

Обработка и распознавание изображений, 2021 год

Григорьев Илья, 317 группа

Лабораторная работа №1

Отчет

Постановка задачи

Задача заключалась в распознавании на изображении фишек тримино, определении их местоположения и их классификации по количеству точек, нанесенных на них. Для решения этой задачи можно было применять разные методы, в том числе и нейросетевые. Но надо учесть очень ограниченный набор данных, которого даже при сильной аугментации не хватит для обучения сверточной нейронной сети. Так же важна скорость работы алгоритма: недопустимо, чтобы программа работала ощутимо долго для распознавания одного изображения. Представленная программа очень эффективна, время работы на одном изображении из тестового датасета не превышает 10 миллисекунд.

Описание данных

Мной были выбраны самые сложные данные уровня Expert (изображения Pict_4_1.bmp и Pict_4_2.bmp). Сложность этих изображений заключается в том, что на картинке расположено много фишек, при этом они располагаются на пестром фоне с множеством других объектов. То есть появляется дополнительная сложность отличить объекты на клеенке от фишек тримино. Так же ко всему этому на картинках неоднородное освещение, и камера расположена достаточно далеко от стола с объектами.

Описание метода решения

Для решения данной задачи был сделан выбор в пользу классических алгоритмов компьютерного зрения, так для обучения сверточной нейронной сети нужно больше данных. Первая часть программы направлена на поиск самих фишек на изображении и определение их центра. Был применен алгоритм выделения границ Canny. Он состоит из нескольких шагов: сначала применяется фильтр Гаусса для размытия изображения, потом горизонтальный и вертикальный оператор Собеля для детектирования границ, потом удаляются пиксели, не относящиеся к ребрам, и на последнем шаге оставляются только границы, удовлетворяющие условию с двумя пороговыми значениями. Эти пороги являются параметрами алгоритма, и были подобраны для достижения наилучшего качества. После детектирования границ были применены операции замыкания с разными структурирующими элементами: крест размера 5 на 5 и эллипс размера 5 на 5. После этого был применен алгоритм поиска контуров на бинаризованном изображении, его суть заключается в объединении найденных границ в связанные контуры. Далее контуры были

преобразованы алгоритмом Ramer–Douglas–Peucker для уменьшения числа вершин с заданной погрешностью, которая подбиралась с точки зрения максимизации качества алгоритма. Далее был произведен поиск треугольников и вычисление их центров масс.

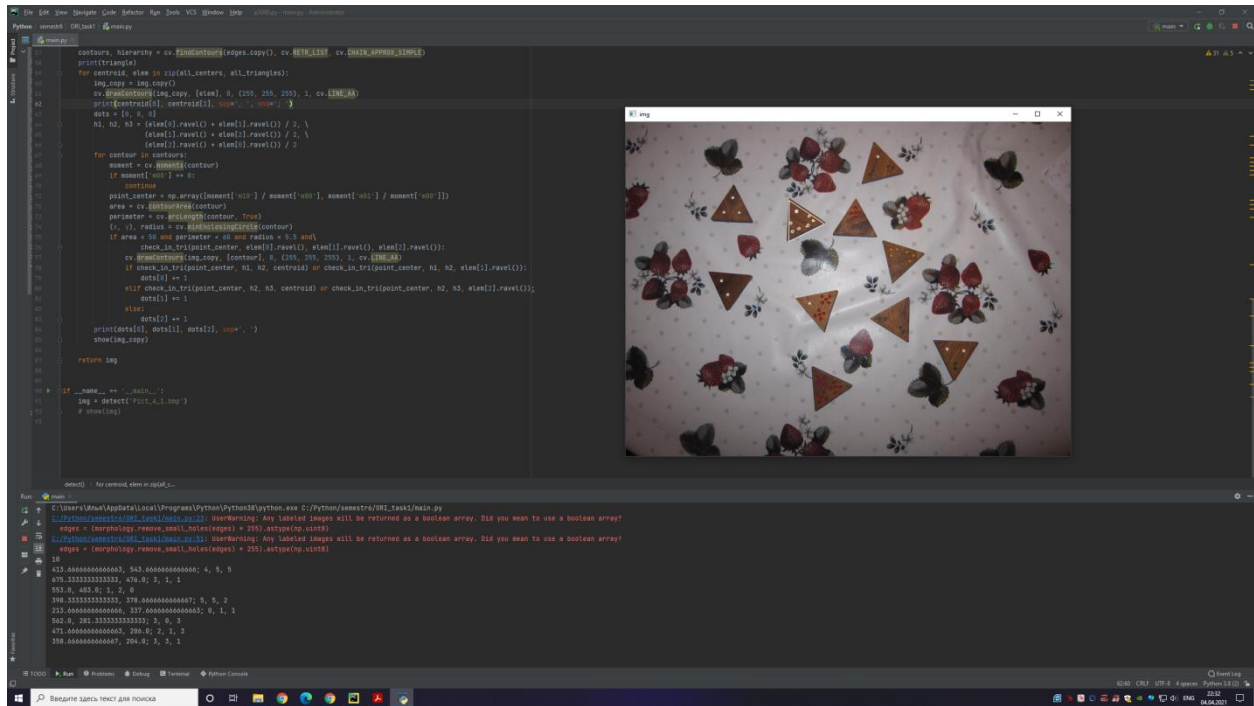
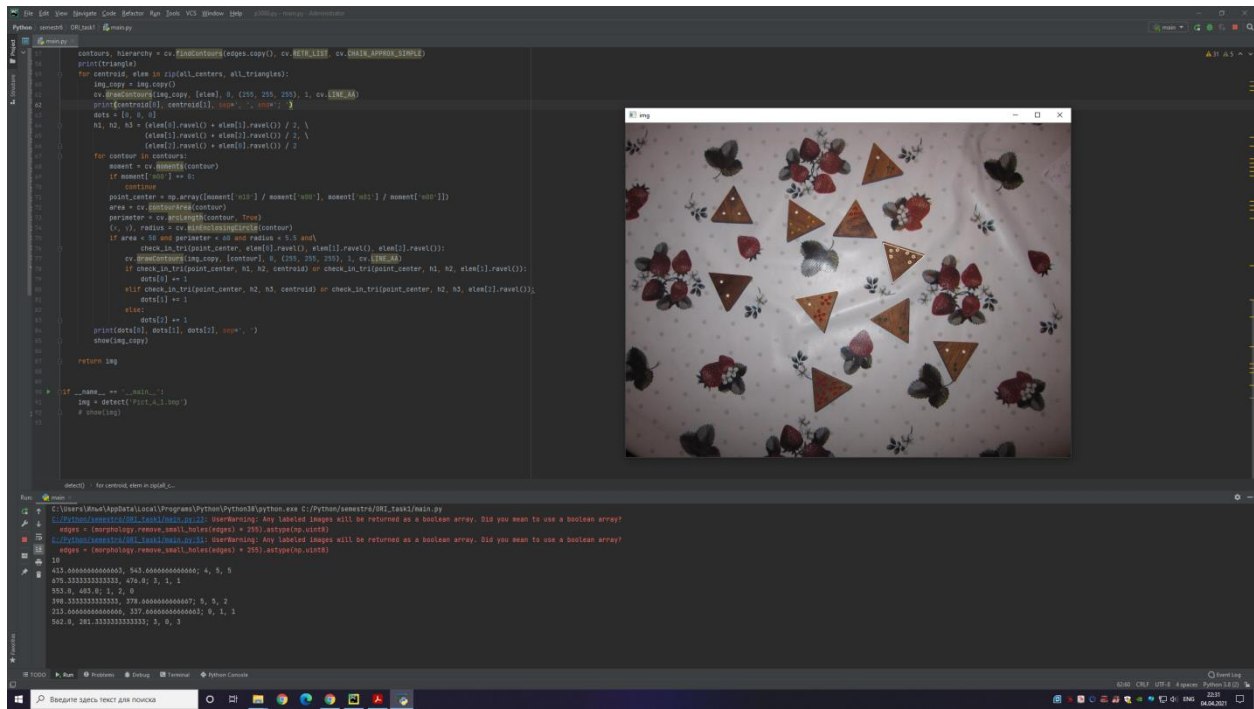
Вторая часть программы ищет точки внутри фишек и производит классификацию фишек. В каждом углу фишки может быть от нуля до пяти точек, при этом они разного цвета. Цвет определять не требовалось, но надо заметить, что какие-то цвета определяются алгоритмом лучше, а какие-то хуже. Для данной подзадачи применялся аналогичный предыдущей подзадаче пайплайн преобразований, но все методы имеют другие параметры, так как надо детектировать уже не треугольники, а точки. После нахождения точек программа проверяет их принадлежность какой-то фишке, а далее вычисляет в какой именно области треугольника лежит данная точка и формирует класс данной фишки. Программа дополнительно выводит промежуточные изображения с обведенной фишкой и найденными точками внутри нее.

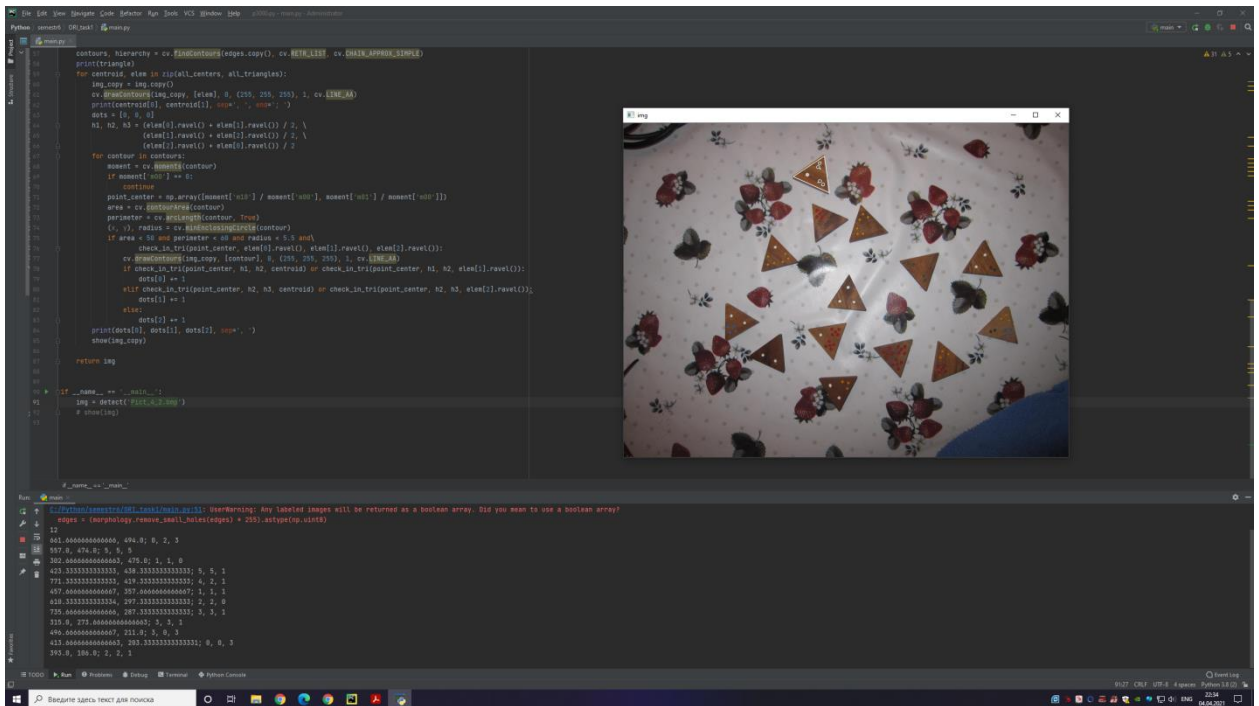
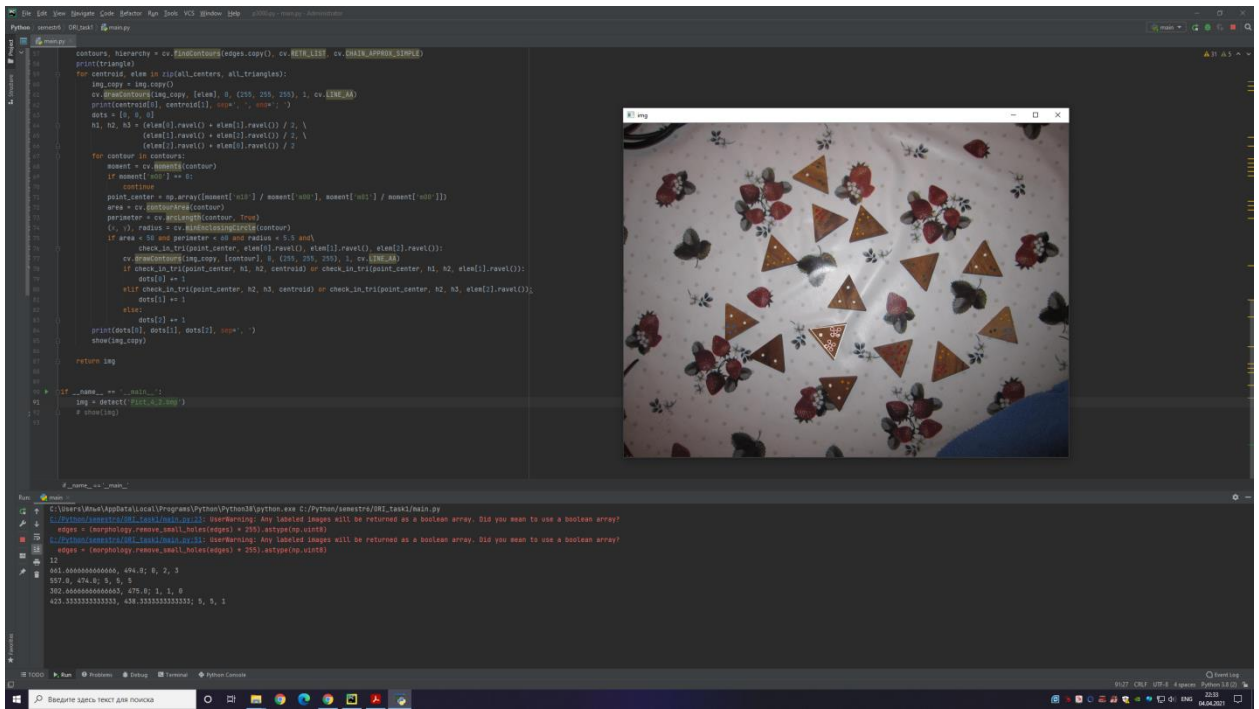
Описание программой реализации

Для программной реализации был выбран язык программирования Python 3 благодаря наличию огромного количества библиотек для машинного обучения и компьютерного зрения. Из библиотек использовались: Numpy (для линейной алгебры), OpenCV (алгоритмы компьютерного зрения), Scikit-image (алгоритмы компьютерного зрения). Алгоритм детектирования границ Canny и алгоритм нахождения контуров были взяты из OpenCV. Вывод программы производится в stdout, для показа промежуточных изображений создается дополнительное окно.

Эксперименты

Эксперименты были произведены с изображениями Pict_4_1.bmp и Pict_4_2.bmp. Программа показала очень высокое качество работы. Были детектированы все фишки, их центры определяются почти без погрешности. При классификации фишек программа допускает ошибки не больше, чем на 10% фишек. Подробности экспериментов и работоспособность программы можно посмотреть на скриншотах.





Выводы

В данном задании было испробовано большое количество современных алгоритмов компьютерного зрения. Высокое качество и скорость работы, а так же отсутствие необходимости обучать алгоритм еще раз доказывают актуальность не нейросетевых подходов компьютерного зрения наравне с нейросетевыми подходами.