

安価な組み込み教材で高度な画像認識を可能にするサーバー連携方式の教材開発

伊藤匡祐[†] 漆原宏丞[†] 大村基将[†] 疋田真一[†] 兼宗進[†]

[†]大阪電気通信大学

1 はじめに

画像認識処理を伴うプログラムの実行には一定以上の演算能力を必要とする。このことはロボット制御を目的としたプログラミングの学習においては、教材として利用できる制御コンピュータ等の選択肢を狭め、教材価格の高騰につながる。また、コロナ禍に伴う半導体不足により、教材となるコンピュータ自体が供給不足に陥り、画像認識等を含む制御プログラミング授業実施の障害となる事案も起こっている。このことから、画像処理等の負荷の高い処理等を利用する場合であっても、安価かつ安定的にそれらを利用することができる教材システムが必要であると考えた。

そこで、サーバーで高度な画像処理などを行うことで、安価な小型マイコンで高度な処理を行う仕組みを開発し、実際に大学の授業で使用したので報告する。

2 サーバ連携型制御教材システム

本システムは、画像認識等の負荷の高い処理や機能を WebAPI を介してインターネット上のサーバーに処理を移譲することで、Wi-Fi 接続可能なマイコンであればその性能に依らず利用できる制御プログラミング教材を目指した。図 1 にシステムの全体構成を示す。

本教材は、マイコンに接続したカメラで静止画を撮影し、それを画像認識を利用しながら車輪型のロボット制御に適用するプログラミング学習を想定した。学習者がプログラミングを行う制御対象としては、Wi-Fi 機能を持ちコロナ禍においても低価格で比較的手取り性がよく、授業実施がマイコンの供給状況に左右されない点をふまえて、esp32 マイコンとカメラが一体となっている esp32UnitCAM を対象とした。

Development of teaching materials using a server-connected method that enables advanced image recognition with inexpensive embedded teaching materials
Kyoosuke Ito[†], Kosuke Urushihara[†], Motomasa Omura[†], Shinichi Hikita[†], KANEMUNE Susumu[†]
[†]Osaka Electro-Communication University
572-8530, Neyagawa, Japan

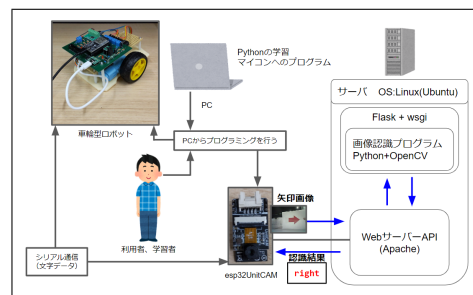


図 1: 全体構成

2.1 機能提供サーバ

画像処理機能を提供するサーバについては、Python 言語と OpenCV ライブラリで実現した。WebAPI を提供するため、Python のプログラムを Flask と WSGI で Web サーバー (Apache) と連携して動作するように設定した。上記は教員が容易に準備ができるよう、レンタルの仮想専用サーバ (VPS) を用いて構築した。表 1 サーバが提供する機能を示す。WebAPI は提供機能ごとに URI をもち、機能利用時はパラメータとして「ユーザ ID と HSV 色空間での抽出色の範囲、および静止画データ」を設定して送信する。このとき、サーバに対しては WebAPI を介して画像データや画像認識のためのパラメータを POST 通信として与える。サーバは指定された色範囲に含まれた色を抽出し、そこから URI に応じて色の面積や矢印の向きを求め、その結果を文字列および数値で返す。

表 1: サーバが提供する画像認識機能

機能	内容
矢印検出	画像から指定色の矢印を検出し向きを判定する
面積取得	画像から指定色のピクセル数を算出する

なお、矢印の判定については、OpenCV を用いて画像全体に対して指定範囲色のピクセルの偏り（画像の重心）を求めることで判断する。この方法は学生が理

```
//esp32UnitCAM側プログラム
String url = "http://app.garakuta-lab.net:5000/get-arrow-direction";
String payload;
String param = "user=Eh00A000&h_lower=160&s_lower=50&v_lower=50
&h_upper=180&s_upper=255&v_upper=255";
uint8_t httpCode = callWebAPI(url,payload,param,fb);
serial.print2(payload);

//ロボットカー側プログラム
if (mySerial.available() > 0) {
  delay(20);
  String command = mySerial.readString();
  if(command == "right"){
    motorOutputL(150);
    motorOutputR(-150);
  }
}
```

