論　文

YOLOと顔認証を用いた挙手者の検出及び PTZカメラを用いた授業等における指名の自動化

荻野敦史[a], 田中雅博[b]

[a] 甲南大学 大学院 自然科学研究科 知能情報学専攻
神戸市東灘区岡本８－９－１, 658-8501
[b] 甲南大学 知能情報学部 知能情報学科
神戸市東灘区岡本８－９－１, 658-8501

（受理日 2021 年 5 月 11 日）

概 要

挙手は授業やイベント等でしばしば行われ，授業の発表の際には，教員が挙手した学生や生徒の中から指名を行う場面がよく見られる．しかし大学の講義等の多人数での授業や多数決などで，挙手者数を把握する必要がある状況では，挙手の見落としや挙手者数の計測に時間がかかる問題がある．このような問題の解決として挙手を自動的に検出することが挙げられ，授業では教員の負担を減らし生徒の評価や理解度の指標を得ることができるなど多くの応用が可能となると考えられる．

本論文ではPTZカメラを用いて撮影を行い，YOLOを用いて人の顔と挙手のリアルタイム物体検出を行う．そしてYOLOによる人の顔と挙手の検出を基に顔認証を用いて個人を把握することや，常に顔認証を行うことによる負荷を軽減するための個人追跡の手法を提案する．またこれらの応用として挙手者を自動的に指名するシステムを構築する．

はじめにYOLOv3を用いて学習及び検出を行い，精度向上のための改善を行った．次に検出性能が改善されたモデルのYOLOv4を用いて学習と評価を行い，より高い精度での検出が可能となった．

またこのシステムをリアルタイムで動作させるためにはGPUを搭載した教壇等に持ち運びが可能なマシンが必要となるため，本研究ではJetson AGX Xavierを用いた利用を想定しシステムの動作実験とリアルタイム処理性能を検証した．

キーワード: 挙手者検出，Deep Learning，顔認証

1　はじめに

挙手は，学校教育等において主に問題提示に対して，教員が挙手を要求した際に学生や生徒によってしばしば行われ，その他には討論や会議において賛否を問う際や，アンケートの回答等で挙手が行われる．

しかし多人数または大教室での授業では挙手を見落としやすく，教員は授業を行いながら生徒全員への配慮が困難であるため，より一方向的な授業になりやすい問題がある．また挙手者を指名する際

に，指名者に偏りが出ることは好ましくないが，偏りを無くすためには指名者を記録し指名者を選択する手間がある．同様に多数決やアンケートの挙手回答においては，挙手者の計測に時間がかかる問題もある．更に今後普及する可能性がある遠隔授業でも，一人の教員がより多人数を相手に授業を行うことが可能または必要となる状況が多くなると考えらえる．

これらの問題を解決するため，挙手者を自動的に検出することで，教員に挙手者の存在を知らせて挙手者の検出を応用し挙手者を自動的に指名することや，瞬時に挙手者数の計測などが可能となる．本研究では，リアルタイム処理が可能な深層学習の物体検出アルゴリズムである YOLO (You Only Look Once) [1] を用いて，挙手と人の顔の検出及び検出精度向上を行う．また顔認証システムを用いた個人認証によって挙手者の同定を行い，挙手回数や指名回数等の記録を基に挙手者を自動的に偏りなく指名するシステムを構築する．

また本研究で構築するシステムは，教室等で利用することから持ち運び可能なマシン上で稼働することが求められるため，Jetson Xavier 環境でのシステムの実験とリアルタイム処理性能について検証を行った．

本論文の構成は以下の通りである．第 2 章にて YOLO の概要と本研究の深層学習の学習環境等について，第 3 章にて YOLO を用いた挙手と人の検出及び精度向上について，第 4 章にて顔認証システムを用いた個人の特定と個人追跡の手法について，第 5 章では挙手者の指名システムと Jetson AGX Xavier でのシステムの動作実験及びリアルタイム処理性能について述べる．

2 YOLO の概要と研究環境

物体検出アルゴリズム YOLO は複数のバージョンがあり，本研究では YOLO，YOLOv2 [2] に続く YOLOv3 [3] と YOLOv4 [4] を用いて物体検出を行う．本章では YOLOv3 と YOLOv4 を含む YOLO についての説明と研究環境について示す．

2.1 YOLO の概要

YOLO は 2016 年に Joseph Redmon らが発表した，リアルタイム処理が可能な物体検出アルゴリズムである [1]．検出対象の物体を含む画像を多数収集し，物体を囲む矩形の座標とクラスをタグ付けするアノテーションを行い，それらを教師データとして学習を行う．その学習をした結果を用いて，学習していない画像にある検出対象の物体の位置とクラスを予測し，検出することが可能である．YOLO には検出性能や処理速度が改善された後継バージョンがある．YOLO の初期モデルでは多数の物体が画像内にある場合や，物体のスケールが異なる場合には検出が困難となるため，本研究では YOLOv3 と YOLOv4 を用いた検出を行う．

YOLOv3 は，これまでのバージョンで問題となっていた異なるスケールの物体に対しての検出を行えるようになり，撮影環境による大きさの異なる物体に対しての検出精度の改善や，画面内に多数の物体がある場合でも検出を可能としたモデルである．

YOLOv3 までは，Joseph Redmon が中心となって改良したモデルであるが，YOLOv4 は Alexey Bochkovskiy らによって 2020 年 4 月末に発表された YOLO の後継バージョンで，YOLOv3 に CSPNet

やSSP，PANet等の物体検出の性能や処理速度の向上を可能にする構造を追加しており，YOLOv3よりも検出精度が向上している．

2.2 システム環境

本システムで，YOLOv3やYOLOv4を用いた物体検出を行うための学習や推論を行う環境と使用するPTZカメラについて記す．

2.2.1 YOLOの学習フレームワーク

YOLOv3及びYOLOv4の学習フレームワークは，C/C++で書かれた機械学習フレームワークのDarknetを用いる [5]．Darknetは YOLOv3, v4開発者が提供しているフレームワークであることや，C/C++で書かれているため学習速度が高速であり，学習において役に立つ機能が多いためDarknetでの学習を行う．ネットワークの構成や学習の設定など，本稿で言及していない点は，Darknet内のyolov3.cfg及びyolov4.cfgの初期値 [3], [4] を利用している．

2.2.2 応用システムのフレームワーク

応用システムを構築するにあたりC言語で書かれたDarknetは応用開発が難しい．よって応用システムの構築はPython環境で行うため，Darknetで学習した重みをPyTorchで利用できるように開発されたPyTorch版 YOLOv3及びYOLOv4を利用する [6].

