

2018年度 卒業論文 2019/02/12 04:07

# 忘れ物返却支援システムの開発

大阪産業大学 デザイン工学部 情報システム学科  
情報教育システム研究室

15H072 長瀬正也



# 忘れ物返却支援システムの開発

15H072 長瀬正也

## 1 はじめに

大阪産業大学(以下、本学とする)では学生が利用する施設が多数存在する。施設を利用する中で、忘れ物や落し物をした事がある学生は少なくない。各施設に届けられた忘れ物や落し物は1週間をめぐり、本学の学生生活課に移動される。忘れ物の中には、家の鍵やICカードなど日常に支障をきたす貴重品といったものから、消しゴムや折り畳み傘といった安価で大量に流通しているものまで届けられる。

## 2 目的

内閣府の「遺失物に関する世論調査」[1]の結果では、安価で大量に流通しているものほど落し物の意識が低くなっている。本学でも同様に通学定期や自転車の鍵など貴重品は持ち主に返却されやすいが、折り畳み傘や充電ケーブルといった安価で大量に流通しているものは返却されない事が多い。

忘れ物の返却される割合(以下、返却率とする)が低いという問題には「探すまたは取りに行くのが面倒くさい」という原因の他に、「忘れ物、落し物をした事に気付いていない」といった原因があると考えられる。

本研究では上記の課題を解決すべく「忘れ物を早期に通知する」、「忘れ物をどこからでも確認できる」ことで忘れ物の返却率の向上および、簡単に使う事ができる忘れ物の返却支援システムの開発を行う。

## 3 忘れ物返却支援システム

本研究は本学15号館5階演習室(以下、演習室とする)の忘れ物を学生にメールで通知しWebページでどこでも確認できることで忘れ物の返却を支援するシステムである。しかし、忘れ物が届くたびに管理者が手作業でメールを作成しWebページを更新するのは手間である。そこで、一連の作業を自動で行うことにより、忘れ物管理者の負担にならないシステムを実現する。本研究では一例として、演習室の忘れ物を対象と

した返却支援システムの開発を行う。

## 4 検証と結果

本学の施設に届けられる忘れ物は、1週間をめぐり本館の学生生活課に移される。そこで1週間を検証期間にし、情報システム学科事務室に本システムを導入し2度の検証を行った。本システム導入前と本システム導入後の、1週間に届いた忘れ物の返却率の比較を行う。

表1 本システムの導入前と導入後で、1週間毎の忘れ物の返却率を示す。システム導入によって返却率の向上が見られる。

	届いた数	返却された数	返却率
導入前	6個	1個	18%
第1回検証	6個	4個	67%
第2回検証	3個	2個	67%

本システム導入後では返却される割合が高くなった。

## 5 まとめ

本研究では、情報システム学科事務室に届けられた忘れ物の返却率を高めるため、メールとWebを用いて忘れ物の返却を支援するシステムを開発した。その結果、学生にメールで忘れ物を通知し、過去の忘れ物をWebページで確認する事が可能となった。

また、システム導入前と導入後では忘れ物の返却率が向上傾向にある結果となった。

今後、本システムを利用する学生が増える事により、忘れ物の返却率を高めていく事で、忘れ物の管理者が本館へ忘れ物を移す手間が省け、負担軽減に貢献できるのではないかと考えられる。

## 参考文献

- [1] 内閣府政府広報室. 遺失物に関する世論調査. <https://survey.gov-online.go.jp/tokubetu/h28/h28-lost.pdf>, 2016年.



# 目次

1	はじめに	1
2	目的	2
3	忘れ物管理支援システムの概要	5
3.1	システムの詳細	6
3.2	システムの起動	6
3.3	画像の取得	6
3.4	画像の物体認識	7
3.5	データの格納	10
3.6	Web ページの設計	11
3.7	メールの送信	12
3.8	設置図	13
4	検証	16
4.1	検証方法	16
4.2	検証結果	17
5	結果と考察	19
5.1	回収率における結果と考察	19
5.2	画像認識における結果と考察	21
5.3	本システムの改善点	22
6	まとめ	24
7	謝辞	25
付録 A	ソースコード	27
A.1	忘れ物の写真を撮影し、サーバに転送するプログラム	27

# 1 忘れ物返却支援システムの概要

本研究は本学 15 号館 5 階プログラミング演習室 (以下、演習室とする) の忘れ物を学生にメールで通知し、Web ページでどこでも確認できることで忘れ物の返却を支援するシステムである。しかし、忘れ物が届くたびに、忘れ物の管理者が手作業でメールを作成し、Web ページを更新するのは手間であるため、一連の作業を自動で行うことにより、忘れ物管理者の負担にならないシステムを実現する。本研究ではシステムの一例として、演習室の忘れ物を対象とした支援システムの開発を行う。

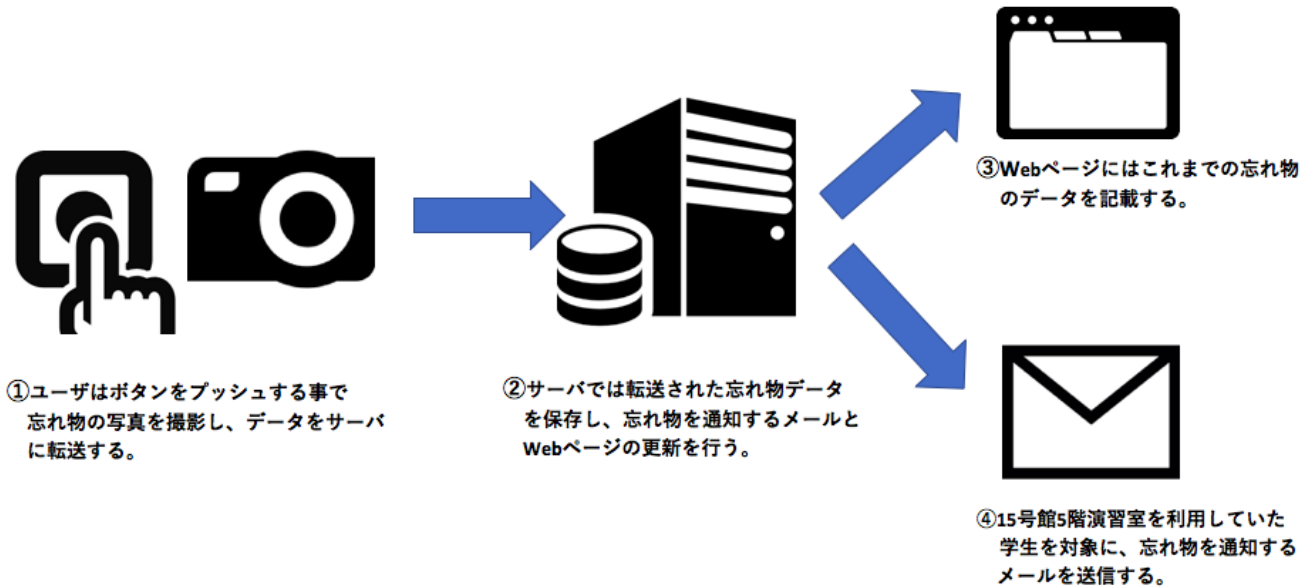


図 1 本システムの利用者が簡単に操作できるように、一連の処理をボタン一押しで出来るよう設計した。1 番で撮影された忘れ物の画像データは、2 番のサーバに転送される。転送されたデータは、忘れ物の届け日や物体名といった情報と共に保存される。保存されたデータを Web ページで表示し、メールで忘れ物の通知を行う。