図 2: プログラム上での WebAPI 利用例

解しやすく、実際に学生が容易に実現できるアルゴリズムである。また、ユーザがデバッグをする際の利便性を考慮し、機能提供サーバが行った画像認識処理の結果を PC やスマートフォンのブラウザでリアルタイムに取得できる Web アプリも併せて開発した。

2.2 制御プログラム

学習者は、Arduino 言語を用いて WebAPI を用いた制御プログラムを記述する。WebAPI の呼び出しについては、HTTP リクエスト/レスポンス等の知識がなくても利用できるようにするため、WebAPI 呼び出しのための機能をライブラリとして提供する。図 2 にプログラムでの記述例を示す。図中で callWebAPI() は利用のための関数であり、利用する機能に応じた URL、パラメータ、データを与える。委譲した処理の内容は引数(図中の payload)に格納され、以降の処理に利用することができる。

3 授業実践

2022 年度に工学部の 2 年生を対象とした「メカトロニクス実践演習 2」の演習授業の中で本教材を使用した。

3.1 授業内容

表 2 に実践授業の概要を示す。授業では、前半の 8 回で事前に Python による画像処理のプログラムを学習し、後半部にて本システムを用いてカメラで読み取った矢印の向きを判定し、自律走行するプログラム制作実習を行った。前半の学習では「機能提供サーバ」の内部処理を実際と同じ内容を学習することで、制御実習時の利用する WebAPI の処理がブラックボックスにならないように配慮した。自立走行ロボットの制御演習では、H 型のコース上に矢印等を記載した障害物を設置し、自立走行させる課題を指示した。

表 2: 授業内容

授業回数	授業内容
1,2	ガイダンスおよび開発環境の設定
3	Python による画像処理の基礎
4	色範囲を指定して画像から色の抽出
5	指定した色面積の取得
6	矢印の重心の算出
7	画像認識処理の関数作成
8	カメラ画像の画像認識実践
9	ロボットカー制御方法の確認
10	WebAPI 利用方法の確認 (遠隔制御)
11	esp32UnitCAM の動作確認
12	WebAPI による画像認識の利用
13-15	自律走行ロボットの制御演習

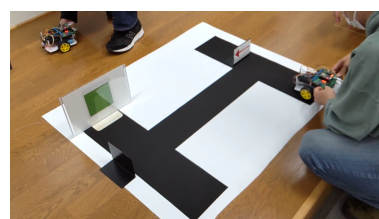


図 3: 授業風景

3.2 授業結果

図 3 に授業の様子を示す。授業では、WebAPI のプログラム上での利用方法に困難を感じる学生は少なかった。ただし、事後に実施したアンケートでは、WebAPI を利用した画像認識制御プログラムの作成の難易度を問う質問で 24.3% の学生が困難を感じたと回答した。アンケート回答の中には処理遅延やネットワーク起因のトラブルの対応に困難を感じる学生も多く、これが画像認識プログラム作成を困難と感じる要因となった可能性が考えられる。

また、WebAPI の利点を問う記述式の質問の回答文に共起ネットワーク分析を行ったところ、「処理の軽減や委譲に関する回答群」、「開発コスト効率に関する回答群」、「開発難易度の軽減に関する回答群」を確認することができ、一定の学習効果を得られたことが示唆された。

4 おわりに

画像認識などの高度な処理をサーバー側にデータを転送して処理することで、安価な小型マイコンを用いて画像認識を導入した制御プログラミングの演習を行える教材の仕組みを開発した。今後は、実践にて確認された処理遅延問題を解消し、中学校や高等学校を含めた授業で利用できる WebAPI の検討や授業内容の検討を行いたい。