スマートデバイスを入力として用いた情報集積・分析システムの構築

Construction of the Information Collection and Analysis System Using a Smart Device as an Input

長倉 貴洋[*1]
Takahiro NAGAKURA[*1]
[*1] 国立大学法人 琉球大学 工学部 情報工学科
[*1]Department of Information Engineering, Faculty of Engineering, University of the Ryukyus
Email: e125719@ie.u-ryukyu.ac.jp[*1]

あらまし：2010 年度から爆発的な普及を遂げたスマートデバイスにより，スマートデバイスは高いポータビリティを誇っている．また，スマートデバイスは，以前に普及していた PHP やフィーチャーフォンと比較して，圧倒的な性能差があり，非常に多機能なデバイスである．その多機能性を利用したシステムも多く開発されており，スマートデバイス普及以前では実現が難しかった様々なシステムが実用化され，多くのユーザに利用されている．本研究では，スマートデバイスのポータビリティの高さとそれらの多機能性に着目し，「ポータブルデバイスを入力とする情報収集システムの開発」を行なった．
キーワード：スマートデバイス，ポータビリティ，情報収集

1. はじめに

総務省の平成 26 年度通信利用動向調査[1]によると，2010 年におけるスマートフォンの保有率が 9.7%，タブレット型端末が 7.2%であるのに対し，2015 年ではスマートフォンが 64.2%，タブレット型デバイスが 26.3%となっており，過去 5 年間でスマートフォンでは 6.62 倍，タブレット型デバイスでは 3.65 倍の爆発的普及を実現している．近年では，1 人がスマートデバイスを複数台所有していることも珍しくない．

また，スマートデバイスは GPS による位置情報の送受信や複数端末との通信など，多くの機能を実装しており，非常に機能性に優れている．

本研究では，スマートデバイスのポータビリティの高さとその多機能性に注目し，「ポータブルデバイスを入力とする情報収集システムの開発」を行なった．

2. 開発環境

2.1 iOS アプリケーション

1) Objective-C

Objective-C とは，Mac OS X/iOS アプリケーション開発言語であ

if/for/while などの制御文や，int/char/float などのスカラー型，関数記法，宣言や代入といった基本的な文法は C 言語に準じているが，オブジェクト指向は Smalltalk の概念を借用している．

2) OpenCV

OpenCV とは，オープンソースのコンピュータビジョンライブラリである．C/C++，Java，Python，MATLAB で利用可能であり，BSD ライセンスで配布されているため，商用目的でも使用可能である．

OpenCV には，フィルター処理，変形処理，構造解析と形状ディスクリプタ，物体検出，モーション解析と物体追跡，領域分割，カメラキャリブレーション，特徴点検出，機械学習が実装されている．

3) Tesseract-OCR

Tesseract-OCR とは，光学的文字認識ライブラリである．

Tesseract-OCR は 2016 年 1 月現在，64 の言語に対応している．また，ニューロンネットワークによる機械学習を実装しているため，対応言語の教師データを与えることで，その言語の光学的認知度を向上させることが可能である．

2.2 Web アプリケーション

1) Ruby on Rails

Ruby on Rails とは，オープンソースの Web アプリケーションフレームワークである．

Ruby on Rails はその開発理念に「同じことを繰り返さない(DRY：Don’t Repeat Yourself)」と「設定より規約(CoC：Convention over Configuration)」を掲げており，実アプリケーションの開発が他のフレームワークよりも少ないコードで簡単に行える．

2) PostgreSQL

PostgreSQL とは，オープンソースのオブジェクト関係データベース管理システムである．

データベースの操作には SQL データベース操作構文を利用しているが，PL/pgSQL，PL/PSM，スクリプト言語(PL/Perl, PL/php, PL/Python, PL/Ruby, PL/Tcl, PL/Lua)，コンパイラ言語(C 言語, PL/Java)，統計処理言語(PL/R)の関数にて実行することも可能である．

3) Heroku

Heroku とは，PaaS(Platform as a Service)を提供している企業，および提供されているサービスの名称である．

PaaS とは, アプリケーションの公開に必要な「ハードウェア/ネットワーク/仮想環境/OS/DB/アプリケーションフレームワーク」のうち, アプリケーションフレームワーク以外のすべてをサービス事業者が管理し, サービス利用者はアプリケーションフレームワークのみを管理する形態である.

3. システム概要

本システムの利用の流れは, 以下のようになっている.

a) iOS アプリケーション
   1. iOS アプリケーションにて, 車両のナンバープレートを撮影
   2. iOS アプリケーションの内部処理にて, 撮影画像から車両ナンバーを文字列として取得
   3. 取得した車両ナンバーを Web アプリケーション側の PostgreSQL に送信

b) Web アプリケーション
   1. Web アプリケーションにログイン
   2. PostgreSQL データベースを操作し, 登録情報を確認
   3. Web アプリケーションからログアウト

今回構築したシステムのフローは, 以下のようになっている.

c) iOS アプリケーション
   1. iOS デバイスのカメラを利用して, 撮影画像を取得
   2. 取得した画像を, OpenCV の「特徴点検出」を利用して, 注目したい領域を制限(ナンバープレートのみに注目)
   3. 注目した領域内の文字列(ナンバープレート内の文字情報)を, Tesseract-OCR の光学的文字認識を用いて取得
   4. 取得した文字列(車両ナンバー)を, PostgerSQL に POST

d) Web アプリケーション
   1. ユーザに対してログイン要求
   2. データの閲覧, 検索, 追加, 削除を行うための SQL 構文を PostgerSQL にリクエスト
   3. 利用後はユーザに対してログアウト要求

図 1　iOS アプリケーション

図 2　Web アプリケーション

4. 現時点での評価

現時点では, iOS アプリケーションにおける Tessarect-OCR の光学的文字認識によるナンバープレートの文字列認識率が低い.

その原因として, 以下が考えられる.

“ナンバープレートに注目した際, 「〇〇-〇〇」部分が, 他の文字よりも大きいため, それを数字として認識することが難しく, 黒塗りの領域であると Tesseract-OCR が判断してしまう.”

この問題の解決策として,

“接写しなくてもよいように, 特徴点検出の精度を調節し, Tesseract-OCR がナンバープレートの数字を文字として認識できるようにする.”

ことが考えられるが、今回は実装に至っていない.

5. まとめ

本研究にて構築した「車両認識システム」であるが, 車両ナンバーの認識率が甘く, まだ実用には至らない段階である.

今後は, 認識率を向上させるために, 画像処理部分のアルゴリズムの再検討や実装が大きな課題となっている.

参考文献

(1) Wikipedia. Objective-C. “https://en.wikipedia.org/wiki/Objective-C” (2016 年 1 月 16 日アクセス).

(2) Wikipedia. OpenCV. “https://en.wikipedia.org/wiki/OpenCV” (2016 年 1 月 18 日アクセス)

(3) tesseract-ocr. README. https://github.com/tesseract-ocr/tesseract/blob/master/README.md (2016 年 1 月 27 日アクセス).