## 1.1 システムの詳細

本システムは忘れ物を撮影する Raspberry Pi3 及びカメラを搭載した筐体型の組み込みシステム側と Web ページの更新やメールを送信するサーバ側の二つに分かれる。システムの構成を図 4 に示す。

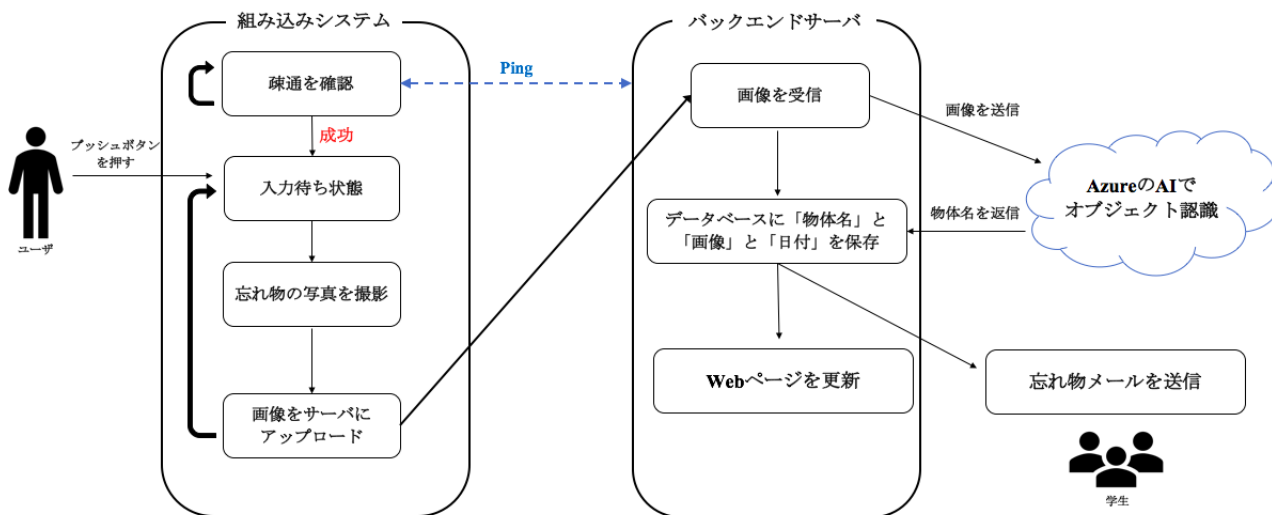


図 2 本システムの一連の処理の流れである。忘れ物の撮影を行う組み込みシステム側と、画像の認識や Web ページの更新やメールの送信を行うバックエンドサーバ側に分かれる。

## 1.2 システムの起動

本システムはディスプレイやマウス、キーボードなど外部インターフェースは必要ないため、Raspberry Pi3 を起動させると同時に、本システムが入力待ち状態になる必要がある。そこで、`crontab`<sup>\*1</sup>の`@reboot`機能を使用した。`crontab`では時刻指定の部分を`@reboot`と記述する事で、OSの起動時に指定したプログラムを一度だけ実行させる事ができる。

また、入力待ち状態をユーザに目視で確認できるように LED を取り付けた。始めに Ping でサーバとの疎通の確認を行い、問題がなければ入力待ち状態であることを LED を点灯させることでユーザに知らせる。

## 1.3 画像の取得

忘れ物を撮影する為、Raspberry Pi3 に Web カメラとプッシュボタンを取り付けた。Web カメラで忘れ物を撮影し、画像として保存するために `fsWebcam` を用いた。`fsWebcam` は Raspberry Pi が公式で配布しているソフトウェアであり、コマンドを実行するだけで Web カメラで撮影、保存する事ができる。

本システムではプッシュボタンの入力を確認し、忘れ物の撮影と画像データをサーバに転送するプログラムを実装した。また、Raspberry Pi3 とサーバ間の通信は `Curl`<sup>\*2</sup>を用いて POST メソッド<sup>\*3</sup>で送信する。サーバ側では、Web サーバを動作する Web フレームワーク<sup>\*4</sup>に画像データがルーティングされ、POST メソッドに反応し画像物体認証プログラムが呼び出される。

\*1 定期的にジョブを実行するようスケジュールする `cron` を設定するコマンド

\*2 URL を使用してデータ転送するためのコマンドラインツール

\*3 HTTP 通信でクライアントから Web サーバへ送るリクエストの一つ

\*4 システム開発を効率化する機能群

## 1.4 画像の物体認識

サーバに転送された画像の物体認識を行うため、本学の忘れ物データに特化した独自の学習モデルを作成する。本研究では Microsoft Azure の Custom Vision API という機械学習モデルを採用して物体認識を行った。Google Cloud の Cloud Vision や AWS の Rekognition など、事前に提供元が学習させたモデルでの物体認識を行ってみたが、学習範囲が広いいためか一致率が 50% を下回る信頼性に欠ける制度となってしまった。

Custom Vision API では、ユーザーが学習させたい画像を登録し、それぞれの画像にラベル情報を付加する事により、物体のイメージとラベル情報の関係を学習させることができる。学習したモデルは、サービスに登録した際に発行される ProjectID\*<sup>5</sup>と IterationID\*<sup>6</sup>を指定し、物体認識したい画像データを Curl で Custom Vision API の URL に POST することで JSON 形式で評価が返ってくる。1つの画像データで複数の予測が出力されるケースもある。Custom Vision API では、JSON 形式で返されるデータは適合率を表す Probability 値で降順に並べられているため、本システムは、一番上のデータを選択する Custom Vision API の画像認識の実行例を図 5、図 6 にそれぞれ示す。

```
curl -sX POST "https://southcentralus.api.cognitive.microsoft.com/customvision/v1.0/{ProjectID}/url?iterationId={IterationID}" \
-H "Content-Type: application/json" \
-H "Prediction-key: {Prediction-key}" \
-d '{"Url": "画像のバス"}'
```

図 3 指定した画像データを Curl を用いて API の URL に POST する。Curl コマンドの ProjectID と IterationID を自身の ID に変更する。

```
{
  "Created": "2018-11-30T07:22:29.2698916Z",
  "Predictions": [
    {
      "Probability": 0.976153,
      "Tag": "Mobile battery",
      "TagId": "28bdc70f-26f2-47cf-91a1-730905b3366a"
    },
    {
      "Probability": 0.730135,
      "Tag": "USB memory",
      "TagId": "a662bff1-06f8-4ac5-be49-22861237e8cc"
    }
  ],
  "Project": "250d2d53-2d1a-4ca5-9b35-95c215206e84"
}
```

