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

Институт кибербезопасности и цифровых технологий

КБ-4 «Интеллектуальные системы информационной безопасности»

Отчет по лабораторной работе №2

по дисциплине: «Анализ защищенности систем искусственного интеллекта»

Выполнил:

Студент группы ББМО-02-22

Кузьмин Владимир Дмитриевич

Проверил:

К.т.н. Спирин Андрей Андреевич

Москва 2023

Содержание

Задание на лабораторную работу

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. Вы можете загрузить набор данных (152 МБ) по ссылке.

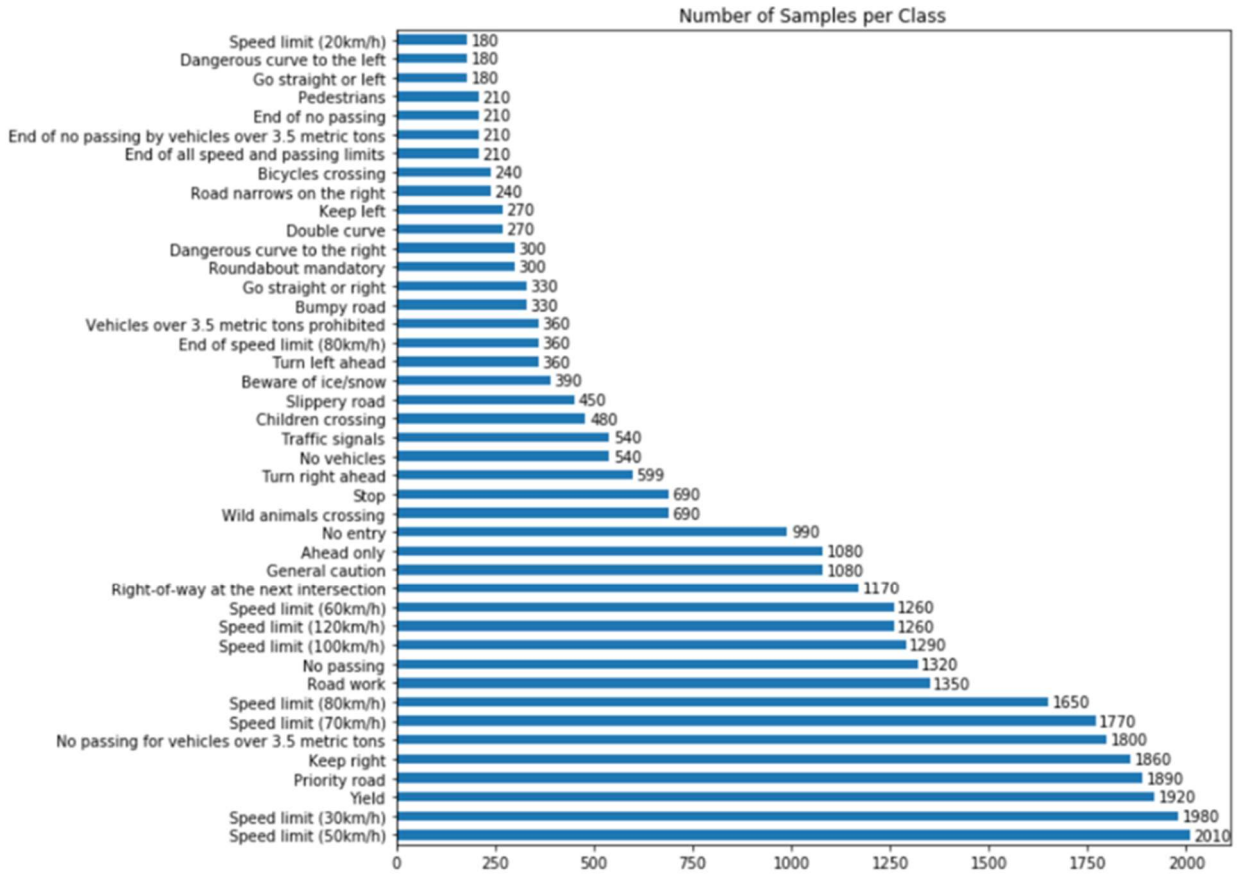

Рисунок 1 – Распределение изображений в GTRSB

Задание 1:

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.

2. Использовать предобученные сети (например, на ImageNet).

3. Выполнить поиск наилучших гиперпараметров моделей.

4. Использовать бесплатные ресурсы GPU сервиса Google Colab.

