コンピュータ理工学特別研究報告書

題目

IoT カメラの研究開発

-YOLOv3 を使った男女と正面顔の認識に関する研究-

学生証番号　1544601

氏名　齋木 健介

提出日　平成 31 年 1 月 22 日

指導教員　蚊野 浩

京都産業大学
コンピュータ理工学部

要約

人通りの多い商店街や,広告やディスプレイがある待ち合わせ場所は経済的効果が期待でき，価値が高い空間である．しかし，宣伝・広告用の空間には明確な定価を決める手段がなく，空間の価値を比較することが難しいのが現状である.

本研究の目標は空間価値を定量化するための IoT カメラの開発である．これは，一般の通路などの映像から人間の数や，男女の比率，人が広告やディスプレイを見ているかどうかの判別をリアルタイムで実行できるカメラデバイスである.本研究では，判別のために画像認識のための深層ニューラルネットワークである YOLOv3 を使用する.YOLOv3 は他の最新の深層ニューラルネットワークと比較して，同等の認識性能で処理速度が速い．YOLOv3 の学習済みのモデルは，人に関して人体（全身）の検出しかできない.そこで，人体に加えて男性，女性，正面顔／非正面顔を判別させるために独自のデータを用意して学習させた.

学習には VOC2007 から人物画像 4010 枚と男女比のバランスを調整するために ImageNet から女性の画像 1368 枚を加え，合計 5378 枚の画像を学習に使用した.各種ファイルの作成，調整を行い,学習させた.学習の様子をグラフで確認し，60000 回の繰り返しで十分に収束した判断し，学習を終了させた．COCO dataset で学習された既存モデルとの比較と 300 枚の画像を認識にかけ，どの程度検出ができていたか検証を行った．検証の結果から学習を行うことで人を検出できるだけでなく広告やディスプレイがあるカメラ側を向いているか否かと男性，女性の区別をすることはできたが,男性と女性の誤検出が想定より多く，体の一部が隠れた場合やカメラから遠い場合，暗い空間での検出精度が低い状態であった．男性や女性などの多様性があり，条件や基準を定めなければ男性と女性の境界が明確にできない．理想的なデータだけでなく様々な状況を考慮した学習データの作成が課題である.

目次

1章 序論

人通りが多い商店街,注目を集める広告やディスプレイ，魅力のある商品が陳列されているウィンドウなどは大きな経済効果をもたらすため，価値が高い空間である．しかし,これらの空間を他の空間と比較して，明確に，どれくらい価値が高いと判断することは難しい．その理由は,常に移動している歩行者などをターゲットに，空間がもたらす効果を正確に測定することが困難であることによる.

商店街や通路にいる歩行者を数える方法として,調査員が人数をカウントするというアナログな調査やアンケートがある．しかし，通行人の数は毎日変動するので，アナログ調査によって特定の日時のデータを正確に得たとしても,それは一時的なものでしかなく,総合的な判断を下すには不十分である.実際，広告やディスプレイ，ショーウィンドウなどを設置する際,定量的な調査データに基づいて立地を検討することは稀である.有名な場所から見える場所であるとか,集合場所としてよく使われている場所であるなど，過去の経験や街の開発状況から推測する場合が多いと思われる.

私は，空間を通行する人の数や，設置される広告やディスプレイに注目する人の数などを数値化することが，その空間の価値を定量化する第一歩であると考えた．それらの数値に基づいて，目的に合わせた広告やディスプレイ，ウィンドウの設置が可能になるであろう.数値化するための手段として人によるアナログ的な調査は，一般の商業用途に用いるには,あまりにもコストが高い.そこで,現場をカメラによる動画で撮影し，その映像から人間に関する情報を自動的に検出し，記録することができれば良いと考えた.

