

Искусственный интеллект

2022/23 учебный год

Второй отборочный этап

Задача IV.1. Классификация видео (100 баллов)

Данная задача позволяла участникам освоить основы работы с видео как источником данных для анализа и адаптироваться к особенностям формата.

Задачи классификации, к которым относится и эта, являются важнейшим классом задач обучения с учителем (*supervised learning*). Задача классификации коротких видеофрагментов имеет широкое практическое применение, а успешные модели могут быть использованы, например, для фильтрации видеоконтента или тематической классификации видео, а также в задаче поиска (*information retrieval*).

Условие

В рамках задачи 2-го этапа участникам нужно было разработать алгоритм, который распределит короткие видео (< 1 минуты) на 9 заданных классов:

1. **animal** — в видео присутствует одно или несколько животных;
2. **car** — в видео присутствуют автомобили;
3. **cloud** — в видео показано небо в облаках (тучах);
4. **dance** — в видео есть танцующий человек (люди);
5. **fire** — в видео присутствует пламя (огонь, пламя свечи и т. д.);
6. **flower** — в видео присутствуют цветы;
7. **food** — в видео готовят еду, либо просто присутствует еда;
8. **sunset** — в видео показан закат или рассвет;
9. **water** — в видео показана поверхность воды.

Для работы участникам был предоставлен обучающий набор данных, который включает в себя 864 коротких видео и метки, соответствующие этим видео.

Критерии оценивания

Обученные модели участников запускались на тестовых данных, формат которых соответствует обучающей выборке, однако сами данные были скрыты от участников.

Тестовые данные представлены двумя наборами: **Public** и **Private test**, на которых была рассчитана метрика **accuracy**.

Метрика **accuracy** является стандартной метрикой для задач классификации. Она характеризует долю точных совпадений среди пар предсказанных и истинных ответов, то есть отражает отношение числа совпавших ответов (когда модель участника предсказала такой же ответ, как истинный) к общему числу ответов (независимо от того, в определении какого именно класса модель допустила ошибку). Эта метрика изменяется от 0 до 1, где 0 — наихудшее значение, 1 — наилучшее.

Метрика, рассчитанная на данных **Public test**, была доступна участникам в процессе соревнования на платформе. Финальное значение метрики было рассчитано на данных **Private test**, на основании которого был сформирован итоговый рейтинг

участников.

Результаты итогового рейтинга участников были переведены в стобалльную систему в соответствии с формулой, где участник, занявший первое место рейтинга, получал 100 баллов:

$$Total = \frac{Result \times (RankParticipants + 1 - RankPlace) \times 100}{MaxResult \times Participants},$$

где $Total$ — итоговый балл за решение задачи;

$Result$ — результат решения задачи;

$RankParticipants$ — общее количество участников в рейтинге;

$RankPlace$ — место участника в рейтинге;

$MaxResult$ — максимальный результат решения задачи среди всех участников;

$Participants$ — общее количество участников.

Решение

Базовое решение от разработчиков: <https://storage.yandexcloud.net/ds-ods/files/materials/93627f7d/Baseline.ipynb>.

Пример загружаемого на платформу файла: <https://storage.yandexcloud.net/ds-ods/files/materials/09d4f50a/submit.zip>.

В проверяющую систему было необходимо отправить код алгоритма, запакованный в ZIP-архив. Решения запускались в изолированном окружении при помощи Docker. Время и ресурсы во время тестирования были ограничены:

- 5Gb на архив с решением;
- 25 минут на работу решения.

Так как решение не имеет доступ к Интернету, все дополнительные данные, например, веса обученной модели, должны были быть подгружены в контейнер. В качестве примера организаторы предоставляли контейнер для запуска бейзлайн-решения.

Доступные ресурсы для проверки решения:

- 8 ядер CPU;
- 48Gb RAM;
- видеокарта NVidia Tesla V100.

Разбор базового решения от разработчиков: https://vk.com/video-165533067_456239151.