



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. В. ЛОМОНОСОВА
Факультет вычислительной математики и кибернетики

Отчет к Лабораторной работе №1: Подсчет монеток на изображении

Студент 3 курса ВМК (317 группа):
Оспанов А.М.

Москва, 2015

Содержание

1	Введение	2
2	Основная часть	3
2.1	Исследование изображений	3
2.2	Обработка изображений	5
3	Заключение	8

1 Введение

Данный отчет был написан по дисциплине “Обработка и распознавание изображений” студентом 317 группы Оспановым Аятом.

В данной работе нужно было реализовать алгоритм для подсчета монеток на изображении. Для этого использовались морфологические методы обработки изображений. В качестве языка программирования использовался Python версии 2 с библиотекой OpenCV.

2 Основная часть

2.1 Исследование изображений

Перед тем, как начать писать код, надо узнать информацию о картинках. В ходе исследований стало известно, что картинки не подчиняются под один закон, т.е. не одинаково распределено освещение, картинки имеют разные фоны и т.д. Но так же был замечен интересный факт при рассмотрении трех каналов картинок: если фон имеет темный цвет, то для бинаризации хорошо подходит красный канал, если фон светлый - то синий. Это можно увидеть из следующих картинок:

Для темного фона



RED



GREEN



BLUE



RED



GREEN



BLUE



Для светлого фона



RED



GREEN



BLUE



RED



GREEN

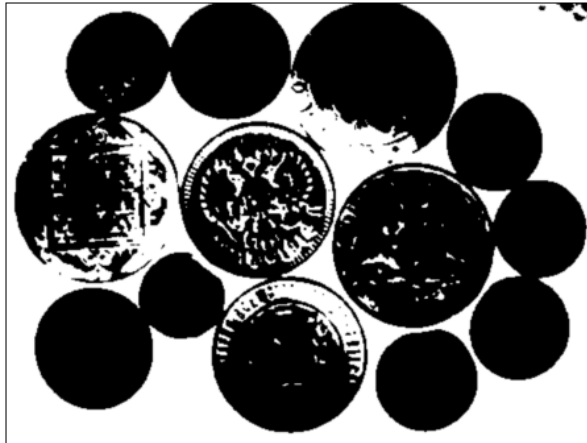


BLUE



2.2 Обработка изображений

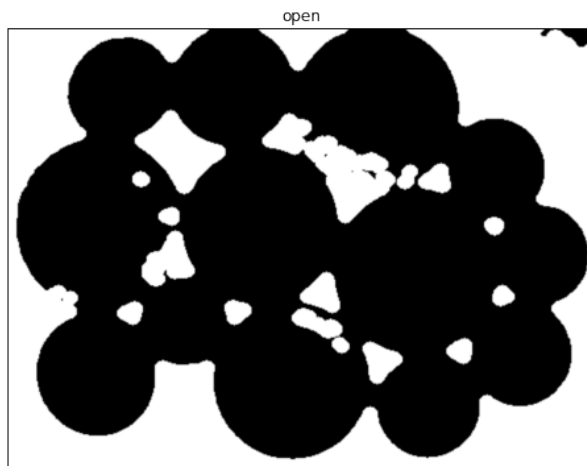
После того, как выяснили характеристики изображений, начнем обработку. Всегда все начинается с бинаризации:



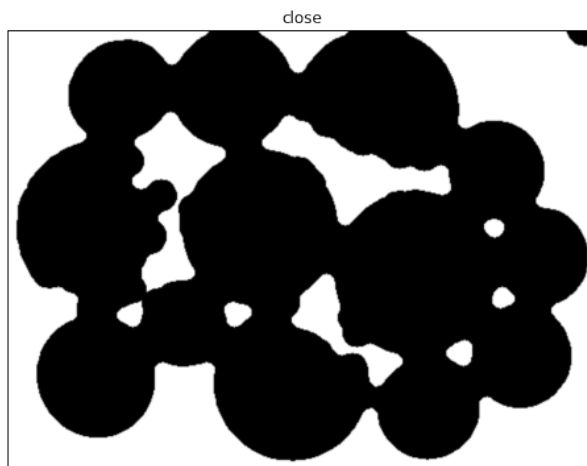
Бинаризация делалась адаптивным методом из библиотеки OpenCV:
функция `adaptiveThreshold`

Затем комбинации открытых и закрытий (или, что то же самое, дилатация и эрозия):

