

Министерство образования и науки РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ
«МИСИС»

Институт информационных технологий и компьютерных наук
Кафедра инженерной кибернетики

Курсовая работа

по дисциплине «Методы и средства обработки изображений»

на тему

«Анализ изображений эпителия методами компьютерного зрения»

Выполнили:
Студенты 3-го курса,
гр. БПМ-20-2
Гущин В.В., Моисейченко К.А., Медведев К.Д.

Проверил:
Садеков Р.Н.

Москва, 2023

Содержание:

1. Аннотация
2. Постановка задачи
3. Используемые технологии
4. Используемые методы и алгоритмы
5. Демонстрация работы программы
6. Заключение
7. Список использованных источников

1. Аннотация

Данная курсовая работа посвящена разработке приложения для обработки изображений эпителия с использованием методов компьютерного зрения. Цель работы состоит в автоматизации процесса обнаружения индикатора красного цвета, определении площади и контуров красного цвета на изображении, а также обнаружении "артефактов" - пятен яркого цвета, не относящихся к клеткам эпителия. В результате работы было разработано удобное для использования приложение, которое значительно ускоряет и упрощает процесс обработки изображений эпителия.

Summary

This course work is devoted to the development of an application for processing epithelial images using computer vision methods. The purpose of the work is to automate the process of detecting a red indicator, determining the area and contours of red in the image, as well as detecting "artifacts" - bright spots that do not belong to epithelial cells. As a result of the work, a user-friendly application has been developed that significantly speeds up and simplifies the process of processing epithelial images.

2. Постановка задачи

Целью данной работы является разработка приложения, которое может выполнять следующие задачи на изображениях эпителия:

- Обнаружение индикатора красного цвета.
- Определение площади красного цвета на изображении.
- Определение контуров красного цвета.
- Обнаружение "артефактов" - пятен яркого цвета, не относящихся к клеткам эпителия.

3. Используемые технологии

Для разработки приложения использовались следующие технологии

- Язык программирования: Python.
- Библиотеки: OpenCV, NumPy, StreamLit.
- Среды разработки: Jupyter Notebook, Docker.

4. Используемые методы и алгоритмы

4.1. Определение площади красного цвета

Изначально планировалось решить задачу определения красного цвета с помощью сегментации по цветам алгоритмом Kmeans^[1], однако такое решение не давало желаемого результата из-за сильной цветовой зашумлённости изображений. Поэтому было решено преобразовывать изображение в цветовое пространство HSV для более удобной работы с оттенками цвета. Задаются нижние и верхние границы для диапазона красного цвета в цветовом пространстве HSV (диапазон разбит на две части, так как красный цвет обернут в цветовом круге)^[2]. Создаются две маски, каждая из которых содержит пиксели, соответствующие красному цвету в изображении. Затем маски объединяются с использованием побитового "или"^[3]. Для определения площади, подсчитывается количество пикселей, соответствующих красному цвету, в маске.

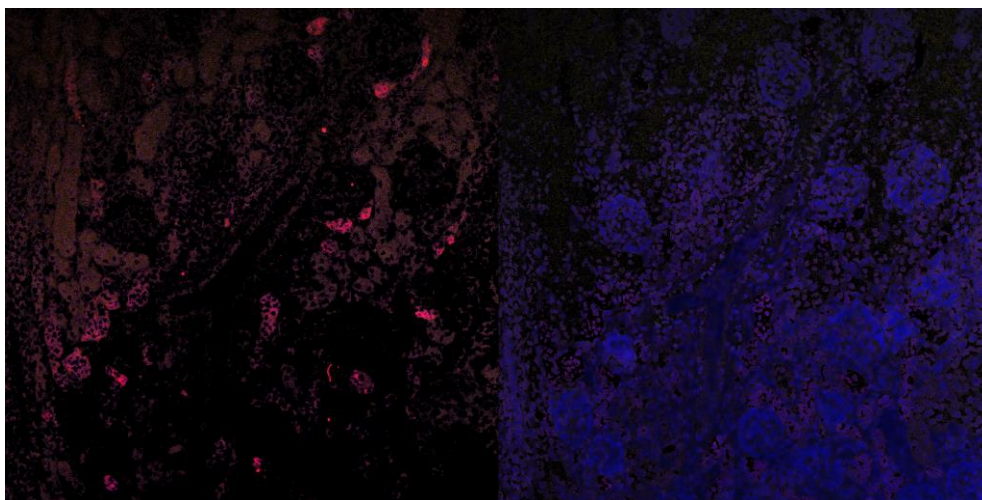


Рисунок 1 – Определение красного цвета на картинке

4.2. Определение контуров красного цвета

Из-за сильной цветовой зашумлённости изображений, для корректного определения контуров красного цвета, потребовалась предварительная обработка полученной маски красного цвета. К маске применяются морфологические операции закрытия и открытия для удаления маленьких шумовых объектов и замыкания областей красного цвета^[4]. Применяется размытие Гаусса для

