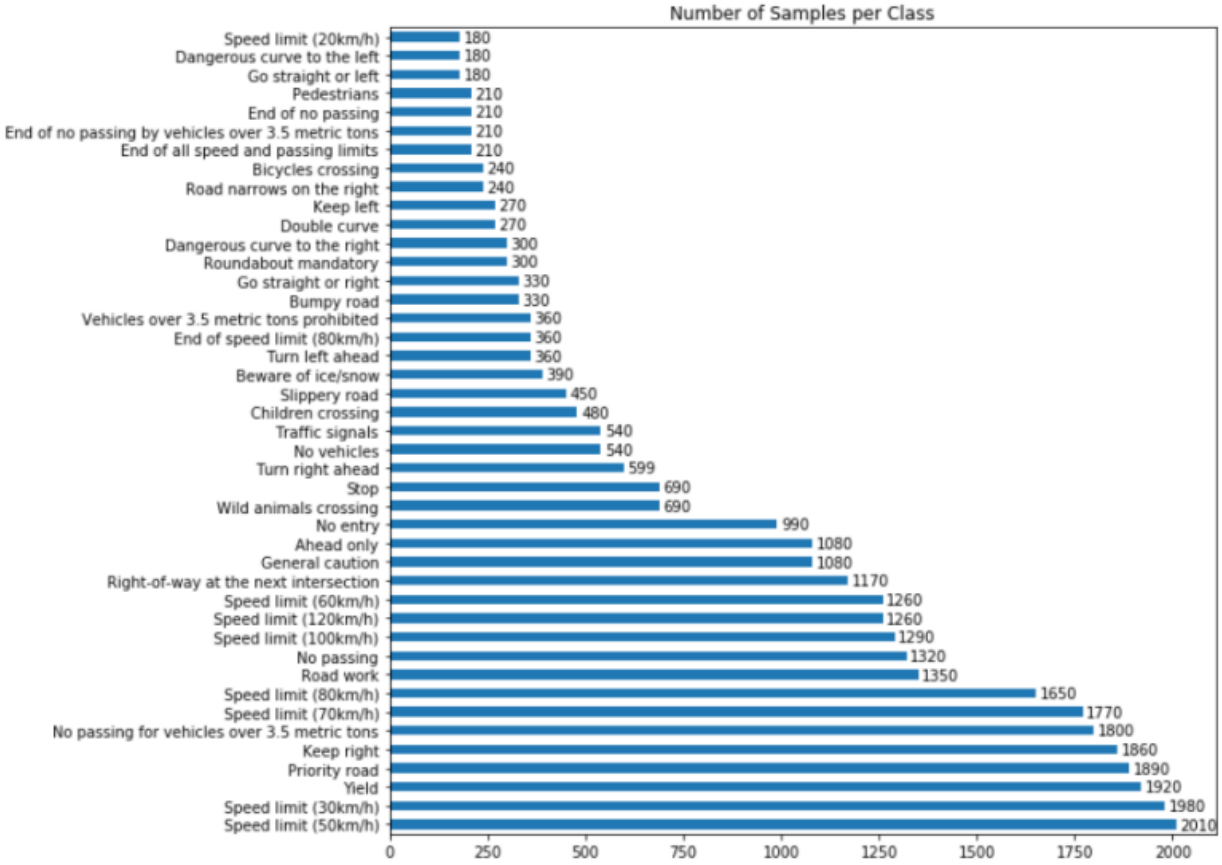


Рис. 1. Распределение изображений в GTRSB

Задание 1.

Обучить 2 классификатора на основе глубоких нейронных сетей на датасете GTSRB. Использовать следующие модели нейронных сетей: VGG16, ResNet50/10X, MobileNet v2/3. Можно использовать фреймворки Keras, TensorFlow, PyTorch, не надо создавать сети вручную и с нуля. Использовать предобученные сети (например на ImageNet). Выполнить поиск наилучших гиперпараметров моделей. Использовать бесплатные ресурсы GPU сервиса Google Colab.

Составить отчёт: (a) Заполнить Таблицу 1. (b) Для каждой модели построить графики функции потерь для данных валидации и тестирования и графики метрики Accuracy(пример на рис. 2).

Таблица 1.

Модель	Обучение	Валидация	Тест
VGG16			
ResNet50			

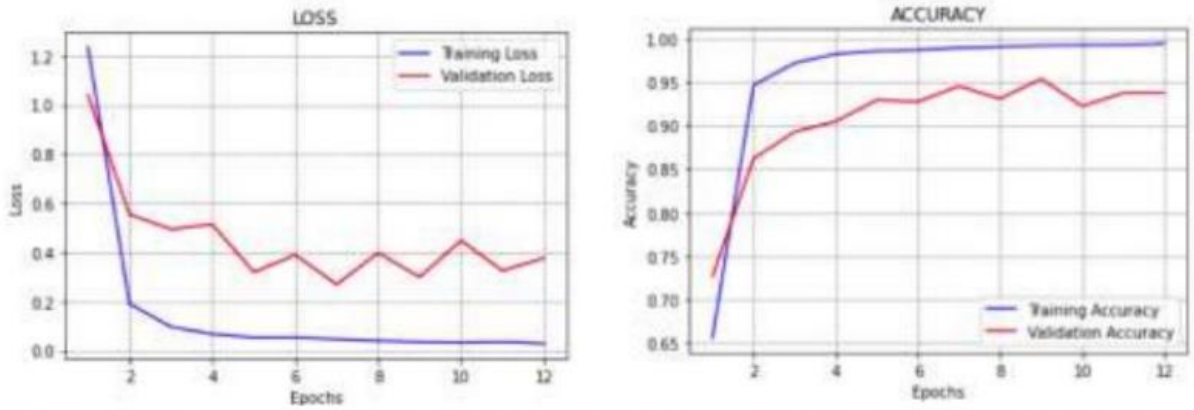


Рис. 2. Примеры графиков функции потерь и графиков точности моделей.

Задание 2.

Применить нецелевую атаку уклонения на основе белого ящика против моделей глубокого обучения. Реализовать следующие типы атак: Fast Gradient Sign Method (FGSM) и Projected Gradient Descent (PGD). Может быть использован код из следующих библиотек: Adversarial Robustness Toolbox ART , Cleverhans CH, scratchai SC.

Наиболее проработанная библиотека – Adversarial Robustness Toolbox, рекомендуется использовать её, но другие также могут быть применены.

Например, https://github.com/Trusted-AI/adversarial-robustness-toolbox/blob/main/notebooks/art-for-tensorflow-v2-keras.ipynb объясняет как использовать ART с помощью Keras. Также есть другие https://github.com/Trusted-AI/adversarial-robustness-toolbox/tree/main/notebooks с примерами атак на основе библиотеки ART.

Используйте атаки FSGM и PGD для создания нецелевых атакующих примеров используя первые 1,000 изображений из тестового множества. Необходимо использовать следующие значения параметра искажения: $\epsilon\epsilon$= [1/255, 2/255, 3/255, 4/255, 5/255, 8/255, 10/255, 20/255, 50/255, 80/255]. Постройте графики точности 2-х моделей в зависимости от параметра искажений $\epsilon\epsilon$ (пример на рис. 3, $\epsilon\epsilon$=80/255≈0.3). Для атаки FGSM, отобразите исходное изображение из датасета и атакующее изображение с указанием величины параметра $\epsilon\epsilon$= [1/255, 5/255, 10/255, 50/255, 80/255], отобразите предсказанный класс атакующего изображения (см. рис. 4).

Отчёт должен содержать: (a) Заполненную таблицу 2. Все модели должны иметь точность менее 60% для $\epsilon\epsilon$=10/255. (b) Для каждой модели постройте график зависимости точности классификации от параметра искажений $\epsilon\epsilon$ (как на рис. 3). (c) Сделать выводы о полученных результатах.

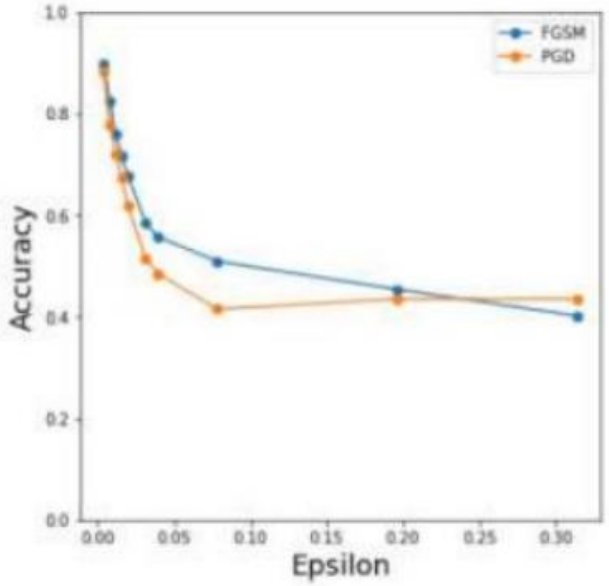


Рис. 3. Зависимость точности классификации от параметра искажений эпсилон

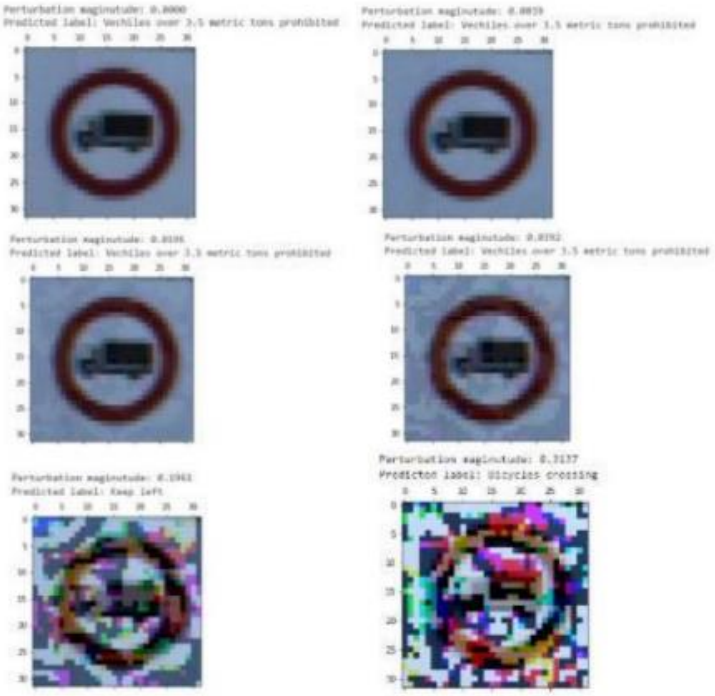


Рис. 4. Пример исходных и атакующих изображений

Таблица 2.

