

多様な言語から使用できる教育用WebAPIライブラリの開発

Development of educational Web API that can be used from various languages

鈴木 達也† 島袋舞子† 小関啓子† 兼宗進†

tatuya suzuki Maiko simabuku keiko koseki susumu kanemune

1. はじめに

平成30年に告示された高等学校情報科の学習指導要領[1]では情報 I において「コンピュータとプログラミング」が追加され、すべての高校生がプログラミングを学ぶことになっている。その学習内容の1つに外部ライブラリやOSやサーバが提供するAPIを用いたプログラムがある。

本研究ではサーバが提供するWeb APIに着目した。既存のWeb APIとして、各国の気象データを提供するもの[2]や政府の統計データを提供するもの[3]がある。これらの多くは利用にあたってユーザー登録が必要となるため、授業で扱おうとすると生徒一人ひとりが登録する必要がある。利用登録が不要なWeb APIも存在するが授業内容に即したものが多くないため、Web APIは授業で扱いにくいといった課題が存在する。

そこで本研究ではユーザー登録が不要で利用するとともに、情報 I で扱う内容に即した機能を提供する教育用Web APIを提案する。今回はその1つである画像処理の機能の開発について報告する。

2. 画像処理機能の開発

画像処理の機能は画像の加工が可能な外部ライブラリである「OpenCV」を使用して開発した。OpenCVには画像処理に関する関数が多く用意されている。今回はグレースケールや二値化といった画像処理や顔検出の機能を教育用Web APIにて提供する。現在対応する機能を表1に示す。実装にはPythonのWebアプリケーションフレームワークのライブラリである「Flask」を使用した。

表1において、「顔検出」のカスケード分類器はあらかじめ用意したものを使用している。「ぼかし」は「平滑化フィルタ」、「メディアンフィルタ」、「ガウシアンフィルタ」、「バイラテラルフィルタ」の4種類でWeb APIに対応している。また、二値化を行うには閾値の下限と上限を決める必要がある。そこで、Flask内にある「request」機能を用いて学生から任

意の下限と上限を決めてもらいWeb APIを呼び出すことで二値化ができるようにしている。

表1 現在対応している処理内容の一覧

処理名	処理内容
グレースケール	カラー画像を白黒(グレースケール)にする。
二値化	カラー画像の明度を計測し、条件に応じた白黒(灰色なし)画像を生成する。
ハフ変換	画像の中から直線や円などの図形を検出する。
マスク画像を利用した画像加工	マスク画像を利用し元画像に対しての画像加工を行う。
顔検出	カスケード分類器を用いて、人物の画像から顔を検出する。
ぼかし加工	画像に対してぼかし加工を行う。
ネガポジ反転	画像の色を調べ、色相環において反対の色に変換を行い加工する。

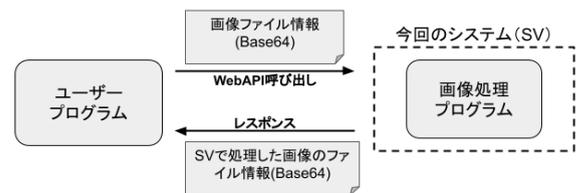


図1 システムの概要

†大阪電気通信大学,

システムの概要を図1に示す。まず、ユーザーのプログラムからWeb APIを呼び出し、画像ファイルをWeb APIに送信している。ただ、今回のWeb APIの開発環境であるFlaskでは画像ファイルをそのまま受け取ることが出来ない。そこで、Base64と呼ばれるエンコーダ方式に変更することで画像ファイルの受け取りを行っている。Web APIが画像を受け取り、処理を行う。処理した画像ファイルをユーザのプログラムに返す際もBase64のエンコーダ方式でレスポンスを行っているため、ユーザー側でBase64を画像ファイルに変換するクラスファイルを準備する必要がある。

具体例として、「ドリトル」でグレースケール加工のWeb APIを使用した際のソースコードを図2に、処理前と処理後の画像を図3、4に示す。

```
wa:WebAPI!作る。
wa:url="http://グレースケールのURL"。
ret=wa!読む。
画像=ret.graph。
こまんど=( ".jre\bin\java Base64Decode" )!(画像)連結。
システム! (こまんど)実行!
ラベル! "<html><img
src=file:result.png></html>"作る。
```

図2 ドリトルでのグレースケールの実行例

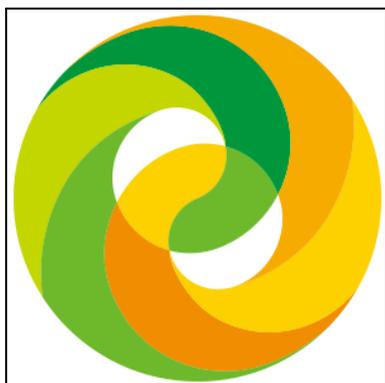


図3 グレースケール加工前の画像



図4 グレースケール加工後の画像

ドリトル[4]とは兼宗研究室で開発を行っているプログラミング言語で日本語でのプログラムを可能としている。図2で1行目にWeb APIオブジェクトを作成し、2行目で画像の処理方法をWeb APIに送信している。3行目で処理結果を呼び出し、4行目で処理結果の文字コードの入手を行っている。5行目と6行目で処理結果の文字コードを画像ファイルに変換して7行目に変換した画像ファイルを表示している。

3. おわりに

今回は画像処理を行える教育用WebAPIの提案を行った。今後の展望としては、WebAPIのコンテンツの充実を目指している。今回紹介している画像処理のほかにもアルゴリズム学習用の乱数データやMediaPipeなどの骨格検出をWebAPIにしていきたい。

参考文献

[1]文部科学省.情報編 高等学校学習指導要領(平成30年告示)解説.2018-7.(参照2023-07-17)
https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_01831.html

[2]OpenWeatherMap(<https://openweathermap.org/>)

[3]e-Stat(<https://www.e-stat.go.jp/api/>)

[4]兼宗研究室,プログラミング言語「ドリトル」,2022/11/26.
<https://dolittle.eplang.jp/>