

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

# РТУ МИРЭА

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

Кафедра КБ-4 «Интеллектуальные системы информационной

безопасности»

# Лабораторная работа №2

По дисциплине

“Анализ защищенности системы информационной

безопасности”

ГруппаББМО-01-22

Выполнил: Челышев Д.И.

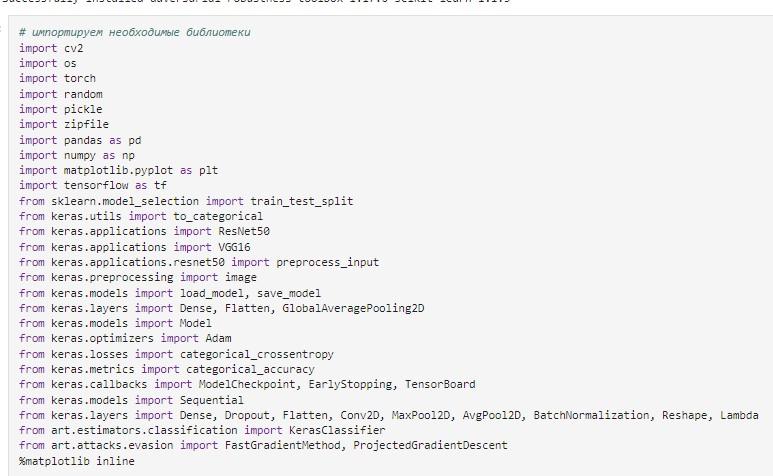
Проверил: Спирин А.А.

Москва 2023

# Задание 1



Импортируем необходимые библиотеки.



Далее нам нужно подключить гугл диск для возможности корректно и быстро работать с датасетом. Разархивируем датасет

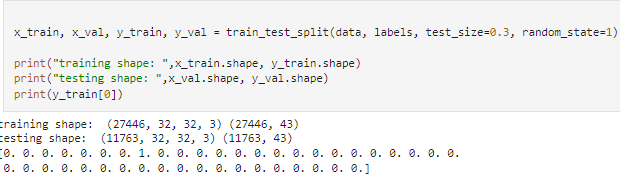
Прочитаем данные и выполним предварительную обработку изображений из тестового набора.

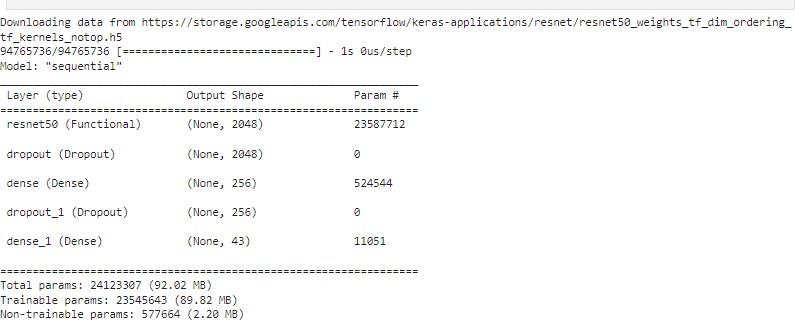
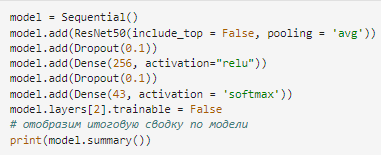


Отобразим первый элемент в виде картинки

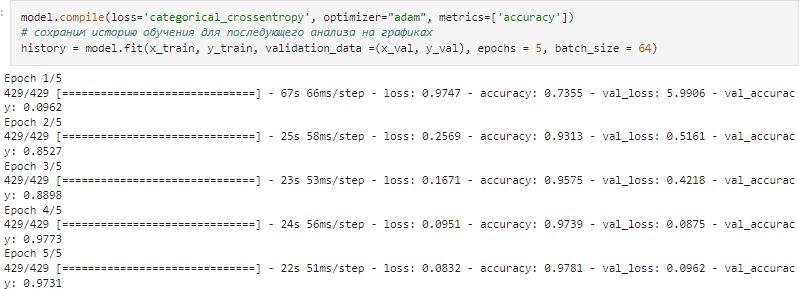


Далее выполняем разделение данных на тренировочный и тестовый наборы и отобразим размерности этих наборов



Создадим модель глубокого обучения ResNet50 для классификации изображений и отобразим сводку по ней