図 4 モバイルバッテリーの画像データを POST し、モバイルバッテリーである確率が 97.6% という結果が出力された。POST された画像データが解析され、JSON データとして返ってくる。Probability という項目が物体認識の適合率となっている。また、適合率が 73.0% の確率で USB メモリという予測も出ている。Custom Vision API では、JSON 形式で返されるデータは Probability 値で降順に並べられているため、本システムは、一番上 (モバイルバッテリー) のデータを選択する。

\*<sup>5</sup> Custom Vision API のプロジェクトを作成すると生成される ID

\*<sup>6</sup> Custom Vision API では学習させた単位をさす ID

## 1.5 データの格納

物体認識が完了した画像と認識された物体名を保存しておく必要があるためデータベースを使用する。本研究では、オープンソース\*7で公開されている関係データベース管理システムのMySQLを用いた。Web ページに必要な「物体名」「忘れ物の画像」「忘れ物が撮影された日時」の三つの情報を保存するデータベースを作成する。

また、画像データはデータベースに直接保存すると非常に大きなリソースが必要なため、画像が保存されているディレクトリパスを指定する事で、データベースから間接的に画像にアクセスするよう設計した。本システムを表 2 に示す。

表 1 忘れ物返却支援システムのデータベースのテーブル

object	time	file_path
glasses	2018-12-19 16:29:15	./static/images/20181219162910.jpg
adapter	2018-12-19 16:30:32	./static/images/20181219163021.jpg
gloves	2018-12-21 10:25:59	./static/images/20181221102546.jpg
pencilcase	2018-12-21 10:27:13	./static/images/20181221102708.jpg
handkerchief	2018-12-25 15:48:31	./static/images/20181225145415.jpg
USBmemory	2018-12-25 15:48:32	./static/images/20181225154120.jpg
book	2018-12-25 15:51:22	./static/images/20181225154010.jpg

画像認識されたラベルを保存するテーブルでは 3 つのカラムを作成した。object カラムは画像認識された物体名を保存する。object カラムには Type の「varchar(30)」と「NOT NULL」が設定されており、必ず物体名を指定する必要がある。

time カラムには「timestamp」を指定しており、画像データを登録した時点で日時が登録されるように設定した。

file\_path カラムは忘れ物の画像データを相対パスで登録する設定をしている。画像データは、Web ページを更新する Web フレームワークのディレクトリの下に保存していく。

---

\*7 ソフトウェアのソースコードが一般公開されており、ソースの変更や再配布を行えるライセンスを指す

## 1.6 Web ページの設計

学生が Web ページから忘れ物を確認できるよう Web ページを作成した。Web ページを作成するにあたって、Web ページの作成を支援する Web フレームワークの Flask\*<sup>8</sup>を用いた。Web ページが表示されるまでのプロセスは、クライアント側の PC から“<http://www0.ise.osaka-sandai.ac.jp/ks150/>”にアクセスすると、サーバ側の Web フレームワークが動作し、クライアント側の PC に表示するプログラムが読み込まれる。また、Web ページの URL は忘れ物メールの本文にも記載する。

Web ページはデータベースに保存されたデータを取得し、忘れた日付の昇順で表示する。表示するデータ項目は「物体名」「忘れ物の画像」「忘れ物が撮影された日時」の三つである。Web ページの更新は、ボタンを押して忘れ物の写真が撮影されてから約 3 秒で行われる。

また、忘れ物は事務室に届けられた日付から 1 週間を目処に本館に移されるため、表示するデータも 1 週間の期限を設ける。これにより返却された忘れ物データは一定期間消えないものの、1 週間後には自動で削除されるためシステムの管理者が手動で削除する手間を省く事ができる。実際の Web ページを図 9 に示す。

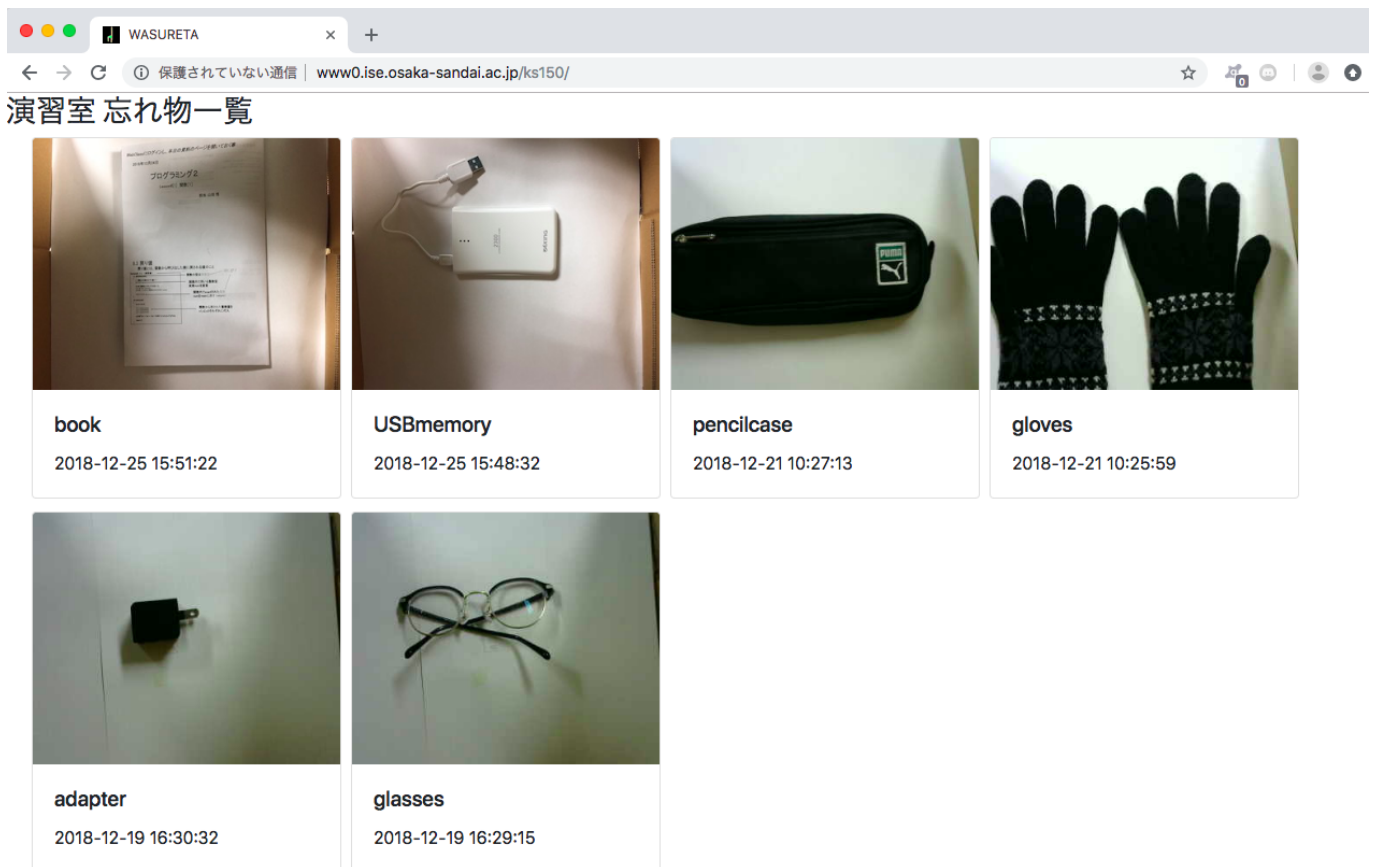


図 5 忘れ物返却支援システムの Web ページである。忘れ物の写真が撮られた日付から降順で並べられており、更新されてから 1 週間経過した忘れ物は掲載されなくなる。また、物体名は AI で認識をしているため誤認している場合がある。

