

令和6年 3月31日

令和5年度 学生自主研究成果報告書

教育本部長様

学生自主研究グループ名	WUNO (ウノ)	
研究課題名	パンの自動販売機における在庫自動認識	
研究代表者 (学生)	学籍番号	B25P041
	氏名	渡邊 集翔
指導教員	学科	経営システム工学科
	氏名	嶋崎 真仁

学生自主研究の報告書を別紙のとおり提出します。

パンの自動販売機における在庫自動認識

システム科学技術学部 情報工学科

1年 渡邊 集翔

1年 宇津野瑞季

1年 落合 壮太

1年 成瀬 葵音

指導教員 システム科学技術学部 経営システム工学科

教授 嶋崎 真仁

1 背景と目的

日本の少子高齢化に伴う働き手の減少を解決するためには機械やシステムによる自動化が必要である。

自動販売機に焦点を当てると、今なお目視確認による在庫管理を行っているものがある。人間が現地へ赴くのではなく遠隔かつリアルタイムで在庫が分かるということが必要になる。

そこで本研究では、アクリル越しに商品が直接確認、認識できる自動販売機を対象に、画像解析を利用することで商品の在庫管理を自動化するシステムの開発を最終目的とする。

2 先行事例

上田[1]はドローンを用いた小売店在庫管理システム開発に向けての基礎研究をおこなっている。本研究はこれを参考に窓拭き用掃除機による在庫管理システムの開発を行う。

3 開発方法

3-1 ハードウェアとシーケンス

図1にシステムの流れを示す。この図の上段はシステムのブロック図を表し、下段はM5Stack UnitV2(以下「UnitV2」という)内のシーケンスを表す。

開発には以下の機器を用いた

- ・カメラ付きマイコンUnitV2
- ・窓拭き用掃除機 (YIHAO社製[2])
- ・赤外線通信装置 Nature remo[3]
- ・モバイルWi-Fi[4]

UnitV2にプログラムを組み込み自立させ、パソコンとWi-Fi接続で繋げることでパソコンから無線で操作可能にする。

窓拭き用掃除機にUnitV2を取り付け、自動販売機内を撮影する。掃除機は赤外線を受信することで操作可能であり、Nature remo を用いて操作する。

色検出を行いながら掃除機を操作し、撮影を行う。撮影された画像は画像処理・解析を行い、在庫管理へとつなげるシステムである。

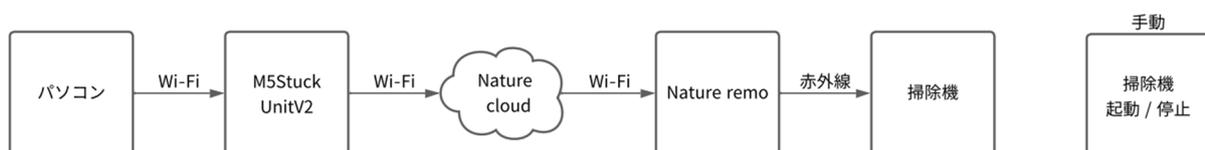


図1：システムの流れ

3-2 Pythonライブラリの構成

本研究は UnitV2 にプログラムを組み込み、撮影、画像処理、赤外線を発するための指示などを行う。

システム開発にあたり、使用したライブラリなどを表1に示す。

表1：開発環境

カテゴリ	名称	バージョン
OS	Linux	UnitV2標準
プログラミング言語	Python	3.8.6
ライブラリ	OpenCV	UnitV2 標準 装備
ライブラリ	Nature-remo	0.3.3
モジュール	os	
モジュール	time	
ライブラリ	ipython	7.9.0
ライブラリ	Pillow	7.0.0

4 実験に使用したプログラム

4-1 UnitV2のWi-Fi接続

UnitV2 は2つの操作方法が存在し、パソコンと直接USBで接続するか、同一のWi-Fi環境の下でクラウドを経由して操作することが可能である。また、M5Stackには電源が入るとWi-Fiを発する機能が備わっている。これを利用して無線での操作も可能である。

本実験では UnitV2 起動時に指定のWi-Fiに接続させた。図1にあるようにNature cloudにはWi-Fi環境が必要であり、UnitV2の発するWi-Fiではクラウドの利用が不可能なためこのような方法を取る。

4-2 UnitV2の活用方法と実験の流れ

パソコンとUnitV2の接続を確立したら、ブラウザを使用してドメイン名(unitv2.py)にアクセスする。設定ボタンからノートブックモードに切り替え、Jupyter NotebookからPythonでプログラムする。

プログラムで撮影と移動を行う。図2にUnitV2と掃除機の一連の動作を示す。

4-3 撮影

撮影と画像ファイル保存、ディスプレイに画像出力を行うphoto()関数を作成した。

4-4 画像合成 (パノラマ化)

