

Интерактивная сегментация

Андрей Москаленко, Андрей Стоцкий, Влад Шахуро



В данном задании необходимо реализовать модель интерактивной сегментации изображений. В качестве взаимодействий рассматриваются только клики: положительные (для добавления объектов), отрицательные (для удаления объектов). Каждое взаимодействие называется раундом, при добавлении новых кликов-взаимодействий качество (IoU) получаемой маски увеличивается.

Данные

Для обучения вам выдается датасет, являющийся объединением датасетов сегментации COCO и LVIS. Для скачивания данного датасета необходимо выполнить следующие команды:

```
mkdir -p COCO_LVIS/

wget -N https://github.com/saic-vul/ritm_interactive_segmentation/releases/download/v1.0/cocolvis_annotation.tar.gz
tar -xf cocolvis_annotation.tar.gz -C COCO_LVIS/

wget -N http://images.cocodataset.org/zips/train2017.zip
unzip -j train2017.zip -d COCO_LVIS/train/images/

wget -N http://images.cocodataset.org/zips/val2017.zip
unzip -j val2017.zip -d COCO_LVIS/val/images/
```

Скачайте все файлы из тестовой системы и разархивируйте их. После этого, у вас должна получиться следующая структура файлов:

```
├── COCO_LVIS/
│   ├── tests/
│   │   └── 00_test_davis_input/
│   ├── utils/
│   │   ├── datasets.py
│   │   ├── points_sampler.py
│   │   ├── sample.py
│   │   ├── clicker.py
│   │   └── misc.py
│   ├── run.py
│   └── solution.py
```

Шаблон

В шаблоне задания в качестве бейзлайна для вас уже реализована модель **DeepLabv3** с увеличенным числом каналов первой свертки для возможности передачи пользовательских кликов (в виде изображений с кругами фиксированного радиуса), а также маски сегментации прошлого раунда взаимодействий.

Базовая процедура обучения основана на модели **RITM**.

Для каждого объекта в обучающей выборке семплируется случайное число точек внутри маски (положительные клики) и снаружи (отрицательные клики). Затем модель выполняет итеративное предсказание маски случайно выбранное число раундов, где каждый новый клик производится в область с наибольшей ошибкой между предсказанной маской прошлого раунда и ground-truth.

При тестировании клики ставятся последовательно в центры областей с наибольшей ошибкой между предсказанной маской прошлого раунда и ground-truth. Центр области определяется с помощью distance-transform.

Тестирование

Вы можете проверить корректность своего решения используя выданный вам публичный тест на датасете **DAVIS**. Вы можете запустить автоматическую проверку на этом датасете запустив скрипт `python ./run.py tests`. После выполнения тестов, в папке `tests/00_test_davis_check/output` будут также сохранена визуализация результатов работы вашей сети.

Обратите внимание!

- Для обучения разрешается использовать **только** данные из выданного вам датасета **COCO_LVIS**. Другие источники данных использовать **нельзя**.
- В качестве предобученных весов разрешается использовать **только** веса моделей, предобученные на задаче классификации (например, на **ImageNet**). Веса моделей, предобученных на других задачах (в том числе, на задачах сегментации) использовать **нельзя**.

Финальное тестирование будет проводится на отдельном приватном тестовом датасете. Гарантируется, что приватный датасет имеет схожую структуру файлов, размеры изображений и т.п., как в публичном тесте с датасетом **DAVIS**, однако используемые изображения и маски из приватного датасета вам не доступны.

Для оценки вашего решения будет использована метрика **IoU@3** (качество сегментации после 3-х интерактивных кликов). Точность **IoU@3** на приватном тесте конвертируется в итоговый балл:

$\text{IoU@3} \geq 0.81$ – 10 баллов,

$\text{IoU@3} \geq 0.80$ – 9 баллов,

$\text{IoU@3} \geq 0.79$ – 8 баллов,

$\text{IoU@3} \geq 0.78$ – 7 баллов,

$\text{IoU@3} \geq 0.77$ – 6 баллов,

$\text{IoU@3} \geq 0.76$ – 5 баллов,

$\text{IoU@3} \geq 0.75$ – 4 балла,

$\text{IoU@3} \geq 0.74$ – 3 балла,

$\text{IoU@3} \geq 0.72$ – 2 балла,

$\text{IoU@3} \geq 0.70$ – 1 балл.

В проверяющую систему необходимо загрузить файл с обученной моделью `checkpoint.pt` и файл `solution.py` с реализованными функциями создания, обучения и тестирования модели и предсказания карты сегментации для изображения (классы `ISModel`, `Predictor` и функция `train_segmentation`).

Классы `ISModel` и `Predictor` будут использованы для автоматического тестирования в системе. Функция `train_segmentation` в системе запускаться не будет, однако она все равно должна присутствовать в вашем решении и должна позволять воспроизвести сданную в файле `checkpoint.pt` модель с небольшой погрешностью, связанной со случайностью в процессе обучения. Решения без функции обучения или с не рабочей функцией обучения приравниваются к плагиату.

Дальнейшие шаги по улучшению

1. Больше идей можно узнать в статьях [RITM](#), [SimpleClick](#), [Segment Anything](#) и т.д.
2. Попробовать другие архитектуры и функции потерь
3. Обучаться больше эпох, возможно, добавив scheduler
4. Добавить аугментации (например flip, zoom-in, resize), в том числе test-time аугментации
5. Вы также можете явно учитывать при обучении, что модель будет тестироваться на 3-х кликах