\*8 プログラミング言語 Python 用の、Web フレームワーク

## 1.7 メールの送信

本システムでは Web ページで忘れ物を公開すると同時に、忘れ物情報の通知を学生にメールで行う。メールには Web ページ同様、「物体名」「忘れ物の画像」「忘れ物が撮影された日時」の三つが記載されている。また、Web ページのリンクを載せることで、過去の忘れ物一覧にメールの本文からアクセスできるようにする。

また、全学生にメールを一斉に送信するのは対象が多く効率的ではない。本システムでは、メールの送信は“忘れ物が届いた当日の時間以前に演習室の PC にログインしていた学生”と“忘れ物が届いた前日に演習室の PC にログインしていた学生”を対象とする。本システムのメール本文を図 10 に示す。



図 6 忘れ物返却支援システムのメール内容である。メール本文には「物体名」「忘れ物の画像」「忘れ物が撮影された日時」の三項目が記載されている。また、Web ページ同様に物体名は AI で認識をしているため誤認している場合がある。

## 1.8 設置図

本システムの検証は事務室で行うため、Raspberry Pi3 の電源は設置場所を限定されないようモバイルバッテリーを使用した。組み込みシステムの筐体は簡易的に段ボールで作成し、その上部に Raspberry Pi3 と Web カメラ、入力のプッシュボタンと入力待ちを表す LED を取り付けた。本システムの全体像を図 11 に、実際の動作風景を図 12 にそれぞれ示す。



図 7 忘れ物管理支援システムの組み込みシステムの全体像である。段ボールの筐体には Raspberry Pi3 と Web カメラ、入力のプッシュボタンと入力待ちを表す LED を取り付けている。

システムが起動し、入力待ち状態を示す LED が点灯している。プッシュボタンを押す事で処理が始まるが、処理中はプッシュボタンの入力を受け付けない。そのため、処理中は LED を点滅させる事でシステム利用者に目視で確認できる設計を行った。

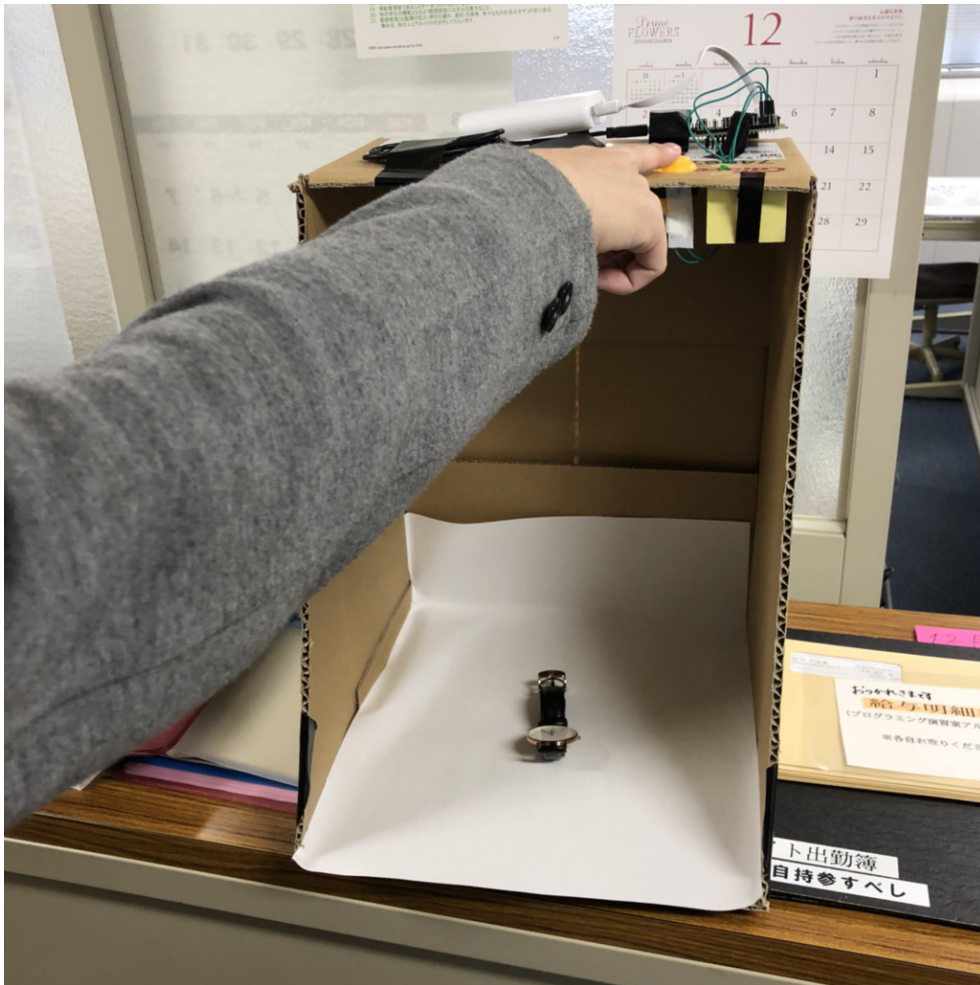


図8 実際の動作風景である。押しボタンを押す事で、筐体の中の忘れ物(時計)が撮影される。また、押しボタンを押して忘れ物の撮影から通知のメールを送信する一連の処理が終わるまで約5秒かかる。

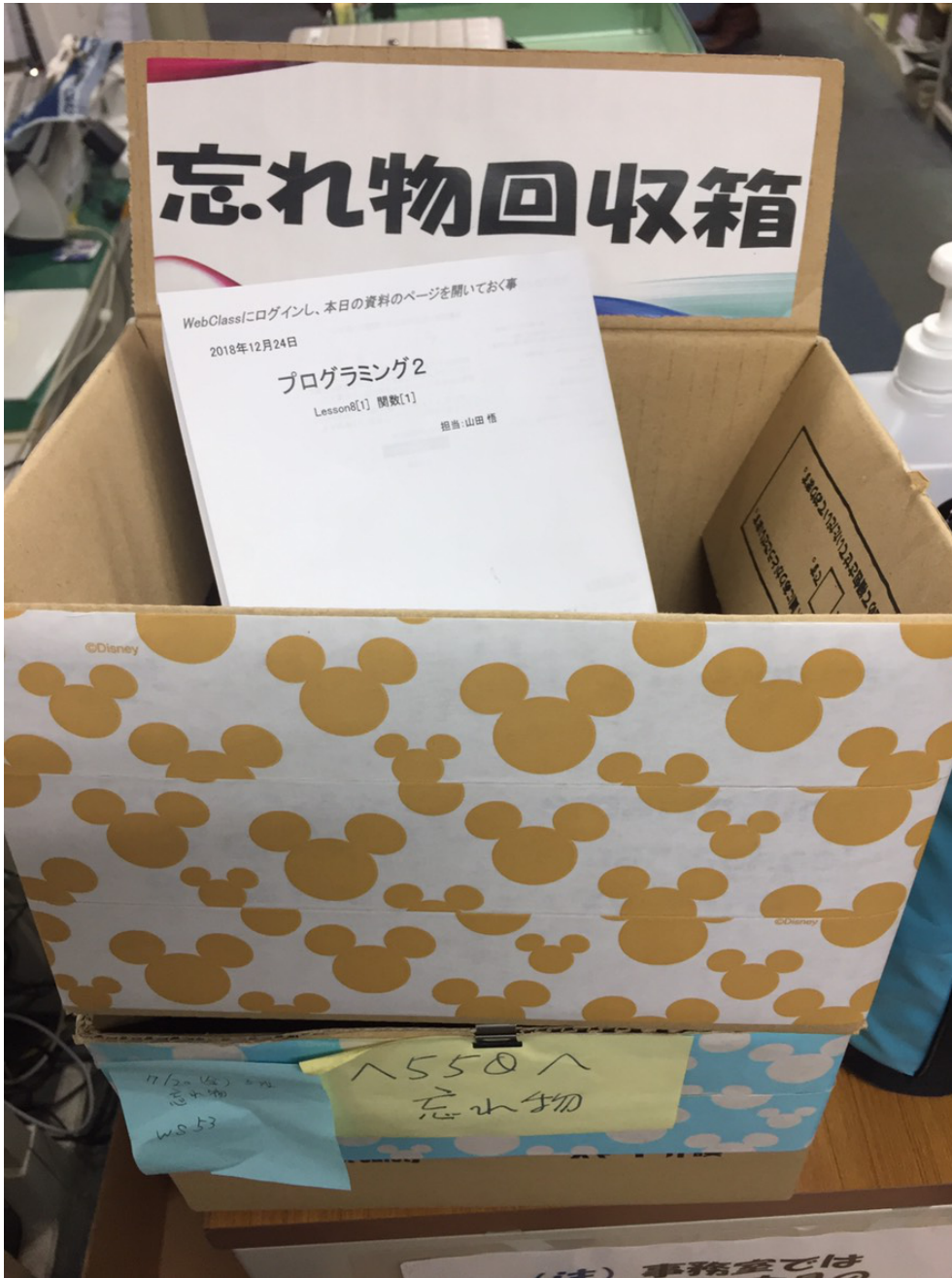


図9 忘れ物の撮影が終わると、忘れ物を回収箱に入れる。

## 2 検証

事務室に届けられる忘れ物は、1週間を期間に本館の学生生活課に移される。そこで1週間を期間にし、事務室に本システムを導入し検証を行った。本システム導入前と本システム導入後の、1週間に届いた忘れ物の返却率の比較を行う。

### 2.1 検証方法

検証期間の1週間に事務室に届けられた忘れ物を、本システムを用いて忘れ物の返却支援を行う。そこで、10時30分、12時30分、14時30分、16時30分と、事務室が閉まる18時に事務室にて忘れ物の確認を行う。忘れ物が確認されると、本システムを用いて忘れ物を撮影し、Webページの更新と学生にメールの送信を行う。また、演習室の忘れ物は、授業合間の開放時間に常駐スタッフが事務室に届けるケースが多い。そこで、常駐スタッフに協力してもらい忘れ物を見つけると連絡してもらうようにした。

#### 2.1.1 メール送信

3.7でも説明した通り、「忘れ物が届いた時間以前に演習室のPCにログインしていた学生」と「忘れ物が届いた前日に演習室のPCにログインしていた学生」を対象にメールを送信する。また、メールの本文には「\*\*\*本メールは自動で生成されたメールです。返信しないでください。\*\*\*」と注意書きをしているが、メールを返信する学生がいる事も考えられる。さらに、多くの学生が返信した際に「User unknown」になってしまうとGmailでスパム判定になってしまい検証ができなくなる。そこで、返信されたメールはaliasを設定することで、返信されたメールを自分宛のメールアドレスに転送する設定をした。実際の返信を図14に示す。



図10 本システムから送信されたメールに返信すると、aliasの設定によりシステム側ではなく、本システムの管理者宛にメールが届く。

## 2.2 検証結果

今回、本システムを導入した検証を期間を分けて2度行った。1週間の忘れ物の届いた件数と返却された件数をまとめる。

### 2.2.1 2018年12月19日から同年12月25日の検証結果

本システム初めの本番稼働である。2018年12月19日から同年12月25日の検証データを表3と表4に、画像認識AIの適合率を表5にそれぞれ示す。

表2 2018年12月19日から同年12月25日の忘れ物件数と返却された件数。土日は事務室が開いていないため検証日に含めない。

	水	木	金	月	火
日付	12/19	12/20	12/21	12/24	12/25
忘れ物数	2	0	2	0	2
返却数	0	0	2	0	2

表3 2018年12月19日から同年12月25日の忘れ物の固有名と届けられた日時と返却された日時

	忘れ物	届け日	時間	返却日	時間
1	Glasses(眼鏡)	12/19	14:30 ~16:30	—	—
2	Adapter(アダプタ)	12/19	14:30 ~16:30	12/25	14:30 ~16:30
3	Gloves(手袋)	12/21	9:00 ~10:30	12/21	16:30 ~18:30
4	Pencilcase(筆箱)	12/21	10:30 ~12:30	12/21	12:30 ~14:30
5	Book(本)	12/25	9:00 ~10:40	—	—
6	USBmemory(USBメモリ)	12/25	14:30 ~16:30	12/25	16:30 ~18:00

2018年12月19日から同年12月25日の間に届いた忘れ物は合計6個であったが、その内返却された忘れ物は4個であった。4個の返却された忘れ物の中で、表4中の3,4,6番の3個がその日のうちに返却される結果となった。

### 2.2.2 画像認識AIの適合

表4 2018年12月19日から同年12月25日の忘れ物の画像認識結果の適合率

予測	正解	正誤
Glasses(眼鏡)	Glasses(眼鏡)	正
Adapter(アダプタ)	Adapter(アダプタ)	正
Glove(手袋)	Gloves(手袋)	正
Pencilcase(筆箱)	Pencilcase(筆箱)	正
Book(本)	Print(プリント)	誤
USBmemory(USBメモリ)	Mobilebattery(モバイルバッテリー)	誤

