

ВАРИАНТ 1

Тестовое задание для кандидата на должность “Инженер – программист в отдел распознавания образов и обработки изображений”

Задача: необходимо произвести обучение модели нейронной сети для детектирования объектов с использованием фреймворка Darknet. Объекты могут принадлежать классу “человек”, “защитный жилет” или “каска”, при этом метрики точности должны быть выше минимальных значений: $MaP \geq 0.9$, $IoU \geq 0.7$.

Ссылка на фреймворк – <https://github.com/AlexeyAB/darknet>.

Обучение модели необходимо производить на linux-подобных системах. В качестве настойчивой рекомендации предлагается использование ОС Ubuntu.

Необходимые данные для выполнения задачи доступны по ссылке:

https://drive.google.com/file/d/11RVb4dFyiWoAT_vQHv2GBhevIQcIXWbV/view?usp=sharing

Метрики точности рассчитываются на закрытом тестовом датасете

Результатом выполнения описанной задачи является файл с параметрами модели (*.weights) и набор конфигурационных файлов к нему (*.cfg, *.data и *.names).

В качестве некоторой вспомогательной информации может рассматриваться обобщённый алгоритм обучения:

- 1) Сбор данных для формирования датасета;
- 2) Конвертирование датасета в необходимый для обучения формат. Например, [* .jpg, *.json] >> [* .jpg, *.txt];
- 3) Выбор архитектуры детектирующей сети. Например, Yolo_v3;
- 4) Подготовка конфигурационных файлов для обучения;
- 5) Обучение модели;
- 6) Выбор наиболее точной модели.

ВАРИАНТ 2

Тестовое задание для кандидата на должность “Инженер – программист в отдел распознавания образов и обработки изображений”

Задача: необходимо произвести обучение модели нейронной сети для детектирования объектов с использованием фреймворка Darknet. Объекты могут принадлежать классу “человек”, “защитный жилет” или “каска”, при этом метрики точности должны быть выше минимальных значений: $MaP \geq 0.9$, $IoU \geq 0.7$.

Ссылка на фреймворк – <https://github.com/AlexeyAB/darknet>.

Обучение модели необходимо производить на linux-подобных системах. В качестве настойчивой рекомендации предлагается использование ОС Ubuntu.

Необходимые данные для выполнения задачи доступны по ссылке: https://drive.google.com/file/d/11RVb4dFyiWoAT_vQHv2GBhevIQcIXWbV/view?usp=sharing
Метрики точности рассчитываются на закрытом тестовом датасете

Результатом выполнения описанной задачи является файл с параметрами модели (*.weights), набор конфигурационных файлов к нему (*.cfg, *.data и *.names) и файл *.py для обработки произвольного количества тестовых изображений.

Программа для вывода должна реализовывать следующий функционал:

- импорт необходимого функционала из библиотеки Darknet;
- загрузка обученной модели;
- возможность определять нижний порог вероятности принадлежности искомого объекта классу (т.н. thresh);
- последовательная обработка массива изображений из указанной директории;
- отрисовка рамок разного цвета (в соответствии с классом) на найденных объектах с указанием вероятности принадлежности классу;
- сохранение обработанных изображений в указанной директории.

Пример сигнатуры запуска программы для вывода модели:

```
python inference.py -images_dir images/ -output_dir output/ [-thresh 0.8]
```

В качестве некоторой вспомогательной информации может рассматриваться обобщённый алгоритм выполнения подобных задач:

- 1) Сбор данных для формирования датасета;
- 2) Конвертирование датасета в необходимый для обучения формат. Например, [* .jpg, *.json] >> [* .jpg, *.txt];
- 3) Выбор архитектуры детектирующей сети. Например, Yolo_v3;
- 4) Подготовка конфигурационных файлов для обучения;
- 5) Обучение модели;
- 6) Выбор наиболее точной модели;
- 7) Написание программы на языке Python (версии 2.7 и более свежие) для обработки произвольного количества изображений с использованием функционала библиотеки Darknet.