УДК004

ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ СИМВОЛОВ

Потапов В.К., студент гр. ИТб-161, IV курс
Научный руководитель: Асанов С.А.
Кузбасский государственный технический университет имени Т.Ф. Горбачева, г. Кемерово

На сегодняшний день, распознавание символов по изображению является одной из самых важных задач. Автосервисы, службы геолокации, автозаправки, ГИБДД, закрытые объекты - все они нуждаются в достойной технологии, которая будет автоматически определять гос. номер автомобиля или его VIN-номер.

Одной из самых главных задач является поиск границ для определения номера. Существует несколько способов поиска. Один из них - поиск прямоугольного контура, но при условии хорошо читаемого номера, явно отображенного контура и отсутствии преград между камерой и номерной рамкой. В основе данного метода лежит фильтрация изображения для нахождения границ, а затем производится выделение всех найденных контуров и их анализ. Минусы - искомый контур может быть грязным, сливаться с цветом автомобиля, так же может быть преграда между камерой и искомым контуром. Следующий - поиск, при котором анализируется только часть рамки номерного знака. Выделяются контуры, затем производится поиск вертикальных прямых. Исходя из полученных данных, на основе двух вертикальных прямых, выдвигается гипотеза о том, что номер находится где-то между ними. Также, существует «гистограммный анализ регионов» - выдвигается предположение, что частотная характеристика региона с гос.номером/VIN-номером отлична от частотной характеристики фона вокруг автомобиля. Один существенный недостаток данного метода - автомобиль должен быть сопоставим с размером изображения, по которому ведётся поиск, т.к. фон может иметь надписи или другие высокочастотные элементы.

Лучшими методами являются те, которые основаны на различных классификаторах. Они позволяют анализировать определенную область изображения на наличие в ней характерных для номера отношений, точек или градиентов. Могут быть основаны на методе Виолы-Джонса, примитивах Хаара. На данный момент, существует большое кол-во библиотек для распознавания.

1. **Проект Opos.** Реализован на C#. Для распознавания номера используется библиотека компьютерного зрения opencv, а для распознавания самих символов, библиотека CuneiFrom - puma.NET. Одним из минусов является то, что изображение нужно сохранить на диск для передачи в puma.NET. http://opos.codeplex.com

2. **Проект JavaANPR.** Реализован на Java. Основной плюс - кроссплатформенность, все алогритмы написаны без использования каких-либо библиотек, легко сортируется на Android, скорость распознавания 0.2 - 0.8 сек. Но, при всех плюсах, плохо работает с зашумлёнными изображениями. http://javaanpr.sourceforge.net

3. **Проект Automatic License Plate Recognition.** Реализован на C#. Использует две библиотеки: opencv - Emgu для поиска номера и tesseract OCR для распознавания самого номера. Работает с кириллическими символами.

Для реализации проекта был выбраны библиотеки Opencv и Tesseract.

-Одним из главных критериев является скорость работы. Пользователь должен как можно скорее загрузить требуемое изображение, а приложение выполнить распознавание. Проект Opos требует достаточно большого количества времени для загрузки изображения на диск, сам поиск тоже занимает некоторое время. Также, хорошим и быстрым решением был бы JavaANPR.

-Также, учитывался язык реализации, Opos - C#, Tesseract + Opencv - C#, JavaANPR - C#. В приоритете был C# из-за его удобности и простоты.

-Следующим критерием являлась кроссплатфроменность. В дальнейшем, заказчик может потребовать перенести разработанное ПО на Android/IOS. Из вышеизложенных предложений, только Opencv + Tesseract удовлетворяют требованиям. Также, возможен порт JavaANPR на мобильные устройства, но только на Android.

-Учитывалась возможность распознавания изображений плохого качества. JavaANPR имеет проблемы с зашумлёнными изображениями, с перегруженными изображениями, на которых много мелких символов и деталей. Opos - не всегда точно определяет изображения, на которых цвет фона совпадает с цветом искомых символов. Оптимальный выбор это Tesseract + Opencv.

Итог:

Как показывают эксперименты, скорость обработки картинок связкой Opencv + Tesseract OCR достаточно высока, вероятность ошибки сведена к минимуму. Это позволяет использовать данные библиотеки в системах распознавания в режиме реального времени. Реализация может быть кроссплатформенной, что позволит перенести данное приложение на Android/iOS девайсы. Язык реализации - C#, что позволит, при необходимости, легко и быстро редактировать программный код.

Список литературы:

1. https://docs.microsoft.com/ru-ru/azure/cognitive-services/computer-vision/quickstarts/csharp-print-text

2. https://habr.com/ru/post/471542/