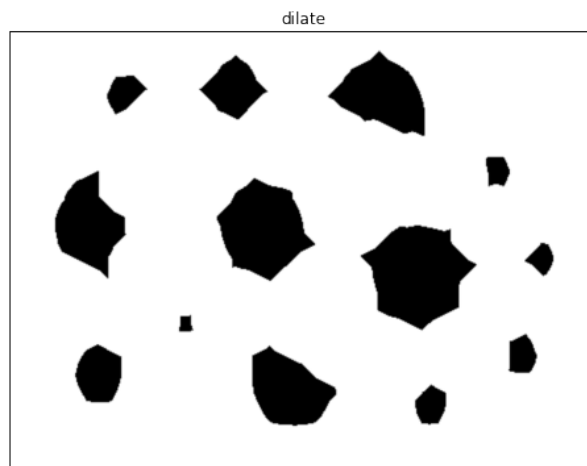
Открытие с эллипсоидальным ядром первого радиуса:



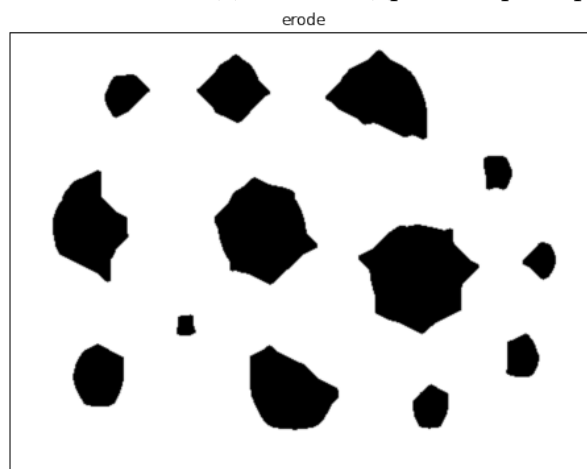
Закрытие с эллипсоидальным ядром первого радиуса:



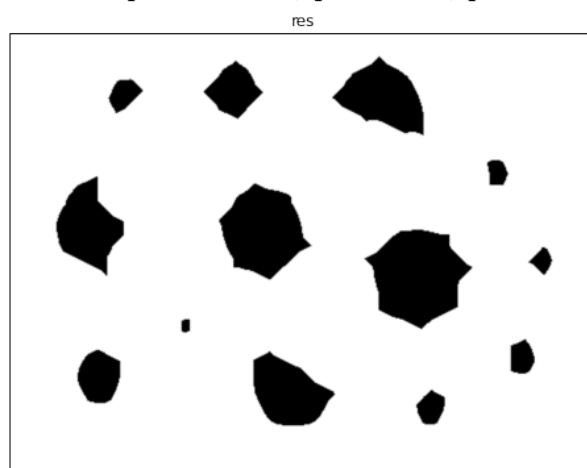
Заккрытие с эллипсоидальным ядром второго радиуса:



Открытие с эллипсоидальным ядром второго радиуса:



Заккрытие с квадратным ядром:



Открытие и закрытие делались следующими функциями: morphologyEx, erode, dilate.

Радиусы выбирались с целью покрыть этим алгоритмом большее число картинок. Т.к. картинки имеют разные характеристики, то не удастся получить единый алгоритм для всех картинок сразу. В ходе исследований выяснилось, что оптимальные параметры следующие:

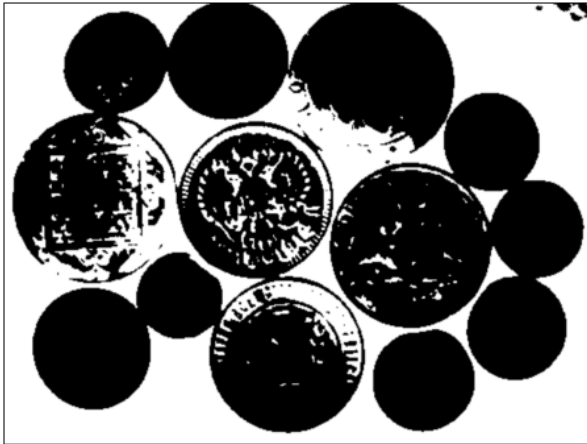
Преобразование	Ядро	Радиус (повторения)
Первое открытие	Эллипс	13 (1)
Первое закрытие	Эллипс	25 (1)
Второе закрытие	Эллипс	7 (10)
Второе открытие	Эллипс	5 (1)
Третье закрытие	Квадрат	7 (1)

После того, как монеты отделились на компоненты связности, считаем эти компоненты связности. При этом не считаются компоненты, прилегающие к краям. Для этого использовалась функция `findContours` которая находит контуры и далее остается просто подсчитать количество этих контуров.

3 Заключение

Данный алгоритм сработал на всех изображениях, кроме 5, 13, 14 изображений, при хорошем подборе констант. Результаты можно увидеть в следующей таблице:

Бинарное изображение



Конечное изображение

