



ЗАДАЧА 10 О МОНЕТАХ

Фирсов Сергей
МФТИ, Б05-105



Знакомство

Фирсов Сергей Андреевич



3 курс бакалавриата МФТИ, ФПМИ ПМФ ПМ

Студент направления прикладная математика и физика

Хобби и интересы:



Волейбол



Программирование



Кино



Покер



Преподавание





План



Постановка задачи



Анализ и идеи реализации



Теоретические исследования



Идея практического решения



Продвижения в реализации



Постановка задачи

- Требуется научиться определять суммарный номинал монет по фотографии.
- Особенности:
 - *Разные цифры/текст*
 - *Разный общий рисунок*
 - *Разный диаметр:*
 - 10р -- 22 мм
 - 5р -- 25 мм
 - 2р -- 23 мм
 - 1р -- 20.5 мм



Варианты решения

- Выявление окружностей и/или рисунка монет
- Разделение по
 - *размеру*
 - *изображению*
 - *надписям*
 - *другим отличительным признакам*
- Подготовка базы данных картинок монет
- Обучение модели для определения сходства с базой



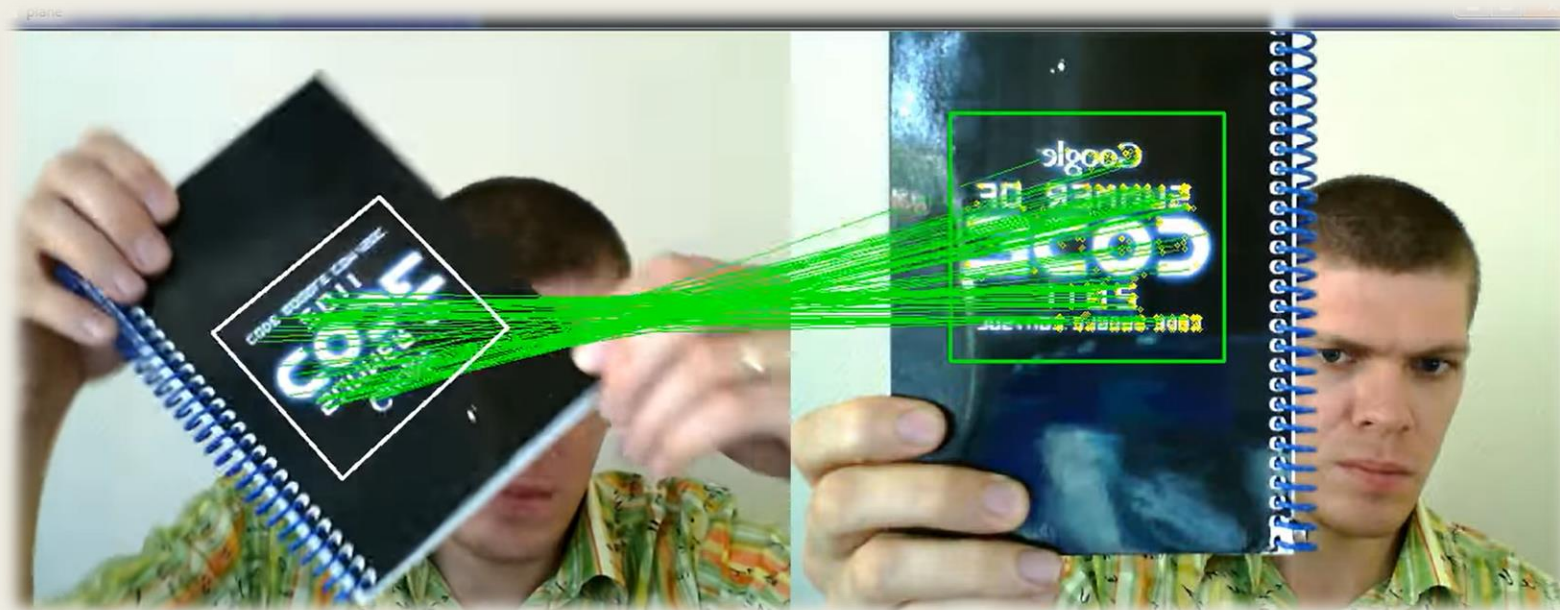
Идеи анализа сходства картинок

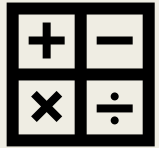


Гистограммы

Шаблоны

Функции



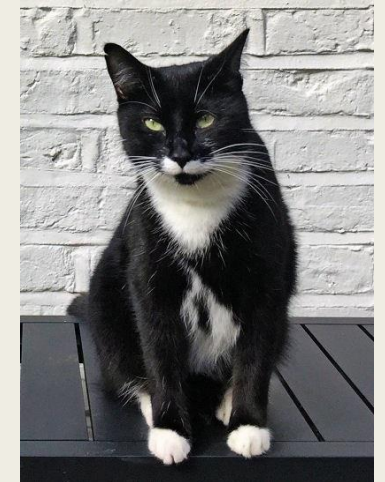


Численные методы

SSIM индекс структурного сходства:

- Метод учитывает «восприятие ошибки» благодаря учёту структурного изменения информации
- Недостатки:
 - *Входные изображения имеют одинаковый размер*
 - *Проблемы с масштабированием, переносом, вращением и искажениями, так как сравнение поточечное*

Векторные представления:

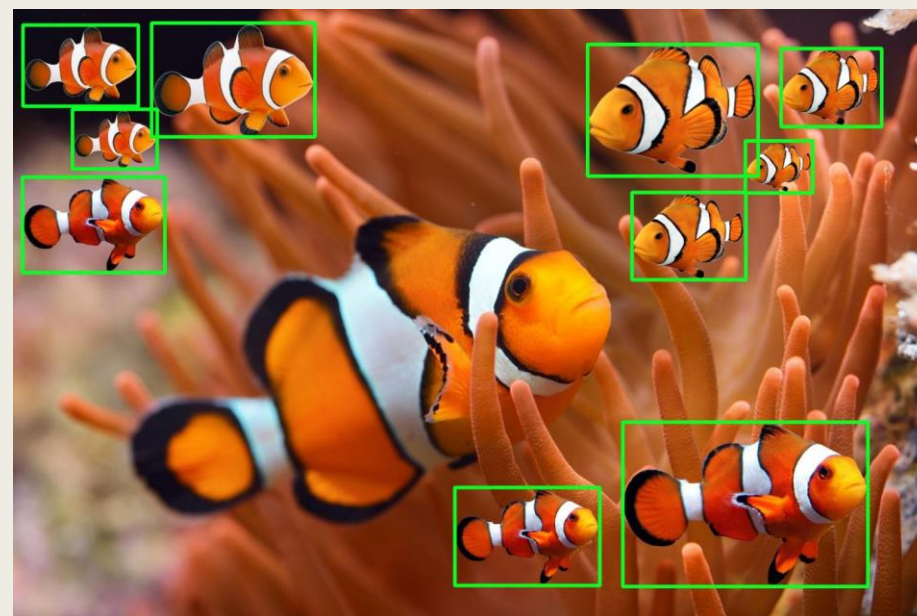
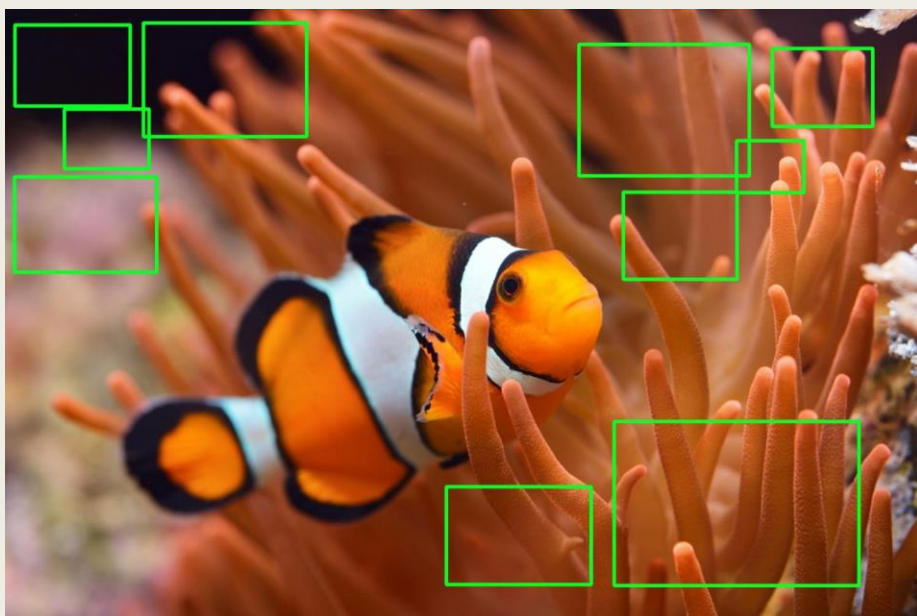


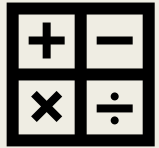
Score: 97.141%
.\cat_1.jpg
.\cat_2.jpg



Оценка сходства:

89,462%



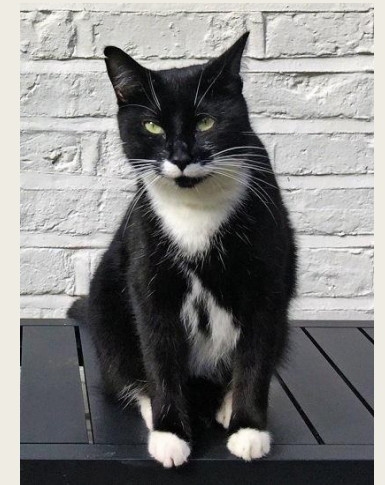


Численные методы

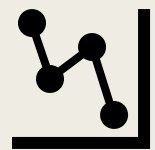
SSIM индекс структурного сходства:

- Метод учитывает «восприятие ошибки» благодаря учёту структурного изменения информации
- Недостатки:
 - *Входные изображения имеют одинаковый размер*
 - *Проблемы с масштабированием, переносом, вращением и искажениями, так как сравнение поточечное*

Векторные представления:



Score: 97.141%
.\cat_1.jpg
.\cat_2.jpg

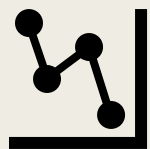


Детектирование границ Canny

- **Края**(границы) — это такие кривые на изображении, вдоль которых происходит резкое изменение яркости или других видов неоднородностей.

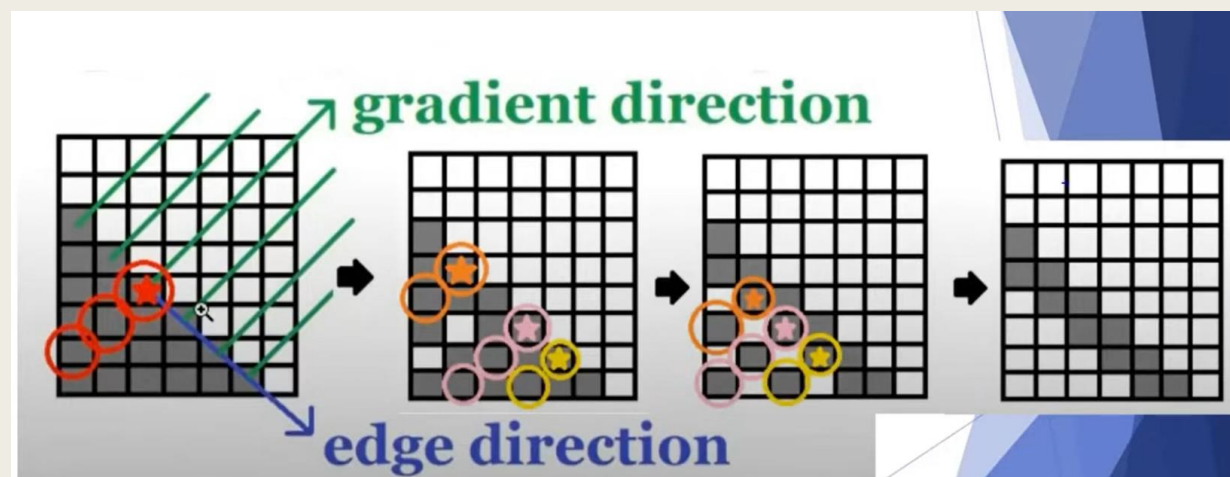
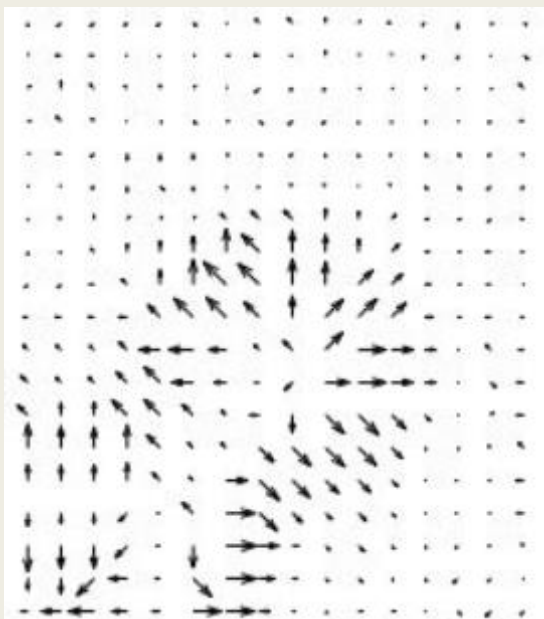


<https://en.wikipedia.org/>



Детектирование границ Canny

- Шаги детектора:
 - Сглаживание
 - Поиск градиентов
 - Подавление немаксимумов
 - Двойная пороговая фильтрация
 - Трассировка области неоднозначности



↻ Преобразование Хафа

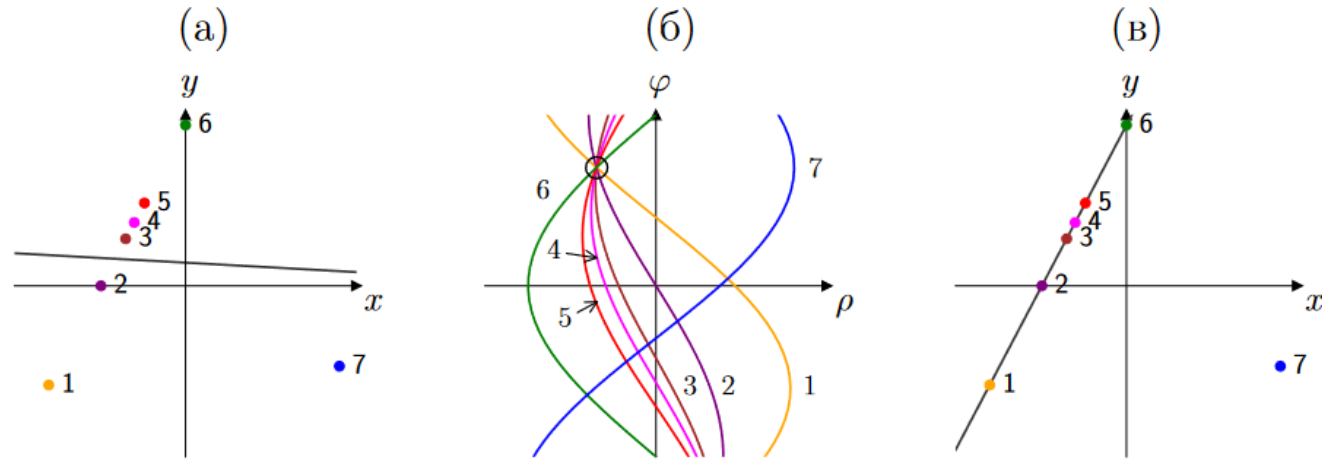


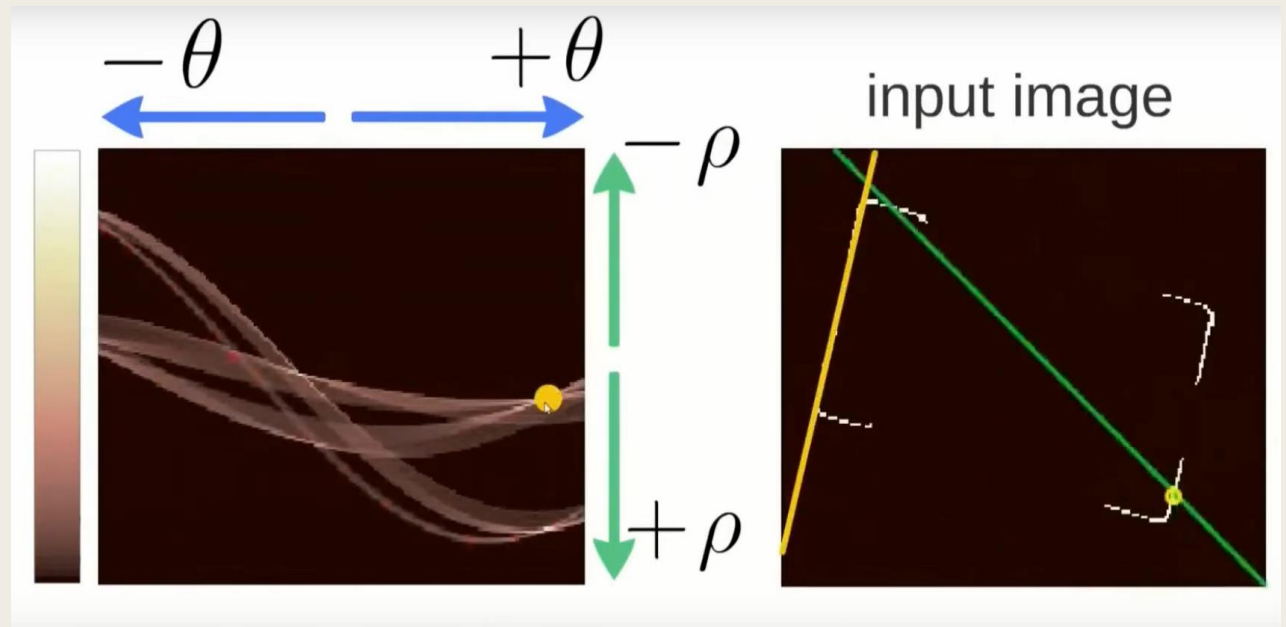
Рисунок 1.1 — Преобразование Хафа с нормальной параметризацией:
(а) шесть точек, лежащих на одной прямой в исходном пространстве и одна, являющаяся выбросом; отмечена прямая, являющаяся решением задачи определения параметров прямой методом наименьших квадратов;
(б) Хаф-образ исходного изображения, кружком отмечено его максимальное значение; (в) прямая в исходном пространстве, с параметрами, равными координатам максимума Хаф-образа.

Диссертация Дмитрия Петровича Николаева

Методы и алгоритмы Хаф-анализа изображений в системах технического зрения

↻ Преобразование Хафа

- Записываем параметрическое уравнение
- Берём двоичное изображение*
- Перебираем все точки
- Голосуем за все параметры из каждой точки (с заданным шагом)
- Ищем локальные максимумы

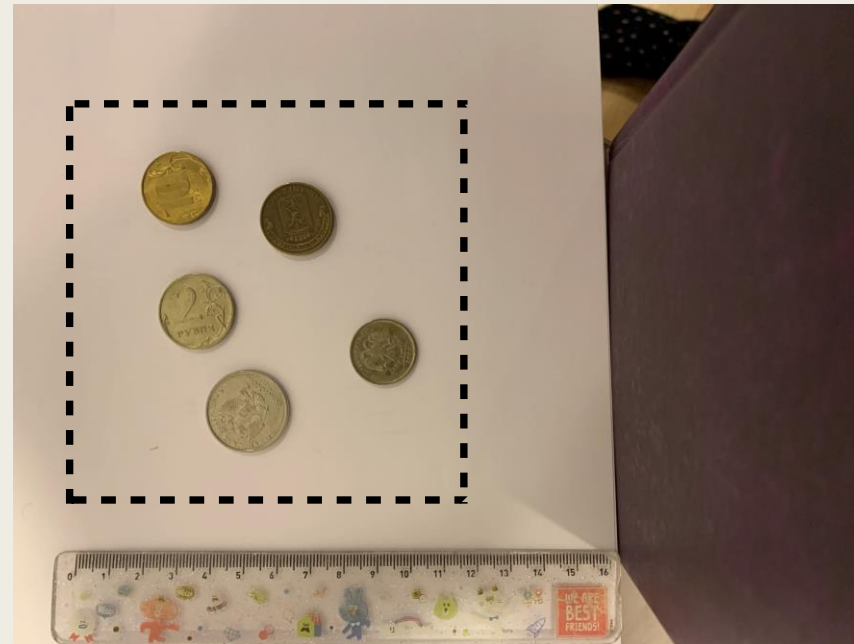


Реализация

- Будем использовать библиотеку OpenCV для python
- Разделим задачу на этапы и будем учиться их выполнять для итогового решения результата

Параметры:

- *Фотографируем с высоты 16 см*
- *Фиксированные параметры камеры*
- *Фон равномерный*





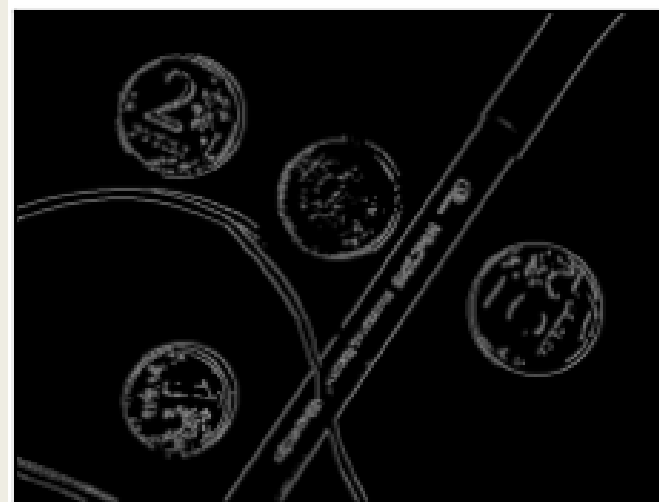
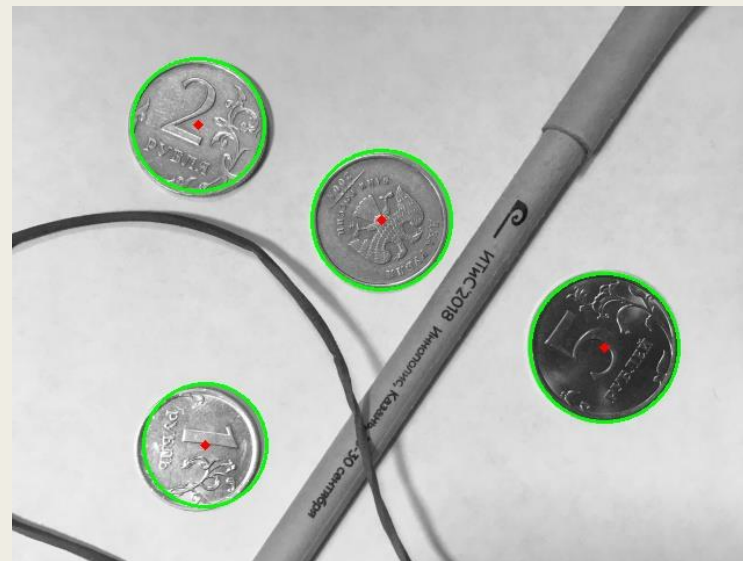
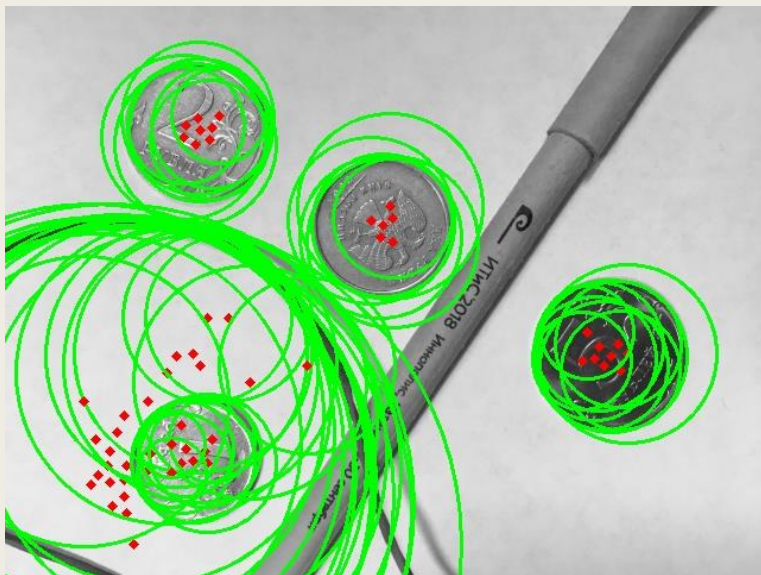
Реализация

- Научимся различать монету и выделять её контура
- Научимся обрабатывать сразу несколько монет с фото
- Научимся вычислять диаметр и площадь одной монеты
- Исследуем погрешности измерения диаметра и площади монеты для выявления лучшего метода *
- Соединим этапы и получим готовую программу **

Выделение монеты и получение контура

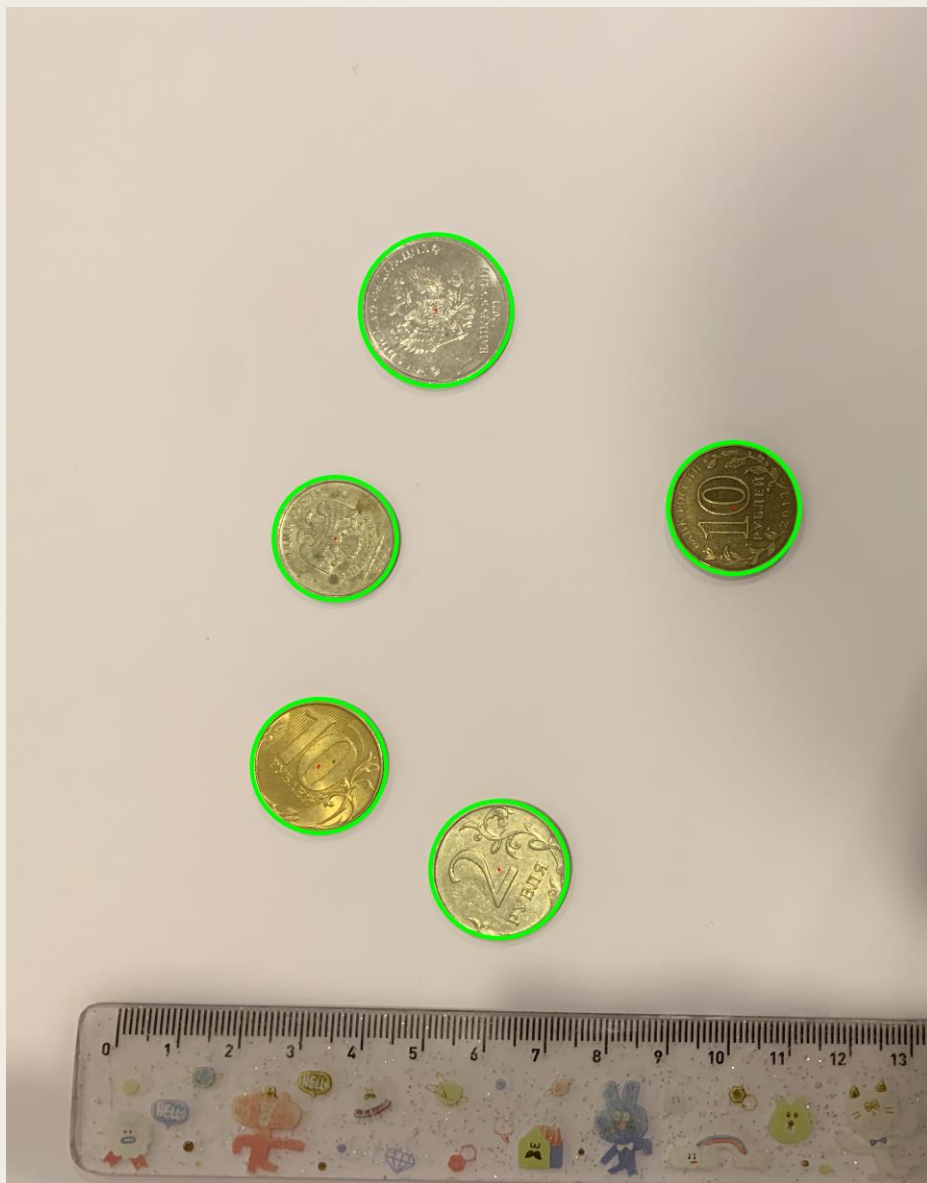


Работа с несколькими монетами





Определение диаметров



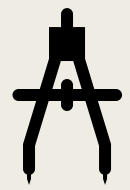
радиус 195 п -- 5 р

радиус 176 п -- 2 р

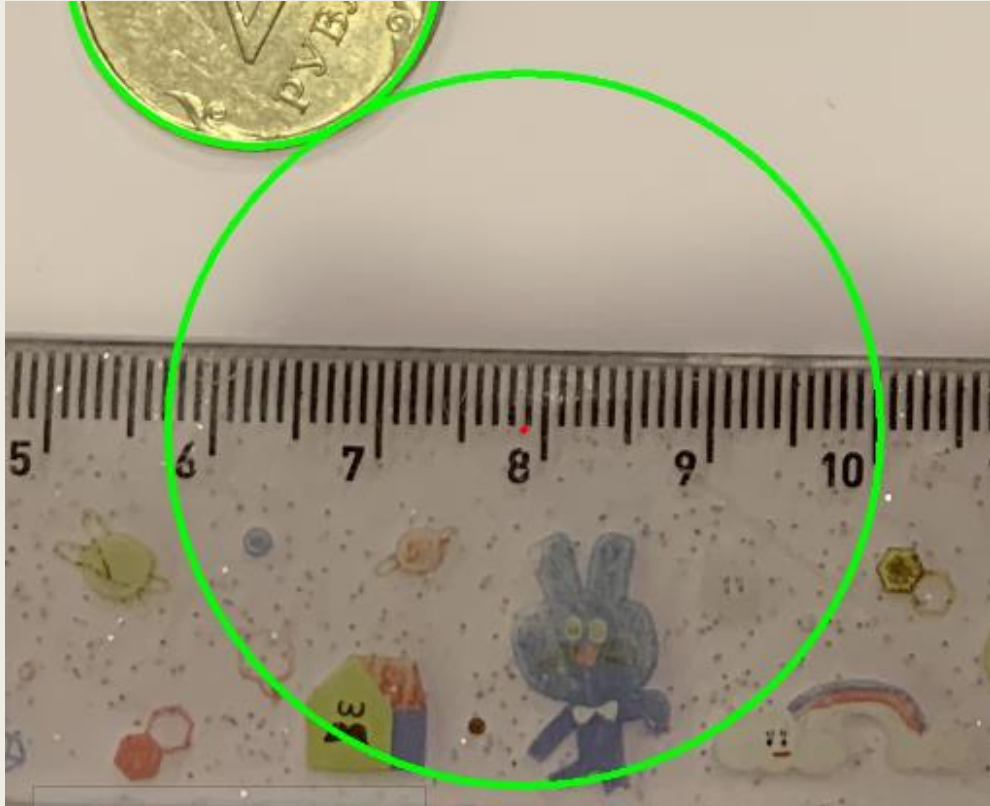
*радиус 172 п -- 10 р (чуть больше, так
как коллекционная)*

радиус 169 п -- 10 р

радиус 158 п -- 1 р

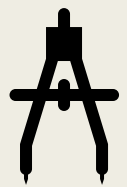


Определение диаметров

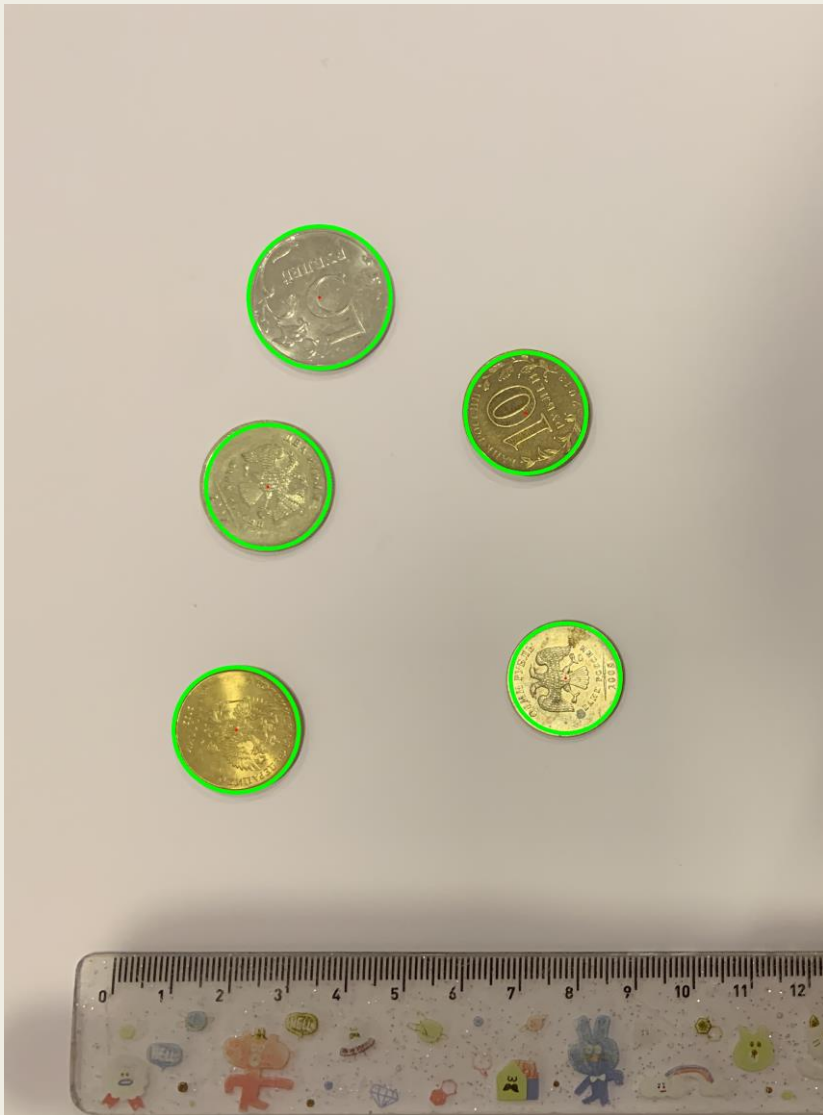


- Импирически (доп экспериментом) был получен коэффициент пересчёта пиксели -> миллиметры ($k \approx 16$)
- 337 пикселей == 21 мм

Монета	R пиксели	R мм (эксп)	R теор
5	195	12,19	12,5
2	176	11	11,5
10*	172	10,75	11*
10	169	10,56	11
1	158	9,875	10,25



Определение диаметров



радиус 193 п -- 5 р

радиус 170 п -- 2 р (видно что контур
определился не очень точно)

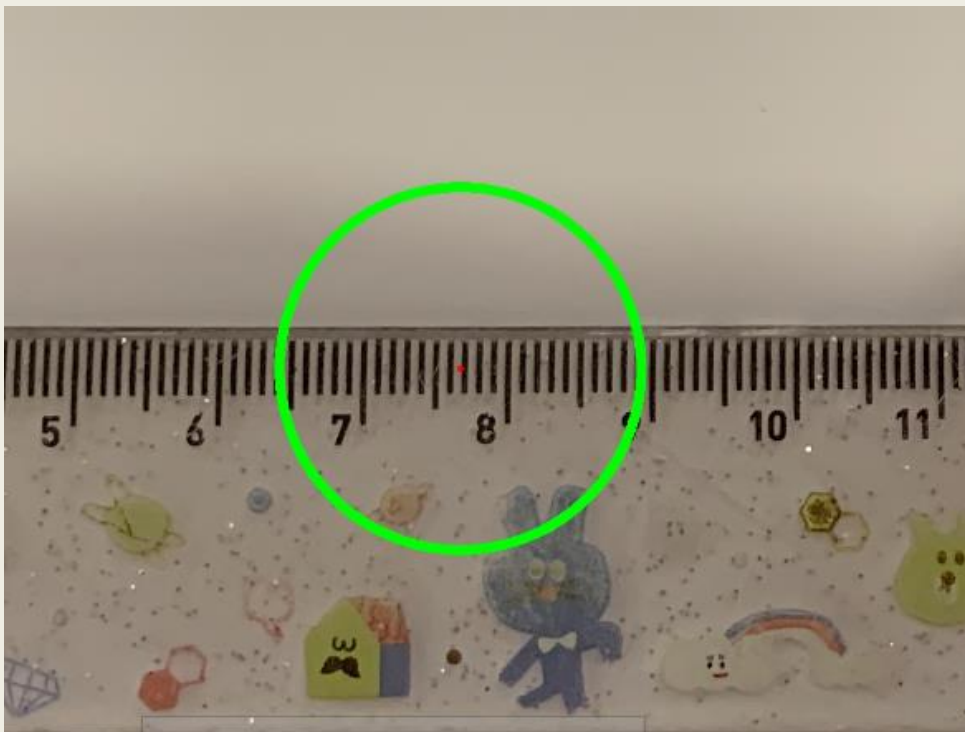
радиус 168 п – 10 р *

радиус 164 п -- 10 р

радиус 149 п -- 1 р



Определение диаметров



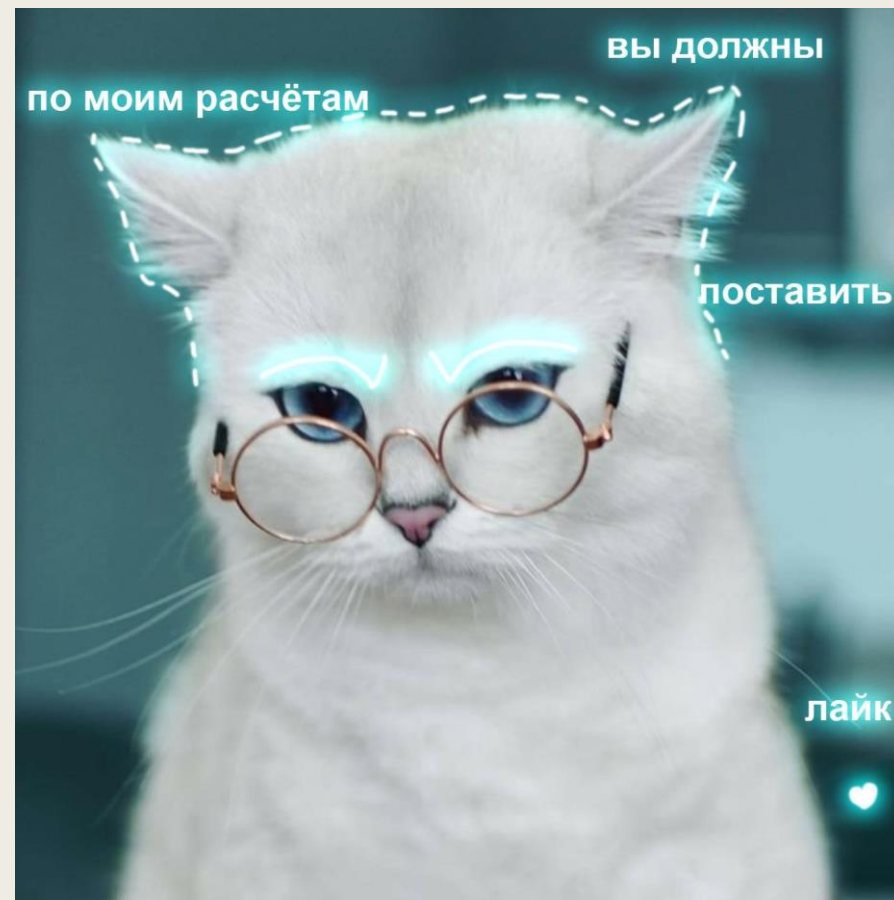
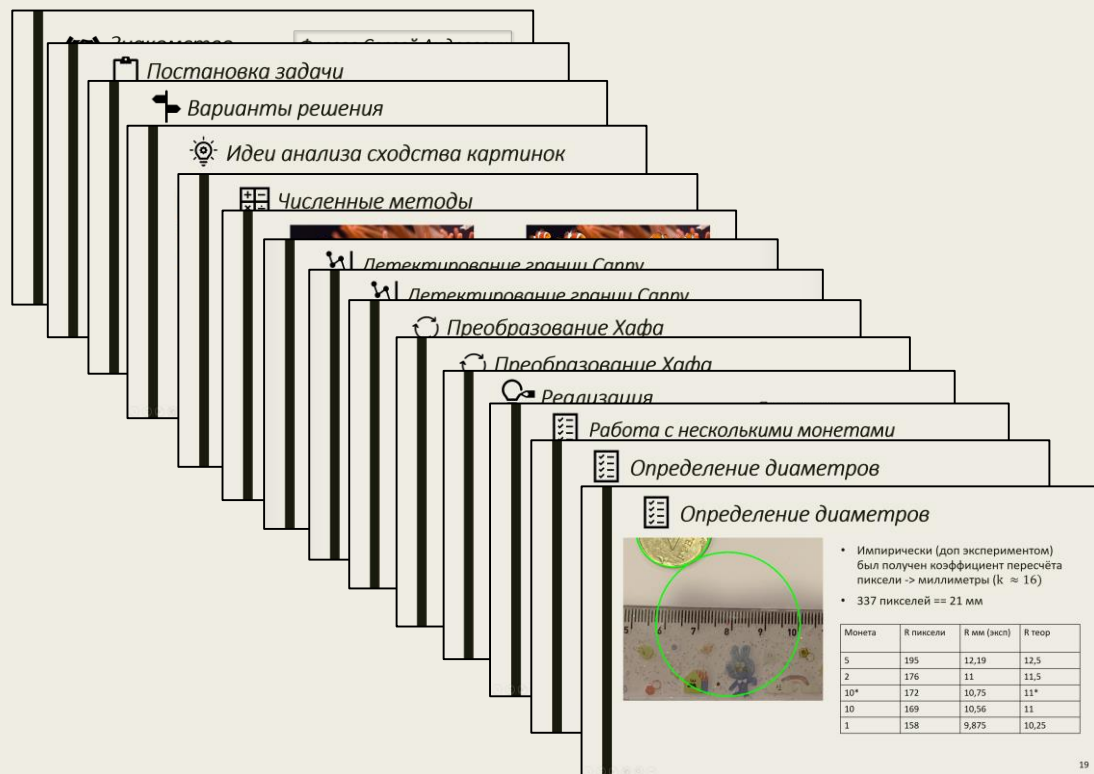
- Импирически (доп экспериментом) был получен коэффициент пересчёта пиксели -> миллиметры ($k \approx 16,4$)
- пикселей 197 == 12 мм

Монета	R пиксели	R мм (эксп)	R теор
5	193	11,8	12,5
2	170	10,37	11,5
10*	168	10,24	11*
10	164	10	11
1	149	9,1	10,25



Исследование улучшения точности при подсчёте площади





Приложение 0



Screenshot of a PyCharm IDE window showing a Python script for circle detection and a corresponding image of coins with detected circles.

The script, located in `osnovnoe.py`, performs the following steps:

- Imports `cv2` and `numpy` as `np`.
- Loads an image `col_img` from `IMG_1705.JPG`.
- Applies a bilateral filter to the image.
- Uses `cv2.HoughCircles` to detect circles in the filtered image.
- Iterates over the detected circles and draws them on the original image using `cv2.circle`.
- Saves the result to `hough_circles1.jpg`.

The image `hough_circles1.jpg` shows six coins with green circles drawn around them, indicating successful detection. A ruler is visible at the bottom of the image for scale.

The PyCharm interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for file operations and running, and a status bar at the bottom showing the current file, encoding, and subscription information.

Приложение 1 – Численные методы

Среднеквадратичная ошибка MSE

$$MSE(x, y) = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \|x(i, j) - y(i, j)\|^2$$

MSE измеряет среднеквадратичную разницу между оценочными значениями (предсказанными значениями) и фактическими значениями (истинными значениями). Итак, мы просто вычисляем квадрат разницы пиксель за пикселем.

Пиковое отношение сигнал / шум PSNR

$$PSNR(x, y) = 10 \log_{10} \left(\frac{MAX}{MSE(x, y)} \right)$$

Структурное сходство SSIM

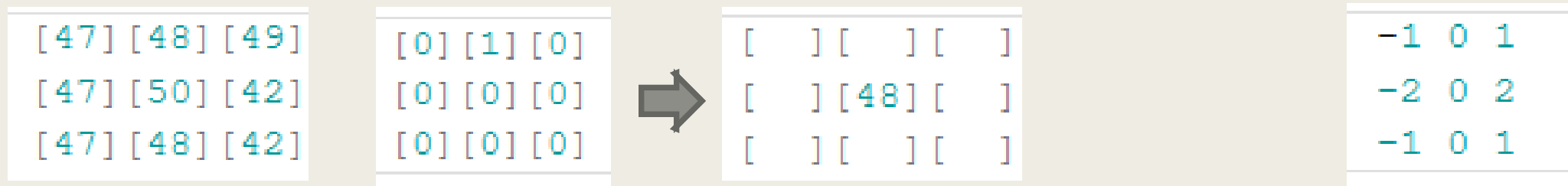
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2)}$$

Приложение 2 – свёртка и оператор Соболя

Свёртка— это операция, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой.

В случае работы с изображениями — свёртка — это операция вычисления нового значения заданного пикселя, при которой учитываются значения окружающих его соседних пикселей.

Главным элементом свёртки является т.н. **ядро свёртки** — это матрица (произвольного размера и отношения сторон; чаще всего используется квадратная матрица (по-умолчанию, размеры 3x3)).



Оператор Соболя — это дискретный дифференциальный оператор, вычисляющий приближение градиента яркости изображения.

Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы.

Т.е. результатом работы оператора Соболя в точке области постоянной яркости будет нулевой вектор, а в точке, лежащей на границе областей различной яркости — вектор, пересекающий границу в направлении увеличения яркости