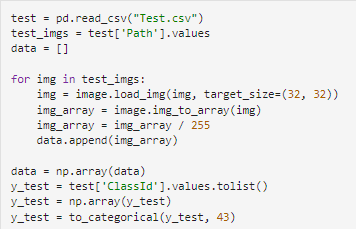
Обучаем модель в течение 5 эпох, используем оптимизатор Adam и функцию потерь categorical crossentropy

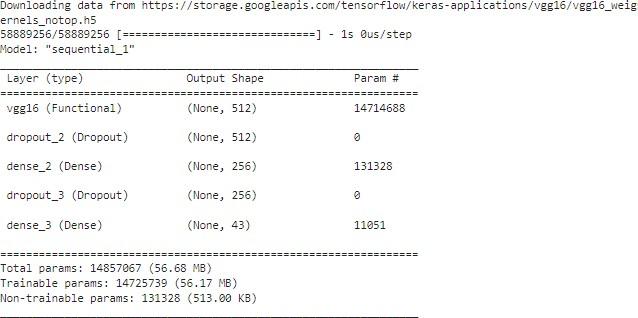
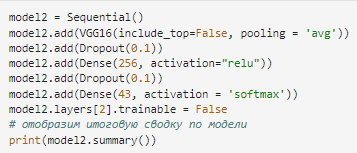


Сохраняем модель для последующего использования

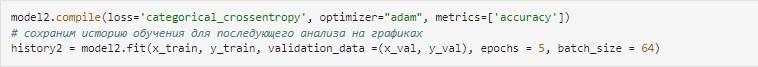


Считаем данные из csv в датафрейм, в ней содержится оригинальная метка класса и путь к изображению



По аналогии с предыдущей, создаем модель для классификации изображений (VGG16)

Обучаем модель в течение 5 эпох, используем оптимизатор Adam и функцию потерь categorical crossentropy

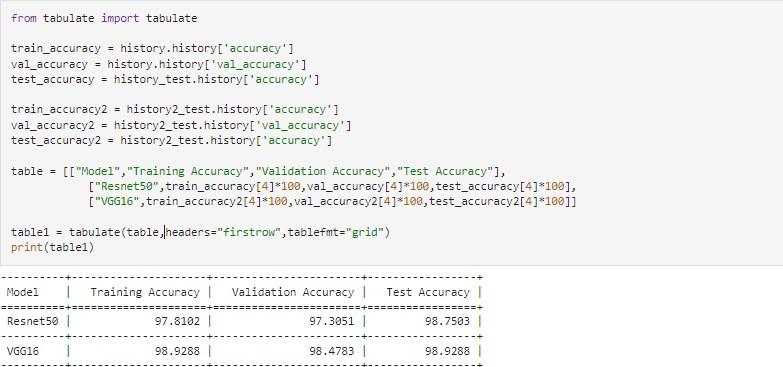


Сохраняем модель для последующего использования

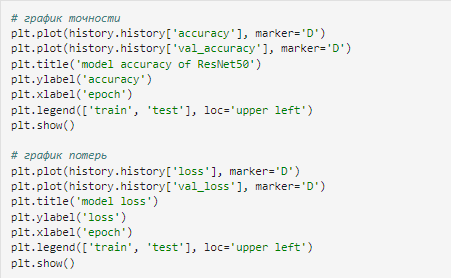


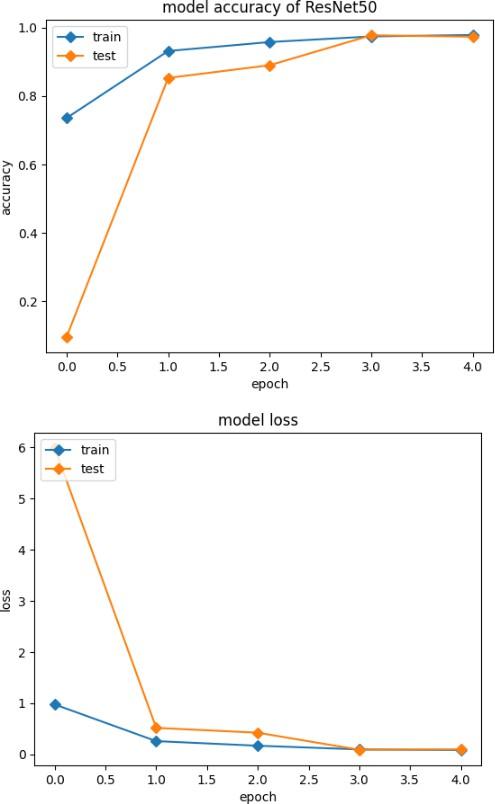
Выполним оценку производительности двух моделей на тестовом наборе данных