2.2.3 学習画像の収集方法

YOLOで学習を行うために挙手を行っている人の画像が必要となる．多様な学習データが必要となるため，主にインターネット上から挙手を行った画像を集めて学習データとしている．

2.2.4 システムで使用するカメラ

本システムは多人数や広い教室等で挙手者の検出や顔認証を行うことを目的としているため，そのような状況でも顔認証が行えるようにパン，チルト，ズーム機能を搭載したPTZカメラ (TEVO-NV10U) を用いたシステムを構築する．本研究で用いたPTZカメラ (TEVO-NV10U) はパン角度が $-175° \sim 175°$，チルト角度が $-35° \sim 55°$，最大10倍の光学ズーム可能なカメラである．

3 挙手と人の検出及び精度向上

本章では，主に YOLOv3 を用いて挙手と人の検出を行い，アノテーション方法や画像枚数，入力解像度の違いなどによる精度の評価と比較を行う．まず YOLOv3 を用いて，顔と前腕及び手を含むアノテーションと顔と肩から肘を含むアノテーションでの学習を行い，それらの検出精度の比較や，入力解像度を変更した際の検出精度の比較を行う．また YOLOv4 を用いて学習と精度の評価を行い，YOLOv3 との検出精度の比較を行う．

3.1 評価指標

YOLO などの物体検出の評価指標には，Precision，Recall，Average Precision (AP)，Mean Average Precision (mAP) が用いられる [7]．ここでは，検出精度の評価指標として AP，mAP の値を用いる．

3.2 YOLOv3 を用いた顔と前腕及び手を含むアノテーションでの検出

まず図 1 の矩形で囲われている領域のように，顔と前腕及び手を含めることを条件としたアノテーションを用いて学習を行う．挙手の腕が他の物体に隠れる等で，一部見えていない場合には，その他物体に被せるようにアノテーションを行う．このアノテーション方法をアノテーション法 A とする．またアノテーション法 A での学習におけるクラスの種類は，挙手のみとしており，挙手の検出精度のみを評価する．

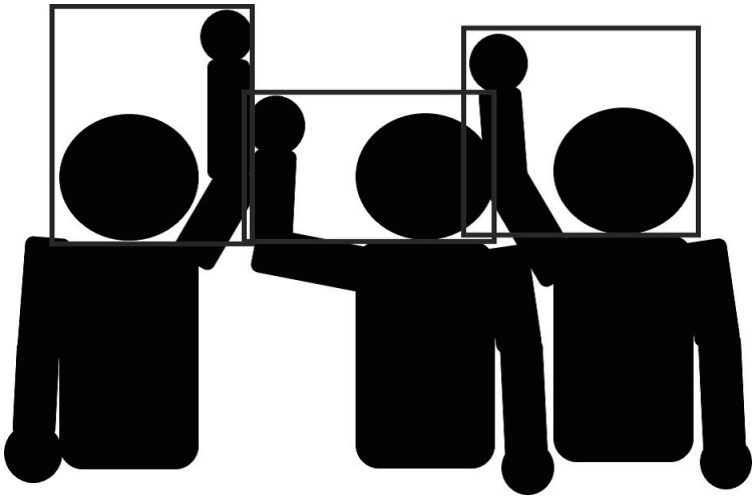

図 1: アノテーション法 A の例

3.2.1 学習設定

本章の学習に利用した画像の枚数と学習設定は以下のように行った．

- 訓練用画像: 662 枚
- 検証用画像: 284 枚
- バッチ数: 64
- イテレーション数: 20000 回
- エポック数: 1819 回
- 学習時解像度: 608×608

これらの画像は右手を挙げた挙手者及び左手を挙げた挙手者の画像であり，いずれの挙手もアノテーションを行っている．訓練用と検証用画像は，収集した画像の 7 割を訓練用画像とし，3 割を検証用画像としている．

モデルへの入力解像度は YOLO のモデルの制約により正方形画像にする必要があるため，608×608 とする．入力画像がこの解像度でない場合はリサイズ処理を行う．Darknet では，その際にアスペクト比を固定したままリサイズして，空白となる部分を RGB=(128, 128, 128) で埋めるレターボックス処理を行っている．また利用する重みは，学習過程での検証用画像を用いた評価の mAP が最大となった重みを採用する．

3.2.2 学習結果による検証用画像の評価と考察

学習後の検証用画像による精度評価は，挙手の AP が 0.6730，Precision は 0.79，Recall は 0.61 であった．mAP はクラスが 1 種類なので AP と同じ値である．これらの評価値は信頼度閾値 (confidence threshold) を 0.25 として信頼度が 0.25 以下の予測結果は除外して計算している．評価の結果から Precision が 0.79 に対して，Recall が 0.61 と少し低くなっているため，全体的に検出すべき挙手の検出漏れが多い傾向にある．

この学習結果を用いて挙手の推論結果を確認すると，挙手者が大きくはっきりと写っている画像は，右手挙手者と左手挙手者どちらも良い検出結果となっている．しかし，後方に写る人の顔や腕の一部が，前方の人の腕や顔等と被る状況では，検出率が低い問題がある．また特に画像全体のサイズに対して，挙手者の大きさが比較的小さく写る挙手の検出率が低い問題もある．学習画像には数人が大きく写り挙手を行っている画像や，多人数が写っていても低解像度で挙手判別が難しい画像が多かったため，より画像に対して小さく写る挙手者の画像を追加することや，より解像度が高い画像の追加が必要である．

