Министерство науки и высшего образования Российской Федерации

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»

по теме:

«Распознавание дорожных знаков на изображениях»

Выполнил:
Студент 3-го курса,
гр. БПМ-21-2 Сироткин С.Ю.

Проверил:
доцент, к.т.н. Полевой Д.В.

Москва, 2024

СОДЕРЖАНИЕ

1 Описание задачи

1.1 Краткое описание задачи

Для работы современных беспилотных автомобилей и систем помощи водителям необходимо находить и корректно распознавать знаки ПДД на изображениях с камер. Работа заключается в исследовании методов детекции и классификации дорожных знаков на изображениях и разработке алгоритма, применяющего эти методы.

1.2 Формальное описание задачи

Формальная задача состоит в разработке методики детекции и последующей классификации знаков ПДД на изображениях с камер. Это включает в себя:

1. Предварительная обработка изображения для удобного и эффективного применения последующих методов
2. Разработка алгоритма детекция участков, содержащих дорожные знаки
3. Разработка алгоритма классификация знаков
4. Подготовка датасета изображений для оценки качества работы алгоритма
5. Проведение экспериментов и вывод метрик

1.3 Описание оценки качества

Для оценки качества был собран датасет из изображений, содержащих ровно 1 дорожный знак.

Введем следующие метрики оценки качества:

1. Точность детекции:

$$Detection\ Accuracy = \frac{M}{N},$$

где M – количество картинок, на которых был обнаружен ровно 1 знак, N – общее количество входных изображений

2. Точность классификации:

$$Classification\ Accuracy = \frac{L}{M},$$

где L – количество корректно классифицированных знаков, M – общее количество распознанных знаков.

3. Частота ложно-положительных распознаваний:

$$False - positive\ rating = \frac{F}{N},$$

где F – общее кол-во ложно-положительных распознаваний, N – общее число входных изображений.

2 Решение задачи

2.1 Детекция знаков

Алгоритм детекции описан в библиотеке *detection* в классе *Detection*

Многие дорожные знаки имеют характерные цветные контуры, которые можно выявить с помощью цветовой сегментации. Например, запрещающие знаки, как показано на рисунке 1, обычно окружены красным контуром. Этот цветовой элемент может быть использован в качестве основного признака для детекции таких знаков.

Рисунок 1 – запрещающие знаки ПДД

Операция детекции знаков с использованием цветовой сегментации включает следующие шаги:

1. Преобразование цветового пространства в пространство HSV (Hue, Saturation, Value) – это позволяет более легко и точно работать с оттенками и интенсивностью цветов
2. Определение диапазона цветов. Для красного цвета были подобраны следующие диапазоны (т.к. красный цвет находится на обеих границах шкалы оттенков, будем использовать два диапазона):

```
std::vector<ColorRange> RED_RANGES = {
    { cv::Scalar(0, 50, 90), cv::Scalar(9, 255, 255) },
    { cv::Scalar(171, 50, 90), cv::Scalar(180, 255, 255) },
}
```

3. Создание битовой маски:

```
cv::inRange(image, range.lower, range.upper, tmp);
```

В случае нескольких диапазонов, выполняем объединение результата *tmp* с основной маской *mask*

```
mask |= tmp;
```

4. Применение морфологических операций, таких как закрытие и открытие (включающих в себя эрозию и дилатацию), помогает устранить шум и улучшить качество маски, что способствует более точной детекции контуров дорожных знаков.

```
cv::Mat kernel = cv::getStructuringElement(cv::MORPH_RECT,
cv::Size(5, 5));
cv::morphologyEx(mask, mask, cv::MORPH_CLOSE, kernel);
cv::morphologyEx(mask, mask, cv::MORPH_OPEN, kernel);
```

5. Поиск контуров и удаление внутренних контуров

```
auto contours = getContours(mask);
mask = cv::Scalar(0);
cv::drawContours(mask, contours, -1, {255}, 1);
```

Рисунок 2 – исходное изображение и получившаяся маска

Помимо цветовой характеристики, знаки имеют строго определенную геометрическую форму. В случае запрещающих знаков (рис. 1) – это окружность. В работе остановимся только на запрещающих и предупреждающих знаках (имеют округлую и треугольную форму соответственно). Для детекции окружностей основным алгоритмом было выбрано преобразование Хафа. Для треугольников – аппроксимация контура с помощью алгоритма Рамера-Дугласа-Пекера. В случае, если контур является треугольником – на выходе алгоритма получится 3 точки. Обнаружение окружностей и треугольников реализовано в методах *getCircles* и *getTriangles* класса *Detection*.

2.2 Классификация знаков

Для классификации будем использовать те же качественные характеристики знаков, что и для детекции: цвет и геометрическую форму.

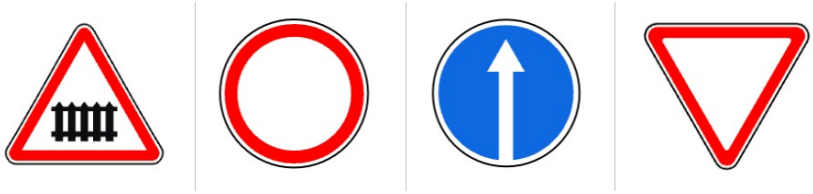

Рисунок 3 – виды знаков (предупреждающие; запрещающие; предписывающие; знак «Уступите дорогу»)

В данной работе остановимся на определении вида предупреждающих, запрещающих и знака «Уступите дорогу» (рис. 3).

3. Результат

3.1 Инструкция по сборке и описание структуры проекта

Исходный код: https://github.com/sssemion/misis2024s-21-02-sirotkin-s-u

Требования к окружению:

- CMake минимальной версии 3.24
- Язык C++ стандарта 17
- Библиотека opencv2
- Библиотека CLI11

Инструкция:

1. Склонировать репозиторий

```
git clone https://github.com/sssemion/misis2024s-21-02-sirotkin-s-u
cd misis2024s-21-02-sirotkin-s-u
mkdir build && cd build
```

2. Собрать проект

```
cmake .. && cmake --build
```

3. Запустить бинарник

```
./bin/main --src <Путь к файлу с изображением> [--dst <Путь к файлу с размеченным изображением>]
```

3.2 Результат

В простых случаях (с хорошим освещением и отсутствием побочных объектов красного оттенка) (рис. 4, 5) алгоритм справляется отлично. Например, на рис. 6 алгоритм не обнаружил знак ограничения скорости, т.к. его контур сливается с задним фоном, вследствие чего его контур на маске не является окружностью.

Рисунок 4

Рисунок 5

Рисунок 6

На датасете из порядка 20 изображений алгоритм показал следующие метрики оценки качества:

Detection accuracy: 58.82%

Classification accuracy: 80.00%

False positive per image: 0.00

Список использованных источников

1. Booysen, M. J., De Weerdt, D., Kroon, D., & Herbst, B. M. (2009). - "Towards Detection of Road Traffic Signs in a Mobile Environment" - B: “Proceedings of the IEEE Intelligent Vehicles Symposium”
2. Gonzalez, R. C., & Woods, R. E. (2018). – “Digital Image Processing (4th Edition)”.
3. OpenCV Documentation - URL: https://docs.opencv.org/
4. Kaggle Datasets - URL: https://www.kaggle.com/
5. ПДД РФ (Правила дорожного движения Российской Федерации).