НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«КЛАССИФИКАТОР ИЗОБРАЖЕНИЙ ОВОЩЕЙ»

Выполнил студент

Павлов Сергей Владимирович

Научный руководитель:

Блуменау Марк

Москва, 2023 г.

Оглавление.

Глава 1. Постановка и описание задачи. Ожидаемые результаты.

1.1. Постановка задачи.

В рамках данной работы планируется использовать открытый набор данных с фотографиями овощей и выполнить следующие подзадачи:

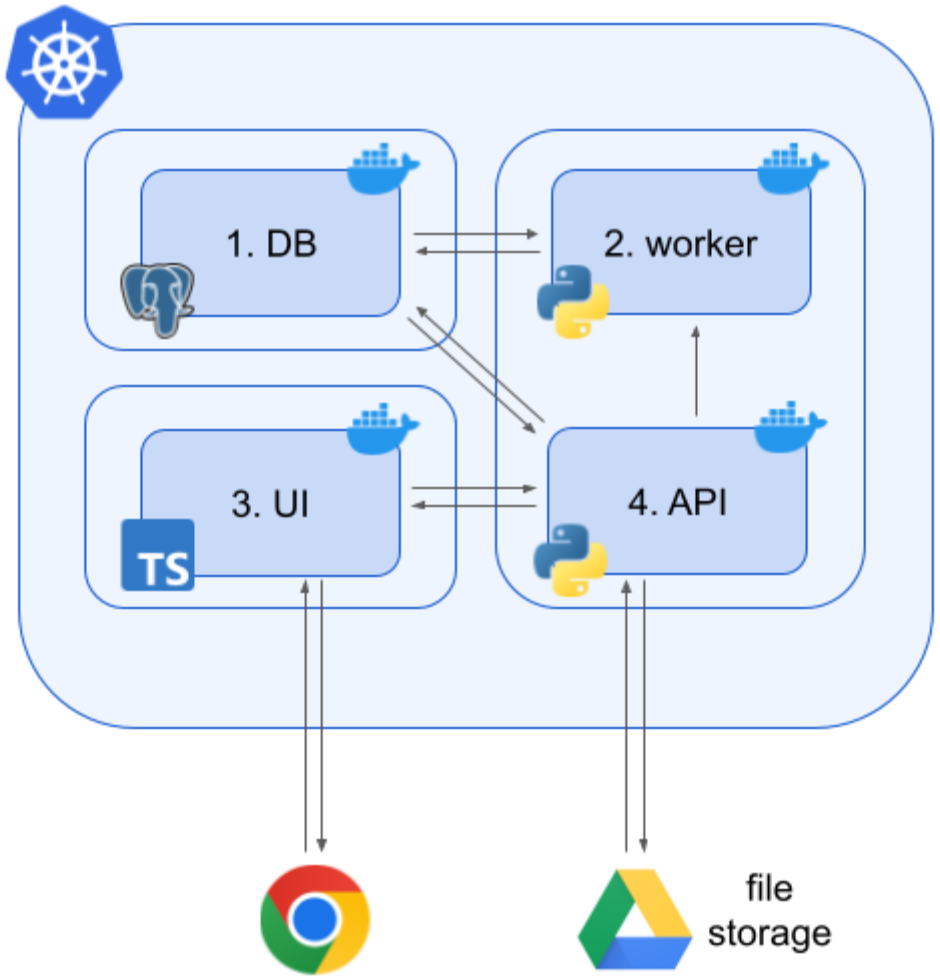
1. Решить задачу классификации овощей в двух измерениях:
   - 1.1. Вид овоща (огурец, помидор и т.д.).
     - 1.1.1. Для выбора архитектуры воспользоваться примерами успешных решений участников сообщества Kaggle.
     - 1.1.2. Адаптировать имплементацию архитектуры выбранной модели под стек, используемый в данной работе (Python, PyTorch), обучить модель, оценить качество.
     - 1.1.3. Подготовить сериализованный объект с моделью для инференса.
   - 1.2. Цвет овоща.
     - 1.2.1. По результатам разведочного анализа данных подготовить набор цветов для классификации и сделать разметку набора данных в соответствии с выбранным набором цветов.
     - 1.2.2. Самостоятельно разработать архитектуру модели и обучить её, оценить качество.
     - 1.2.3. Подготовить сериализованный объект с моделью для инференса.
2. Разработать API-сервис, позволяющий:
   - 2.1. Отправлять запрос с комбинацией параметров «количество + вид + цвет» и получать вывод изображений овощей, соответствующих предоставленной комбинации параметров (или визуализацию отсутствия подходящих изображений в базе).
   - 2.2. Загружать пользовательское изображение овоща и получать вывод его классификации по виду и цвету, предсказанной обученными моделями.
3. Разработать web-интерфейс сервиса для взаимодействия с API через браузер.
4. Выполнить развертывание разработанной связки API + UI на демонстрационном стенде с публичным сетевым адресом и провести живую презентацию работы сервиса.

