Московский государственный университет
имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ЛАБОРАТОРНАЯ РАБОТА №2
ПО КУРСУ «ОБРАБОТКА И РАСПОЗНАВАНИЕ ИЗОБРАЖЕНИЙ»

КЛАССИФИКАЦИЯ ГЕОГЛИФОВ ПУСТЫНИ НАСКА

Выполнил студент 3 курса
317 группы:

Иванов Сергей Максимович

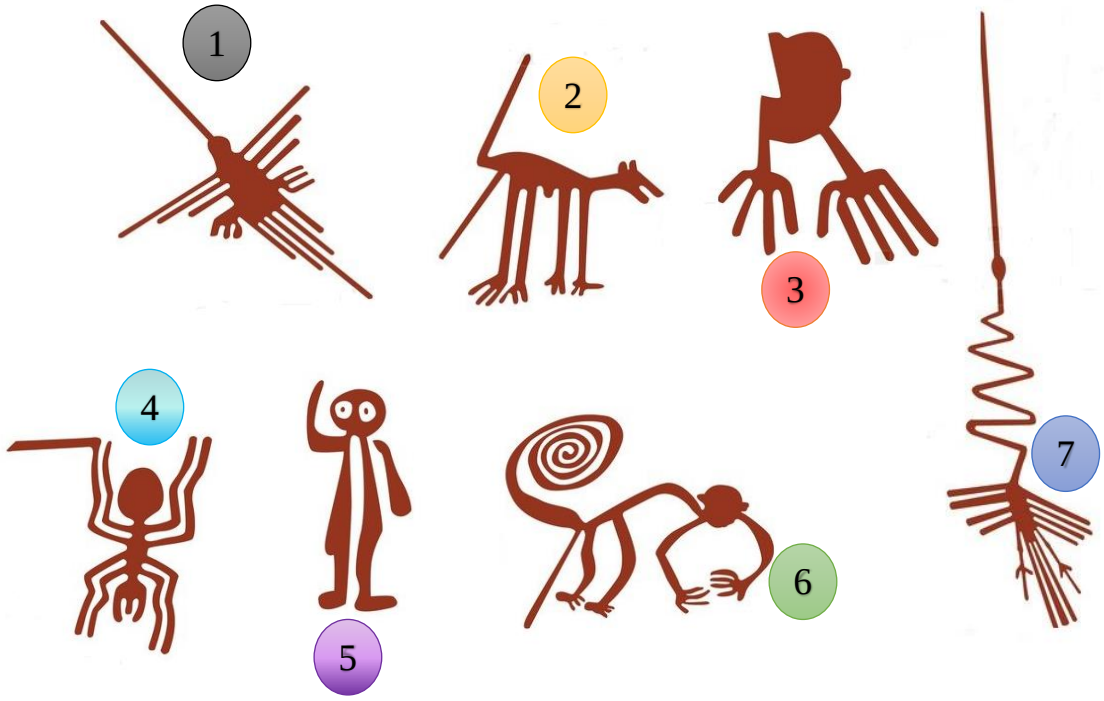
Москва, 2017

Оглавление

1. Задача

1.1. Постановка задачи

По приведённому изображению геоглифа в формате BMP24 требуется определить её класс (1-7). 7 видов геоглифов, которые требуется различать, определены условием задачи; для каждого из класса приведено по 4 примера изображений.



1.2. Входные данные

Примеры входных изображений приведены на рисунке выше, и они же являются эталонами для 7 классов. Все изображения одного класса являются нелинейными трансформациями исходных эталонов. Масштаб изображений при этом относительно сохраняется, хотя соотношение ширины и длины изображений меняется. В силу нелинейности трансформаций, части фигур меняют масштаб по различным правилам, что не позволяет решать задачу тривиальными подходами.

Входные изображения уже представлены в виде готовых к бинаризации фигур. За исключением класса 5, все фигуры представляют собой единую компоненту связанности в бинарном изображении; за счёт этого, в принципе, класс 5 уже можно отделить от остальных классов.

1.3. Программная реализация

Программа реализована на языке Python 3 в среде Jupyter Notebook. Реализация базовых процедур обработки изображений (ввод, вывод,

бинаризация) взяты из библиотеки OpenCV. Также из OpenCV использовался стандартный алгоритм нахождения минимальной выпуклой оболочки контура. Алгоритм получения скелета фигуры взят из библиотеки scikit-image.

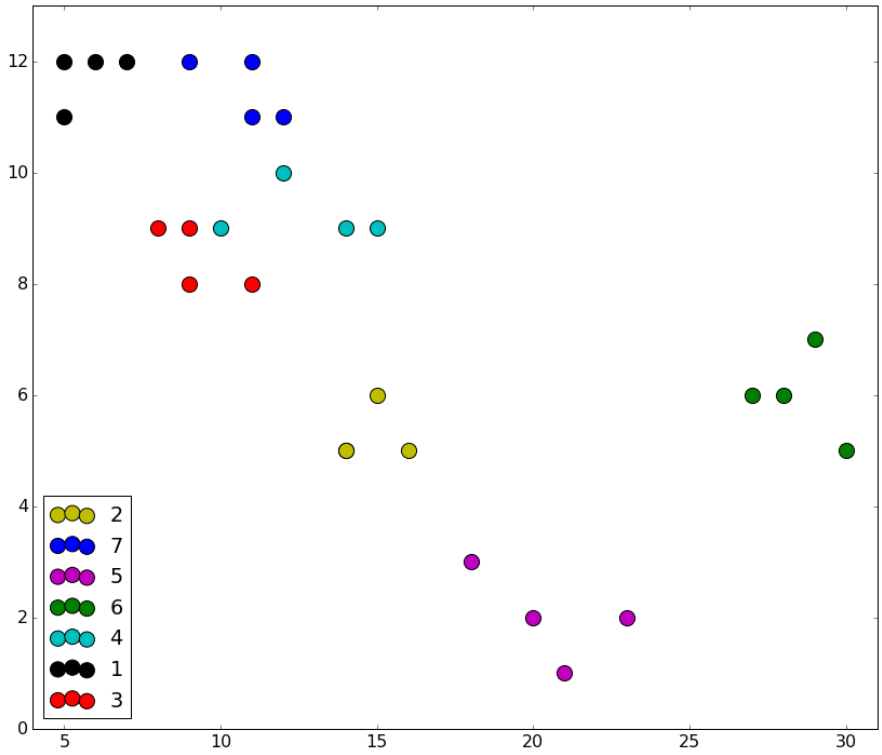
Программа состоит из следующих функций:

- show – для показа изображений средствами OpenCV
- load – для загрузки геоглифа и последующей тривиальной бинаризации
- skelet – для преобразования изображения в формате OpenCV в изображение формата scikit-image, вызова функции получения скелета из последней библиотеки и обратного преобразования
- getFeatures – основная функция программы, в которой реализована генерация двух признаков по бинаризованному изображению геоглифа согласно описанию в разделе 2.
- solve – решает задачу для входного изображения согласно описанию в разделе 3.

В программе также содержится «демонстрационный» модуль, в котором алгоритм запускается на предоставленных входных изображениях, а результат работы отображается графически. Результаты его работы приведены в данном отчёте.

1.5. Эксперименты

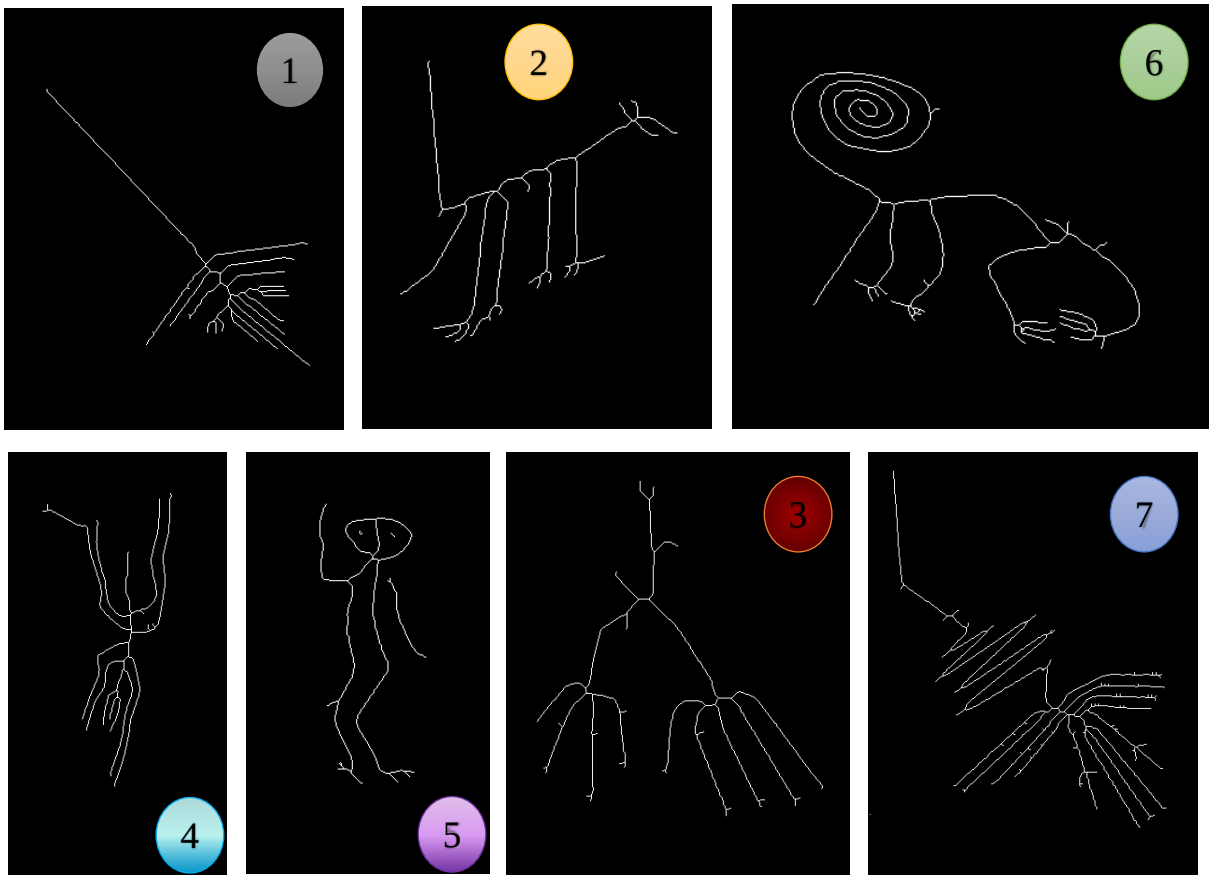
На графике ниже продемонстрированы значения двух признаков, описание которых представлено в разделе 2, для всех входных изображений. Цвет точек означает принадлежность соответствующему классу.



Видно, что все классы успешно отделились, за потенциальным исключением в виде классов 3, 4, которые, хотя и отделены, но уверенной границу между ними назвать нельзя.

2. Генерация признаков

Перед выделением обоих признаков к изображению применяется стандартный алгоритм выделения скелета. Дальнейшая работа ведётся с изображением фигуры, «утончённой до ширины в один пиксель» – множеством центров вписанных в фигуру окружностей.



Первый признак получен нахождением количества вершин в выпуклой оболочке скелета.

Идея второго алгоритма заключается в подсчёте числа точек, в которых скелет «разветвляется» на несколько путей. Подсчёт такого числа затруднён тем, что число точек скелета с числом соседей, больших двух, не отражает его истинного значения из-за шума и приближённого отображения диагональных линий, а также «веток» скелета с малой длиной. Поэтому для получения второго признака запустим следующий алгоритм.

Пример коричневого цвета (не знаю зачем)

Алгоритм начинает работу в самой верхней точке скелета и работает по следующему принципу:

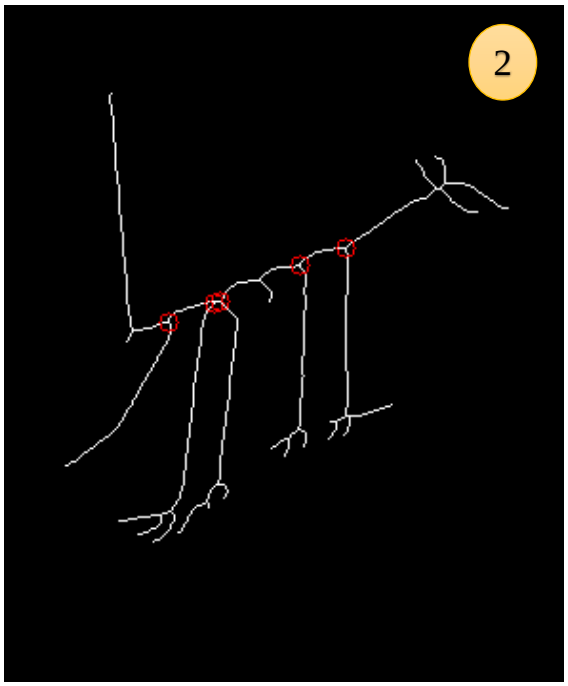
1) если точка не принадлежит скелету, возвращает ноль и заканчивает работу.

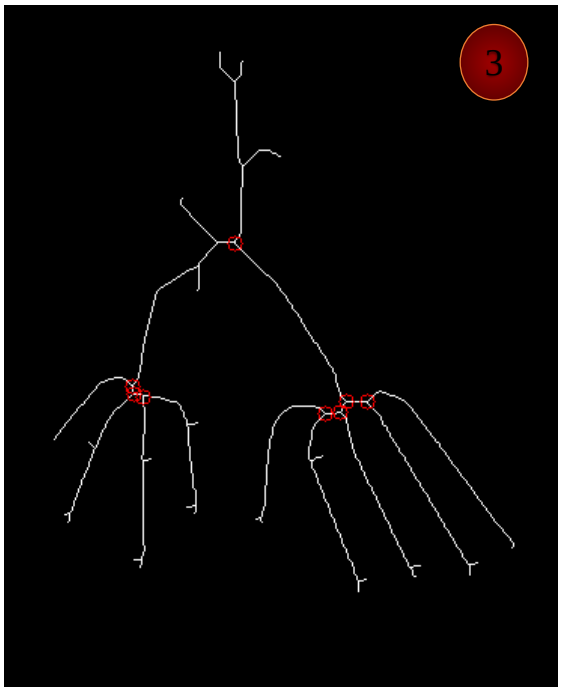
2) иначе удаляет точку из скелета и запускает алгоритм рекурсивно для всех соседних точек по 8-смежности. Все 8 возвращённых значений запоминаются.

3) Если среди этих 8 значений хотя бы два превышают 30, точка объявляется «точкой развилки».

4) Возвращает максимум из этих 8 значений плюс один.

Смысл алгоритма заключается в следующем. В данной точке подсчитывается длина максимального пути, который можно совершить по оставшейся части скелета, начиная от этой точки; сам путь при этом «удаляется». Запуская алгоритм рекурсивно из всех соседних точек, можно получить длины всех «веток», идущих от данной точки. «Ветки» длиной меньше 30 игнорируются и считаются шумом; длина самой длинной плюс один является возвращаемым значением по определению. Если количество веток равно двум и более, точка – «развилка».

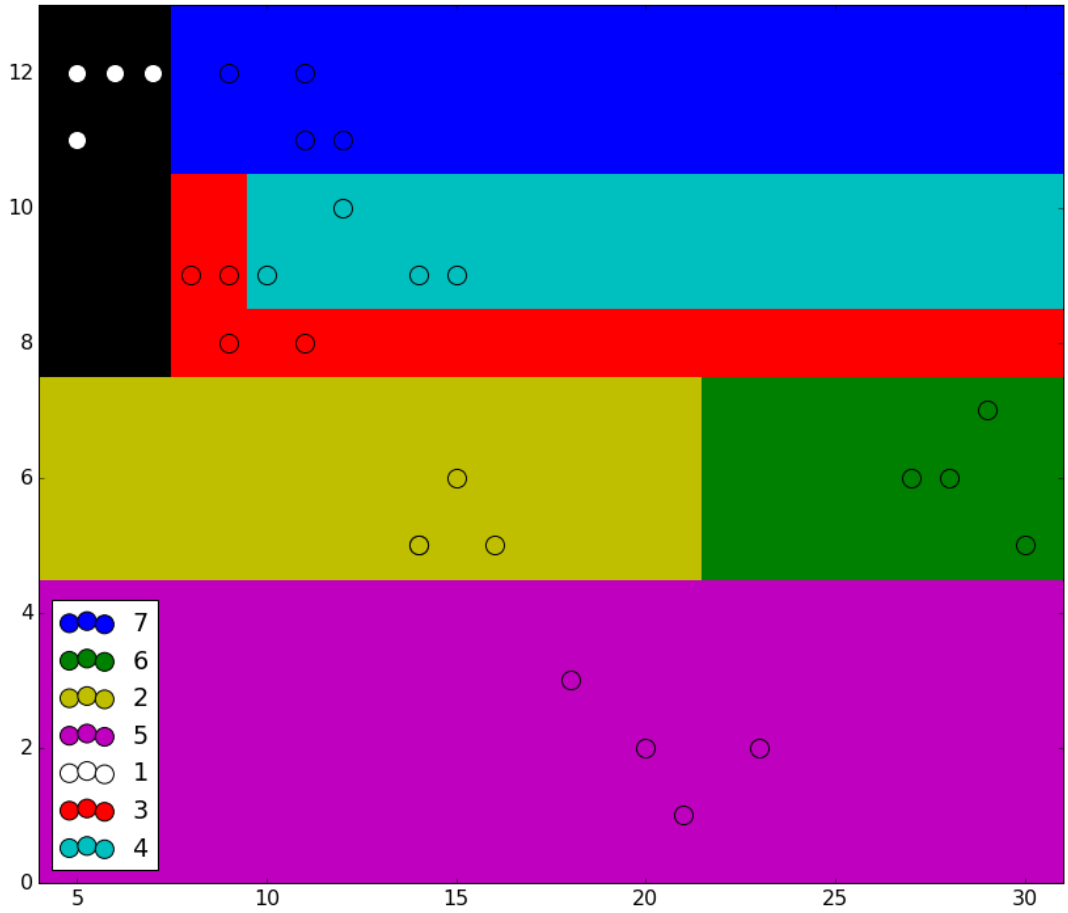




Несмотря на то, что алгоритм не совсем корректен, а выбор порога в 30 довольно условен, количество детектированных развилок довольно стабильно (отличается в рамках одного класса не более чем на 2). Этот признак позволяет разделить почти все классы, за исключением пар 1-7, 2-6 и 3-4. Для разделения этих трёх пар используется первый признак, количество вершин в выпуклой оболочке скелета.

3. Принятие решения

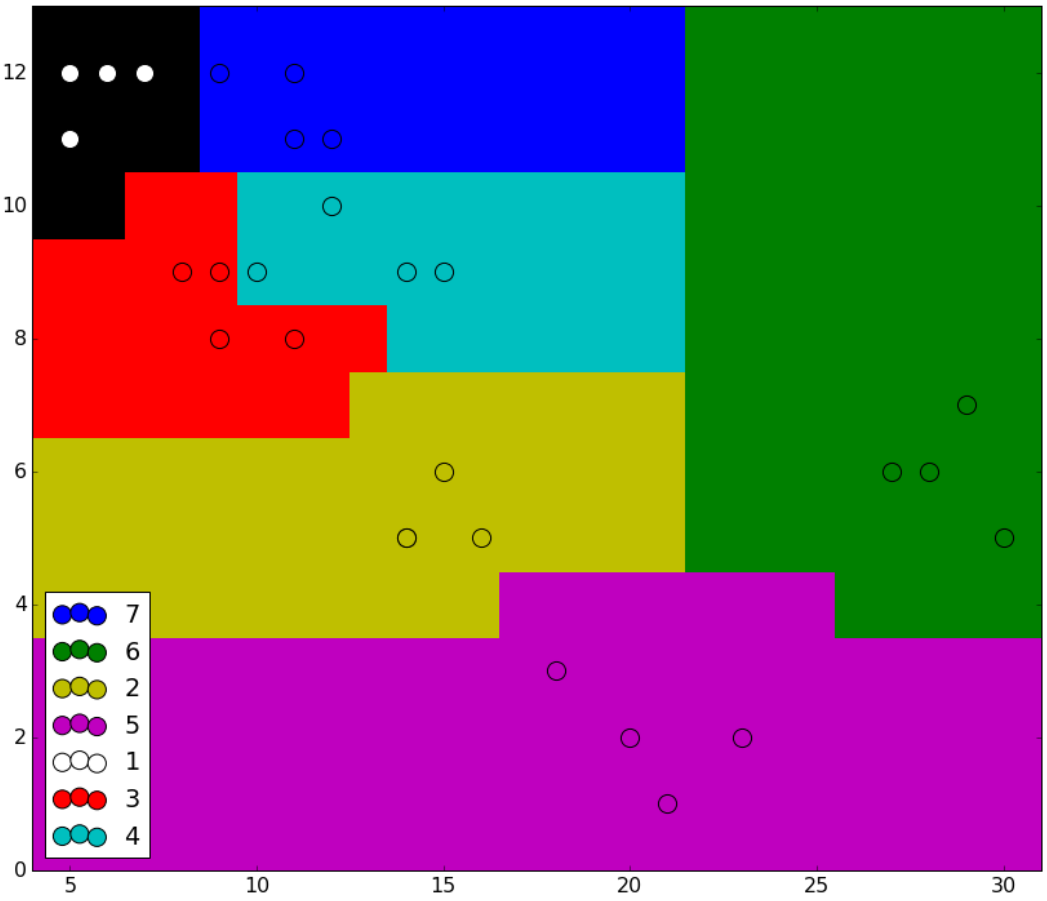
Принятие решения, к какому классу относится изображение, по двум выделенным в разделе 2 признакам, технически является задачей классификации. С другой стороны, в силу дискретности обоих признаков, задача попросту является следующей: для каждой точки $\{x, y \mid x, y \in \mathbb{N}\}$ выбрать метку, то есть номер класса для ответа. Хотелось бы построить такие разделяющие правила, которые были бы интерпретируемы, и при этом разумно реагировали на потенциальные отклонения в значениях признаков. Например, если применить стандартный алгоритм решающего дерева, то можно получить следующий результат:



Такое разделение плохо тем, что далёким от имеющейся выборки точкам дерево присваивает, можно сказать, случайный ответ. Хотелось бы провести разбиение с большей опорой на потенциальные дисперсии признаков для каждого класса, а значит – и с большей опорой на их потенциальные значения. Например, раз для класса 6 велика дисперсия по второму признаку, а по первому она мала, стоит отдать 6-му классу больше пространства вдоль второго признака.

Я решил, что формализовывать это понятия для предложения некоторой математической модели с учётом размера выборки и количества классов

довольно неплодотворно, и провёл разбиение собственным решающим деревом, построенном по моему усмотрению, со следующим результатом:



4. Выводы

Классификация фигур требует выделения признаков, характеризующих каждый из требующих разделения классов. Однако, большинство первых приходящих на ум идей, от тривиальных подсчётов площади до исследования морфологического спектра, неприменимы из-за их неинвариантности к нелинейным трансформациям. Первым и основным шагом преодоления этой проблемы является выделение скелета, который для всех трансформаций одной картинки достаточно неизменчив.

Тем не менее, даже такие признаки, как количество вершин графа скелета степени выше двух («количество развилок») и число вершин в выпуклой оболочке, подвержены шуму от трансформаций. В данной конкретной задаче семь предоставленных классов оказались разделимы по этим признакам, однако при усложнении задачи (например, добавлении новых классов) может потребоваться выделения ещё большей информации, например, о длине рёбер между узлами графа или соотношениях между ними.