

# Отчет о лабораторной работе №2

Обработка и распознавание изображений, ММП ВМК МГУ.

Аристархов Данила Дмитриевич.

Апрель 2024.

## Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
<b>2</b>	<b>Описание данных</b>	<b>2</b>
<b>3</b>	<b>Описание метода решения</b>	<b>2</b>
3.1	Сегментация изображения . . . . .	2
3.2	Поиск фишек . . . . .	3
3.3	Подсчет точек . . . . .	3
<b>4</b>	<b>Описание программой реализации</b>	<b>4</b>
<b>5</b>	<b>Эксперименты</b>	<b>4</b>
<b>6</b>	<b>Выводы</b>	<b>4</b>

# 1 Постановка задачи

Необходимо разработать и реализовать программу для работы с изображениями фишек игрового набора Тримино. Программа должна производить сегментацию изображения, а также подсчет количества точек на фишках. Также необходимо разработать пользовательский интерфейс для работы с программой, обеспечивающий выбор изображения, выполнение операций преобразования и выдачу результата.

## 2 Описание данных

Данные представляют из себя растровые изображения, на которых изображены фишки Тримино на различных фонах. Фишки представляют из себя треугольники с нанесенными на 3 углах точками. Количество точек варьируется от 0 до 5.

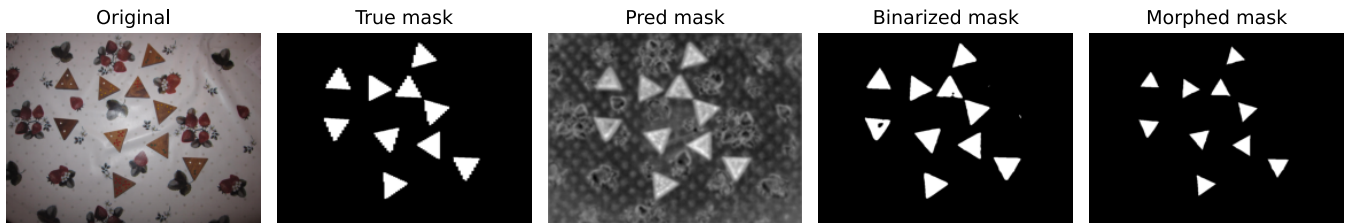


Рис. 1: Пример работы модели сегментации

## 3 Описание метода решения

### 3.1 Сегментация изображения

Для сегментации изображений был применен нейросетевой подход. Для этого изображения предварительно были размечены вручную. В качестве архитектуры была выбрана LinkNet (рис. 2) с использованием энкодера, основанного на VGG13. Обучение производилось с помощью различных аугментаций, таких как вырезание случайного фрагмента изображения, горизонтальное отражение, изменение яркости и т.д.

Для улучшения результата работы модели при постобработке были использованы следующие морфологические операции:

1. Закрытие с ядром (13,13) для устранения полостей в полученной маске сегментации.
2. Эрозия с ядром (13,13) для разъединения на отдельные компоненты близко расположенных треугольников и удаления шума. Также это преобразование позволяет отделиться от границы треугольника, что поможет на этапе подсчета точек.

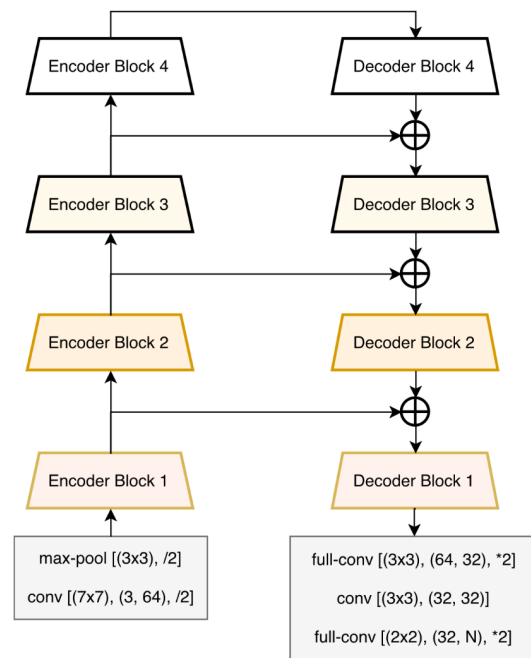


Рис. 2: Архитектура LinkNet

## 3.2 Поиск фишек

Для поиска отдельных фишек сначала выделялись границы полученной маски. Затем эти границы приближались многоугольниками. Из этих многоугольников выбирались треугольники. Таким образом удастся выделить вершины фишек, и с помощью этого вычислить центр фишки.

## 3.3 Подсчет точек

Подсчет точек происходит в несколько этапов:

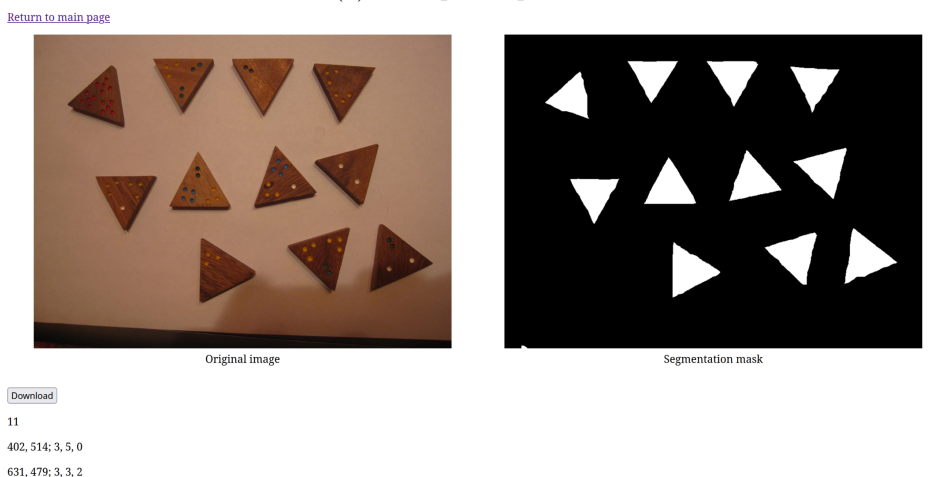
1. Считается градиент изображения по горизонтали  $G_x$  и по вертикали  $G_y$  отдельно по цветовым каналам.
2. Вычисляется общий градиент  $G = |G_x| + |G_y|$ .
3. Получаем маску с помощью максимума градиента по каналам.
4. Эта маска размывается по Гауссу с ядром размера (3, 3).
5. Производится бинаризация Оцу полученной маски.
6. Для каждой вершины треугольников берется ее окрестность. Эта окрестность представляет из себя пересечение маски треугольника, полученной при сегментации, и окружности с радиусом, равным расстоянию от вершины до центра треугольника.
7. В этой окрестности ищется количество связных компонент. При этом происходит отсев слишком маленьких компонент.

Число связных компонент и является количеством точек возле вершины фишки.

# Triangles

Select image:  Pict\_2\_1.bmp

(a) Выбор изображения



(b) Результат работы программы

Рис. 3: Этапы работы с программой

## 4 Описание программой реализации

Программа была написана на языке программирования **Python**. Для работы с изображениями использовалась библиотека **OpenCV**. Для обучение нейросети был использована библиотека **PyTorch**, для аугментаций — **Albumentations**. 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. 3](#)): сначала пользователь выбирает изображение для обработки. Далее происходит выдача результата в виде маски сегментации исходного изображения, а также координаты центров фишек и результат подсчета количества точек в углах. Имеется возможность скачать результат в виде файла формата `txt`.

## 5 Эксперименты

Алгоритм отлично себя показал на этапе сегментации исходного изображения. Нейросеть хорошо выделяет фишки на любом фоне при любом освещении. С помощью морфологии удастся устранить различные неточности нейросети: убрать шум, отделить треугольники друг от друга и заполнить полости внутри них. Также выделение только треугольных компонент позволяет убрать лишние компоненты и получить координаты вершин треугольника.

Алгоритм подсчета точек тоже хорошо справился с задачей. Однако на некоторых иногда результаты получаются неточными. В первую очередь это происходит из-за неоднородности фишки, что влечет появление градиента на фишке не в местах расположения точек. Также алгоритм имеет трудности с классификацией 5 точек, поскольку они плотно расположены и их трудно отделить друг от друга. Возможным решением данной проблемы является фотографирование фишек более крупным планом.

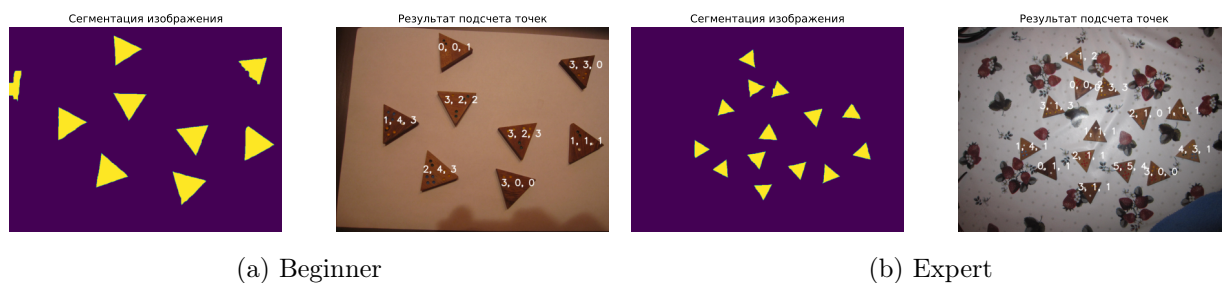


Рис. 4: Эксперименты

## 6 ВЫВОДЫ

Предложенный алгоритм смог добиться высокой точности при сегментации изображения. Классификация фишек происходит менее точно, однако выдает результат, близкий к реальному количеству точек. В целом алгоритм смог справиться с поставленной задачей.