撮影では、同一物体が別々の画像に映ってしまうことがあり、在庫のカウントを誤る恐れがある。そのため、画像合成を行い1枚の画像にまとめる必要がある。

今回作成したパノラマ化させるスクリプトは、画像ごとの特徴点をマッチングさせて複数の画像から1枚の画像を作成するというものである。

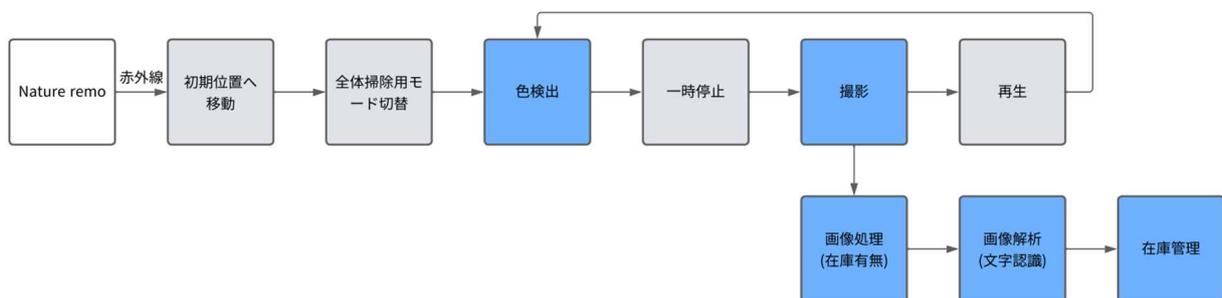


図2：プログラムした動作の流れ

4-5 色の検出と在庫の有無

掃除機が移動している状況下での撮影はブレが生じてしまい、サンプルを取ることができないためシャッターを切る際に停止している必要がある。実験で使用した自動販売機では写真1と2に示すように商品ごとに番号が割り振られている。



写真1 (左), 写真2 (右) : 販売スロット

黄色の文字で商品番号が記載されている。この色をカメラが検出した場合に掃除機が一時停止するスクリプトを作成する。

また、在庫の有無には二値化を活用する。二値化は画像を白と黒の二つの値に分けることで、画像の黒または白の割合を計算することができる。本実験の自動販売機は、商品購入時に前方へ押し出されるような形である。写真1と2を比較すると商品が存在する場合は、自ずと白の割合が大きくなり、商品が存在しない場合は黒の割合が大きくなる。画像の黒の割合をヒストグラムに起こし、有無を判定する。

4-6 赤外線を用いたロボット制御

本実験で利用した Nature remo はスマートフォンのアプリから家電のリモコン情報を登録し利用するものである。登録された指示は、Nature-remo という専用のライブラリからプログラミングによる操作が可能になる。しかしNature cloudを経由させるために Nature remo と同一のWi-Fi環境の下である必要がある。

これを利用して窓拭き用掃除機の方向転換を行う。

4-7 文字認識で商品の割り当て

商品番号における文字認識で商品を割り当てる。

5 実験の様子

実験の様子が動画で撮影されたものを写真3のQR CODEに示す。



写真3 : 実験の様子

6 結果と考察

各工程の結果を表2にまとめた。

表2 : 開発結果

行程	条件	結果
無線操作	Wi-Fi環境	無線で操作できた
撮影	なし	撮影できた
画像合成	特徴点が存在する	合成できた
色検出	なし	検出できた
二値化	販売スロットが画像の中心にある	判定できた
ロボット制御	Wi-Fi環境	操作できた
文字認識	専用ライブラリの使用可能な環境	メモリ不足により使用不可

パノラマ化は画像ごとの特徴点をマッチングさせて行うため、撮影対象との距離を保つことが重要になる。しかし、本研究では撮影対象との距離が非常に近く、撮影地点が1cmほど移動させた画像もパノラマ画像の生成に失敗した。このためパノラマ化は不向きであることが明らかとなった。

二値化を用いた在庫の有無には、判別対象となる販売スロットが撮影されているものは正しい判別ができた。しかし、色検出による撮影は販売スロットを確実に撮影範囲内に納めることはできなかった。そのため、物体検出のような技術を用いることで撮影の精度を上げられると考える。

7 開発における課題と対策

以下に今後の開発における課題を挙げる。

【 課題 】

- ・掃除機の起動は人力である
- ・処理が大きいとメモリのオーバーフローによりカーネル（システムの核となるプログラム）が落ちる
- ・遠隔での操作ができていない
- ・撮影の不確かさ
- ・UnitV2のメモリ不足

【 対策 】

- ・重い処理はデータを転送して他のパソコンで行う
- ・遠隔で操作するためサイトを作りURLをタップすることでプログラムを実行できるようにする

8 まとめ

本研究では、在庫管理システムとして、撮影、画像合成、赤外線を用いたロボット制御まで可能であることを示すことができた。最終目的を達成できなかったが、開発過程における課題を抽出することができ、開発可能であることが分かった。

参考文献

[1] 上田迅(2024):” ドローンを用いた小売店在庫管理システム開発に向けての基礎研究” , 秋田県立大学卒業研究.

[2] <https://amzn.asia/d/1I4WkcW>

[3] <https://amzn.asia/d/7TDIdbc>

[4] <https://amzn.asia/d/iM1bhFY>

(付録) 写真を撮るスクリプト

```
import os
import cv2
import IPython.display as display
from PIL import Image

def capture_image(camera): # 写真を撮る
    ret, frame = camera.read()
    if ret:
        return frame

def photo():
    camera = cv2.VideoCapture(0) # カメラ
    番号0を使用
```

```
output_folder = "Images/1"
os.makedirs(output_folder,
exist_ok=True)
# フォルダが存在しない場合は作成

frame = capture_image(camera)
# frameに画像データを組み込む
if frame is not None:
# frameが何らかのデータを持っている場合
# すでに存在する画像の最大番号を取得
existing_images = [f for f in
os.listdir(output_folder) if
f.endswith('.jpg')]
max_number =
max([int(f.split('_')[1].split('.')[0])
for f in existing_images], default=0)

# 新しい画像のファイル名を生成
new_number = max_number + 1
image_name=
f"{output_folder}/image_{new_number}.jpg"
# (フォルダ名+写真の番号)をimage_nameに

cv2.imwrite(image_name, frame)
# 画像ファイルの読み込み
print(f"Captured and saved
{image_name}")
# 画像の詳細を出力
display.display(Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)))
# ディスプレイに画像を表示
camera.release() # カメラを閉じる
```