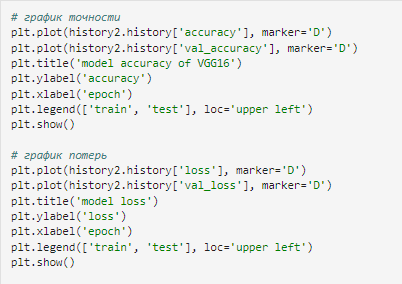


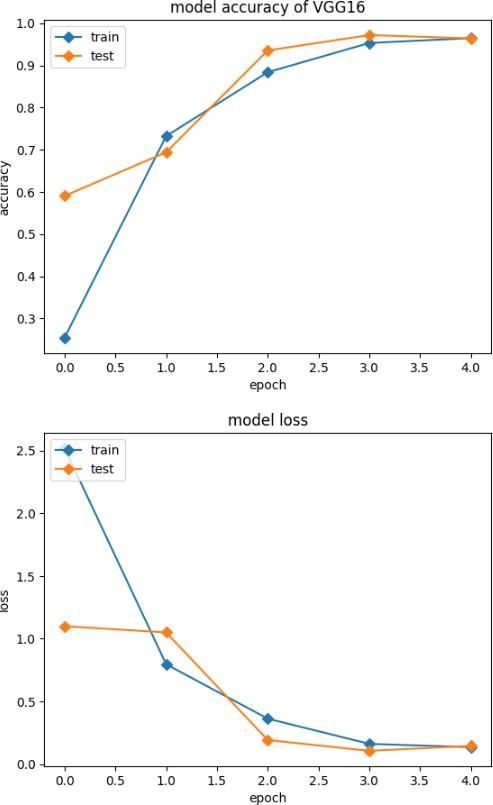
Создаем и выводим таблицу, которая показывает точность обоих моделей на треннировочном, валидационном и тестовом наборе данных

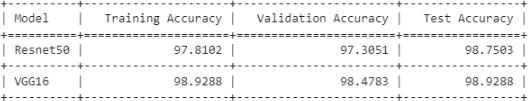
Строим два графика процесса обучения модели ResNet50 графики отражают зависимость метрики от эпохи для тренировочного и тестового наборов





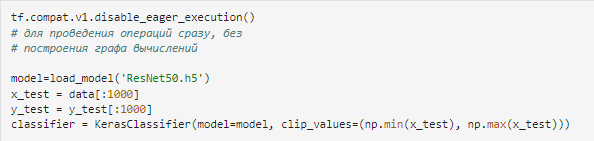
Визуализируем процесс обучения модели VGG16 аналогично с предыдущим



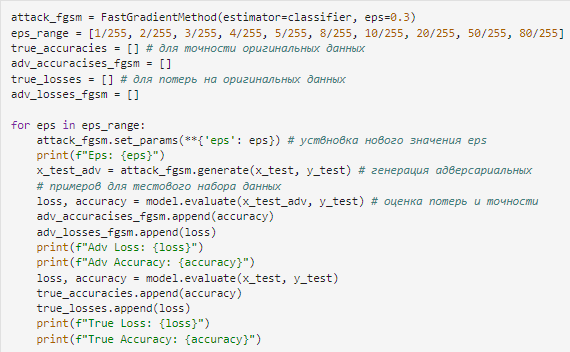


Как видно из таблицы, модели показывают приблизительно схожие результаты, немного лучше оказалась модель VGG16 (по валидации)

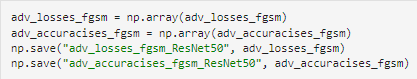
# Задание 2

Загрузим модель из предыдущего задания и берем тысячу первых элементов из тестового множества, создаем классификатор ART

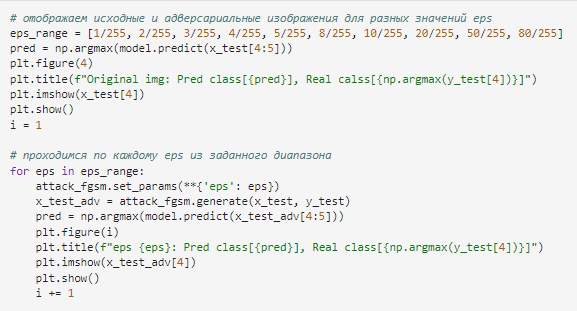
Создадим атаку FGSM. Проходимся по диапазону значений eps, который представляет размер шага, с которым FGSN изменяет оригинальные данные для создания адверсариальных параметров



Сохраним эту атаку для дальнейшего анализа.