1.2. Ожидаемые результаты.

В качестве результата классификации по виду овоща ожидается PyTorch модель с качеством по accuracy score не менее 0.97 на данных для валидации. Данная задача уже хорошо отработана коллегами из сообщества Kaggle, поэтому целесообразно воспользоваться лучшими архитектурами решений для получения хорошего качества.

В качестве результата классификации по цвету овоща ожидается PyTorch модель с качеством по accuracy score не менее 0.90 на данных для валидации. Для этой задачи базы наработанных решений нет, поэтому ожидаются подводные камни и более низкое качество.

В качестве результата разработки веб-сервиса ожидается приложение со следующей архитектурой:



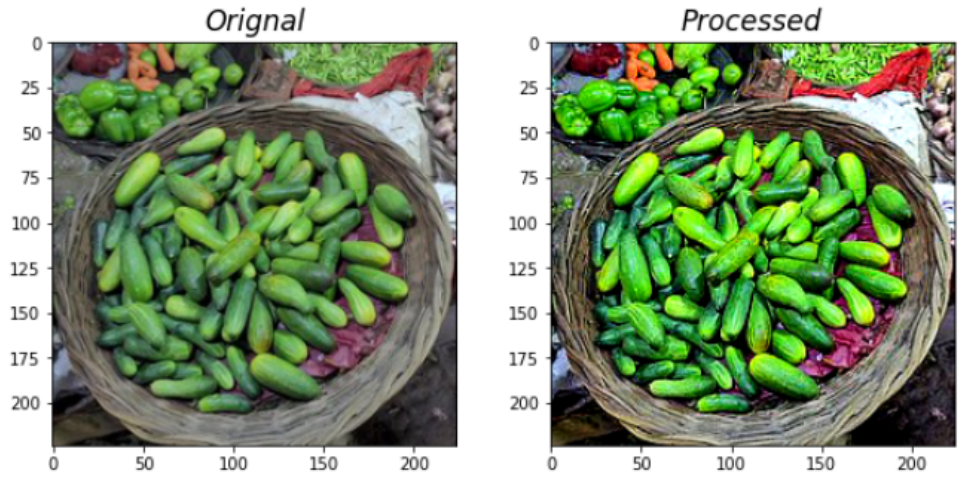
1. На схеме: «DB». База данных PostgreSQL для хранения метаинформации об изображениях из набора данных (имя / путь файла и классификация изображения).
2. На схеме: «worker». Вспомогательный Python-сервис для выполнения задач в фоновом режиме на базе Celery или аналогичного инструмента. В частности, для получения предсказаний от обученных моделей без задержки ответа основного API.
3. На схеме: «UI». Веб-сервер упакованных статических файлов для отрисовки пользовательского интерфейса в браузере. Исполнение кодовой базы UI на TypeScript с использованием одного из актуальных SPA-фрэймворков.
4. На схеме: «API». API-сервер на базе одного из актуальных веб-фрэймворков на Python, обеспечивающий ядро основного функционала.
5. Сборка образов при помощи GitHub CI/CD с последующим хранением артефактов в GitHub Container Registry (ghcr.io).
6. Развертывание в кластере Kubernetes в возможностью доступа к приложению через публичный сетевой адрес.

Глава 2. Краткий обзор существующих решений задачи классификации.