届いた忘れ物は合計6個であったが、その中で予測が適合したのはGlasses(眼鏡),Adapter(アダプタ),Gloves(手袋),Pencilcase(筆箱)の4種であった。過去の忘れ物の中にプリントは入っておらず、機械学習にプリントという

項目がないため一番類似している Book(本) という誤認の予測となった。しかし、Mobilebattery(モバイルバッテリー) という項目は事前に学習しているため、USBmemory(USB メモリ) と認識したのは精度が原因である。

### 2.2.3 2019 年 1 月 8 から同年 1 月 14 日の検証結果

冬季休暇を挟み、本システム 2 度目の本番稼働の検証となる。2019 年 1 月 8 日から同年 1 月 14 日の検証データを表 6 と表 7 に、画像認識 AI の適合率を表 8 にそれぞれ示す。

表 5 2019 年 1 月 8 日から同年 1 月 14 日の忘れ物数と返却された件数。土日は事務室が開いていないため検証日に含めない。

	火	水	木	金	月
日付	1/8	1/9	1/10	1/11	1/14
忘れ物数	2	0	0	1	0
返却数	1	0	0	1	0

表 6 2019 年 1 月 8 日から同年 1 月 14 日の忘れ物の固有名と届け日時と返却日時

	忘れ物	届け日	時間	返却日	時間
1	Umbrella(折り畳み傘)	1/8	9:30 ~10:30	—	—
2	USBmemory(USB メモリ)	1/8	12:30 ~12:30	1/8	16:30 ~18:00
3	Pencilcase(筆箱)	1/11	12:30 ~10:30	1/11	16:30 ~18:00

2019 年 1 月 8 日から同年 1 月 14 日の間に届いた忘れ物は合計 3 個であったが、その内返却された忘れ物は 2 個であった。2 個の返却された忘れ物の中で、表 7 中の 2,3 番の 2 個がその日のうちに返却される結果となった。

### 2.2.4 画像認識 AI の適合

表 7 2019 年 1 月 8 日から同年 1 月 14 日の忘れ物の画像認識予測の適合

予測	正解	正誤
Umbrella(折り畳み傘)	Waterbottle(水筒)	誤
USBmemory(USB メモリ)	Mobilebattery(モバイルバッテリー)	誤
Pencilcase(筆箱)	Pencilcase(筆箱)	正

事務室に届いた忘れ物は合計 3 個あったが、その中で認識結果が正しかったのは 1 個であった。学習項目に Waterbottle(水筒) は含まれているが、水筒が大きいために撮影ブースからはみ出してしまい水筒全体が映らなかった。そのため、類似している Umbrella(折り畳み傘) という認識となった。また、前回同様に Mobilebattery(モバイルバッテリー) を USBmemory(USB メモリ) と認識した。画像認識の精度が原因である。

### 3 結果と考察

本システムを導入し、1度目の検証の2018年12月19日から同年12月25日と、2度目の検証の2019年1月8日から同年1月14日の2度の期間に行なった検証の結果を基にシステムの結果と考察をまとめた。

#### 3.1 回収率における結果と考察

本システムの導入前と導入後の検証結果の返却率を図15に示す。

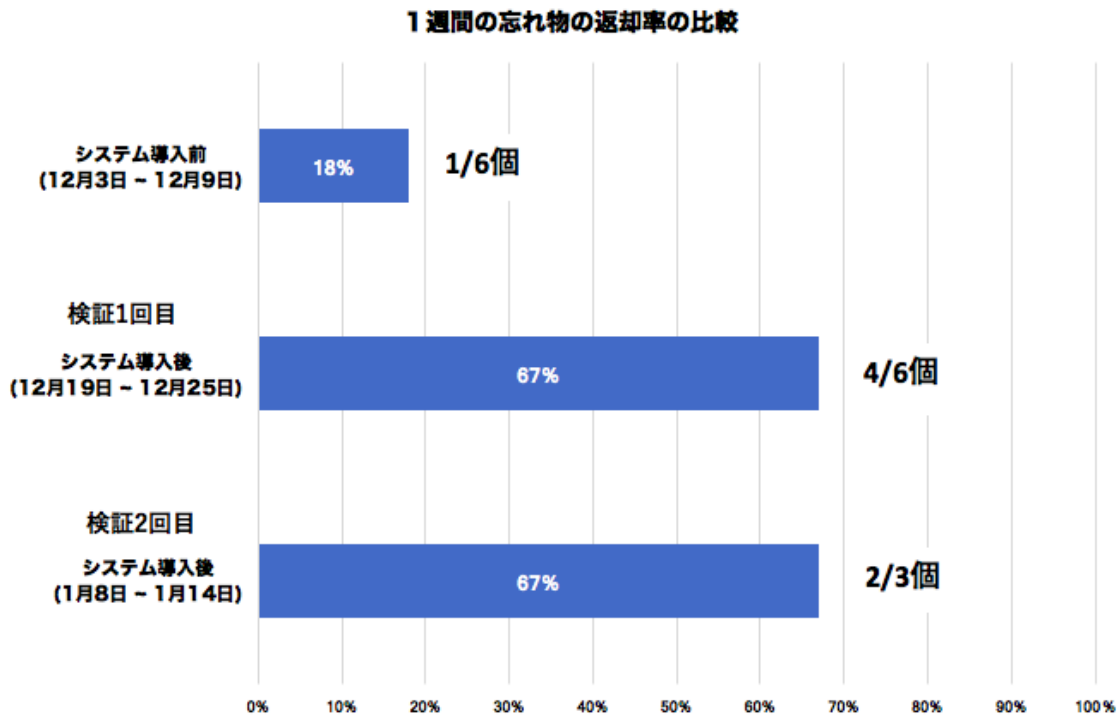


図11 1週間の忘れ物の返却率を比較したグラフである。棒グラフは届いた忘れ物の中で、何個返却されたかの割合である。

本システム使用后、1度目の検証では、忘れ物が6個届けられ、そのうち4個が返却された。そして2度目では、忘れ物が3個届けられ、そのうち2個が返却された。本システム使用前の2018年12月3日から同年12月9日のデータと比較し、システム導入後は返却率が向上する傾向がみられた。

また、1度目と2度目の検証で返却されなかった忘れ物は、Glasses(眼鏡)、Print(プリント)、Waterbottle(水筒)の3個であった。内閣府が公表している「遺失物に関する世論調査」[1]では、「貴重品と安価で流通しているものでは忘れ物の意識は異なる」といった調査結果が示されている。Print(プリント)とWaterbottle(水筒)は安価で流通しているものだが、Glasses(眼鏡)は日常に支障をきたす必需品である。しかし、眼鏡には度が入っておらず伊達眼鏡だった。そのため、結果は最後まで返却されることはなかった。