更にアノテーション方法として顔と前腕及び手を含めることを条件としたため，特に上腕をあまり上に挙げず前腕のみを上に挙げている挙手に対して，図 2 のように，挙手をしていない隣の人も挙手として誤って検出する問題がある．アノテーション法 A では，このような挙手での顔から肩，肘，手までの連続した特徴がアノテーション範囲に入っていないため，特徴を学習する際に，隣の人の腕かを判断する特徴がないことが原因であると考えられる．また学習画像内には，手をあまり高く挙げない挙手画像が少ないことも精度に影響していると考えられる．

アノテーション法 A では手を含めたが挙手の特徴として手の形はあまり重要ではなく，腕が挙がっているかが重要であることから手を含める必要性はないと考えられる．挙手を判断するにあたって重要でない範囲を除外することでより学習で挙手の特徴を捉えやすくし，検出ボックスを小さくしてボックス内に他の物体が多く含まれないようにすることで，検出性能を向上させることができると考えられる．

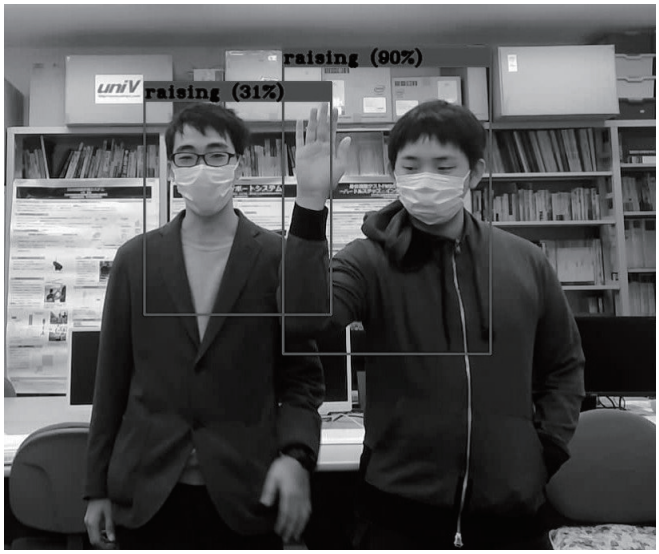

図 2: アノテーション法 A による他人の挙手を検出する問題の例

3.2.3 入力解像度による検出精度向上

推論時にモデルに入力する画像の解像度は設定した解像度にリサイズされるため，比較的高い解像度の画像内にとても小さい物体がある場合にはその特徴がほとんど消えてしまう可能性が高く検出が困難であるため，推論時の入力解像度を高く設定することにより，小さな物体の検出率が向上するか検証を行う．本学習結果を用いて入力解像度を 832 × 832 と，928 × 928 として検証用画像を用いた評価し精度の比較を行った (表 1).

表 1: 様々な解像度における検出精度の比較

	AP	Precision	Recall	mAP
608 × 608	0.6730	0.79	0.61	0.6730
832 × 832	0.7178	0.79	0.65	0.7178
928 × 928	0.7171	0.67	0.67	0.7171

結果として，入力解像度 608 × 608 と 832 × 832 の AP を比較すると，0.6730 から 0.7178 と高くなっており，前者の解像度よりも小さな物体の検出数が増加していることが，推論によっても確認できた．しかし 832 × 832 と 928 × 928 での AP の比較では，0.7178 から 0.7171 と低下しており，Recall

は向上したがPrecisionの値は低下している．Precisionの低下とRecallの向上から，検出すべき物体の検出率は向上したが，検出すべきでない物体を多く検出する誤検出が増加している．

精度が低下した原因は，検証用画像内の解像度の多くは928×928以下であることが挙げられる．全ての入力画像を指定した解像度にリサイズするため，指定した解像度以下ではアップサンプリングが行われて，画質が粗くなってしまうことが原因であると考えられる．これらの結果から入力解像度には832×832を用いる．

3.3 YOLOv3を用いた顔と肩から肘を含むアノテーションでの検出

アノテーション法Aによる学習では，手を高く挙げる人の場合などで検出する矩形が縦方向に大きくなってしまい多人数での検出を行う際に誤認識等の問題となる事が考えられる．そのため可能な限り挙手を特徴づける範囲を小さくし，アノテーション範囲を狭めることにより，挙手以外の物体が矩形内に入ってしまうことを避けることで，誤認識を減らし精度が向上すると考えられる．

また肩や肘を含めることでアノテーション範囲に顔から腕までの連続した特徴が入っており，他人の挙手した腕を検出する問題を解決することができると考えられる．

具体的なアノテーション方法は図3のように顔と肩，肘を含むことを条件として，腕を高く挙げており前腕以降を含まなくてもよい場合には，含まずにアノテーションを行う．顔の位置程度までの挙手の場合には，頭の高さまでを含める．条件である肩から肘までが，他物体により隠れる場合には挙手者を判断することが難しく，誤認識を防止するためアノテーションから除外する．このアノテーション方法をアノテーション法Bとする．

また挙手者を特定するために新たに人の頭をアノテーションした人クラスを追加する．図3の大きな矩形が挙手者クラスで，頭部のみの小さな矩形が人クラスである．

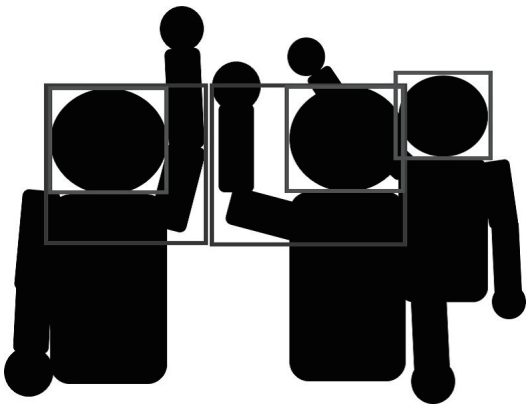

図3: アノテーション法Bの例

3.3.1 学習設定

アノテーション法 B の学習設定は以下のように行った．アノテーション法 A による学習では，低解像度の画像や撮影状況により挙手者の判断が難しい画像が多く含まれていたため，そのような画像は除外を行い，新たに画像に対して小さく写る挙手者を含む画像や，手を高く挙げない挙手者の画像を多く追加した．イテレーション数はこれまでの学習から，20000 回行うまでに収束していることを確認したため，10000 回とし，学習時の解像度設定は 800×800 とした．

