|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА –Российский технологический университет»**  **РТУ МИРЭА**  **Институт кибербезопасности и цифровых технологий**  **Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»** |
|  |

Дисциплина «Анализ защищенности систем искусственного интеллекта»

Отчет

о проделанной лабораторной работе №2

«Атаки уклонения»

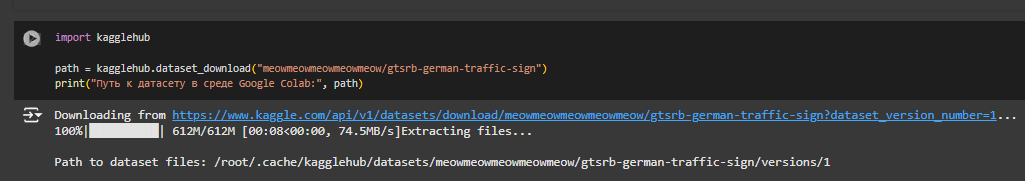
Выполнил студент 2 курса

Группы: ББМО-01-23

Чурсинов Герман Сергеевич

1. Сперва скачаем 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.

Поскольку датасет находится на сайте Kaggle, скачать его оттуда без смс и регистрации можно с помощью внешней библиотеки kagglehub:



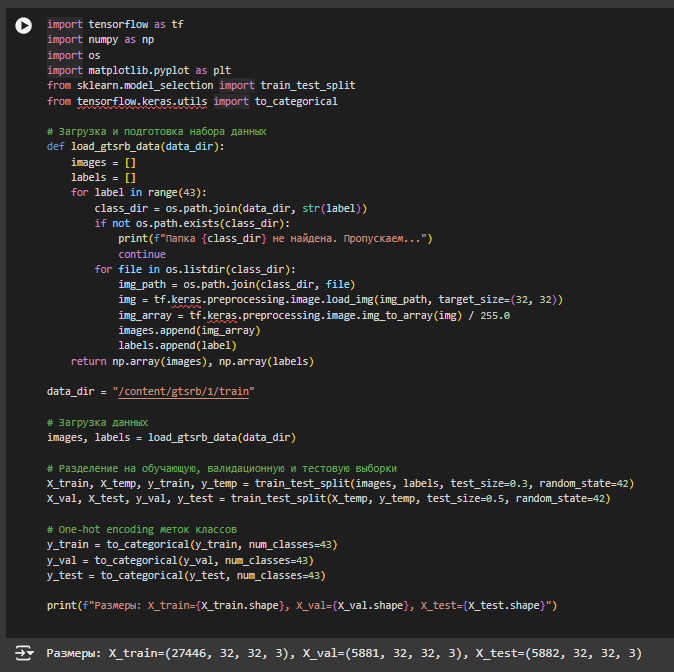
2. Переместим датасет



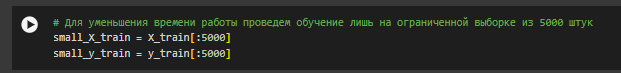
3. Чтобы вычисления были быстрее, включим Cuda-ядра в среде Google Colab на GPU



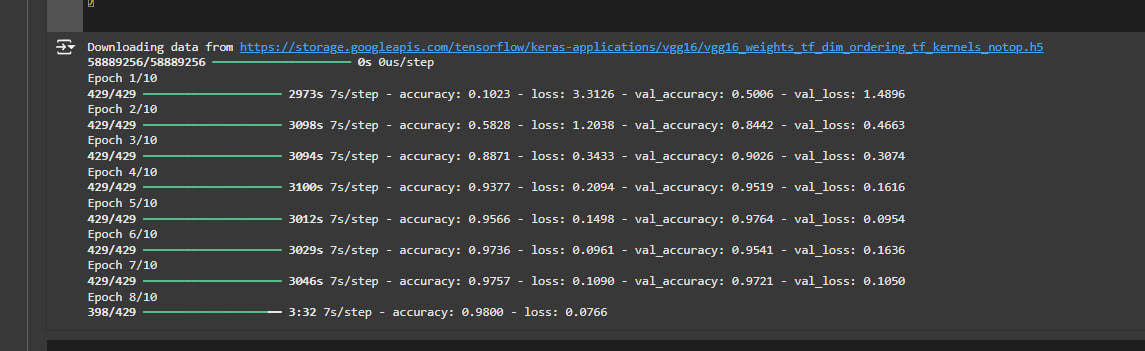
4. Выполнить импорт библиотек и всего необходимого для выполнения работы (датасет, и тд) и разделим датасет на выборки