5. 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 рисунке 2).

Таблица 1 – Таблица по результатам

Модель	Обучение	Валидация	Тест
VGG16			
ResNet50			

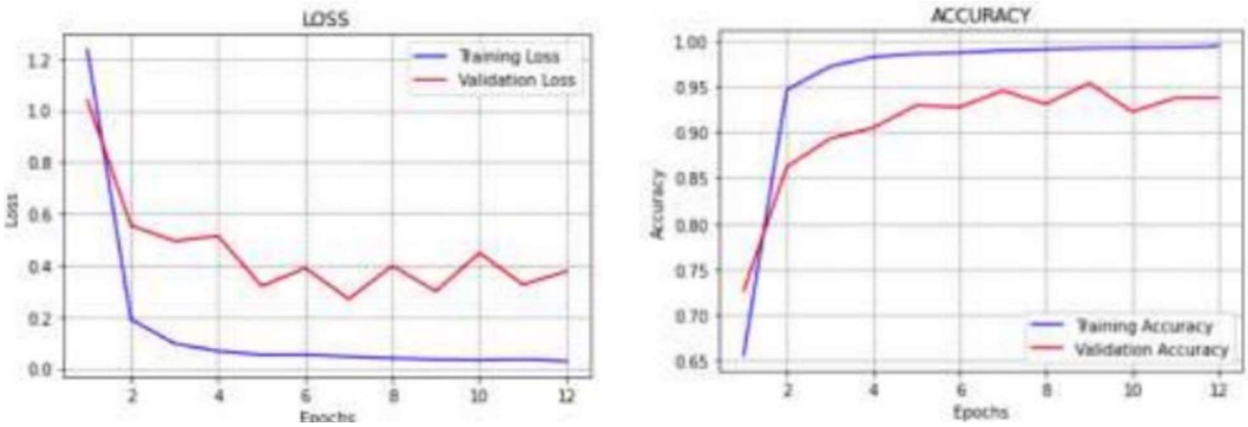

Рисунок 2 – Примеры графиков функции потерь и графиков точности моделей

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, this notebook объясняет как использовать ART с помощью Keras.

Также есть другие 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: ϵ= [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 ϵ= [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 ϵ=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.

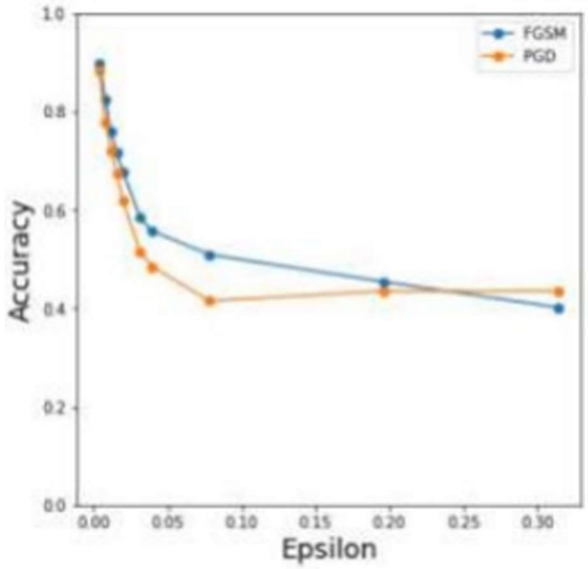

Рисунок 3 – Зависимость точности классификации от параметра искажений эпсилон

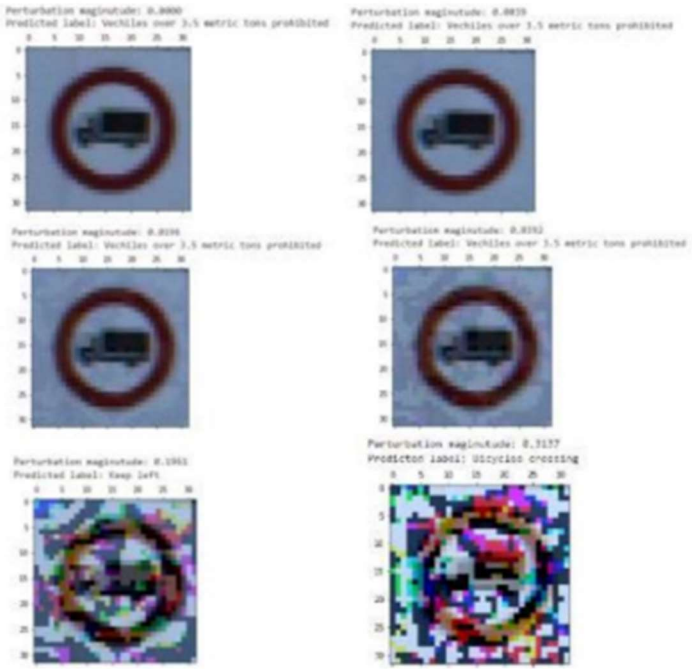

Рисунок 4 – Пример исходных и атакующих изображений

Таблица 2 – Таблица по результатам

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 ϵ= [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 – Таблица по результатам

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		

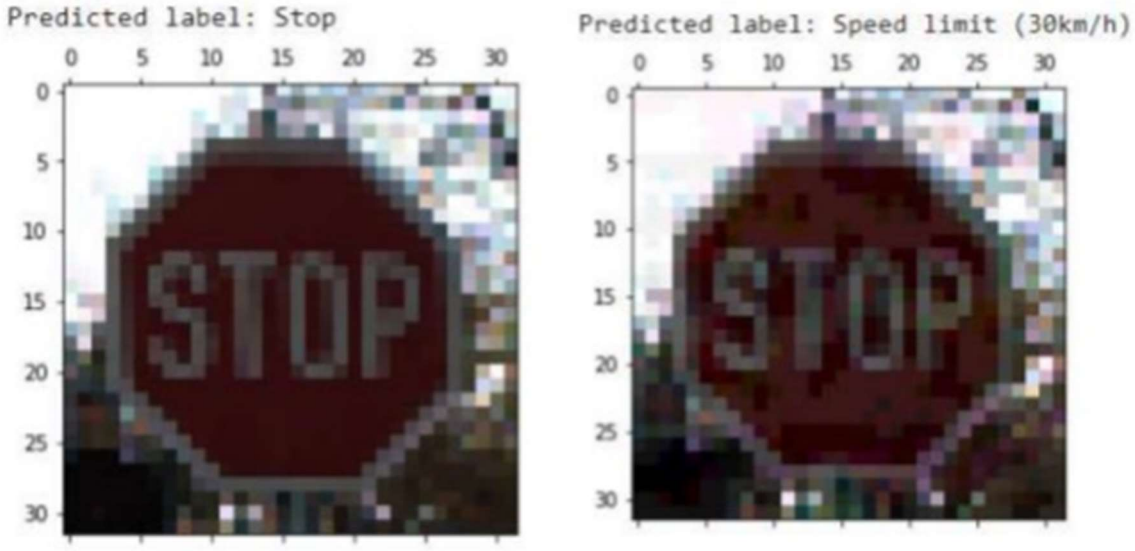

Рисунок 5 – Пример исходных и атакующих изображений

Задание 1

В данном задании нам требуется обучить 2 классификатора на основе глубоких нейронных сетей на датасете GTSRB. В качестве исходных данных была взята лишь часть набора примеров.

Набор данных был поделен на обучающую и тестовую выборки в соотношении 70 к 30.

Первая модель построена на базе ResNet50 и состоит из слоев, представленных на рисунке 1.1.

```
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 1.1 – Модель ResNet50

В результате эмпирического исследования были выбраны оптимальные значения количества эпох обучения и размера пакета, равные 5 и 64 соответственно. Графики процесса обучения представлены на рисунке 1.2. Валидационные показатели представлены на рисунке 1.3.

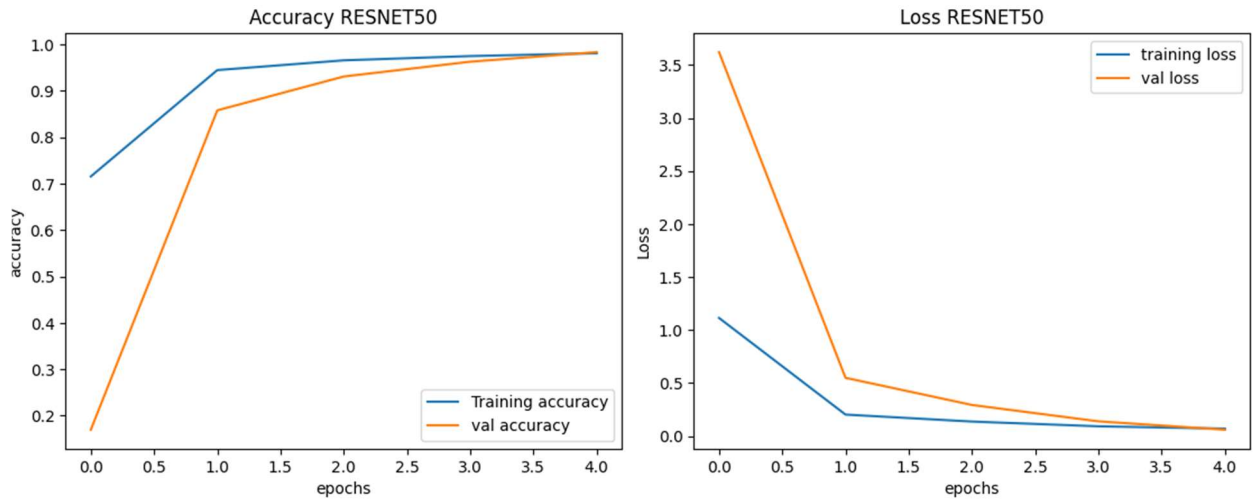

Рисунок 1.2 – Графики ResNet50

```
loss: 0.0705 - accuracy: 0.9812 - val_loss: 0.0606 - val_accuracy: 0.9837
```

Рисунок 1.3 – Валидация ResNet50

После обучения модель была протестирована на тестовом наборе. Показатели валидации приведены на рисунке 1.4.

```
loss, accuracy = model.evaluate(data, y_test)
print(f"Test loss: {loss}")
print(f"Test accuracy: {accuracy}")
```

```
395/395 [==============================] - 6s 13ms/step - loss: 0.3775 - accuracy: 0.9219
Test loss: 0.3775138556957245
Test accuracy: 0.9219319224357605
```

Рисунок 1.4 – Тестирование ResNet50

Вторая модель построена на базе VGG16 и состоит из слоев, представленных на рисунке 1.5.

```
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
```

Рисунок 1.5 – Модель VGG16

Графики процесса обучения представлены на рисунке 1.6. Валидационные показатели представлены на рисунке 1.7.

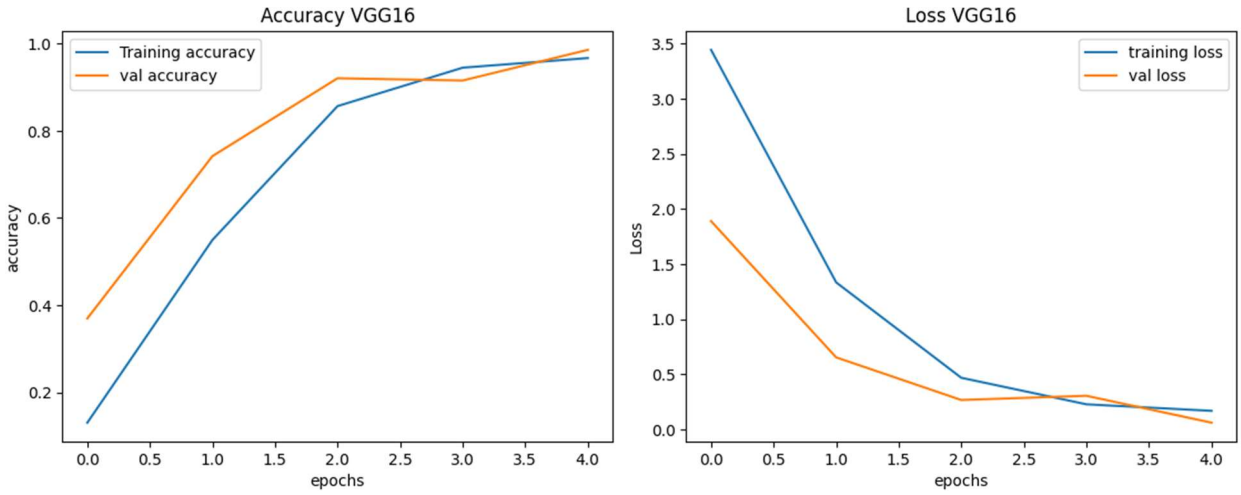

Рисунок 1.6 – Графики VGG16

```
loss: 0.1674 - accuracy: 0.9668 - val_loss: 0.0599 - val_accuracy: 0.9855
```

Рисунок 1.7 – Валидация VGG16

После обучения модель была протестирована на валидационном наборе. Показатели валидации приведены на рисунке 1.8.

```
loss, accuracy = model.evaluate(data, y_test)
print(f"Test loss: {loss}")
print(f"Test accuracy: {accuracy}")
```

```
395/395 [==============================] - 4s 9ms/step - loss: 0.2013 - accuracy: 0.9548
Test loss: 0.20125630497932434
Test accuracy: 0.9547901749610901
```

Рисунок 1.8 – Тестировние VGG16

Результирующая таблица по заданию представлена под номером 1.1.

Таблица 1.1 – Таблица по результатам

Модель	**Обучение**	**Валидация**	**Тест**
ResNet50	loss: 0.0705 accuracy: 0.9812	loss: 0.0606 accuracy: 0.9837	loss: 0.3775 accuracy: 0.9219
VGG16	loss: 0.1674 accuracy: 0.9668	loss: 0.0599 accuracy: 0.9855	loss: 0.2013 accuracy: 0.9548

Задание 2

В данном задании требуется применить нецелевую атаку уклонения на основе белого ящика против моделей глубокого обучения.

Используем атаку FSGM и PGD на базе ResNet50 для создания нецелевых атакующих примеров используя первые 1,000 изображений из тестового множества.

Атаки на изображения проводятся со следующими параметрами искажения [1/255, 2/255, 3/255, 4/255, 5/255, 8/255, 10/255, 20/255, 50/255, 80/255].

График зависимости точности предсказания модели на атакованных изображениях от параметра искажения приведен на рисунке 2.1.

Для атаки FGSM, отобразим исходное изображение из датасета и атакующие изображения с указанием величины параметра ϵ = [1/255, 5/255, 10/255, 50/255, 80/255] (рисунок 2.2).

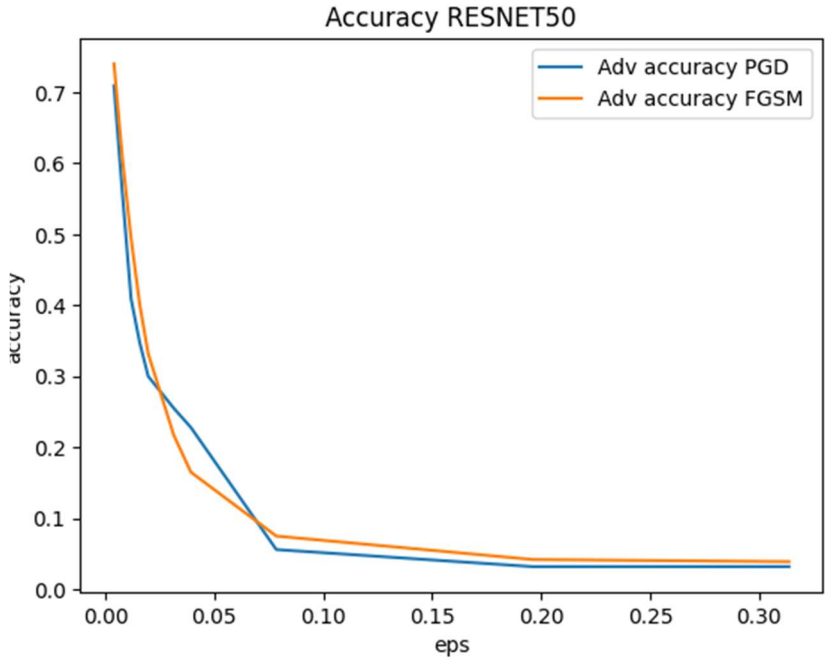

Рисунок 2.1 – График ResNet50

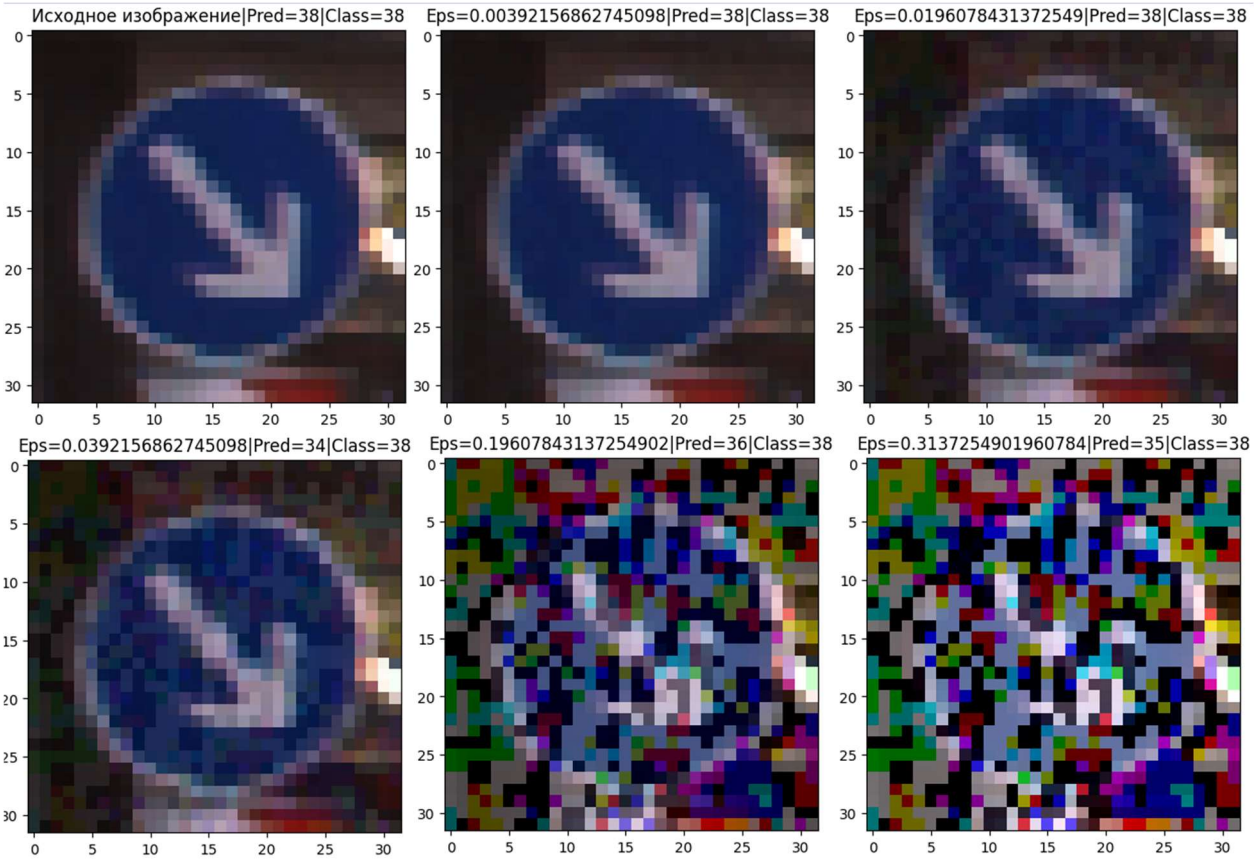

Рисунок 2.2 – Исходное и искаженные изображения ResNet50

Повторим эксперимент с атаками FSGM и PGD на модель на базе VGG16.

График зависимости точности предсказания модели на атакованных изображениях от параметра искажения приведен на рисунке 2.3.

Для атаки FGSM, отобразим исходное изображение из датасета и атакующие изображения с указанием величины параметра ϵ = [1/255, 5/255, 10/255, 50/255, 80/255] (рисунок 2.4).

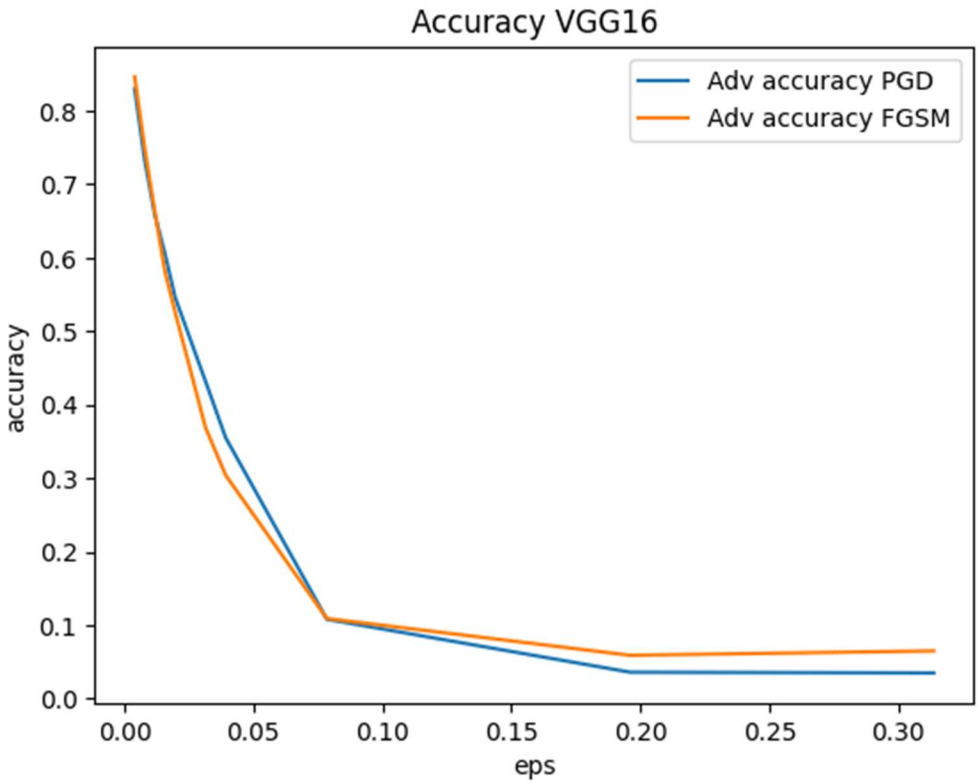

Рисунок 2.3 – График VGG16

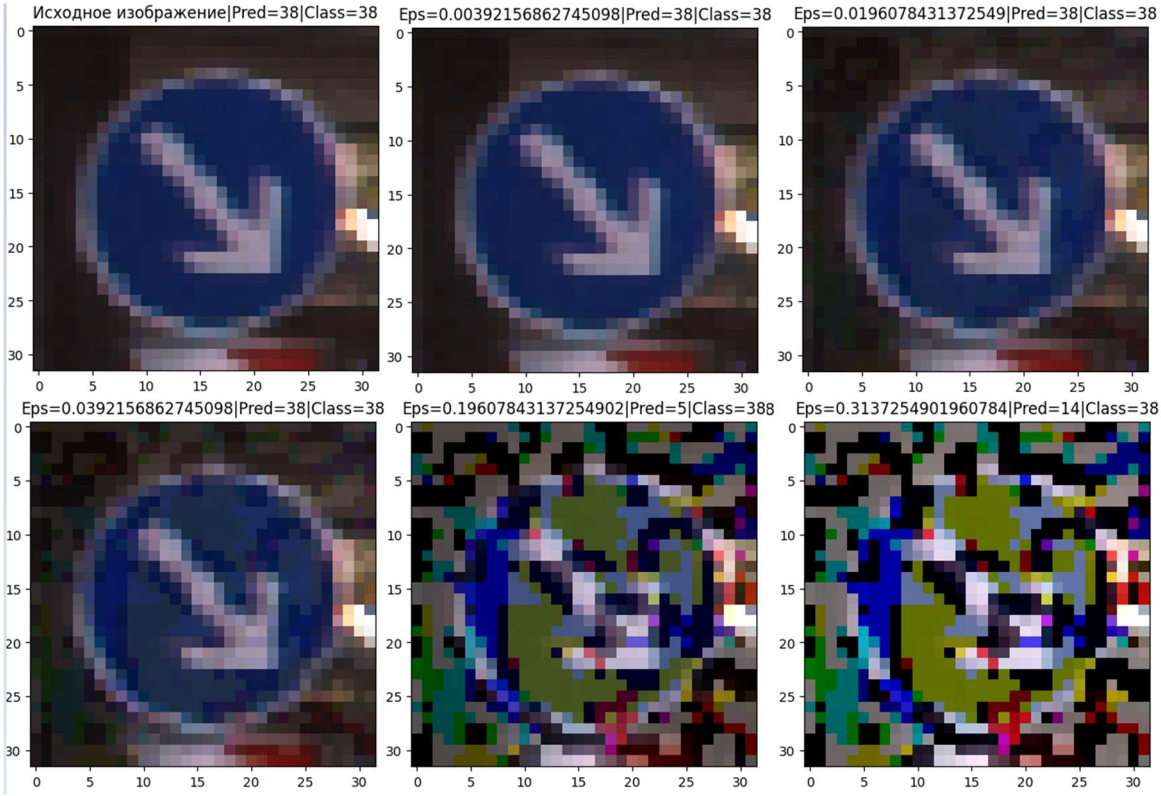

Рисунок 2.4 – Исходное и искаженные изображения VGG16

Результирующая таблица по заданию представлена под номером 2.1.

Таблица 2.1 – Таблица по результатам

Модель	**Исходные изображения**	**Adversarial images ϵ=1/255**	**Adversarial images ϵ=5/255**	**Adversarial images ϵ=10/255**
ResNet50 – FGSM	loss: 0.6115 accuracy: 0.8799	loss: 2.2138 accuracy: 0.6949	loss: 7.1307 accuracy: 0.3269	loss: 9.3106 accuracy: 0.1560
ResNet50 – PGD	loss: 0.6115 accuracy: 0.8799	loss: 2.5462 accuracy: 0.6700	loss: 8.5482 accuracy: 0.3149	loss: 11.4082 accuracy: 0.2029
VGG16 – FGSM	loss: 0.2337 accuracy: 0.9399	loss: 0.8609 accuracy: 0.8460	loss: 3.2187 accuracy: 0.5260	loss: 4.8788 accuracy: 0.3039
VGG16 – PGD	loss: 0.2337 accuracy: 0.9399	loss: 1.0738 accuracy: 0.8299	loss: 5.3848 accuracy: 0.5460	loss: 11.01551 accuracy: 0.3549

Задание 3

В данном задании требуется применить целевую атаку уклонения на основе белого ящика против моделей глубокого обучения.

Используем изображения знака «Стоп» (label class 14) из тестового набора данных. Всего имеется 270 изображений. Применим атаку Projected Gradient Descent (PGD) на знак «Стоп» с целью классификации его как знака «Ограничение скорости 30» (target label class = 1). Переберем значения искажений ϵ = [1/255, 3/255, 5/255, 10/255, 20/255, 50/255, 80/255], и заполним таблицу 3.1 значениями точности классификации изображений знаков "Стоп" и "Ограничение скорости 30".

Повторим атаку методом FGSM, и заполним таблицу 3.1.

Таблица 3.1 – Таблица по результатам

Искажение	**PGD attack – Stop sign images**	**PGD attack – Speed Limit 30 sign images**	**FGSM attack – Stop sign images**	**FGSM attack – Speed Limit 30 sign images**
ϵ=1/255	loss: 0.1199 accuracy: 0.9703	loss: 0.0408 accuracy: 0.9888	loss: 0.2013 accuracy: 0.9555	loss: 0.0408 accuracy: 0.9888
ϵ=3/255	loss: 0.3491 accuracy: 0.9185	loss: 0.0408 accuracy: 0.9888	loss: 1.2275 accuracy: 0.7962	loss: 0.0408 accuracy: 0.9888
ϵ=5/255	loss: 0.7521 accuracy: 0.8777	loss: 0.0408 accuracy: 0.9888	loss: 2.0648 accuracy: 0.6444	loss: 0.0408 accuracy: 0.9888