- 訓練用画像: 719 枚
- 検証用画像: 284 枚
- バッチ数: 64
- イテレーション数: 10000 回
- エポック数: 834 回
- 学習時解像度: 800×800

3.3.2 学習結果による検証用画像の評価と考察

検証用画像を用いた精度評価の検証を行い，推論時の入力解像度は 832×832 とした．精度評価は，挙手クラスの AP が 0.8075，人クラスの AP が 0.9105 となり，Precision が 0.86，Recall が 0.83，mAP が 0.8590 となった．挙手クラスの AP はアノテーション法 A での 0.6730 から 0.8075 と大幅に精度が向上し，新たに追加した人クラスの AP は 0.9105 と高い精度で検出できている．

学習結果を用いた推論から，アノテーション範囲に肩や肘を含めたことで，アノテーション法 A での学習時に問題となった他人の挙手を検出する問題の画像での推論結果は，実際の挙手者のみを検出していることが確認できた (図 4)．またアノテーション範囲を狭くしたことにより，挙手者の後列で挙手を行う人の検出精度や，画像に対して小さく写る挙手者の認識の精度がアノテーション法 A から向上している．

図 4: アノテーション法 A による他人の挙手を検出する問題の解決

3.4 YOLOv4 を用いた挙手検出による精度向上

YOLOv4 [4] は 2020 年 4 月末に発表された YOLOv3 よりも物体検出の精度が高いモデルとなっているため，YOLOv4 を用いた挙手の検出を行うことで精度の向上が期待できる．

3.4.1 学習設定

学習設定や画像，アノテーション法は，アノテーション法 B の学習で行った設定と同様の設定を利用する．ネットワークは YOLOv4 のモデルを利用する．

3.4.2 学習結果と考察

検証用画像を用いた精度評価の検証を行い，推論時の入力解像度は 832×832 とした．精度評価は，挙手クラスの AP が 0.8471，人クラスの AP が 0.9260 となり，Precision が 0.82，Recall が 0.87，mAP は 0.8866 となった．YOLOv3 によるアノテーション法 B での学習から挙手の AP は，0.8075 から 0.8471 へと更に精度が向上した．Precision は 0.86 から 0.82 へと低下しているが，Recall が 0.83 から 0.87 へと向上していることから，YOLOv3 では検出できていなかった挙手を検出可能となっており，検出率は大きく向上している．

学習結果による推論例を図 5 に示す．顔と腕や肩を含む大きな矩形が挙手クラスの矩形であり，頭部のみの小さな矩形が人クラスの矩形である．推論の結果から挙手を行っている人のみを正しく検出していることを確認できた．YOLOv4 を用いた学習では，YOLOv3 のアノテーション法 B では検出

できなかった挙手を多く検出していることが確認された．特に YOLOv3 での画像に対して小さく写る挙手を検出できない傾向が，YOLOv4 での検出により減少し，それらの検出率が向上していた．

YOLOv3 での学習より Precision が少し低下していることから，少し誤検出が多くなっているが，検出できていなかった挙手の検出数の方がより増加したことから，YOLOv4 によって，より高い精度で検出が可能となった．

図 5: YOLOv4 による挙手と人の頭の検出例

4　顔認証による個人の特定と追跡

挙手者の同定と挙手回数及び挙手者数の計測を行うためには，YOLO を用いた挙手及び人の検出を利用し，顔認証を用いて各個人の特定と追跡を行う必要がある．

本章では，まず顔認証システムを用いた個人名の特定を行い，次に特定した個人を追跡および挙手者の同定の手法について記述する．

4.1　顔認証システムによる個人特定

本研究で利用する顔認証システムは，Python ライブラリである Face Recognition を利用する．Face Recognition は機械学習ライブラリである dlib を利用して開発されている顔認証ライブラリである [8].

顔の領域推定と顔の特徴を抽出する CNN により，登録したい人の顔画像一枚を用意するだけで，高い精度での認証が行える．

4.1.1 顔認証システムの処理過程

顔認証の登録の処理過程を図6に示す．まず，登録するための顔を含む画像を入力し，HOG+SVMまたはCNNによって，顔の領域を推定し矩形座標を出力する．HOGは局所領域内の輝度の勾配方向をヒストグラム化した特徴量であり，その特徴をSVMによって分類することで顔の領域を推定している．HOG+SVMは処理速度が速く，CNNは顔が小さく写っている場合や解像度が低い場合でも顔領域を推定できるが，そのような画像では認証精度が低下する．本研究ではこの問題を解決する際に後述する光学ズームを用いた高解像の顔画像を取得を行うため，HOG+SVMの推定で十分である．その後に検出した顔領域を顔の特徴を抽出するResNetをベースとしているCNN [9] に入力し，128次元の顔の特徴ベクトルを出力してリスト上にそれぞれの登録者の特徴ベクトルを保存する．

次に顔認証の処理過程を図7に示す．認証する際には登録と同様に，特徴ベクトルの出力までを行う．その次に登録してある特徴ベクトル全てと，認証する特徴ベクトルとのユークリッド距離の計算を行い，その特徴距離が最小となる人を本人であると判別する．仮に登録外の人が認証を行うと，特徴距離の最小値が大きくなるため，一定の閾値を設けることで，登録外の人物かの判別も可能である．

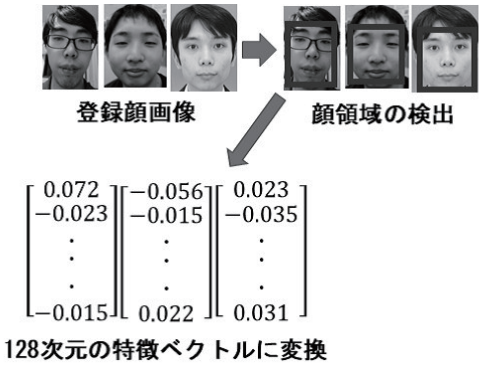

図 6: 顔認証の登録例

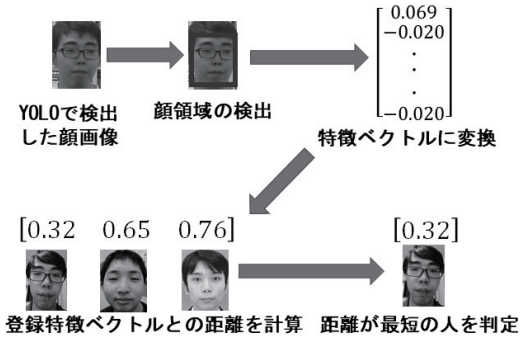

図 7: 顔認証の認証例

4.1.2 顔認証システムの検証

検証として顔画像16人を登録し，登録者の別の顔画像に対して認証した結果は，全員正しく認証した．それぞれの最短特徴距離の平均は0.355で，その中で最長の特徴距離は0.44であり，登録外の人物を認証すると特徴距離は0.58であった．この結果から，本研究では登録外の人物と判断する閾値を0.5としている．

しかし入力する顔画像がカメラから遠い場所で撮影されており，非常に小さく写っている場合には，解像度等の影響により認証精度の低下や顔領域の推定が行えなくなってしまう．本研究は多人数の検出を目的としているため，顔画像が小さい場合でも正しく認証できるようにする必要がある．

また挙手者特定の際の顔認証に入力する画像は，YOLOにより検出した人クラスの画像となる．この画像はほぼ顔画像であるが，毛髪など必要のない部分も含むことや，YOLOでは横や後ろを向いた

人の頭も検出するため，顔認証システムの顔領域推定を行い顔の領域のみを抽出する必要がある．

4.1.3 顔画像の解像度による顔認証の限界

顔認証の精度は，カメラから近く高解像度であれば高い認証精度であるが，カメラから遠く離れた位置で撮影されているなどで顔画像が低解像度になればなるほど顔認証の精度は落ちてしまうため，正しく認証できる顔画像の解像度について検証を行う．

検証に用いたカメラは解像度が 1920 × 1080，画角は 75° のカメラである．検証する顔画像は異なる距離から撮影した図 8 の 4 枚であり登録者名 “kisi” で登録している．それぞれのカメラからの撮影距離と解像度を表 2 に示す．この 4 つの画像に対して顔認証を行った．登録済みの顔画像は本人を含む 23 人を登録している．

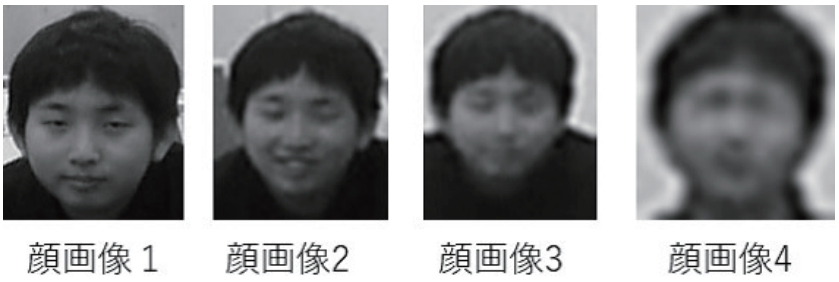

図 8: 異なる解像度の顔画像

表 2: 図 8 の解像度と撮影距離

	顔画像 1	顔画像 2	顔画像 3	顔画像 4
解像度	123 × 146	62 × 76	41 × 51	25 × 27
距離	2 m	5 m	7 m	10 m

それぞれの顔認証の結果は表 3 となり顔画像 1 と顔画像 2 は正しく認証できたが，顔画像 3 と 4 は顔領域の検出が行えず顔認証を行うことができなかった．最も解像度の高い顔画像 1 では特徴距離が 0.298 であるが，解像度が低い顔画像 2 では 0.382 と特徴距離が長くなって精度が低くなっており，解像度が 60 × 60 程度が必要となる結果であった．認証できなかった画像で，顔画像 3 の場合は画像をバイキュービック等の補間処理によるアップスケールを行うことで，認証を行うことができるが，さらに認証の精度が低くなり誤認証する可能性が高くなってしまう．また顔画像 4 の場合は顔の特徴がほぼ消えてしまっているため，アップスケール等の画像処理を行っても正しい認証を行うことは困難である．しかし広い教室で多人数での撮影となると撮影距離 10 m などの広域顔認証が必要となる．

この問題を解決するために本システムでは PTZ カメラを用いることで，通常の広角撮影では認証できない顔画像を光学ズームで高解像度化し認証を行う．

表 3: 図 8 の顔認証結果

	顔画像 1	顔画像 2	顔画像 3	顔画像 4
認証の結果	本人を正しく認証	本人を正しく認証	認証不能	認証不能
最短特徴距離	0.298	0.382	-	-

4.2 個人の追跡と挙手者の同定

YOLO で出力した顔矩形を毎フレーム認証すると，多人数の検出を行う際に処理負荷が大きくなってしまうことや，ズーム機能を用いた顔認証を行うため，広角撮影時に検出された人クラスの顔画像に対して，顔認証を行わずに個人の特定を行う必要がある．個人追跡の手法と同様の手法を用いて挙手者の同定の手法を記述する．

4.2.1 個人追跡の手法

まずメンバーリストを作成し，最初に YOLO により人クラスが検出されたフレームで，人クラスの矩形座標をリストに登録を行う．その次のフレーム以降の追跡手法として，十分なフレームレートがあれば前フレームと現フレームで検出される人クラスの矩形の位置は，ほぼ変移がないことを利用して，前フレームで保存した全ての矩形と現フレームで検出した矩形の一致度をそれぞれ計算して，一致度が最大となった矩形のメンバーへ現フレームの矩形座標を更新することで追跡を行う (図 9)．一致度の計算は重複面積を S_O として，重複していない矩形の面積をそれぞれ S_A，S_B とし（図 10)，一致度を M とすると，一致度は式 (1) である．

$$M = \frac{S_O}{S_A + S_B + S_O} \tag{1}$$

この手法は処理速度が遅い場合では追跡が難しく，必要となるフレームレートは被写体の大きさや，被写体が動く速度によるが，5〜10 FPS 以上の処理速度であれば十分に追跡が可能である．また YOLO の検出漏れ等により追跡できなくなることや，追跡対象がずれることが起こり得るため，一定の秒数毎に再度顔認証による特定と修正を行う．

4.2.2 個人追跡の補正

個人追跡の欠点となる処理速度が低い場合や YOLO 検出漏れが起こると正しく追跡を行えない可能性がある．しかし追跡の過程で前フレームで検出した矩形と重複しない矩形が検出された場合でも，前フレームの矩形と近い距離の矩形があれば，その矩形が追跡対象である可能性が高いため矩形との距離による追跡の補正を行う．また顔認証中に追跡が行えなくなると同名のメンバーが追加されてしまうため，顔認証で同名のメンバーがいる場合には同名のメンバーの座標へ更新を行う．

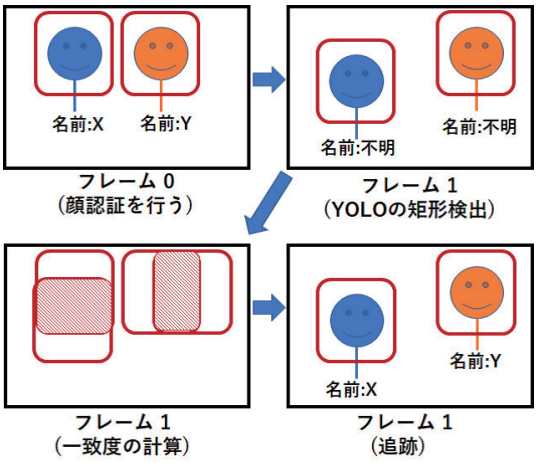

図 9: 個人追跡の例

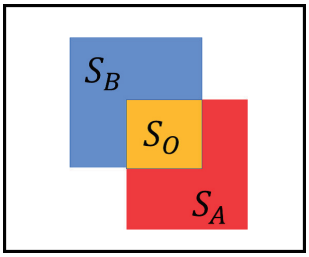

図 10: 重複矩形の例

顔認証中以外の補正方法は，まずそのフレームでの各メンバーの追跡の有無を記録し，前フレームからの追跡を行えない矩形の中心座標と追跡の行われていないメンバーの中心座標の距離をそれぞれ計算する．そして最短距離となる矩形同士の距離が，前フレームからの追跡を行えない矩形の縦と横の長さの平均に 1.5 倍した長さ以内であれば，そのメンバーへ座標等の更新を行う．

4.2.3 挙手者の同定と判定方法

YOLO により検出した挙手矩形では，メンバーリストの誰が挙手したか不明であるため，挙手者を特定する必要がある．挙手者の同定は，検出された挙手矩形内に顔も含まれているため，挙手矩形の画像を用いて顔認証を行えば特定できる．しかし，挙手を検出する度に顔認証システムを利用するには処理負荷が大きくなることから，人クラスの矩形は基本的に挙手クラスの矩形の内側に位置することを利用し，個人追跡で用いた同様の手法によって顔認証を利用せずに特定が可能である．検出した挙手の矩形とメンバーリストに保存されている顔の矩形との一致度を計算し，一致度が最大となったメンバーが挙手したと判定することで，挙手者の同定が可能となる．

また特定したメンバーをすぐに挙手していると判定すると，誤認識の可能性があるため，一定フレーム以上連続して挙手の検出を行うことで，挙手したと判定する．

実際の検出は図 11 のようになり，顔認証と挙手者の同定及び下部に挙手者数と人数を表示している．顔の矩形ラベルに顔認証した際のラベルをつけており，図 11 の右方は “ogino” で，左方は “sakurai_s” で登録している．その結果正しく顔認証が行えており，挙手したと判定されると右のメンバーリストにある名前を赤く表示しているが，これも正しく挙手者を特定している．

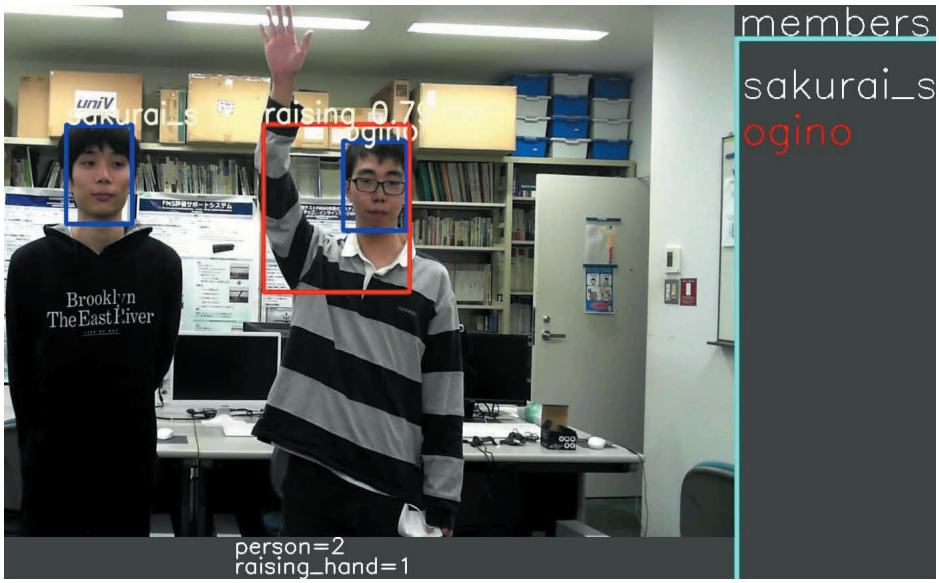

図 11: 挙手の検出と挙手者の同定の例

4.3 PTZ カメラの制御とズームを用いた広域顔認証

顔認証の精度は解像度が低いほど下がってしまうため広角カメラでは後列者の顔認証を行うことが困難である．この問題を解決するために本システムでは PTZ カメラの光学ズーム機能を用いて後列者の顔画像の解像度を高くし顔認証を行う方法を用いる．

顔認証の方法はまずズームを行い後列者の顔認証を最初に行った後，徐々にズームアウトを行い前列者の顔認証を行う．後列者は顔認証後に個人追跡によって追跡が可能であるため，広角撮影時にも YOLO により検出された人クラス矩形から常に個人の特定が可能となる．また PTZ カメラのパンやチルトを利用すると個人追跡による個人の把握を行えないため，本システムではズーム機能のみを利用し広角撮影時に前方の人が画角に入る列を検出や顔認証の対象とする．

PTZ カメラは Pelco-D プロトコルを用いて制御を行ったが，ズーム等の命令を送信すると，次に停止の命令を送信するまで事前の命令を続けるため，処理速度が遅い場合は制御が難しい．そこでズームやパンチルトの位置を記憶するプリセット機能を用いて事前に最大広角の倍率や最大望遠の倍率をプリセット登録し，指定のプリセットへ移動する命令で制御を行う．また徐々にズームアウトしながら顔認証と個人追跡を行うために，最大広角と最大望遠の間で正しく個人追跡を行える間隔のズーム倍率でプリセット登録を行い，それぞれのズーム倍率での顔認証を行う．本システムでは最大望遠と最大広角を含めて 16 分割した間隔の各倍率でプリセット登録を行い，それぞれの倍率で 1.5 秒間顔認証を行うようにしている．

4.3.1 光学ズームによる広域顔認証の実験

実際に大学の講義室での広域顔認証の実験を行った．顔認証を行う前の広角撮影した結果が図 12 となり，通常の顔認証では最前列者より後列の人は顔の特徴が欠損し顔認証が難しい．また右端と左端の列は前方の席が画角に入っていないため，顔認証の対象は中央から左右の机の 2 列である．検出された人矩形のラベルは顔認証前であるため，全員メンバー外として “Not_member” と表示している．

光学ズーム機能を用いた顔認証の過程は，図 13 のようにズーム状態で顔認証を行い徐々にズームアウトをしながら認証と追跡を行った後に，最終的には図 14 のように広角撮影で個人追跡を行う．

結果としては右列の下から二人を除き正しい個人特定が行えていた．その二人は，右の 2 列目の人は顔認証の段階では正しく認証は行えていたが，検出される矩形の距離が近く，ズームアウトの速度の影響により，ズームアウトを行い 1 列目の人が検出された際に，2 列目の人の追跡を 1 列目の人へ誤って行ったため，正しく追跡や個人特定が行えなかった．この問題はプリセット登録によるズームアウトの間隔をより狭くすることや，これらが原因となる誤った追跡が行われないような補正処理を行うことが必要と考えられる．

図 12: 顔認証前の最大広角撮影

図 13: 光学ズームによる顔認証の例

図 14: 顔認証終了後の最大広角撮影

5 応用システムと Jetson AGX Xavier でのシステム動作実験

本研究での挙手者の検出と個人の特定を利用し挙手者を自動的に指名するシステムを構築する．本研究のシステムは教室等での利用が想定されるが，YOLO や顔認証システムは GPU を搭載したマシンでなければリアルタイム処理が難しいため，本研究では持ち運び可能な GPU マシンである Jetson AGX Xavier でのシステムの稼働を想定して行う．

5.1 挙手者指名の手法

挙手を指名する際には挙手者の中で指名された回数が少ない人を優先的に当てることで，様々な人が発表を行いより良い授業になるため，指名回数を管理し挙手を行った人の中でも指名回数が少ない人を優先的に指名するシステムとする．このシステムによって挙手を要求した際の各個人の挙手回数と指名回数を把握することが可能となる．挙手の受付から挙手者を指名するまでのシステムの流れは以下の通りである．

1. 特定のキー入力により挙手の受付を開始する．
2. 受付開始後の数秒間での挙手者を記録し，受付終了と同時に挙手した人をリストアップする．
3. 受付終了後に挙手した人の中からこれまでの指名回数が最も少ない人のリストを作る．最初は全員が指名回数 0 である．
4. 指名回数が最も少ない人のリストの中からランダムで指名者を決定し画面上に表示する．

5. 挙手の受付によって挙手した人の挙手回数と指名された人は指名回数をカウントアップし記録する．

5.2 Jetson AGX Xavier でのシステムの動作実験

実際に大学の講義室での挙手者検出やズーム機能を用いた広域顔認証による個人特定と追跡及び挙手者の自動指名システムの実験を行った．本実験は図 12 から図 14 の顔認証の実験以降に続けて行っているため顔認証の結果は図 14 と同様である．

まず挙手者指名の受付を開始し質問を行い，挙手を要求した結果が図 15 となり，4 人が挙手を行い正しく検出された．次に挙手を行った人の記録を行い，全員の指名回数は 0 回であるため指名の対象は挙手した全員となりランダムに指名を行う．指名の結果は図 16 の下部に示されたとおり “kisi” となった．本システムには右側にメンバーリストとして図 17 のように名前と挙手者指名中に挙手した回数，指名された回数を表示している．今回は挙手を行った 4 人に挙手した回数が 1 回カウントされ “kisi” には指名された回数が 1 回カウントされていることから，想定した動作を行っていることを確認できた．

