Отчет о лабораторной работе №1

Обработка и распознавание изображений, ММП ВМК МГУ.

Аристархов Данила Дмитриевич.

Март 2024.

Содержание

1 Постановка задачи

Необходимо разработать и реализовать программу для работы с фотографиями «Кладбища самолётов». Программа должна производить бинаризацию изображения и оценку количества самолетов. Также необходимо разработать пользовательский интерфейс для работы с программой, обеспечивающий выбор изображения, выполнение операций преобразования, визуализацию работы программы, выдачу результатов подсчёта самолётов.

2 Описание данных

Данные представляют из себя растровые изображения, на которых изображены «Кладбища самолётов»: множество различных самолетов, расположенных на фоне земли.

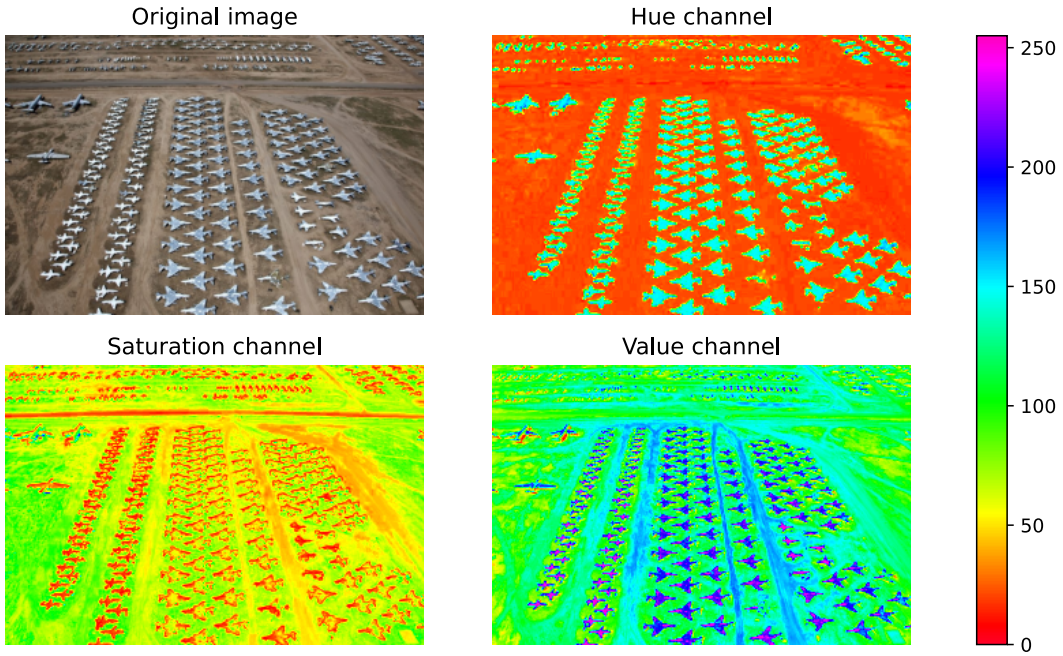


Рис. 1: Разложение изображения по параметрам HSV

3 Описание метода решения

Решение данной задачи происходит в несколько этапов:

1. Сначала изображение переводится в HSV (Hue, Saturation, Value) представление. В таком формате проще сегментировать изображение.

2. Далее проводится бинаризация изображение по HSV параметрам. Данные параметры подбирались путем анализа изображений по отдельным компонентам (см. рис. 1), а также путем экспериментов. Оптимальные значения могут отличаться в зависимости от изображения. Для данной задачи были выбраны следующие пороги:

   (a) Hue $\in [50, 140]$

   (b) Saturation $\in [0, 255]$

   (c) Value $\in [150, 255]$

Замечание: Все параметры HSV были приведены к значениям $[0, 255]$.

3. Для бинаризованной маски проводится подсчет связных компонент.

4. Далее из маски удаляются связные компоненты размером менее заданного порога. Данный порог также выбирается в зависимости от изображения и определяет чувствительность алгоритма к шума. Для данной задачи порог был выбран равным 30 пикселям.

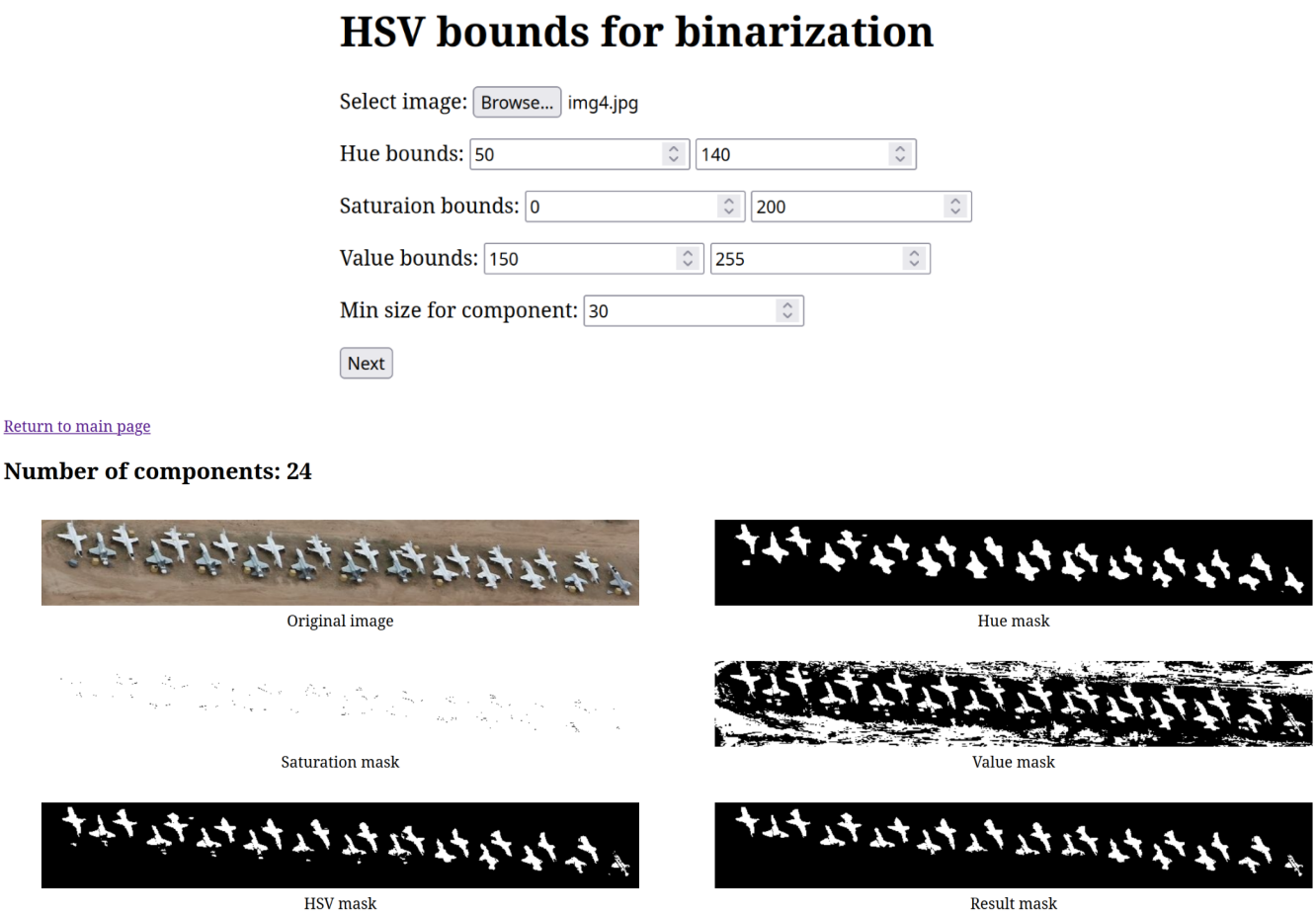


Рис. 2: Этапы работы программы

4 Описание программой реализации

Программа была написана на языке программирования `Python`. Для работы с изображениями использовалась библиотека `OpenCV`. Интерфейс был реализован в виде веб-сервера с помощью библиотеки `Flask`. Для удобство решение было обернуто в docker-контейнер, однако возможна установка программы и всех зависимостей вручную.

Работа с программой происходит следующим образом (см. рис. 2):

1. Выбирается изображение для обработки.

2. Задаются значения порогов бинаризации или оставляются параметры по умолчанию.

3. Задается минимальный размер для связной компоненты.

Далее происходит выдача результата в виде числа связных компонент, а также происходит визуализация всех этапов работы программы: маски бинаризации изображения по отдельным каналам, маска по всем каналам и итоговая маска с удаленными компонентами.

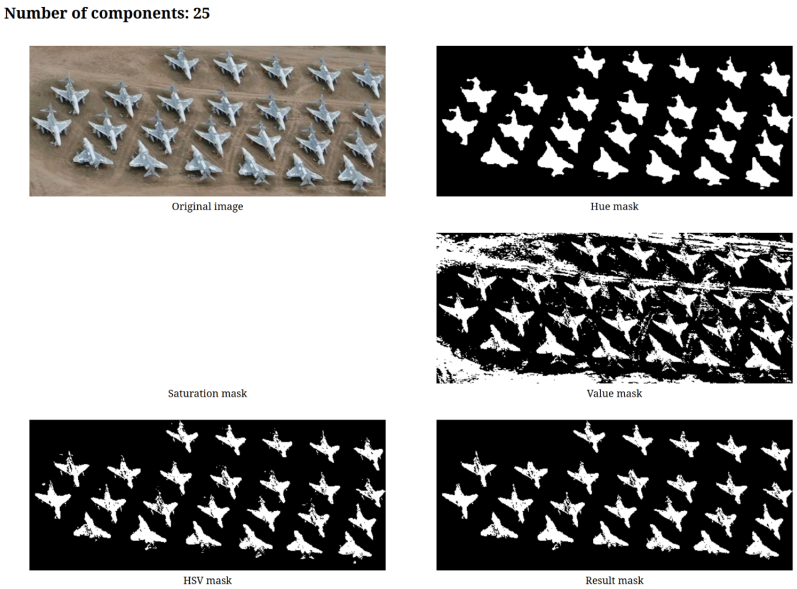
5 Эксперименты

Эксперименты проводились на предоставленных изображениях, обрезанных прямоугольной рамкой (см. рис. 3).

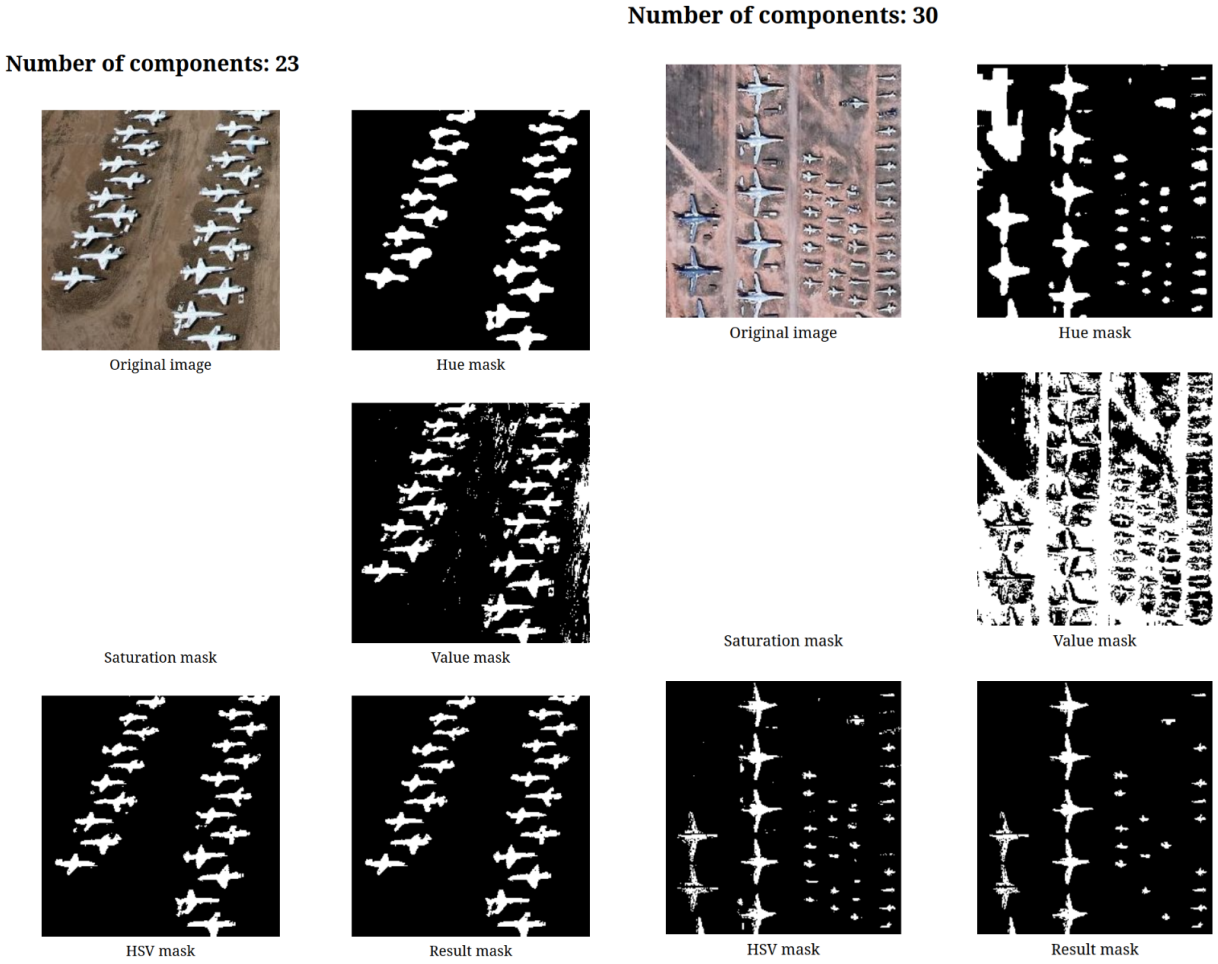
Алгоритм хорошо показал себя на большинстве изображений, корректно составил бинаризованную маску и подсчитал число связных компонент, таким образом точно оценил количество самолетов на фотографии (эксперименты 1, 2). Однако для некоторых изображений у алгоритма возникали трудности (эксперимент 3). В основном это связано с разным размером самолетов на фотографии. В таком случае трудно подобрать минимальный размер связной компоненты, так как слишком высокое значение уберет маленькие самолеты, а слишком низкое — не отсеет шум. Решением может быть разделение изображения на части с самолетами одинакового размера и обработка их в отдельности. Также алгоритм может плохо отделить фон, если его цвет схож с цветом самолета. В таком случае потребуется более тонкая настройка параметров бинаризации.

6 Выводы

Предложенный алгоритм смог добиться высокой точности при решении поставленной задачи. Однако существует область применимости, вне которой качество алгоритма падает. Поэтому необходимо тщательно подбирать входные данные, так как это может существенно повлиять на точность результата.



(a) Эксперимент 1 (истинное количество: 25)



(b) Эксперимент 2 (истинное количество: 23)

(c) Эксперимент 3 (истинное количество: 44)

Рис. 3: Эксперименты