次に、2度の期間に行った検証データを元に曜日別で忘れ物の個数を図16に示す。

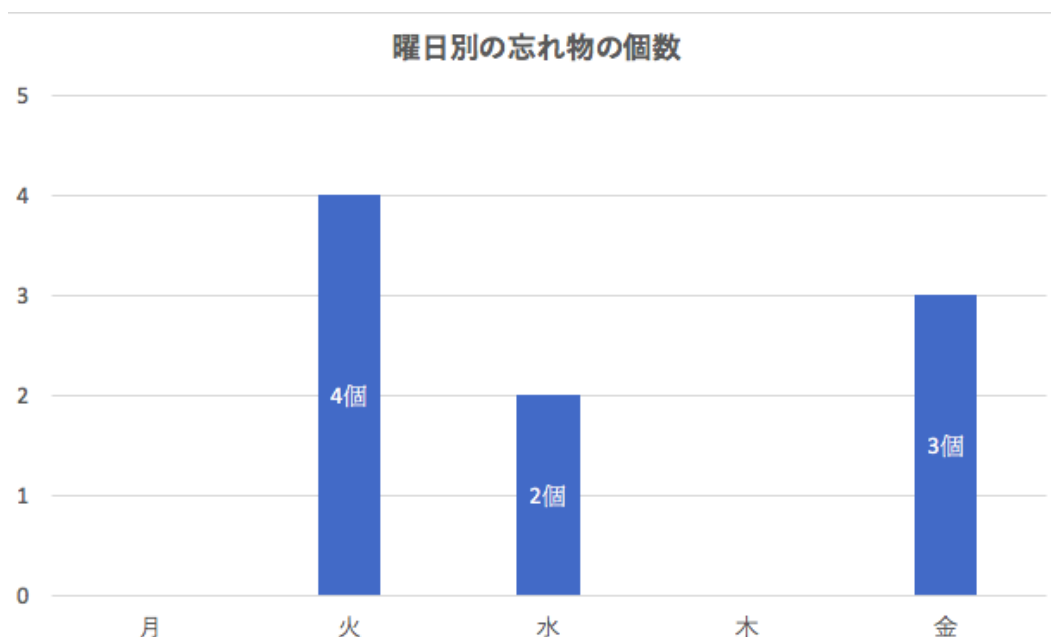


図12 曜日別の忘れ物の個数を比較したグラフである。棒グラフは曜日別の忘れ物の合計の個数である。

2度の検証期間では火曜日が4個と最も多く、次点で金曜日の3個となった。その中で、演習室の自由開放時に届けられた忘れ物は、合計の9個中で6個であった。事務室に届けられる忘れ物は、演習室の自由開放時に常駐スタッフが事務室に届けるケースが多いという事もあるが、常駐スタッフにヒアリングしたところ、自由開放時に演習室を利用した学生が忘れ物をしたというケースも多い。火曜日では演習室は2限から5限まで講義が行われるが、一部の席が自由開放としても使われている。そのため、演習室の自由開放時間が長い火曜日は忘れ物が多くなると考えられる。

また、プリントの忘れ物も確認されているため、プリントを配布する講義が多い曜日では忘れ物が多くなると思われる。

### 3.2 画像認識における結果と考察

2度の検証期間に届けられた忘れ物をまとめた Web ページのスクリーンショットを図 17 に示す。

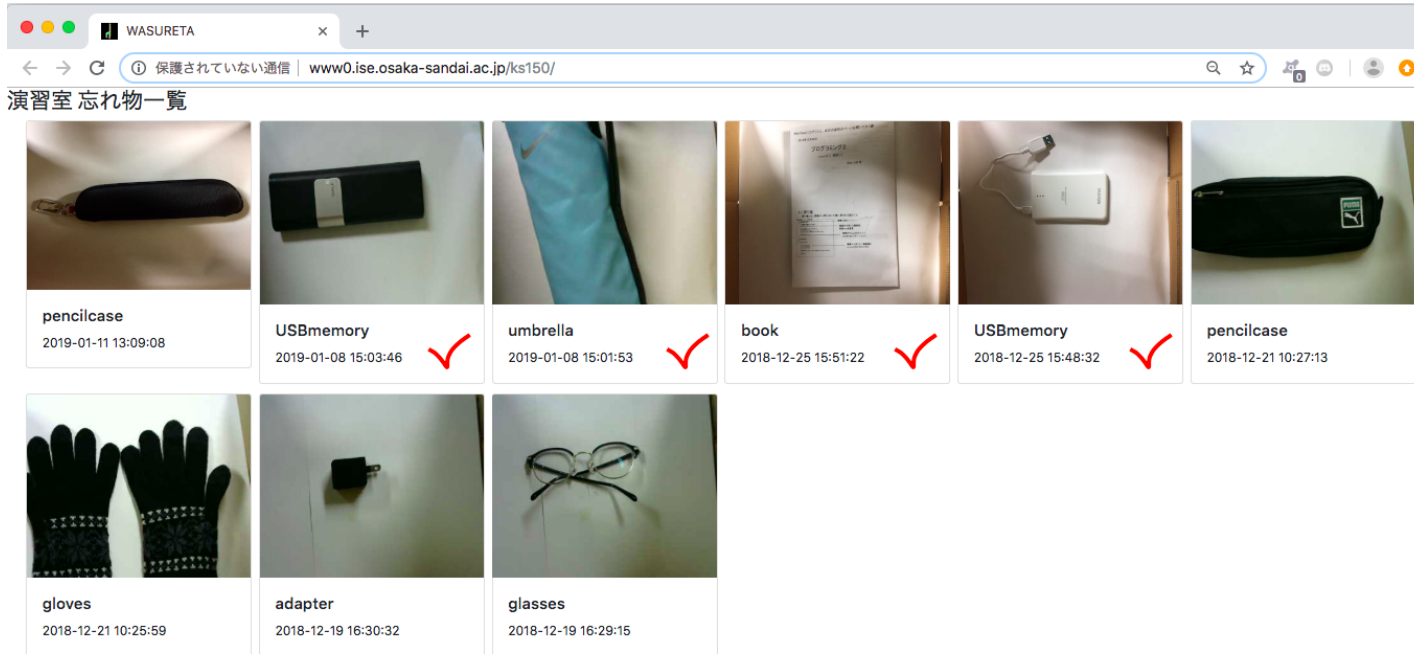


図 13 2018 年 12 月 19 日から 2019 年 1 月 14 日までの忘れ物の Web ページをスクリーンショットし、画像認識が誤っている忘れ物にチェックマークをつけている。

本来は 1 週間の期限で忘れ物を Web ページに記載するが、今回は検証用として全ての忘れ物を表示する設定を行った。また Web ページのスクリーンショットに画像認識の認識が誤っている忘れ物にチェックマークをつけている。

結果として忘れ物 9 個の中で予測が誤っていたのは 4 個であった。その中で予測が外れた項目は Print(プリント)、Mobilebattery(モバイルバッテリー)、Waterbottle(水筒) の 3 項目である。プリントは 1 度目の検証期間中に学習項目に入らなかったため、類似している Book(本) と予測した。しかし、Mobilebattery(モバイルバッテリー) は 2 度の検証期間で 2 回とも USBmemory(USB メモリ) と誤認している。これは学習不足による誤認であると考えられる。Waterbottle(水筒) は忘れ物の撮影時に被写体が撮影ブースからはみ出してしまい、水筒の全体像が写せなかった。水筒の筒を Umbrella(折り畳み傘) と誤認したと考えられる。

