

Задача №7. Метрическое обучение



В задаче Вам предстоит познакомиться с современными инструментами для проведения экспериментов с машинным обучением, включая обучение, тестирование, сравнение и отслеживание качества их работы. Вам потребуется спроектировать и реализовать механизм вычисления точности классификации предварительно обученной модели метрического обучения.

Работа будет проводиться на наборе данных **Market-1501**. Он состоит из изображений людей, снятых с разных камер и разделен на 2 непересекающиеся по людям части: обучающую и тестовую. Набор подготовлен к обучению и избавлен от отвлекающих классов и мусора, его требуется только распаковать в папку `data`.

На обучающей части (с выделением из неё валидационной) была обучена некоторая нейронная сеть с функцией потерь ArcFace (`models/epoch_120.ckpt`). Требуется определить качество работы этой сети на изображениях людей, которые не входили в обучающую выборку.

Тестовая часть была также разделена на две: эталон (**reference**) – изображения с известным соответствием людям и запрос (**query**) – изображения, для которых это соответствие требуется определить. Для определения качества работы сети нужно оценить, насколько точно сеть, зная соответствия эталона, может предсказать соответствия изображениям запроса людям. Для определения соответствия требуется использовать все изображения из эталона, обучающая выборка в процессе тестирования участия не принимает.

Механизм вычисления точности классификатора требуется реализовать непосредственно в коде репозитория. Точность классификации должна логироваться в **MLflow** при запуске `python eval.py`. Можете использовать любые дополнительные библиотеки, хотя предоставленных достаточно для решения. Используемые библиотеки и шаблон репозитория указаны сверху страницы и хорошо документированы. Модифицировать и отправлять (без папок!) можно только файлы `src/models/metric_learning_module.py` и `src/datamodules/market_datamodule.py`.

Для проверки решения можете использовать `pytest`:

```
1 pytest tests/test_metric_learning.py
```