Отобразим исходные и адверсариальные изображения для разных значений eps

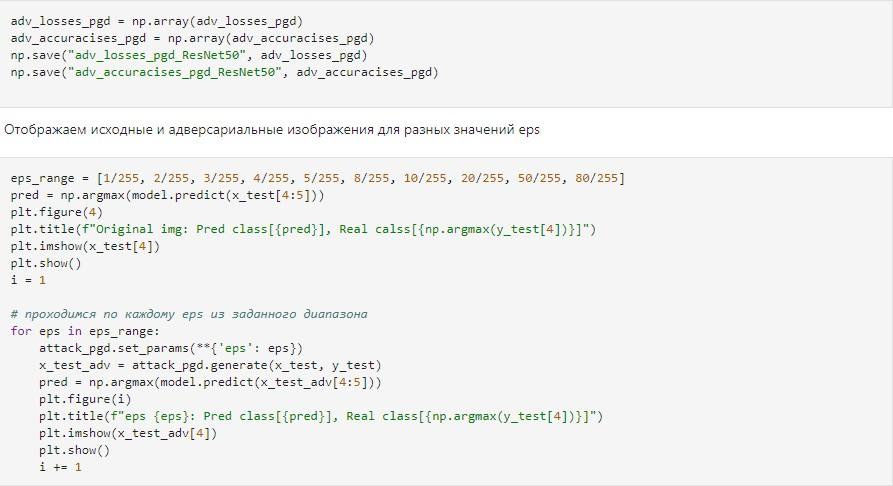




Как видно, ошибки предсказания из-за наложенного шума начались со значения 2/225.

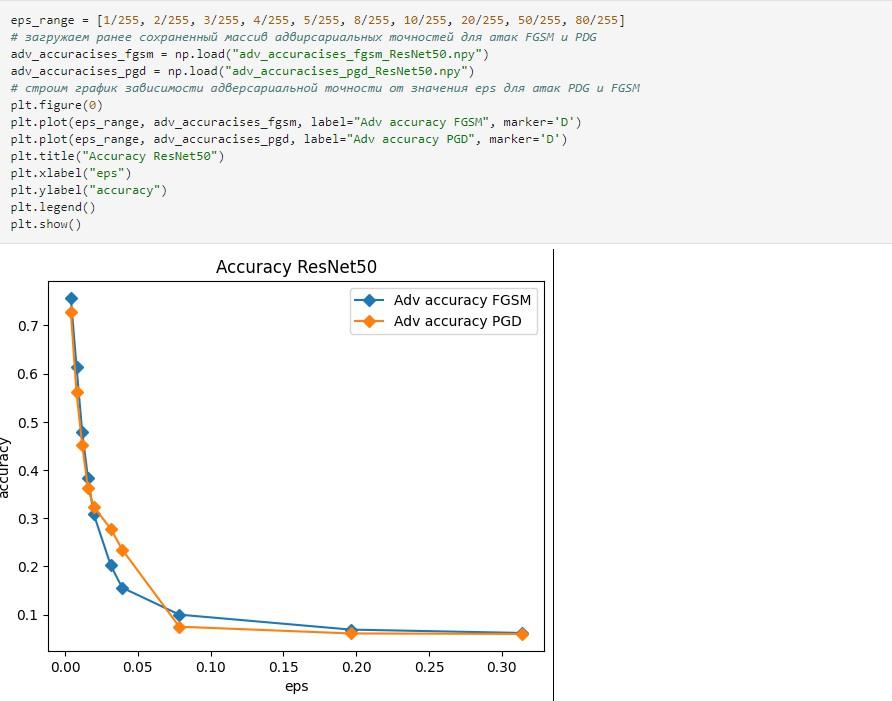
Теперь реализуем атаку PGD для той же модели, создаем атаку по аналогии с предыдущей

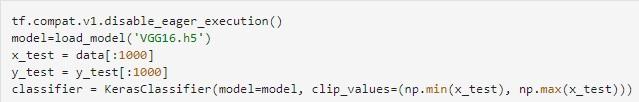






Предсказания стали ложными при параметре 2/255



Проделаем аналогичные атаки для VGG16 реализуем атаку FGSM для модели VGG16, которую сохраняли в первом задании