2.1. Пример 1. InceptionV3 transfer learning.

Решение с использованием модели InceptionV3, предобученной на ImageNet. В качестве предобработки исходного набора изображений автор применил следующие преобразования:

- повышение насыщенности цвета
- повышение контрастности
- повышение резкости



Поверх выходного слоя InceptionV3 были добавлены следующие слои:

- слой 2D average pooling
- полносвязный слой с функцией активации ReLU
- слой Dropout с коэффициентом 0.2
- выходной полносвязный слой для классификации с Softmax

Обучение производилось в 5 эпох. В качестве функции потерь была использована категориальная кросс-энтропия. В качестве оптимизационного алгоритма - Adam.

Полученная модель на наборе данных для валидации показала средний accuracy score в размере 0.992. Худшее значение среди всех классов - 0.97. Лучшее значение среди всех классов - 1.00.

2.2. Пример 2. EfficientNet-b0 transfer learning.

Решение с использованием модели EfficientNet-b0, предобученной на ImageNet. В качестве случайных аугментаций для исходного набора данных были использованы следующие преобразования:

- горизонтальное отражение изображения
- изменение высоты изображения с коэффициентом 0.2
- изменение ширины изображения с коэффициентом 0.2
- поворот изображения с коэффициентом 0.2 и заполнением пустоты ближайшим пикселем
- приближение / отдаление изображения с коэффициентом 0.2

Поверх выходного слоя EfficientNet-b0 были добавлены следующие слои:

- слой 2D average pooling
- выходной полносвязный слой для классификации с Softmax

В качестве функции потерь была использована категориальная кросс-энтропия. В качестве оптимизационного алгоритма - Adam.

Обучение производилось в 2 этапа по 5 эпох.

Сначала были использованы 20% от исходного набора данных для обучения. После этого шага модель на наборе данных для валидации показала средний accuracy score в размере 0.992.

На втором этапе веса из предыдущего обучения использовались в качестве начальных весов модели, а набор данных для обучения был уже полным (100% от исходного набора данных для обучения). Архитектура самой модели относительно предыдущего этапа осталась без изменений. Результатом второго этапа обучения стал средний accuracy score в размере 0.998 на наборе данных для валидации.

2.3. Пример 3. CNN.

Решение с использованием стандартного подхода - построения пользовательской архитектуры сверточной нейронной сети.

Преобразования исходного набора данных и случайные аугментации не использовались. Выбранная архитектура слоев нейронной сети выглядела следующим образом:

Тип слоя	Размерность выхода	Кол-во параметров
Conv2D	(None, 150, 150, 32)	896
MaxPooling2D	(None, 75, 75, 32)	0
Conv2D	(None, 75, 75, 64)	18496
MaxPooling2D	(None, 37, 37, 64)	0
Flatten	(None, 87616)	0
Dense / Linear	(None, 128)	11214976
Dropout	(None, 128)	0
Dense / Linear	(None, 128)	16512
Dense / Linear	(None, 15)	1935

Обучение производилось в 100 эпох с условием остановки при отсутствии улучшения качества модели на протяжении 5 эпох. В качестве функции потерь была использована категориальная кросс-энтропия. В качестве оптимизационного алгоритма - Adam.

Обучение длилось 15 эпох. Полученная модель на наборе данных для тестирования показала средний accuracy score в размере 0.955.

Глава 3. Разведочный анализ данных.

3.1. Примеры классов.

Примеры изображений каждого класса из набора данных для обучения:

3.2. Обзор данных.

Исходный набор данных состоит из изображений 21000 изображений 15 видов овощей. Классы сбалансированы ровно, то есть на каждый вид овоща приходится всего 1400 изображений. Исходный набор данных изначально разделен на группы для обучения, тестирования и валидации (train, test, validation) в отношении 70% / 15% / 15%. Это 15000 / 3000 / 3000 изображений соответственно (1000 / 200 / 200 на каждый класс).

Ручной просмотр изображений не позволил выявить откровенные выбросы, требующие исключения из обучения. Но точноимеют место угловые кейсы, когда несколько разных видов овощей попадаются вместе на одной фотографии, что может привести к ошибкам классификаторов. Примеры: