最終成果報告書

プライバシーを考慮した匿名化カメラによる混雑把握システムの研究開発

鈴木プロジェクト2021

1. 背景と目標

1.1 背景

2020 年以降、COVID-19 の影響で他者との距離をとらなければならないソーシャルディスタンスの時代となった[1]。そのような状況下では、混雑状況を把握し能動的に混雑を回避する必要性が生じる。ニーズの高まりに応じるように、混雑度を把握・掲示するシステムが現れた[2]。これらの混雑度を把握するシステムでは、人を識別できるほど鮮明な写真を撮影しており、個人のプライバシー保護の観点から慎重な扱いが求められる。過去には個人識別が可能な形で撮影された防犯カメラの映像が問題になった事例[3]も発生している。本プロジェクトでは個人のプライバシーを保護した上で、混雑度を計り、混雑を積極的に回避することが可能なシステムが必要であると考えた。

1.2 目標

プライバシーを保護しながら混雑状況を把握する方法はいつくか考えられる。たとえば個人の識別が不可能なセンサ類を用いる方法である。センサを用いた方法では、深度などのセンサを導入する手法[4]や赤外線センサ等で通過を検知する手法[5]があるが高価かつ特別な機材[6]が必要となる。本プロジェクトでは汎用の Web カメラ等を用いて個人が識別できないほどの不鮮明なカメラ画像を取得し利用する方法が必要だと考えた。そこで予め撮影段階でピントをぼかすなど光学的加工を施したカメラ画像を用いて個人のプライバシー保護をしつつ、その画像を使い機械学習によって混雑状況を把握し、混雑状況の通知をスマートフォンを通してユーザーに通知するシステムの研究開発を本プロジェクトの目標とした。

2. 調査

2.1 既存の混雑状況把握システム

現在、監視カメラや店舗内のカメラを利用し混雑状況を把握するシステムの一部の例として以下のものがあるので調査をした(表 1・2)。

表 1.小田急アプリ[7]

システム名
小田急アプリ
使用場所
駅構内
目的
運行支障発生時の混雑負担軽減
懸念点
駅構内のカメラで取得した鮮明な画像データを使用

表 2.VACAN[8]

システム名
VACAN
使用場所
観光地・商店街
目的
客の分散・混雑の回避
懸念点
店舗内のカメラで取得した鮮明な画像データを使用

2.2 既存の混雑状況把握システムの問題点

２つのシステムに共通する問題点は、監視カメラや店舗内のカメラの鮮明な画像データが処理の過程でどのように扱われているか分からず、ブラックボックスになってしまっていることである。システムの注意書きとして使用される画像データには外部からアクセス不可能と示されているが[7]、外部への流出の可能性も捨てきれない[9]。そのため、個人のプライバシーを含む鮮明な画像を用いる学習方法はプライバシー保護の観点において不十分であると考えた。

このことを踏まえ、本プロジェクトは画像の取得の段階で個人を判別不可能なまでぼかしを加える手段をとった。そうすることで、万が一保持していた画像が何らかの理由で外部に流出したとしても個人が特定されることなく一定レベルのプライバシーを保護することが可能と考えた。

3. システム概要

本プロジェクトでは、カメラが取得したぼかした画像をもとに機械学習を利用し、混雑度を算出する手法の研究と、その応用システムの開発をおこなった。システムの構成は以下の図の通りである(図 1)。実際の動作の流れとして次の通りである。

- カメラでプライバシーが保護できる程度にぼかした画像の撮影を行う。
- 撮影した画像をシステムに送信する。
- 機械学習部で画像を解析し混雑度の算出する(図 2)。
- 混雑度の数値をデータベースへ格納する。
- ユーザーが Web サイトにアクセスすると、格納された混雑度を元にユーザーを比較的空いている場所へ誘導する(図 3)。

尚、本プロジェクトでは感染症対策のため、大人数での撮影が出来ず、1 つの教室限定で少人数での混雑度を算出するという制限があった。もし他の場所で混雑度を算出したい場合、その場所ごとに異なる学習器を設置し、新たに画像を撮影及び学習をさせることが必要である。