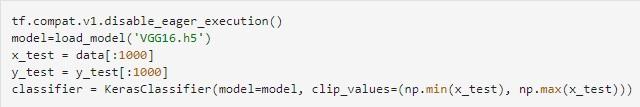
Создаем атаку FGSM по аналогии с VGG16



Сохраним атаку FGSM для дальнейшего анализа с помощью графика





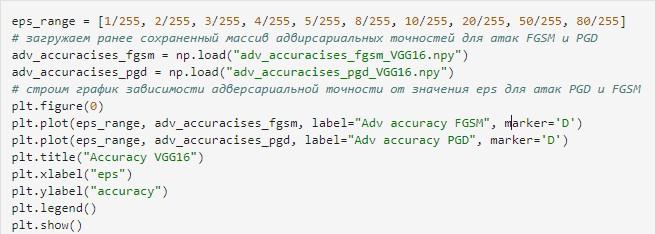
Был выдан ложный результат при значении eps 8/255. Реализуем атаку PGD для модели VGG16

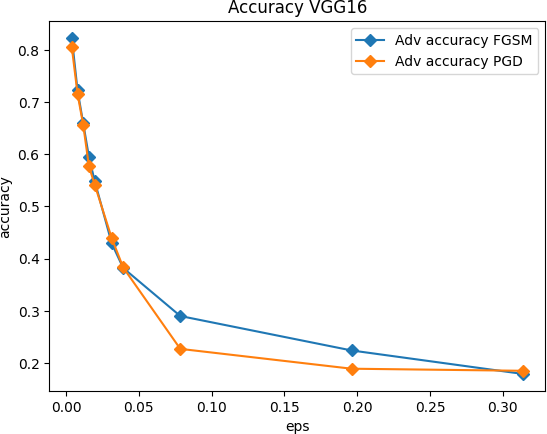
Создаем атаку PGD по аналогии с ResNet50



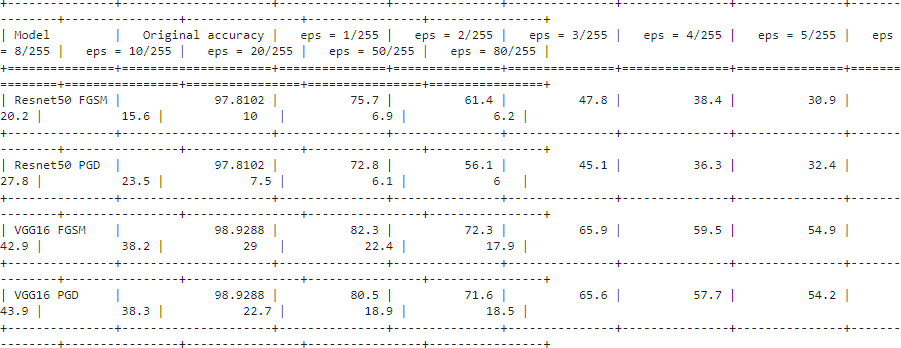


Ошибка предсказания произошла при значении eps 4/255.





В случае VGG6, при атаках PGD и FGSM точность сначала падает одинаково, но с повышением значений eps в какой-то момент точнотсть при атаке PGD начинает падать сильнее, но при максимальном значении eps точность сильнее всего упала при атаке FGSM.



Таким образом была отражена таблица для всех показанных атак и моделей. По таблице видно, что точность выше при всех значениях eps у модели VGG16.

# Задание 3

Создадим две целевых атаки

Загружаем тестовый набор данных из Test.csv и извлекаем изображения с меткой 14

Преобразуем изображения в массив чисел и нормализуем

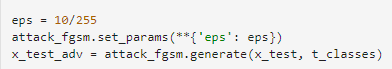


Реализуем целевую атаку FGSM

Сгенерируем адверсариальные примеры и оценим точность модели на них и на исходных тестовых данных