5.3 Jetson AGX Xavier でのシステムのリアルタイム処理性能

本研究では Jetson AGX Xavier でのリアルタイム処理を行うため，Jetson AGX Xavier で YOLO の推論を行うにあたり処理負荷となる GPU 演算の高速化を行った．GPU を利用したニューラルネットワークの推論時間を高速化する手法として演算精度を一般的な単精度 (FP32) から半精度 (FP16) にすることで高速化できる．半精度での演算は特に NVIDIA の第 8 世代 GPU 以降に搭載されている Tensor コアでの演算速度が大幅に向上し，Jetson AGX Xavier にも Tensor コアが搭載されているため高速化が期待できる．また半精度計算を行うことによる推論精度の低下は起こらないため，検出精度を維持したまま高速化が期待できる．前節の実験の際には半精度演算での推論を行った．

半精度演算での高速化を行った実験時の処理速度は図 18 の下部に示されているように，全体を通して 6 FPS 前後での安定した処理を行えた．また本実験では半精度演算での推論でも YOLO による挙手や人の検出は精度良く行われており，リアルタイム処理は可能であることが確認できた．

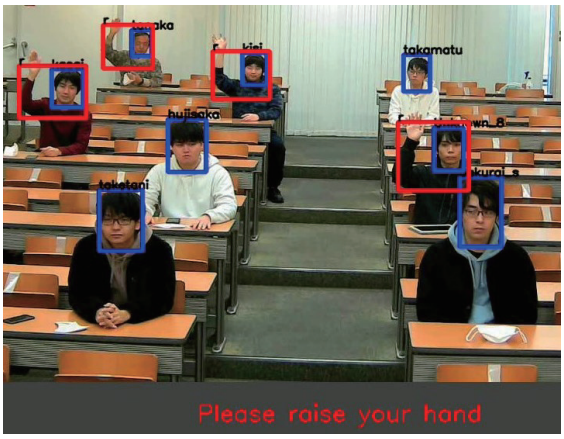

図 15: 挙手者自動指名の挙手受付中

図 16: 挙手者の自動指名

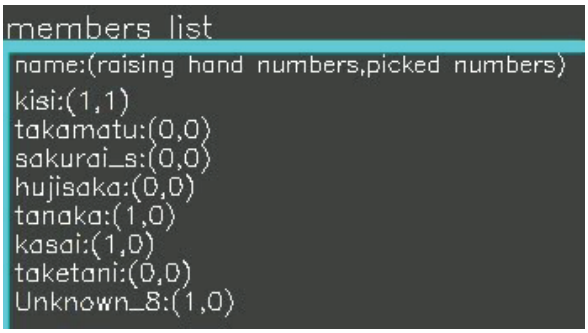

図 17: 挙手者の指名後のメンバーリスト

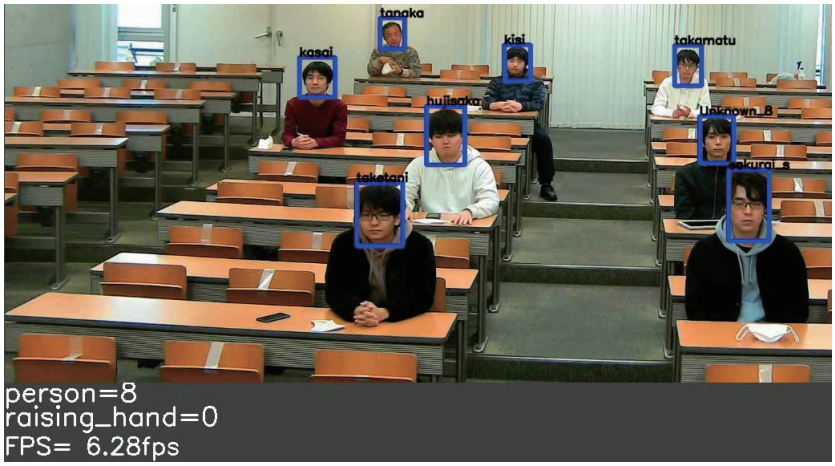

図 18: 本実験時のフレームレート

6 おわりに

本研究ではYOLOv3及びYOLOv4での挙手者の物体検出及び精度の向上を行い，主にアノテーションの方法と入力解像度，検出困難な画像の追加やYOLOv4を用いることにより，挙手クラスのAPが0.8471と高い精度での検出が可能となった．また顔認証システムと個人追跡を用いてYOLOにより検出した挙手と人の矩形から挙手者の同定を行った．さらに挙手者検出を応用し挙手者を自動的に指名するシステムの構築を行い，Jetson Xavier環境での運用を前提としたシステムの実験を行った．

今後の課題は，顔認証時のズームによる誤った追跡の改善や，カメラの広角撮影時に検出対象者全員がカメラの画角に入らないより多人数での環境におけるPTZカメラのパン機能などを使ったシステムの構築で，個人の特定や挙手者の同定を可能にすることである．

参考文献

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” *arXiv preprint arXiv:1506.02640*, 2016.

[2] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *arXiv preprint arXiv:1612.08242*, 2016.

[3] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[4] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[5] AlexeyAB, “Darknet,” `https://github.com/AlexeyAB/darknet`, 参照: 2021/4/18.

[6] Ultralytics, “YOLOv3,” `https://github.com/ultralytics/yolov3`, 参照: 2021/4/18.

[7] R. Padilla, S. L. Netto and E. A. B. da Silva, “A survey on performance metrics for object-detection algorithms,” in *Proc. 2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237-242, 2020, doi: 10.1109/IWSSIP48289.2020.9145130.

[8] Ageitgey, “Face_recognition,” `https://github.com/ageitgey/face_recognition`, 参照: 2021/4/18.

[9] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.