сглаживания контуров маски^[5]. На основе получившейся маски находятся контуры объектов на изображении, которые затем фильтруются на основе их площади, оставляя только те контуры, чья площадь превышает заданный порог^[6]. Отфильтрованные контуры рисуются на копии исходного изображения.



Рисунок 2 – Маска красного цвета до и после обработки

4.3. Определение артефактов

Для обнаружения артефактов применяется фильтр среднего сдвига (mean shift)^[7]. Изображение преобразуется в оттенки серого, после чего применяется пороговая обработка для получения двоичного изображения артефактов.

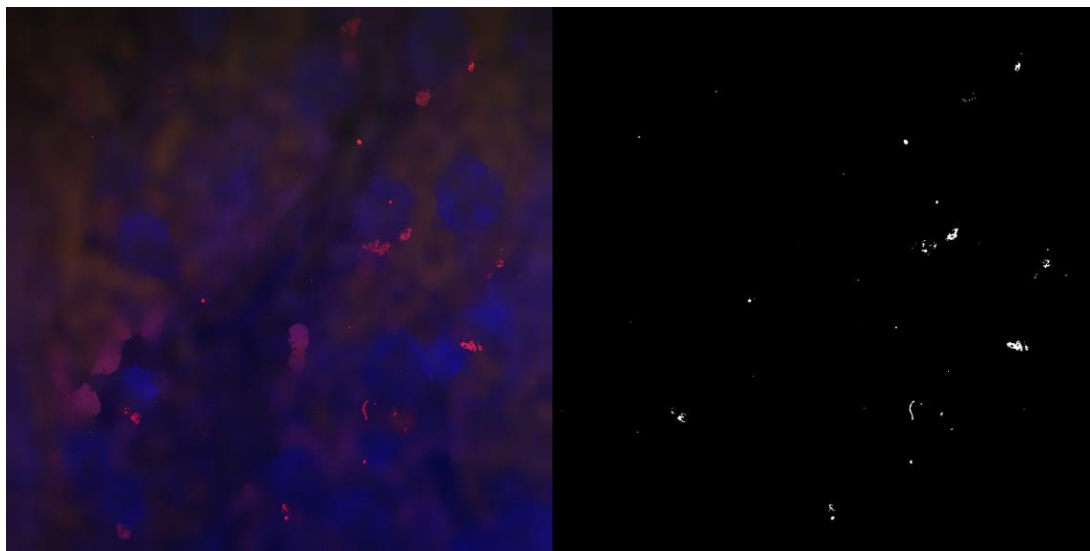


Рисунок 3 – Обнаружение артефактов на картинке

5. Демонстрация работы программы

Приложение разработано с учетом простоты использования. После запуска программы пользователю предлагается загрузить изображение эпителия для обработки и выбрать задачи, которые необходимо выполнить. Результаты обработки отображаются на экране и могут быть сохранены для дальнейшего анализа.