### 3.3 本システムの改善点

当初は押しボタンの接合部のチャタリング<sup>\*9</sup>や、バックエンドサーバのシステムエラーを不安視していたがどちらも起こることがなく検証期間を終えた。しかし、検証を通し別の問題点が浮かび上がった。

#### 3.3.1 メールの誤送信

当初は押しボタンの接合部のチャタリングにより、1度の入力が3度や4度増幅する問題があったが、組み込みシステムに「3秒は入力を受け付けない」というプログラムの処理を設定することでチャタリングの問題は解決した。しかし、押しボタンの接合部の問題は残っており、本体に少しの振動でも加わるとボタンを押したことになる入力になってしまう。誤送信されたメールの例を図18に示す。



図14 振動により入力が入り、シャッターが切られ何も写っていない写真がメールで送信されてしまう。

図18は、組み込みシステム本体に腕が衝突した時に起きたことである。不意に起きた事であり、撮影ブースに忘れ物はなく、下敷きの紙が写っているが画像認識AIはHandkerchief(ハンカチ)と誤認している。改善点として、組み込みシステムに下敷きの紙が写った時のエラー処理をプログラムに追加する事で防ぐ事が出来ると思わ

<sup>\*9</sup> 可動接点などが接触状態になる際に、微細な非常に速い機械的振動を起こす現象

れる。

### 3.3.2 筐体の撮影ブースの幅と奥行き

忘れ物の大きさは、USB メモリや自転車の鍵といった小さい物だけではなく、水筒や傘といった大きい物も届けられる。撮影風景を図 19 に示す。



図 15 水筒を撮影ブースに乗せるが全体に収まりきらず、筐体からはみ出している。

図 19 では 1.5 リットルサイズの水筒が忘れ物で届けられ、写真を撮ろうにも全体が収まりきらず撮影ブースからはみ出している。実際に撮れた写真を図 20 に示す。



図 16 図 19 の際に撮影された写真である。画像が見切れている。

図 20 は、画像認識 AI により Umbrella(折り畳み傘)と誤認されている。本システムの筐体はプロトタイプとして段ボールで作られているため、改善点として筐体を拡張し、撮影ブースの幅と奥行きを広げるとよいと考える。

## 4 まとめ

本研究では、事務室に届けられた忘れ物を、メールと Web ページを用いて忘れ物の返却を支援するシステムを開発した。その結果、学生にメールで忘れ物を通知し、過去の忘れ物を Web ページで確認する事ができるようになった。

検証の結果、期間が短いため断定はできないが、システム導入前と導入後では忘れ物の回収率が向上傾向にあると言える。

事務室に届けられた忘れ物は、1 週間をめどに本館へ移す決まりがある。しかし、忘れ物をわざわざ 15 号館から本館まで届けるのは手間である。今後、本システムを利用する学生が増える事で忘れ物の返却率を高めていく事で、忘れ物の管理者の負担軽減に貢献できるのではないかと考えられる。

## 5 謝辞

本研究を進めて行く上で、担当教員である大垣斉准教授にご指導及びご協力を頂きました。また、情報教育システム研究室所属の学生および卒業生の方々、小山翔平氏、15号館5階プログラミング演習室の常駐スタッフの方々にご協力を賜わり、深く感謝の意を表します。

## 参考文献

- [1] 内閣府政府広報室. 遺失物に関する世論調査. <https://survey.gov-online.go.jp/tokubetu/tindex-h28.html>, 2016 年.

## 付録 A ソースコード

### A.1 忘れ物の写真を撮影し、サーバに転送するプログラム

ソースコード 1 忘れ物の写真を撮影し、サーバに転送するプログラム (capt.py)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import RPi.GPIO as GPIO
import subprocess
import datetime
import os
import glob
import time
from time import sleep

import threading

cmd_ping = "ping -c 3 133.64.170.129"

while(subprocess.call(cmd_ping, shell=True)):

    sleep(3)

GPIO.setmode(GPIO.BCM)
GPIO.setup(14, GPIO.IN, pull_up_down=GPIO.PUD.DOWN)
GPIO.setup(2, GPIO.OUT)

flag = True

def timelag(false_count):
    files = glob.glob("/home/pi/webcam/*.jpg")
    ans=0
    dup_time = 13

    if(false_count % 2 == 0):
        dup_time = dup_time + 1
```

```

for file in files:
    try:
        dirname = os.path.basename(file)
        filename = dirname.strip(".jpg")

        today = datetime.datetime.today()
        now = today.strftime("%Y%m%d%H%M%S")
        t = int(now)-int(filename)
        print(t)
        if(int(now)-int(filename) < dup_time):
            ans =1

    except Exception as e:
        print(e)

print(ans)

return ans

```

```

def led():

    while(flag ):
        GPIO.output(2, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(2, GPIO.LOW)
        time.sleep(0.5)
        print("hikatteru")
    print("owari")

```

```

def camcapslack(self):

    global flag
    flag = True

    false_count = 0
    thread_led = threading.Thread(target = led)

    thread_led.start()

```

```

while timelag(false_count) == 0:
    today = datetime.datetime.today()
    now = today.strftime("%Y%m%d%H%M%S")
    now2 = today.strftime("%Y-%m-%d-%H:%M:%S")
    jpgdir = "/home/pi/webcam/%s.jpg" % now

    cmd_out = "sudo_rmmmod_uvcvideo"
    cmd_in = "sudo_modprobe_uvcvideo"
    cmd_capt = "fswebcam --no-banner /home/pi/webcam/%s.jpg >> /home/pi/
        webcam/CaptLog_2>&1" % now
    cmd_post = "curl -X_POST -F_file=@/home/pi/webcam/%s.jpg http://www0.
        ise.osaka-sandai.ac.jp/ks150/send >> /home/pi/webcam/CaptLog_2>&1"
        % now

    subprocess.call(cmd_out, shell=True)
    subprocess.call(cmd_in, shell=True)

    subprocess.call(cmd_time, shell=True)
    bol_capt = subprocess.call(cmd_capt, shell=True)
    bol_post = subprocess.call(cmd_post, shell=True)

    if bol_capt == 0 and bol_post==0:
        false_count = 0
    else:
        false_count = 1 + false_count

flag = False

```

```
GPIO.add_event_detect(14, GPIO.RISING, callback=camcapslack, bouncetime=3000)
```

```
GPIO.output(2, GPIO.HIGH)
time.sleep(0.01)
```

```
try:
    while True:
```

```
        sleep(3)
        GPIO.output(2, GPIO.HIGH)
        time.sleep(0.01)

except KeyboardInterrupt:
    pass

GPIO.output(2, GPIO.LOW)
time.sleep(0.5)

GPIO.cleanup()
```