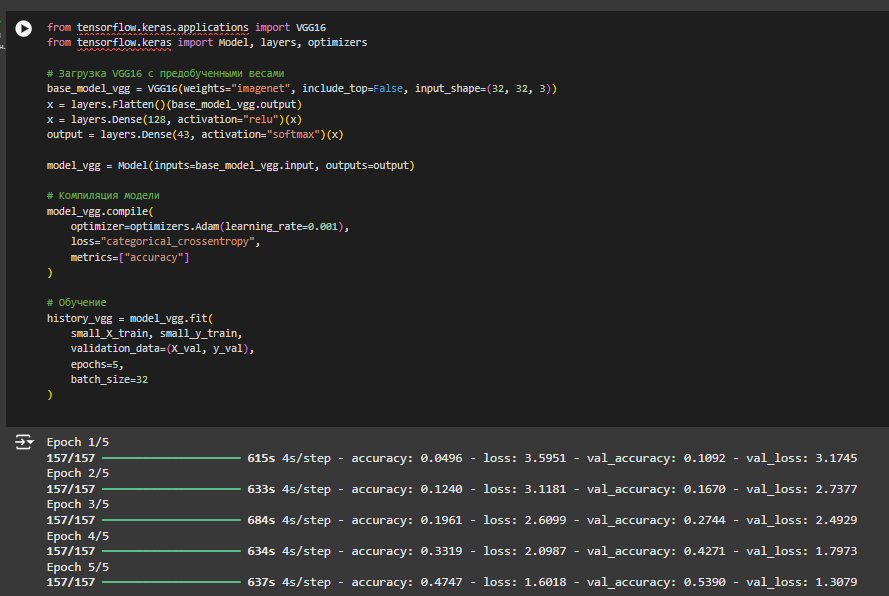
5. Поскольку работа выполнялась очень долго (по часу на каждую эпоху) было принято решение уменьшить выборку до 5000 штук, а также задать batch\_size = 32 и epochs = 5.



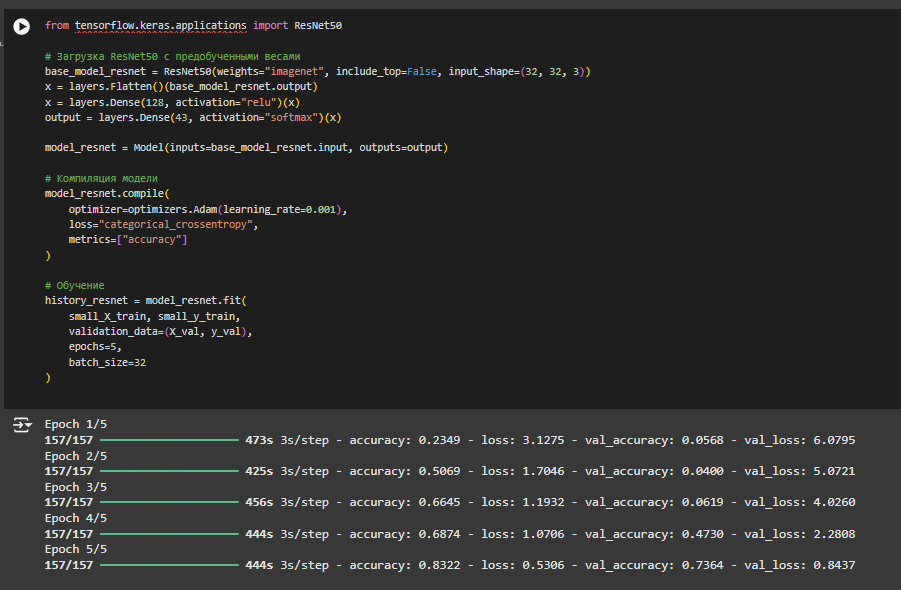
6. Обучаем первый классификатор на основе нейронной сети VGG16, будем использовать готовый фреймворк Tensorflow, а также в среде Google Colab уже автоматически используются предзагруженные весы:



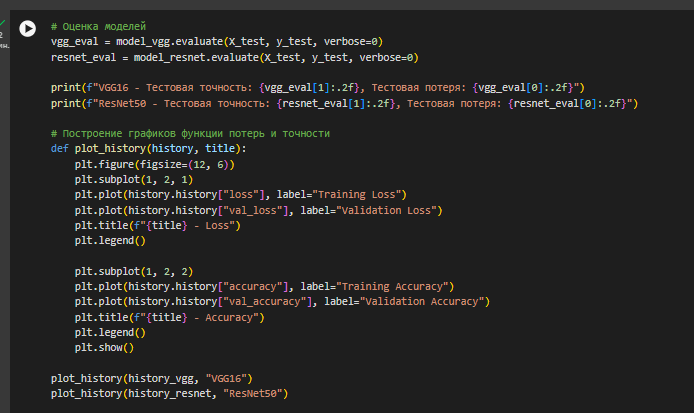
Тут я ссылаюсь на пятый пункт, поскольку в дальнейшем 10 эпох было заменено на 5, и размер выборки уменьшен, из-за чего каждая эпоха длилась всего около 600 секунд вместо 3000.



7. Делаем тоже самое, но второй классификатор обучаем с помощью ResNet50

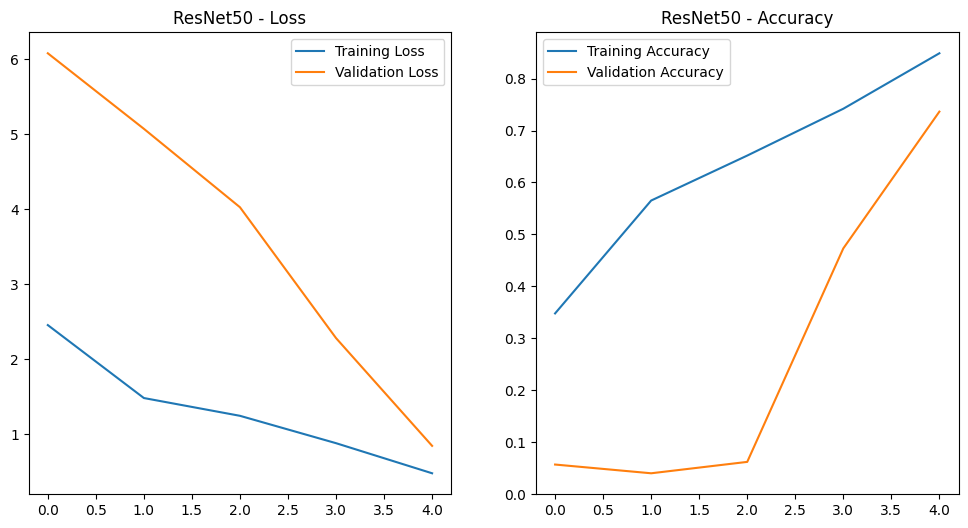


8. Теперь оцениваем модели:



И получаем вот такие результаты для VGG16 и ResNet50 соответственно.:

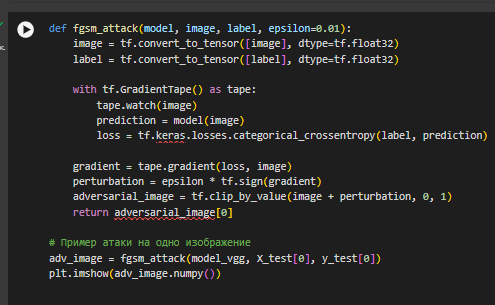


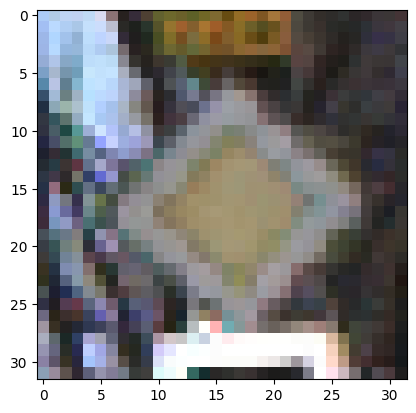


Строим таблицу по формуле: (Accuracy / Loss)

|  |  |  |  |
| --- | --- | --- | --- |
| **Модель** | **Обучение** | **Валидация** | **Тест** |
| VGG16 | 3.54659566 | 10.0227508 | 0.426356589 |
| ResNet50 | 10.3199153 | 1.02086097 | 0.83908046 |

10. После чего производим атаку:





11. Проверяем эффективность атаки:

