Отчет о контесте ImageAnalysis.BarcodesClassification.

Борисов Дмитрий, гр. 394в.

- ***Background***: до этого в подобных контестах не участвовал. Знания в машинном обучении минимальны. Исходя из этого, пришлось выкручиваться подручными средствами, немного тривиальными способами.
- ***Библиотеки***: я решил использовать библиотеку skimage, так как с ней работать оказалось проще, чем с библиотекой OpenCV. Но базовый функционал этих библиотек вроде бы совпадает. О библиотеке skimage и её функциях узнал из курса “Обработка изображений” на Степике(https://stepik.org/course/1280).
- ***Закономерности в классах***:
    1. Класс №0 – схожая центральная часть;
    2. Класс №1 – вертикальные полосы, присутствуют буквы/цифры/символы;
    3. Класс №2 – вертикальные полосы;
    4. Класс №3 – на первый взгляд ничего особо отличительного нет, но правая и левая границы изображений из класса совпадают;
    5. Класс №4 – изображения имеют схожую структуру: 3 одинаковых квадрата в углах;
    6. Класс №5 – сканы рукописного/печатного текста.
- ***Замеченные особенности при решении:***
    1. Изображения имеют разные размеры и различные типы данных. Размер изображений особенных проблем не создает, кроме увеличения время работы алгоритмов. А различные типы данных изображений привносят проблемы. В итоге было решено приводить все изображения к типу uint8, используя функцию img_as_ubyte. Попытки приведения изображений к типу float функцией img_as_float только увеличивали время работы;
    2. Класс №0 – схожая центральная часть штрихкода. Выделим шаблон для классификации к классу №0. Будем искать этот шаблон, изображение поворачивать не нужно;
    3. Класс №4 – 3 одинаковых квадрата в углах штрихкодов. Выделим из них шаблон. Будем искать этот шаблон, изображение поворачивать не нужно;
    4. Класс №2 – штрихкоды состоят из вертикальных полос, используем градиенты. Замечаем, что вертикальный градиент равен нулю везде. Обрежем верх и низ изображения по 5%, так как в тестовой выборке есть артефакты в виде горизонтальных черных полос на границах. Будем использовать такую ориентацию изображения, чтобы ширина была больше высоты. Остальных поворотов и отображений производить не нужно;
    5. Класс №3 – хотелось выделить какую-то отличительную закономерность, но не получилось. Поэтому замечаем, что поля штрихкода справа и слева сохраняются, но они различны. Обрезаем верх и низ для ускорения поиска. Будем искать, по крайней мере, 6 совпадений полей из одних и тех же строк. Так же нужно учесть поворот изображения на 180 градусов. Узнать о повороте изнутри программы автоматически тяжело, поэтому будем дополнительно проверять на совпадение шаблоны, повернутые на 180 градусов;
    6. Класс №1 – есть буквы/цифры. Была идея составить из них множество, а потом искать эти символы в изображениях тестовой выборки, но это оказалось сложным делом (в конце ноутбука-решения есть зачатки алгоритма обнаружения символов). Но на штрихкодах есть вертикальные линии. Тогда обрежем пустое место и символы, получим вертикальные линии, и опять повторим алгоритм для обнаружения класса №2. Но проблема в том, чтобы не обрезать лишнее. Обрезав слишком много, я получил срабатывание функции is_class_1 на одном изображении обучающей выборки из класса №5. Поэтому нужно обрезать у изображения со стороны символов больший кусок. Но мы точно не сможем определить этот случай, поэтому если на входном изображении не получили положительный ответ, то поворачиваем изображение на 180 градусов и повторяем проверку.
- ***Итого имеем последовательную проверку на принадлежность классам, проверка начинается слева направо до первого верного срабатывания:*** **0 → 4 → 2 → 3 → 1 → 5.**