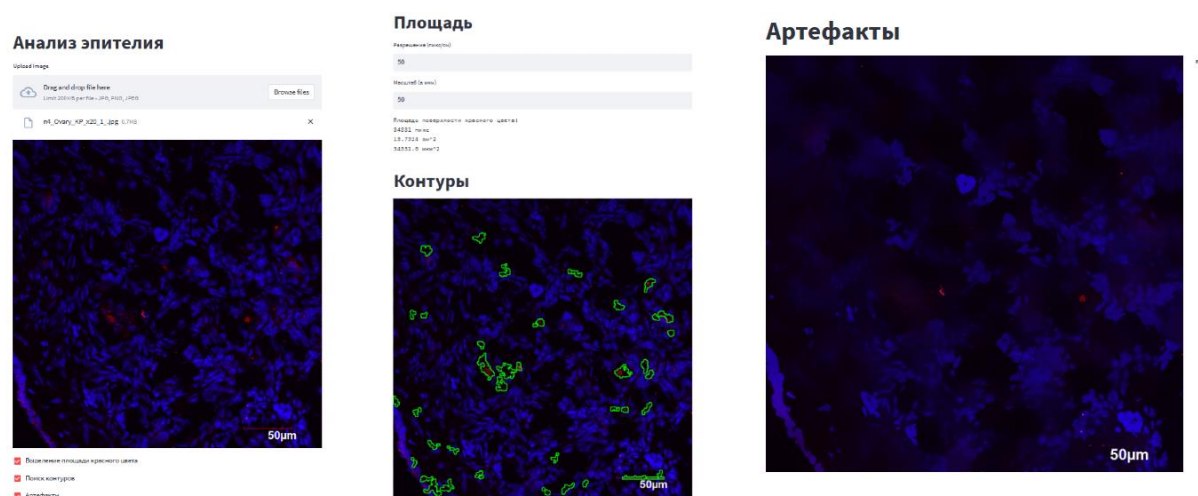


Рисунок 4 – Демонстрация работы программы 1

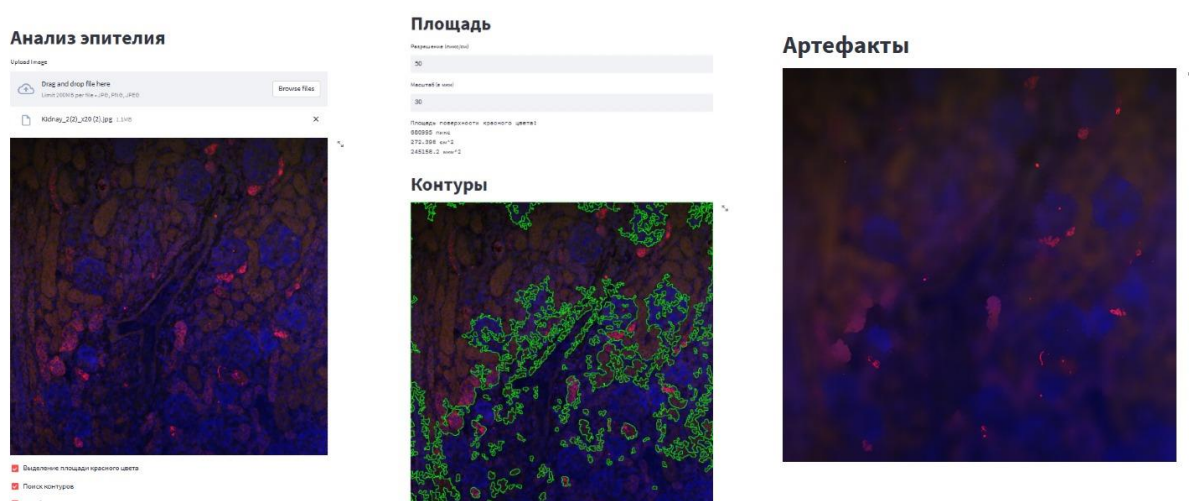


Рисунок 5 – Демонстрация работы программы 2

6. Заключение

В ходе выполнения работы было разработано приложение, которое успешно выполняет задачи обработки изображений эпителия с использованием методов компьютерного зрения. Приложение значительно упрощает и ускоряет процесс обнаружения индикатора красного цвета, определения площади и контуров красного цвета, а также обнаружения "артефактов". Результаты работы могут быть использованы в медицинских и научных исследованиях для анализа изображений эпителия.

7. Список использованных источников

- [1] Stack overflow: OpenCV Color Segmentation Using Kmeans // <https://stackoverflow.com/questions/52802910/opencv-color-segmentation-using-kmeans>
- [2] Stack overflow: Choosing the correct upper and lower HSV boundaries for color detection // <https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv>
- [3] OpenCV documentation: Thresholding Operations using inRange // https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html
- [4] OpenCV documentation: Morphological Transformations // https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
- [5] OpenCV documentation: Smoothing Images // https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html
- [6] OpenCV documentation: Contours in OpenCV // https://docs.opencv.org/3.4/d3/d05/tutorial_py_table_of_contents_contours.html
- [7] OpenCV documentation: Image Filtering // https://docs.opencv.org/3.4/d4/d86/group_imgproc_filter.html