Модель	Исходные изображения	Adversarial images ϵ=1/255	Adversarial images ϵ=5/255	Adversarial images ϵ=10/255
VGG16 - FGSM				
VGG16 - PGD				
ResNet50 - FGSM				
ResNet50 - PGD				

Задание 3:

Применение целевой атаки уклонения методом белого против моделей глубокого обучения.

Шаг 1: Используйте изображения знака «Стоп» (label class 14) из тестового набора данных. Всего имеется 270 изображений. Примените атаку Projected Gradient Descent (PGD) на знак «Стоп» с целью классификации его как знака «Ограничение скорости 30» (target label class = 1). Изменяйте значения искажений $\epsilon\epsilon$= [1/255, 3/255, 5/255, 10/255, 20/255, 50/255, 80/255], и заполните отчёт значениями точности классификации изображений знаков "Стоп" и "Ограничение скорости 30".

Шаг 2: Повторите атаку методом FGSM, и объясните производительность по сравнению с PGD. Отчёт должен содержать: (a) Заполненную таблицу 3. Объясните какой размер искажений достигает максимальной производительности и объясните причины. (b) Постройте 5 примеров исходных изображений знака «Стоп» и соответствующих атакующих примеров (см. рис. 5). (c) Сравните результаты атак PGD и FGSM между собой.

Таблица 3.

Искажение	PGD attack – Stop sign images	PGD attack – Speed Limit 30 sign images
ϵ=1/255		
ϵ=3/255		
ϵ=5/255		
ϵ=10/255		
ϵ=20/255		
ϵ=50/255		
ϵ=80/255		

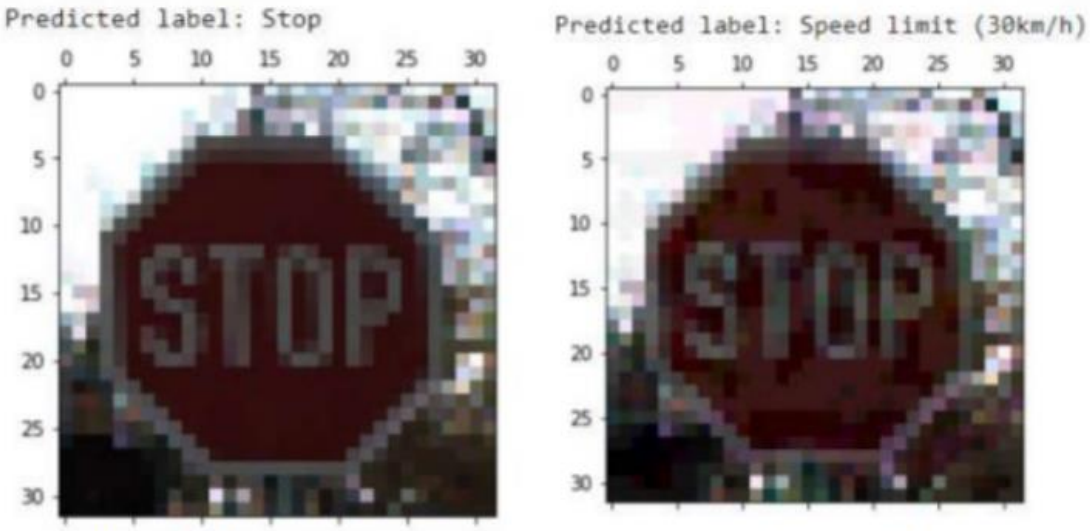


Рис. 5. Пример исходных и атакующих изображений

Задание 1

Создаем модель ResNet50, выборки поделены 70/30, показано на рисунке 6.

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

Рис.6. модель ResNet50

Первый график отображает точность обучения и валидации модели RESNET50, показан на рисунке 7.

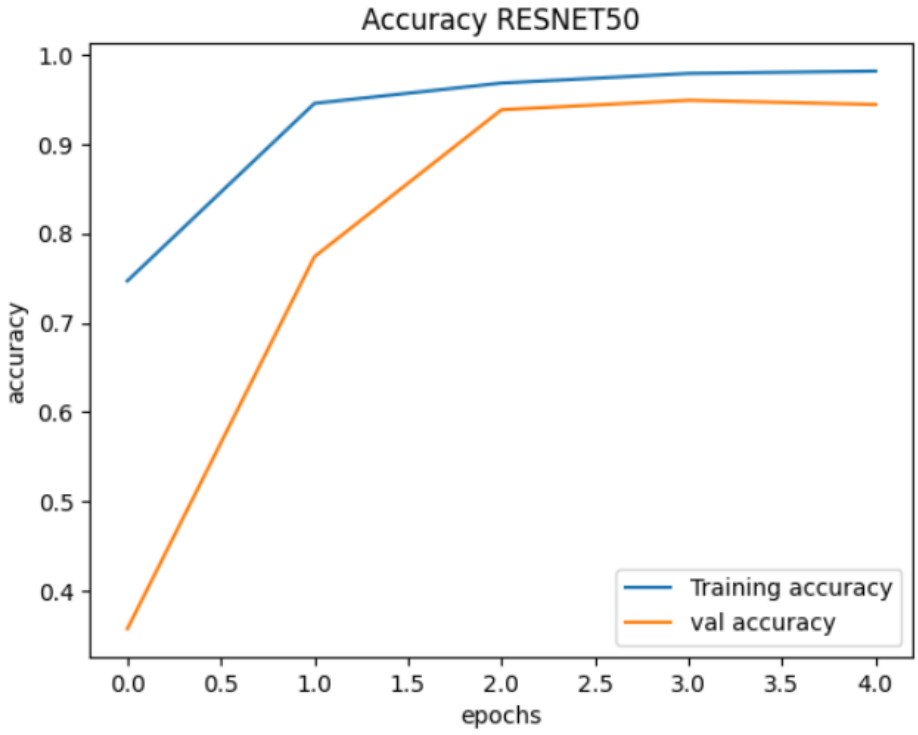


Рис.7. Accuracy ResNet50

Второй график отображает потерю обучения и валидации модели RESNET50, показан на рисунке 8.

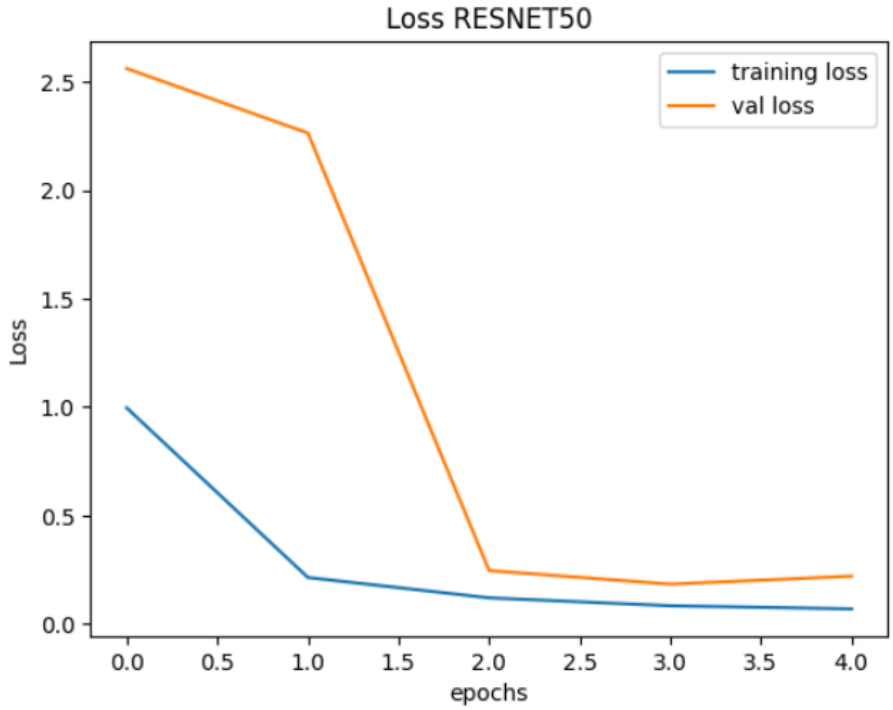


Рис.7. Loss ResNet50

Создаем модель VGG16, показано на рисунке 8.

```
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
```

Рис.8. модель VGG16

Третий график отображает точность обучения и валидации модели VGG16, показан на рисунке 9.

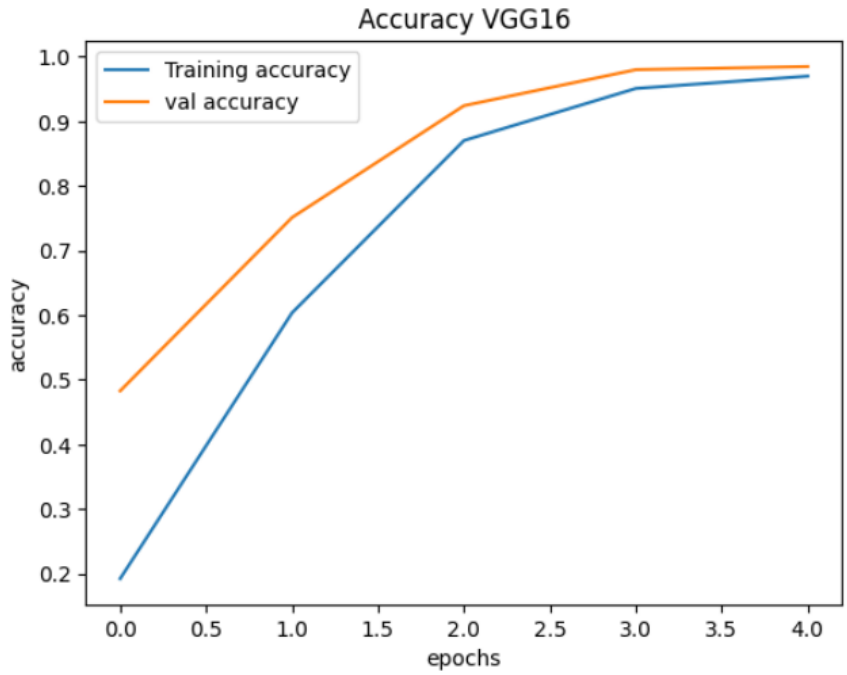


Рис.9. Accuracy VGG16

Четвертый график отображает потерю обучения и валидации модели VGG16, показан на рисунке 10.

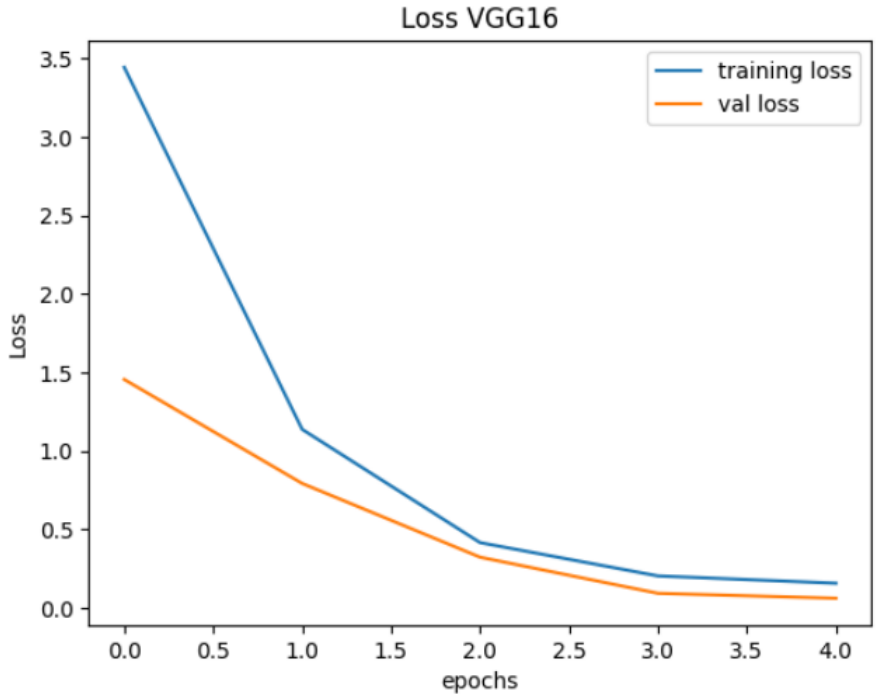


Рис.10. Loss VGG16

Заполним Таблицу 1.

Модель	Обучение	Валидация	Тест
ResNet50	loss: 0.0697 accuracy: 0.9816	loss: 0.2205 accuracy: 0.9442	loss: 0.4797 accuracy: 0.8907
VGG16	loss: 0.1551 accuracy: 0.9698	loss: 0.0592 accuracy: 0.9847	loss: 0.2825 accuracy: 0.9426

Задание 2

Проведем атаки FGSM и PGD на модель RESNET50, используя первые 1,000 изображений из тестового множества. Используем значения параметра искажения:

ϵ = [1/255, 2/255, 3/255, 4/255, 5/255, 8/255, 10/255, 20/255, 50/255, 80/255].

Построим график зависимости точности классификации от параметра искажений эпсилон для RESNET50, показан на рисунке 11.

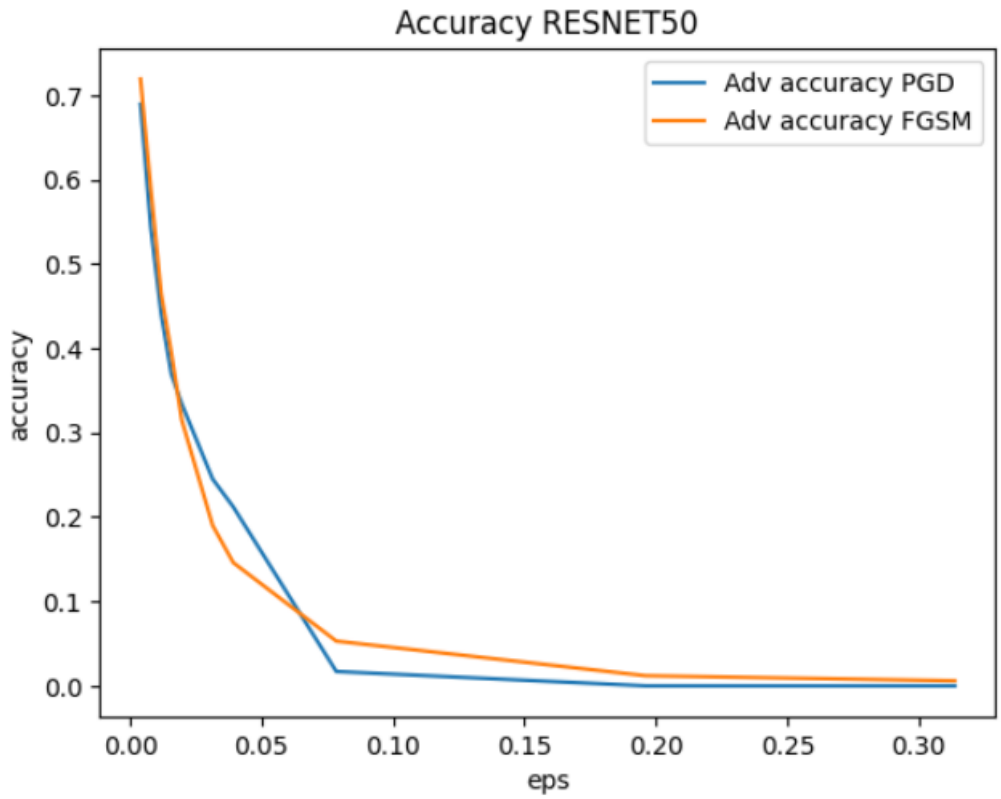


Рис.11. Accuracy ResNet50

Проведем атаки FGSM и PGD на модель VGG16, используя первые 1,000 изображений из тестового множества. Используем значения параметра искажения:

ϵ = [1/255, 2/255, 3/255, 4/255, 5/255, 8/255, 10/255, 20/255, 50/255, 80/255].

Построим график зависимости точности классификации от параметра искажений эпсилон для VGG16, показан на рисунке 12.