ϵ=10/255	loss: 1.1215 accuracy: 0.8074	loss: 0.0408 accuracy: 0.9888	loss: 4.0395 accuracy: 0.2740	loss: 0.0408 accuracy: 0.9888
ϵ=20/255	loss: 4.2444 accuracy: 0.4481	loss: 0.0408 accuracy: 0.9888	loss: 5.4251 accuracy: 0.0481	loss: 0.0408 accuracy: 0.9888
ϵ=50/255	loss: 8.7468 accuracy: 0.0666	loss: 0.0408 accuracy: 0.9888	loss: 5.7564 accuracy: 0.0	loss: 0.0408 accuracy: 0.9888
ϵ=80/255	loss: 9.5345 accuracy: 0.0370	loss: 0.0408 accuracy: 0.9888	loss: 5.9050 accuracy: 0.0	loss: 0.0408 accuracy: 0.9888

Метод FGSM плохо подходит для целевых атак. С ростом искажения классификация начинает давать сбои. Оптимальным значением искажения является 10/255. При больших значения модель будет ошибаться всегда.

PGD отлично подходит для целевых атак. При больших искажениях, модель почти всегда будет определять заданный нами класс, но изображение станет слишком навязчиво искажено. Оптимальным значением искажения является 20/255.

Заключение

В ходе выполнения лабораторной работы были выполнены предоставленные задания, а именно:

– Подготовить 2 классификатора на основе глубоких нейронных сетей на датасете GTSRB;

– Применить нецелевую атаку уклонения на основе белого ящика против моделей глубокого обучения;

– Применить целевую атаку уклонения на основе белого ящика против моделей глубокого обучения.