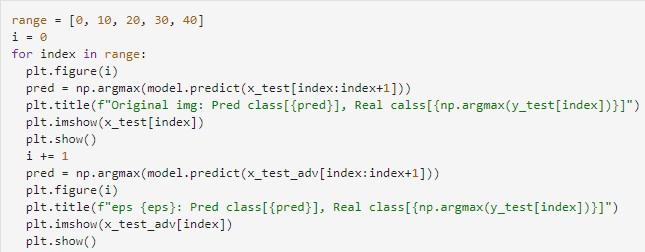


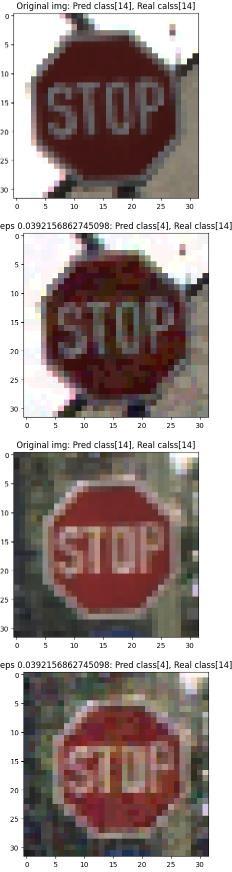
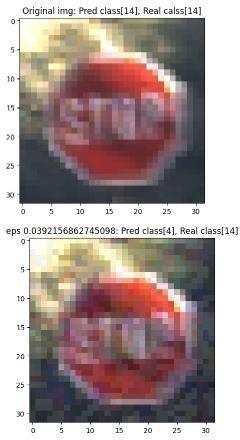
Тут эксперементируем со значениями eps для достижения наилучшего резултата.

Лучше всего себя показывает 10/255

**\**

Отобразим 5 разных изображений для визуализации действия атаки

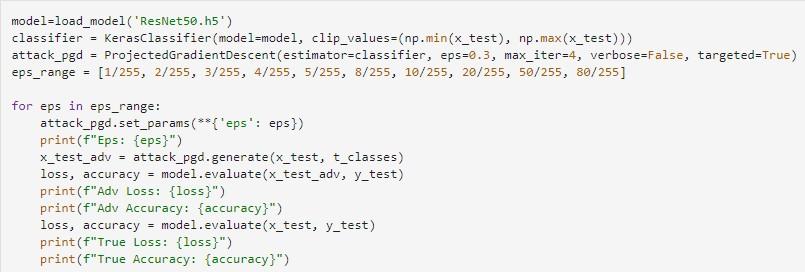


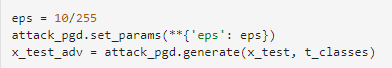
  

Целевая атака FGSM достигает своего пика на eps = 10/255 в нашем случе, при больших значениях eps атака хоть и будет давать больше неточности при предсказании, но это будут разные классы, в большинстве случаев отличные от первого (знак стоп), который мы указали. Можно сделать вывод, что FGSM не очень подходит для целевых атак.

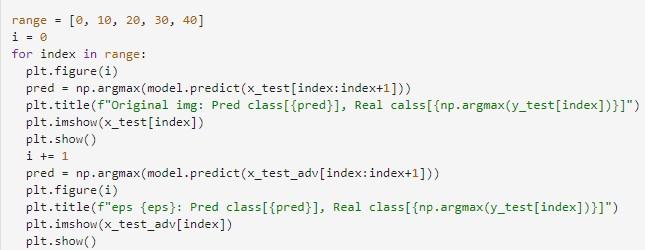
Реализуем целевую атаку PGD

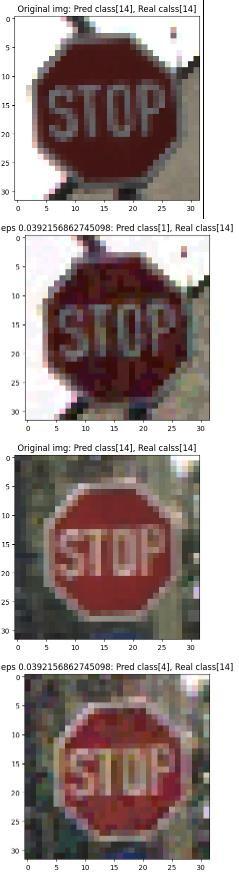
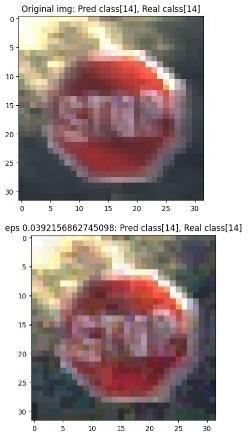
Сгенерируем адверсариальные примеры и оценим точность модели на адвирсариальных примерах и на исходных тестовых данных





Отобразим 5 разных изображений для визуализации действия атаки



Атака PGD достигает отличных значений при eps 50/255, при таком значении очень много требуемых результатов.

Как видим, атака PGD дольше сохранеяет точность, чем FGSM. При этом PGD намного лучше подходит для целевых атак, так как на больших значениях eps выдает лучший требуемый (класс 1 - знак стоп) результат, чем FGSM.