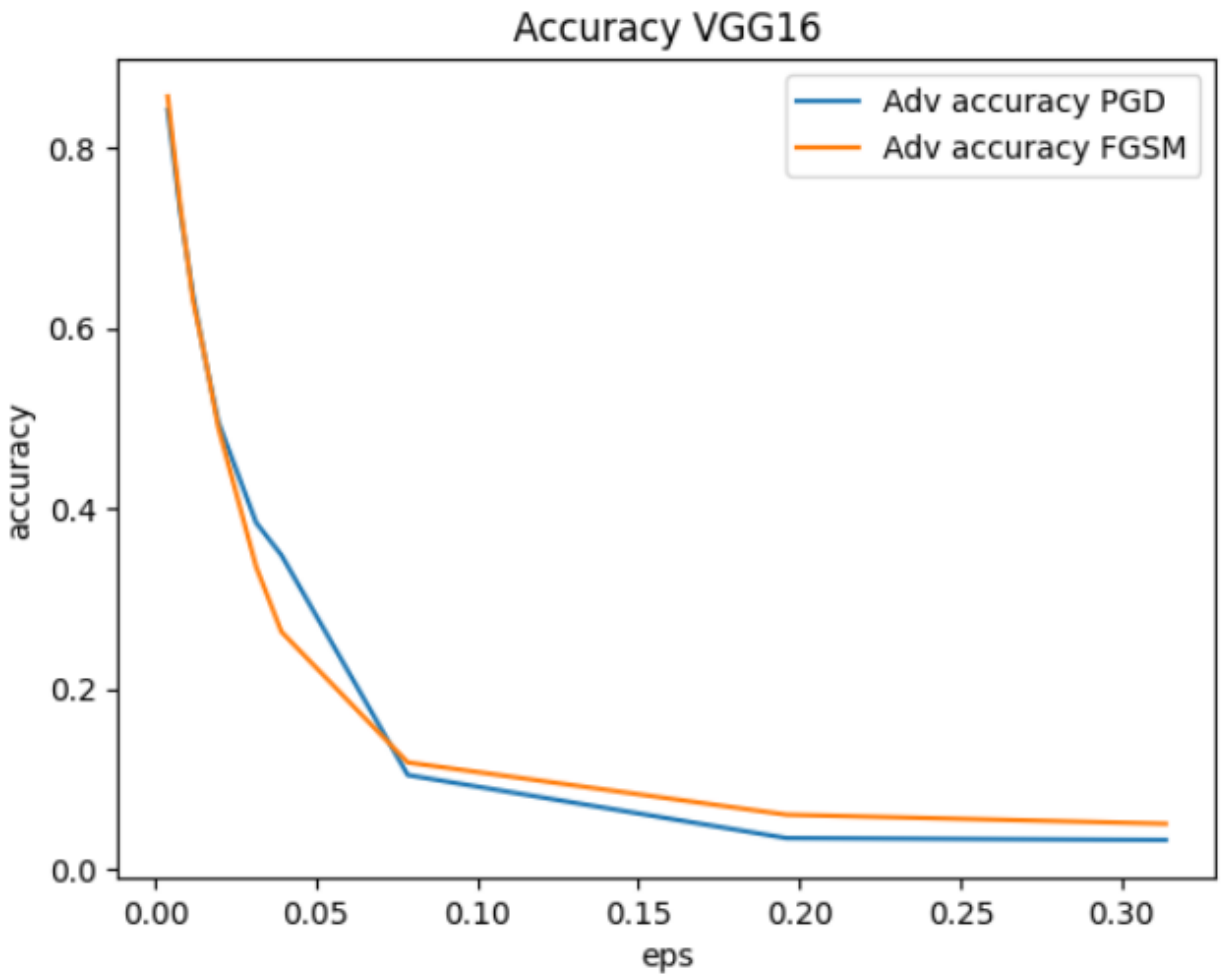


Рис.12. Accuracy VGG16

Для атаки FGSM RESNET50, отобразим исходное изображение из датасета и атакующее изображение с указанием величины параметра $\epsilon\epsilon$= [1/255, 5/255, 10/255, 50/255, 80/255], также отобразим предсказанный класс атакующего изображения, показаны на рисунке 13.

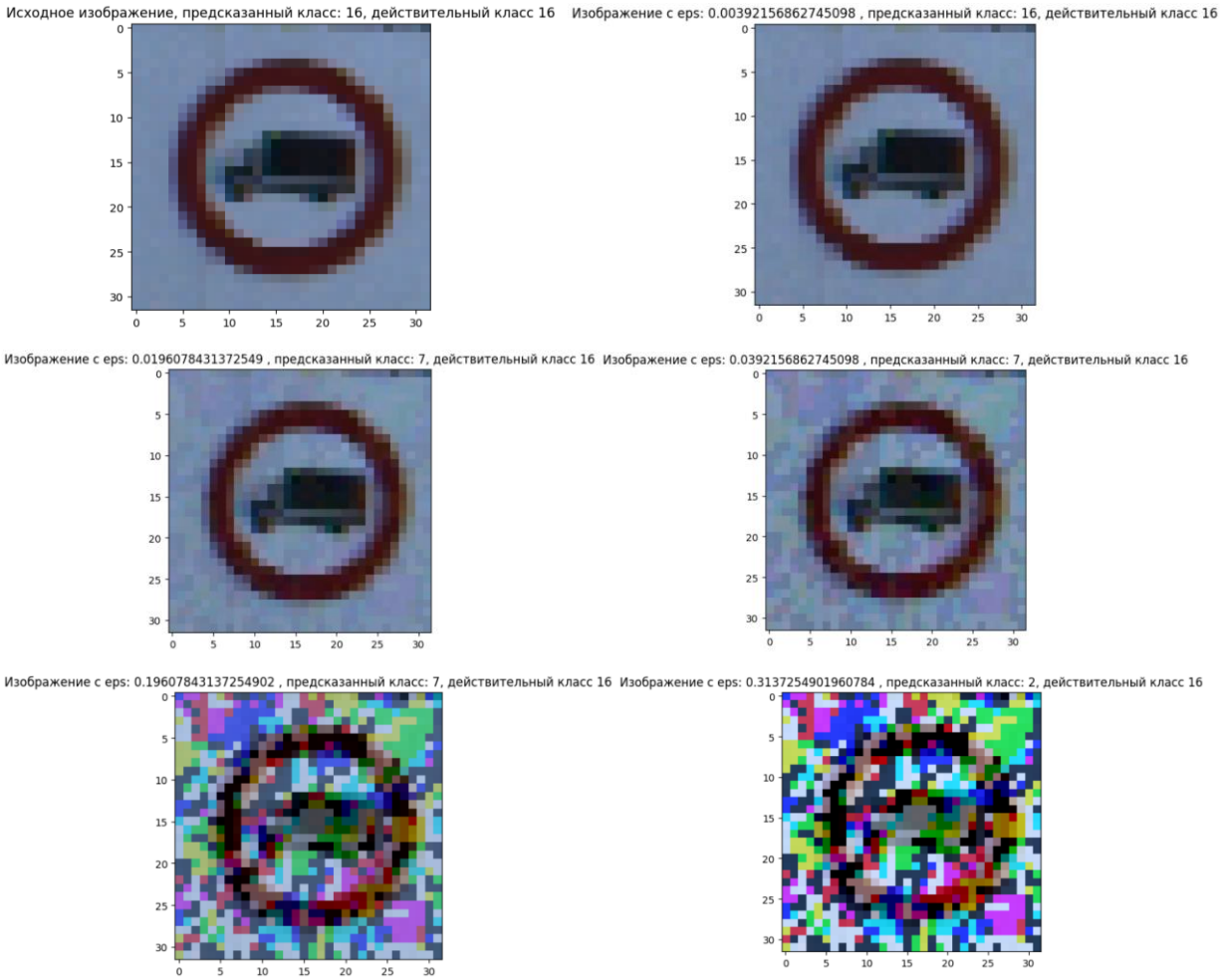


Рис.13. Изображения RESNET50

Для атаки FGSM VGG16, отобразим исходное изображение из датасета и атакующее изображение с указанием величины параметра $\epsilon\epsilon$= [1/255, 5/255, 10/255, 50/255, 80/255], также отобразим предсказанный класс атакующего изображения, показаны на рисунке 14.

Рис.14. Изображения VGG16

Заполним Таблицу 2.

Модель	Исходные изображения	Adversarial images $\epsilon\epsilon$=1/255	Adversarial images $\epsilon\epsilon$=5/255	Adversarial images $\epsilon\epsilon$=10/255
VGG16 - FGSM	89%	79%	44%	21%
VGG16 - PGD	89%	77%	48%	32%
ResNet50 - FGSM	91%	74%	33%	17%
ResNet50 - PGD	91%	71%	30%	23%

Задание 3

Используя изображения знака «Стоп» (label class 14) из тестового набора данных, применим атаки FGSM и PGD на знак «Стоп» с целью классификации его как знака «Ограничение скорости 30» (target label class = 1), изменяя значения искажений ϵ= [1/255, 3/255, 5/255, 10/255, 20/255, 50/255, 80/255].

Выведем 5 пар примеров исходных изображений знака «Стоп» и соответствующих атакующих примеров для атаки FGSM. Рисунки 15-19.

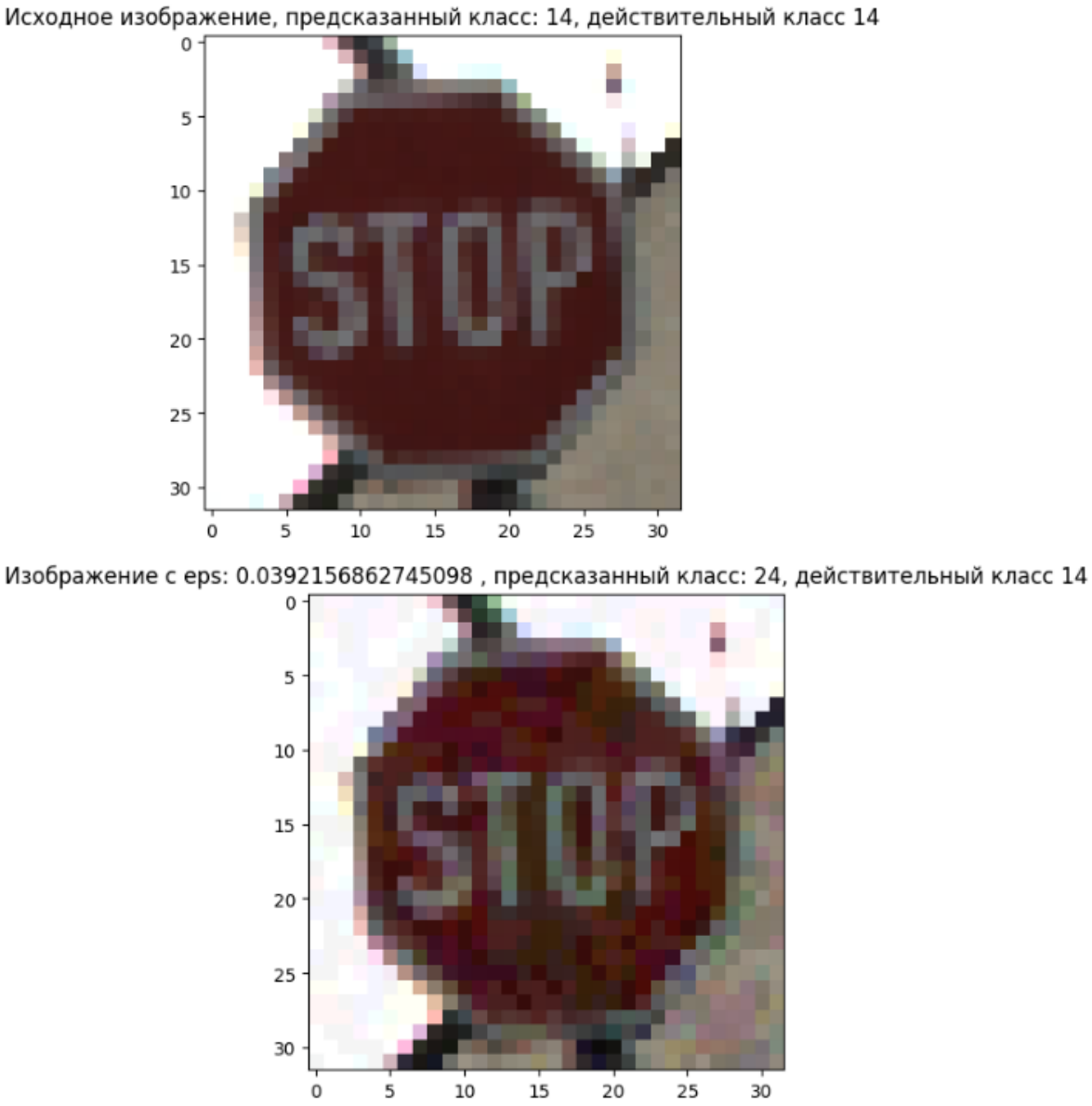


Рис.15. Изображения FGSM

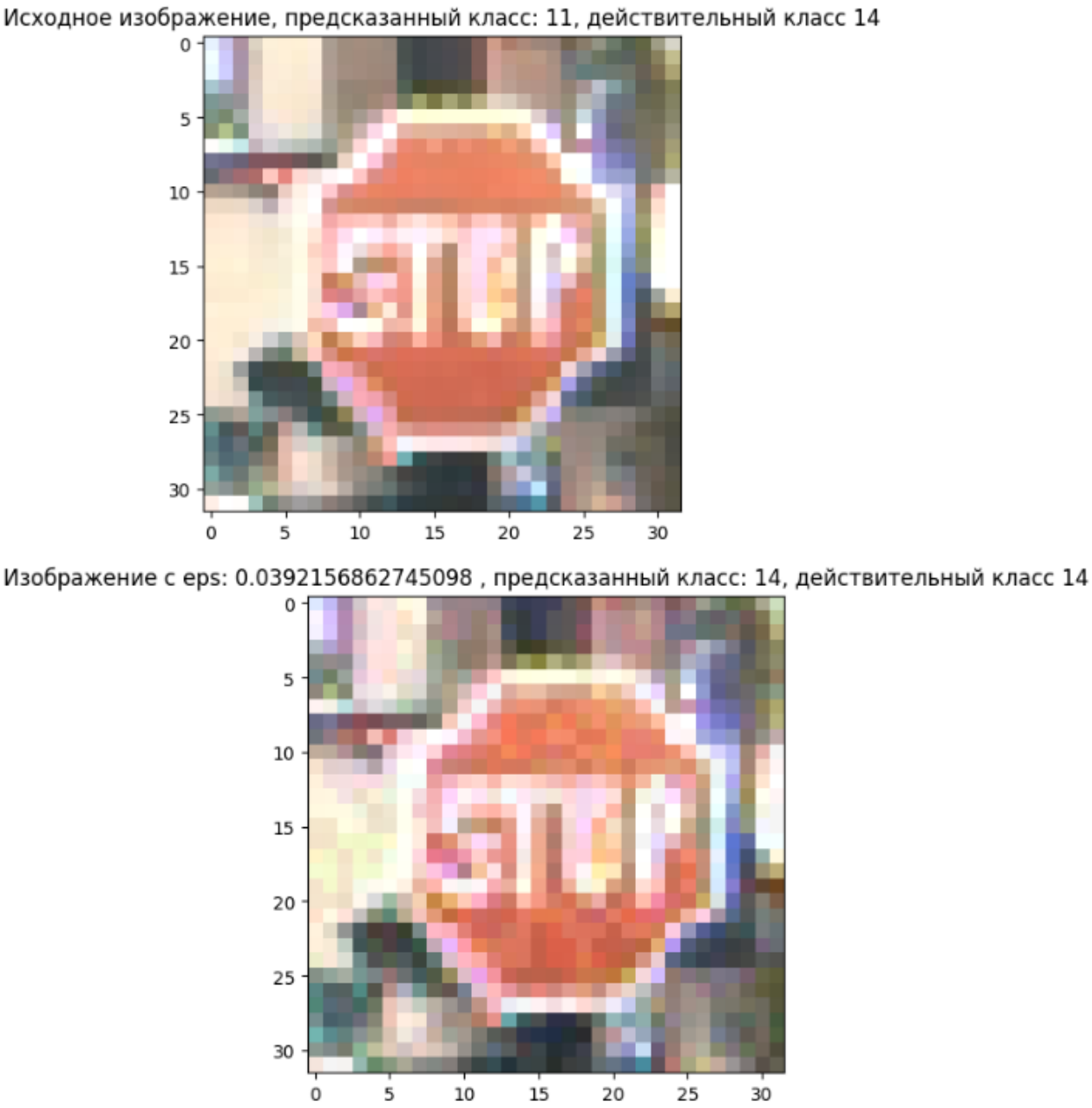


Рис.16. Изображения FGSM

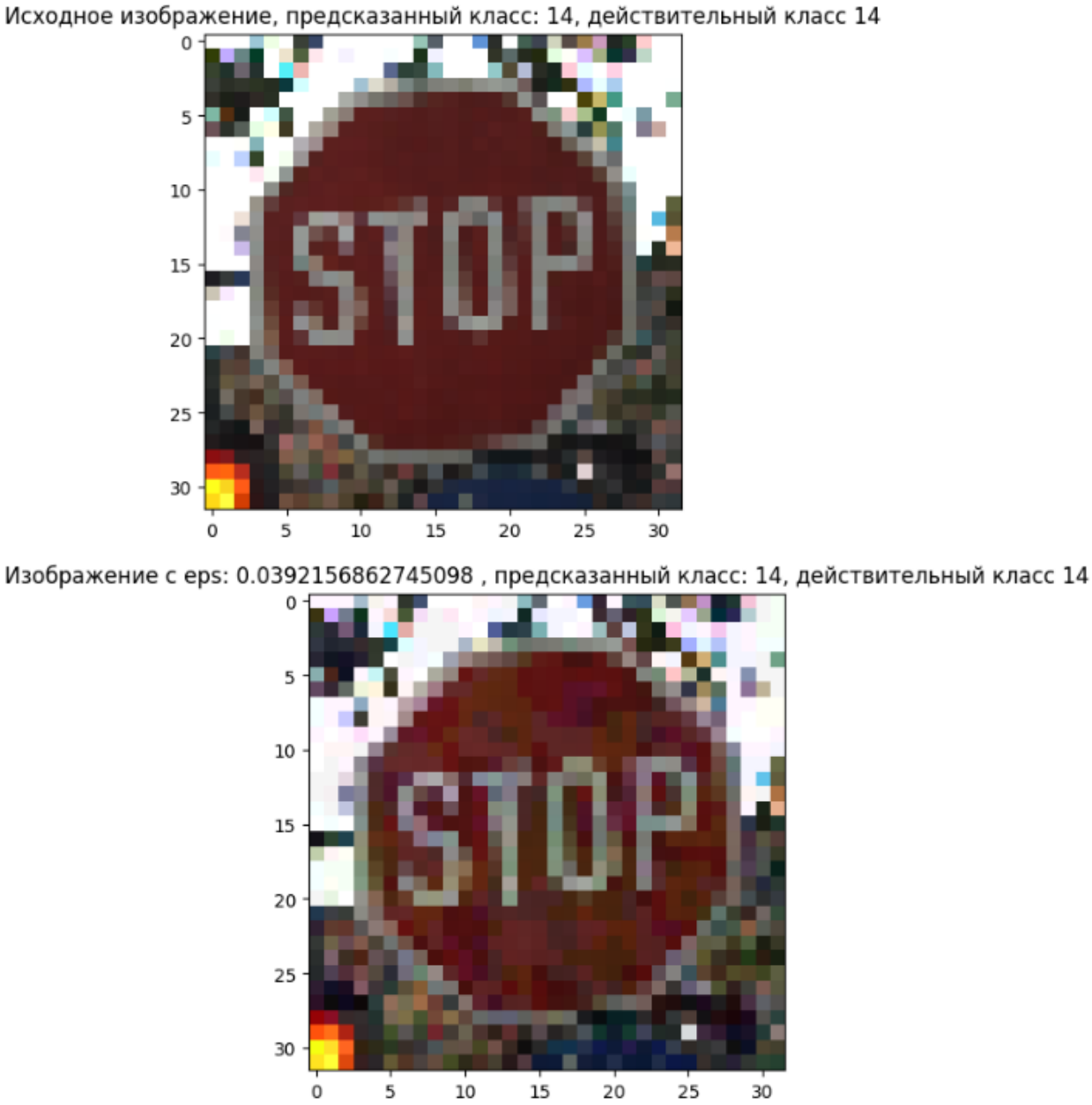
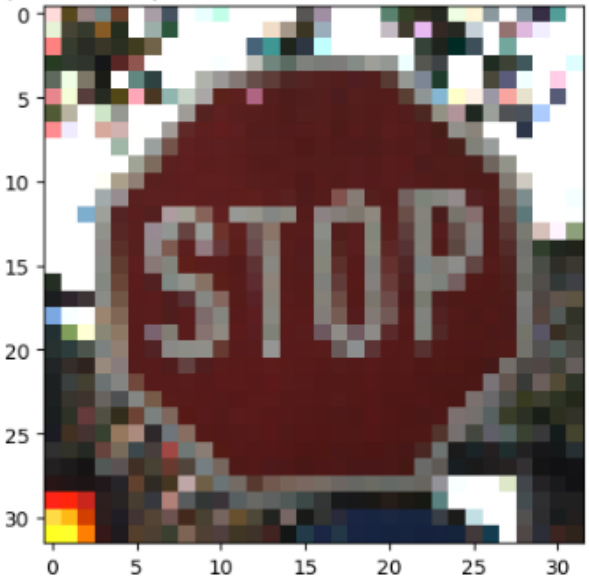


Рис.17. Изображения FGSM

Исходное изображение, предсказанный класс: 14, действительный класс 14



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

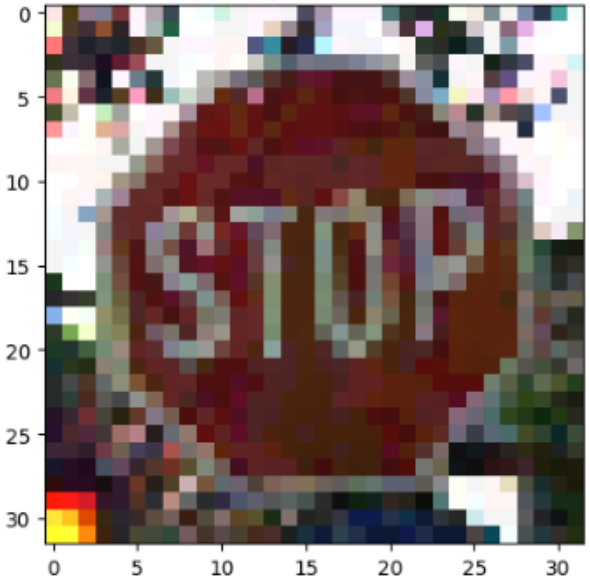


Рис.18. Изображения FGSM

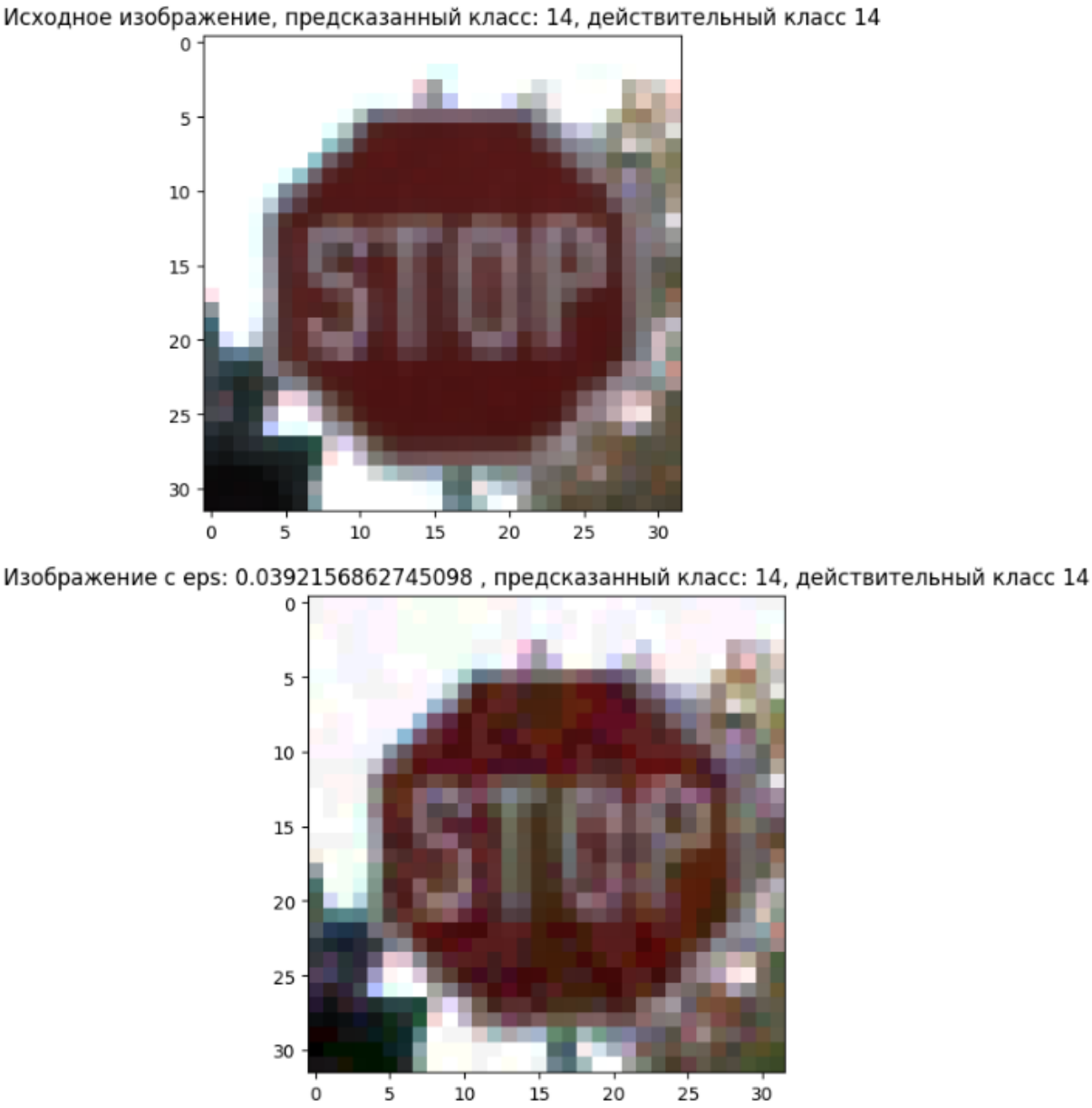


Рис.19. Изображения FGSM

Выведем 5 пар примеров исходных изображений знака «Стоп» и соответствующих атакующих примеров для атаки PGD. Рисунки 20-24.

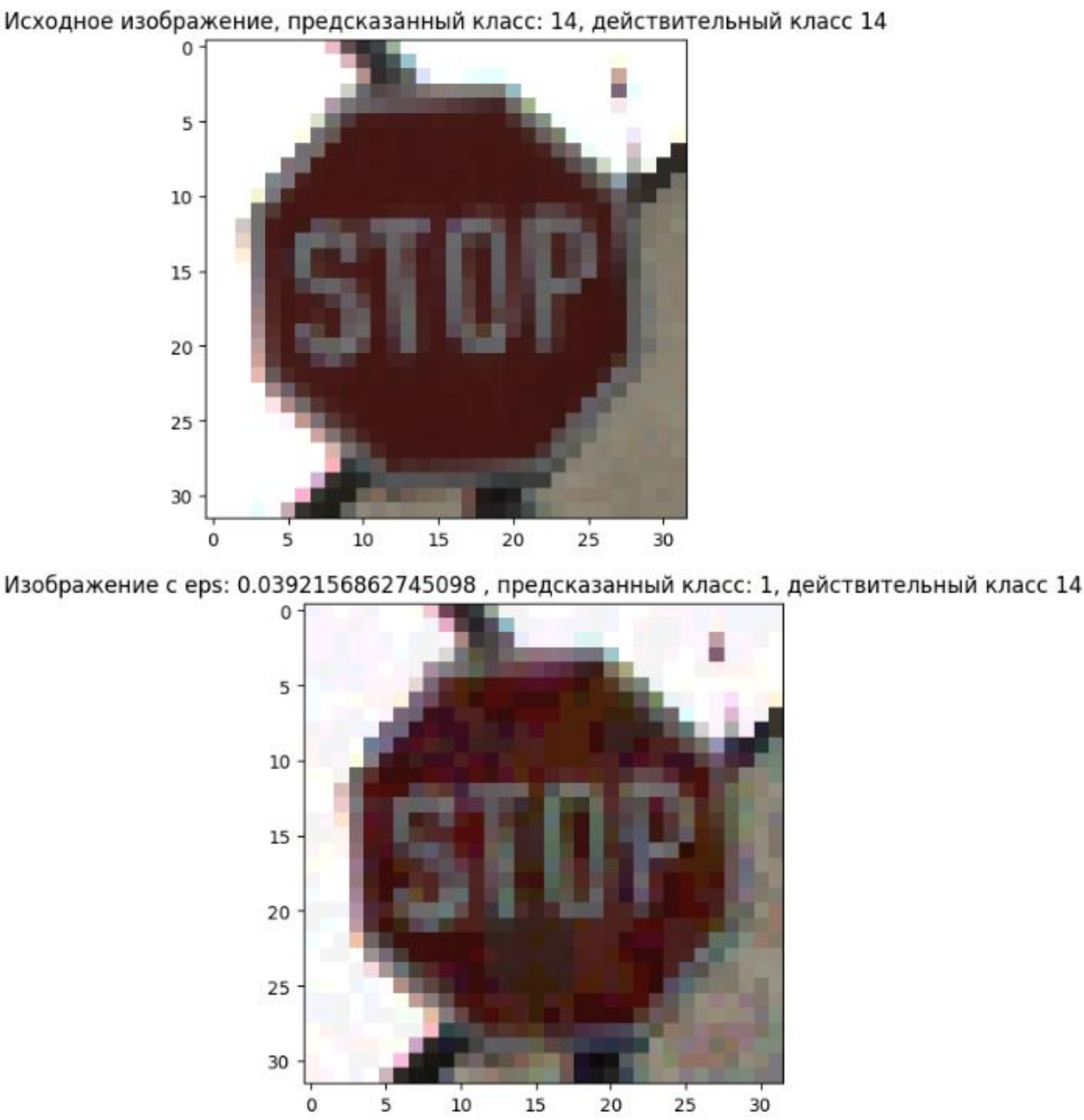


Рис.20. Изображения PGD

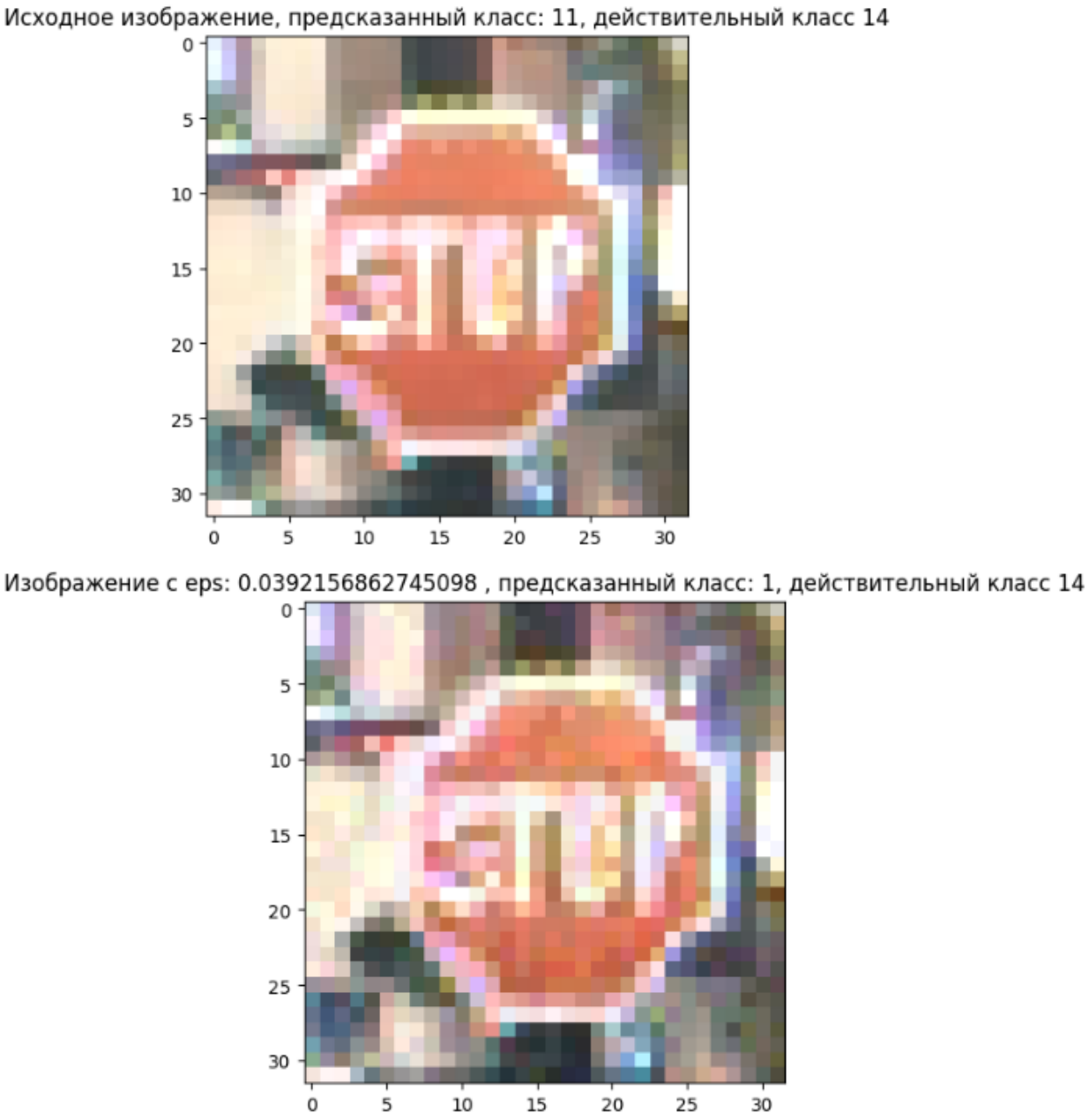


Рис.21. Изображения PGD

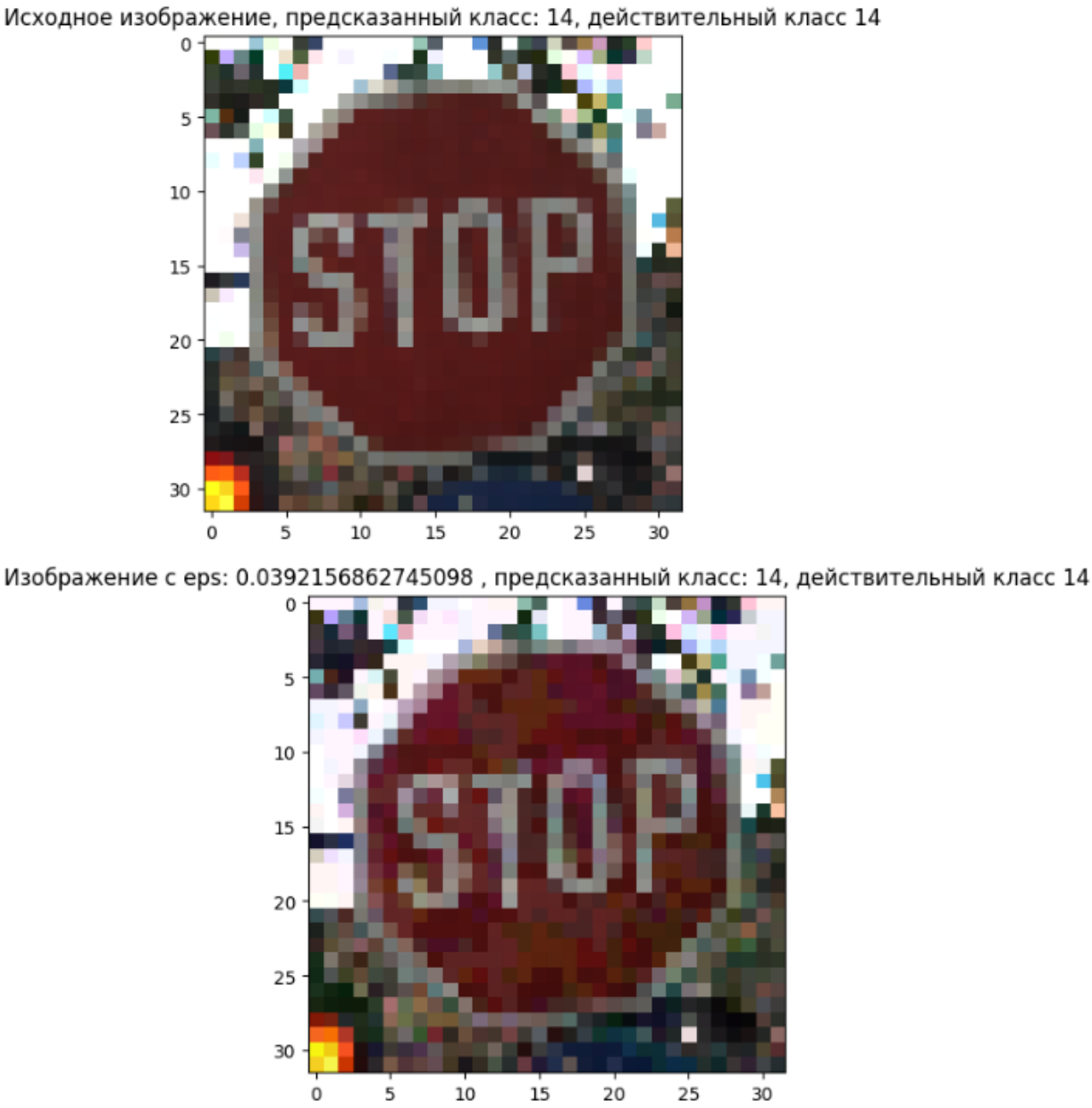


Рис.22. Изображения PGD

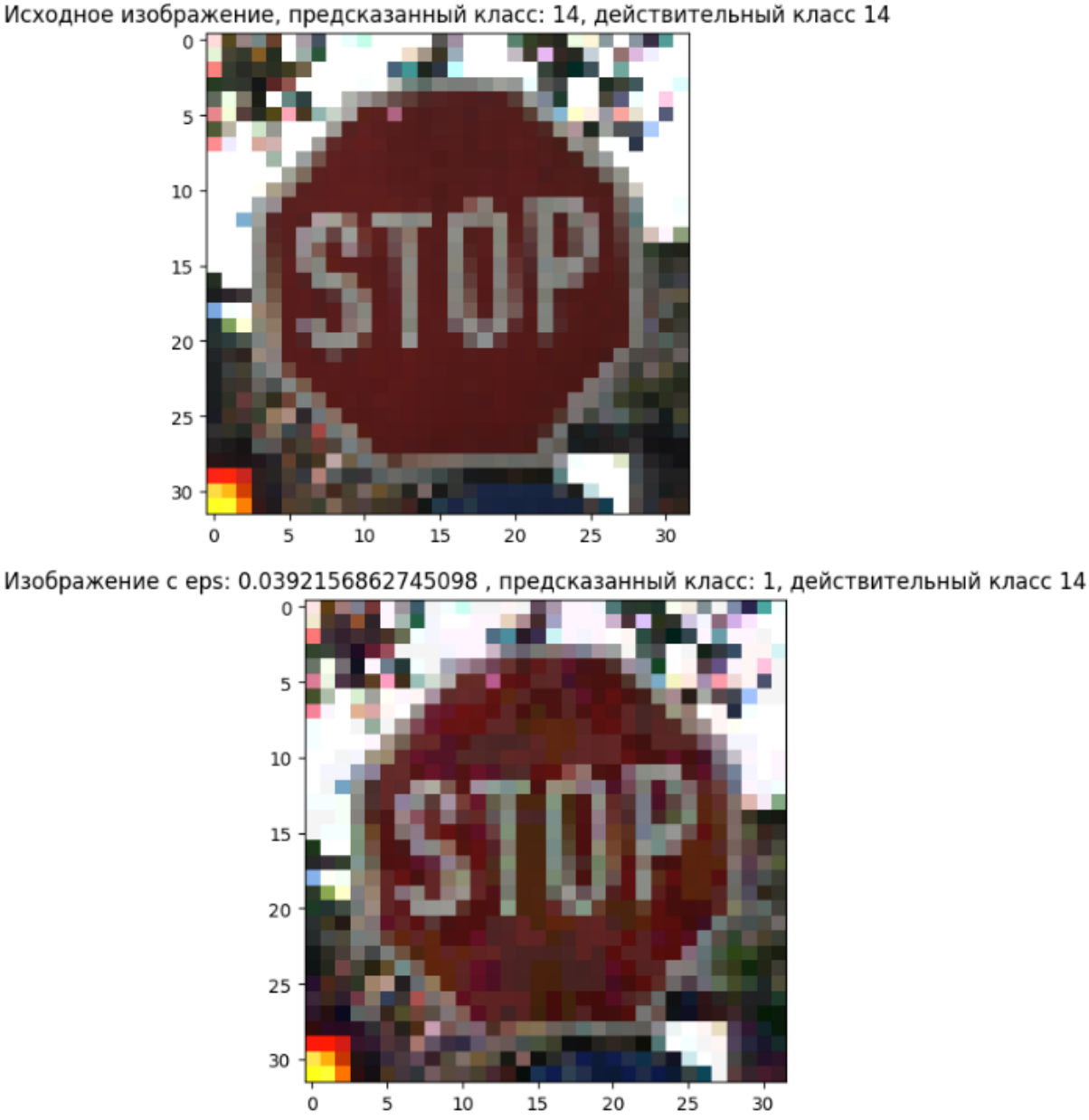


Рис.23. Изображения PGD

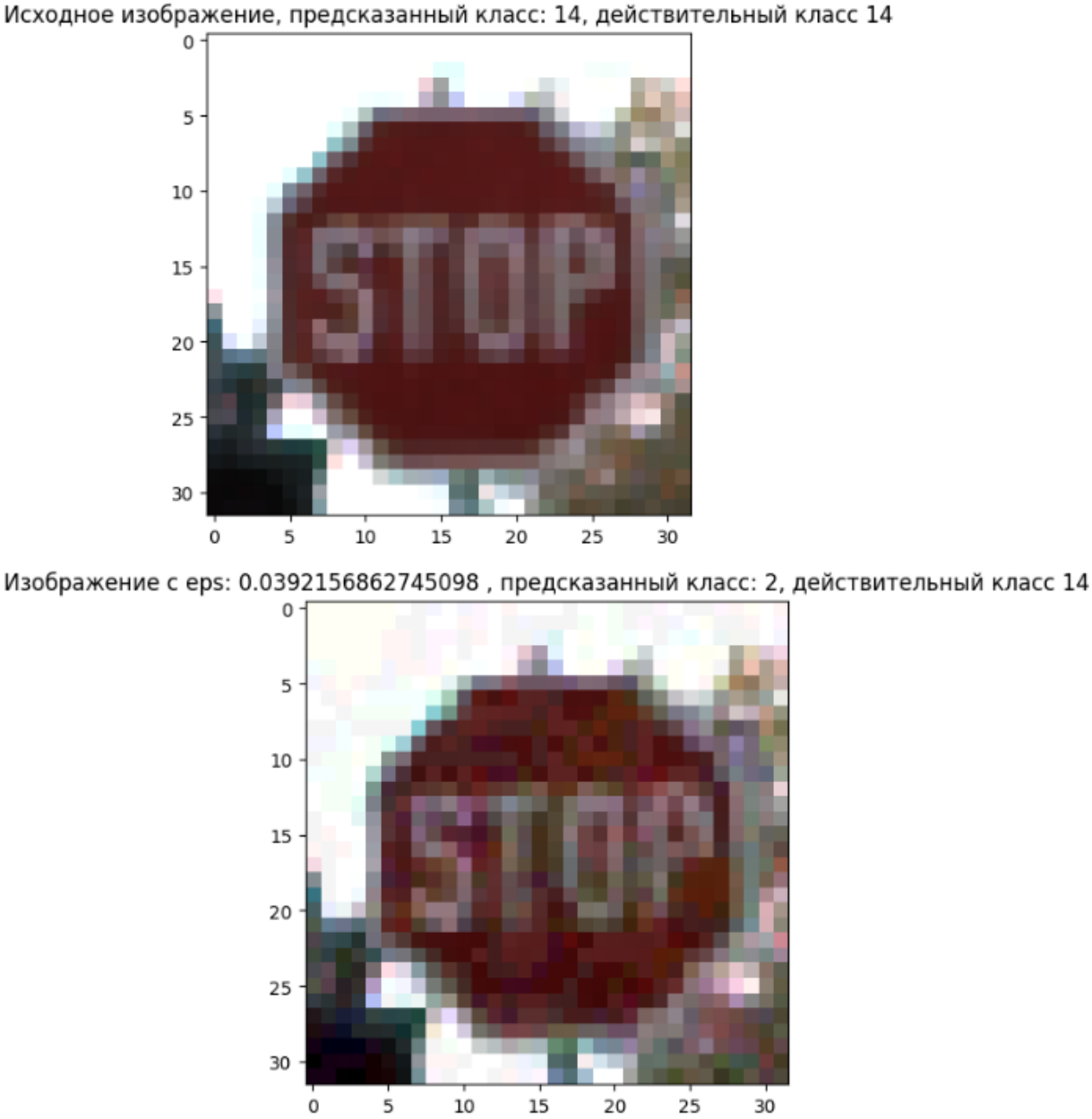


Рис.24. Изображения PGD

Заполним Таблицу 3.

Искажение	FGSM – Stop	FGSM – Limit 30	PGD - Stop	PGD – Limit 30
1/255	99%	99%	97%	99%
3/255	80%	99%	91%	99%
5/255	73%	99%	90%	99%
10/255	26%	99%	71%	99%

По результатам видно метод PGD значительно лучше подходит для целевой атаки, чем метод FGSM.

Выводы

В ходе работы были реализованы атаки уклонения на основе белого ящика против классификационных моделей на основе глубокого обучения и получены практические навыки переноса атак уклонения на основе черного ящика против моделей машинного обучения.

В целом, работа демонстрирует эффективность атак уклонения на основе белого ящика против моделей машинного обучения и необходимость дальнейших исследований в области безопасности систем ИИ.