リアルタイムで画像の撮影を行い、混雑度を算出し、スマートフォンへの表示を行うことで、ユーザーに対してタイムリーかつ有用な混雑状況を提示することが可能になると考えた。

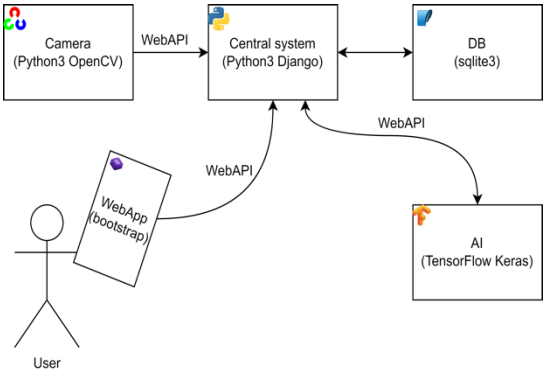

図 1.システムの構成

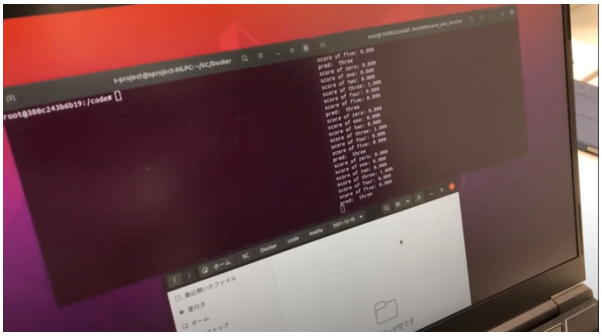
図 2.機械学習部の混雑度算出

図 3.混雑度を表示し、ユーザーを誘導（完成イメージ）

4. プロトタイプ実装

4.1 機械学習

人を検知するために最初は OpenCV を用いて人の輪郭の検出を可能にする「自動着色」と、画像の機械学習でオーソドックスな「画像分類」の 2 つの学習方法を試した。しかし、自動着色による学習方法では好ましい結果が出力されなかったため、本プロジェクトでは画像分類の方法で進めた。今回はプロジェクトメンバーのみでの撮影しか行えなかったため、0 人～5 人の 6 つにクラス分けをして学習を進めた(表 3)。

学習用画像の撮影は 3 段階に分けて同じ教室でそれぞれ別日に分けて行った。学習させた画像は次のようなものである(図4)。1段階目に撮影した画像のみで学習させた学習器では 30%ほどの精度であった。2 段階目には 1 段階目に撮影した画像を含めて学習させ、出力結果の最小値と最大値を除くことで外れ値を弾く変更を加え、さらに 10 枚の画像それぞれから予測された数値の合計を平均化することで精度としては40%程度となった(図 5)。また、教室のライトの反射や、人の服の色で判断している可能性があることから、画像のグレースケール化や輝度の正規化も試す必要があり、3 段階目に撮影した画像を含めてカラーで学習させた学習器では 50%、グレースケールでは 20%となった。グレースケールによる学習では精度が出せないことが分かったため、カラー画像による学習で進めていく方向で固めた。結果として現段階では、過学習が進む前に学習を中断させるアーリーストッピングの実行やバッチサイズを小さくするなどの変更点を加え、精度は 70%程度まで向上させることができており、これまでの中で最も良い精度を出すことができている。

さらに、1 つの学習器を利用して、学習させた教室とは別の教室での混雑度の算出も試してみたが、ほとんど正確な値を出すことができず、教室ごとに学習器が必要であるという結論が出た。

表 3.各段階に用いた学習画像の枚数

	各人数(枚)	合計(枚)
1 段階目	2000	12000
2 段階目	3000	18000
3 段階目	4000	24000

図 4.学習用画像

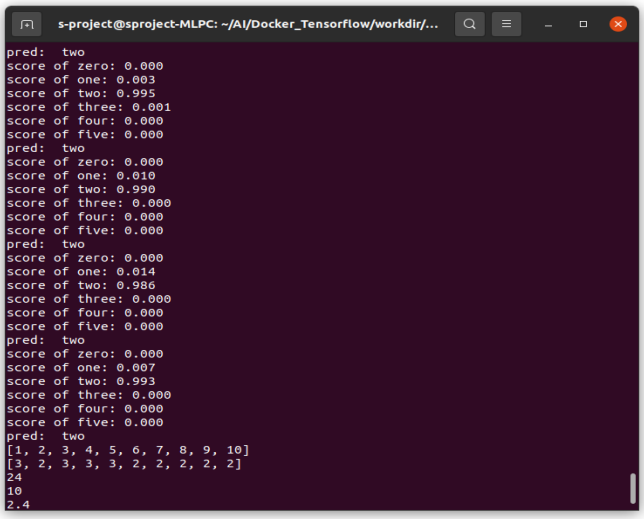

図 5.学習データのテスト結果
Pred:画像ごとに予想された人数
1 番下の数値：平均された人数

4.2　カメラ

4.2.1 ハードウェア

実験に使用するカメラの選定をおこなった。カメラは直接パソコンに接続しなくても使えるものを検討した。比較・検討を行った結果 Raspberry Pi のカメラを使用することにした。また、このカメラが正常に使えなかった時についてはネットワークカメラを使用することにした。Raspberry Pi 本体は無線と有線両方接続できる Raspberry Pi4 (図 6)を購入した。本体と接続するカメラの選定については、光学的にピントをぼかす方法を使うため、自動でピントを合わせるものではなく、マニュアルフォーカスのカメラを購入した。基盤がむき出しの状態だと

ショートの危険性や固定が難しいことから、Raspberry Pi を入れるケースを 3D プリンターで自作した。

4.2.2　ぼかした画像の取得

カメラがぼかした画像を取得するためにぼかす物を選定した。比較検討したものとしては、ピントをぼかす、セロハンテープ、ラップ、ビニール袋、ペットボトル、透明のケース、ジップロック、半透明の養生テープの 8 種類である。これらを比較・検討した結果ピントをぼかすというぼかし方に決定した。また、画像の撮影方法としては Python の OpenCV を使っている。

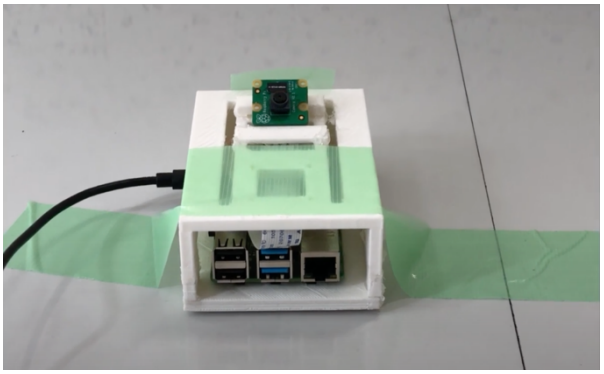

図 6.RaspberryPi4 カメラ

4.3　サーバー・クライアント部

クライアントであるカメラからバイナリーデータを Base64 でアスキー文字化し、Json を用いて画像をサーバーへ送信するシステムを作成した。サーバーシステムに関しては、データベース部・中央処理部・Web アプリケーション部の 3 つに分けて作成し(図 7)、Django という Web アプリケーションフレームワークを用いた。

1 つ目のデータベースでは混雑状況、場所と関連する情報を保存するために Django にデータベースの要素を指定して構築している。

2 つ目の中央処理システムについては複数の機能を取り入れた。まず、カメラからの画像をシステムの中央処理部で保存をしながら AI に転送をする機能と AI から送信された Json のデータを xml に変換してデータベースに保存をする機能を持っている。

3 つ目の Web アプリケーション部は、Web サイトからのさまざまなリクエストに対して対応する情報の検索を行い、その情報をサイトに返す機能である。サイトに表示させる情報としては、人の集まりそうな食堂や休憩所、PC 室の 3 つをピックアップし、それぞれの混雑状況を表示することにした。作成の際にはシンプルかつ見やすいものを意識した。スマートフォンからの利用も可能にするためにレイアウトを変更しないと使いづらいものになってしまうことから、無料かつ制限の少ない Bootstrap を利用した。混雑状況が同じでどちらに行こうか迷ったときのために、場所に関する詳細状況を閲覧できるようにしている。詳細情報としてメニューや座席数、営業時間の情報を提供している。

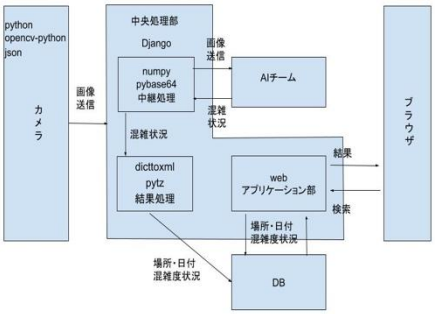

図 7.サーバー・カメラ部のシステム構成

5. 問題点と解決案

機械学習部の精度の問題として以下の点があげられる。

- 出力される混雑度の値に偏りが見られること
- 学習時の画像の状況と少しでも異なる点、例えば日射量や机の配置、服の色などがあると精度が低下してしまう。

それらの今後の改善案として、学習させる画像の場所、人数、時間帯などの種類や枚数を増やすなどして、さらに高い精度を出せるのではないかと考えている。また、現時点での本プロジェクト内の人数の関係上混雑度を人数によって分けているが、実際は食堂といった広い場所に設置し、初めから混雑度を低・中・高のように画像をファイルごとに分類して学習を進めていく方法も考えている。

6. 評価と考察

精度に関しては 70%を超えており、大きな数値の誤差が生じることは少ないため日常使いにおいて問題はないと考えられる。利便性に関しては、画像の撮影からユーザーへ混雑度を表示するまでの時間を計測したところ 30 秒以内だったためストレスを感じることなく、使用することができた。

混雑状況の把握をするシステムは商業施設や駅など多くの場所での使用が考えられる。そこで重要となるのが転用のしやすさである。今回作成したシステムは、全体が WebAPI で動作しているため転用が容易である。また、画像を撮影する頻度や混雑度を算出する枚数を調整することで、人数の多い場所でも少ない場所でも応用できるため混雑度表示のリアルタイム性も向上させることが可能であると考えられる。

7. 今後の展望

現在の本システムの問題点として、コロナ禍におけるプロジェクトだったので本番に近い稼働をさせデータを取ることがむずかしかったことがあげられる。本格的に設置台数を増やし混雑状況を表示できる場所を増やすことが出来なかった。また表示しているマップが画像のみという点もあった。今後の展望としてカメラの設置台数を増やし、多くの場所を表示できるようにすることやマップに道案内の機能を追加していくことを今後の課題としたい。

8.　参考文献

[1]東京新聞｜<新型コロナ>「ソーシャルディスタンシング」→「フィジカルディスタンシング」人との距離
https://www.tokyo-np.co.jp/article/17045

[2]日経クロストレンド｜混雑データが変える人流　街の「密」を丸ごと可視化
https://xtrend.nikkei.com/atcl/contents/18/00341/00003/

[3]ファミリーマートの監視カメラの取り扱いに関する問題
「YAHOO JAPAN ニュース｜ファミマ女性の胸元投稿の 2 人を解雇」
https://news.yahoo.co.jp/articles/574d396643e8b80cffadd3a321bab88f3b63b172

[4]DCS blog｜画像認識技術を用いた混雑状況可視化の取り組み　②深度カメラを用いたトラッキング
https://blog.dcs.co.jp/iot/20210311-cafeteriaIoT-2.html

[5]国土交通省｜リアルタイム混雑情報の提供に係る基本的なシステム構成について
https://www.mlit.go.jp/common/001355034.pdf

[6]SWITCHSCIENCE RealSense
intel realsense depth camera
https://www.switch-science.com/catalog/list/757/

[7]小田急アプリ
https://www.odakyu.jp/odakyuapp/

[8]VACAN
https://corp.vacan.com

[9]日本年金機構における不正アクセスによる情報流出事例について
https://www.nenkin.go.jp/oshirase/topics/2016/0104.html

（最終確認日 2022/01/26）