蚊野研究室では,本研究に先行して,オムロン社の顔認識カメラ HVC-P2 と Raspberry Pi を使い，街角に設置して人数などをカウントする IoT カメラを試作した．これを用いた予備実験では，正面顔を正確に検出することはできるが，一般の通路に設置して人数をカウントするには適していないことがわかった．これに続いて，高速に画像認識可能な深層ニューラルネットとして評価の高い YOLO[1][2]を使い，一般の通路を歩行する人間の数を計測する予備実験を行った．その結果，空間価値の定量化には，人数の定量化だけではなく，男女の区別や顔の向きが重要な情報であることがわかった.

本研究はこれらの予備実験の結果を受けて，一般の通路の映像から人間の数，男女の区別,正面顔か否か,を判別する技術を開発した.そのベースとして YOLO を用い,YOLOv3 に男女，正面顔／非正面顔を学習させた．最終的な目標は，これらをリアルタイムで実行できる IoT カメラを開発し，空間価値を即座に定量化することである.

2 章 IoT カメラに関する予備実験

蚊野研究室では空間価値を定量化する IoT カメラの研究を行っている．これは図 2.1 のような形態で利用されるものであり，商店街や商店の店先，ショーウインドウの前などを観察するように設置し，観察している空間の価値を定量化することが目的である．

図 2.1 IoT カメラの想定利用場面

今回の研究に先立って，オムロン社の顔認識カメラ HVC-P2 を使って IoT カメラを試作したので，これについて簡単に報告する．図 2.2 に IoT カメラの外観を示す．このカメラは同社がヒューマンビジョンコンポという名称で製造･販売している部品の一つである HVC-P2 と，Raspberry Pi 2 Model B を組み合わせたものである．HVC-P2 は，同社の OKAO Vision 技術をカメラモジュール化したものである．HVC-P2 は顔画像の検出・識別，顔画像からの年齢・男女の推定などが可能である．人体全体の検出も可能としているが，基本的に顔画像の処理に特化した顔認識カメラである．

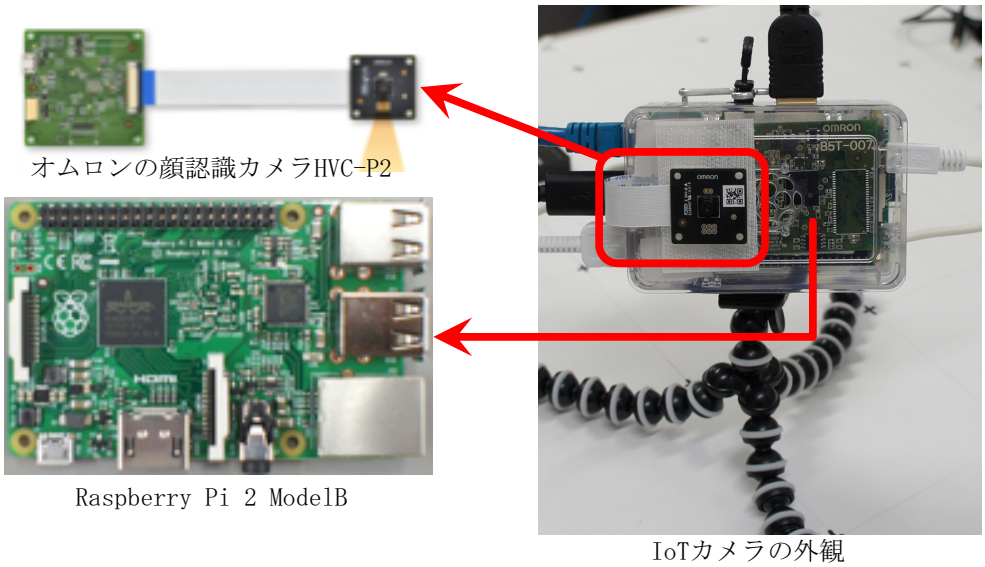

図 2.2 試作した IoT カメラ

試作した IoT カメラを図 2.3 のようにネットワーク化することで，人数・年齢・性別などの数値データをクラウドに蓄積してグラフ化したり，動画を Dropbox に保存したり，スマホで動画を確認することが可能である．

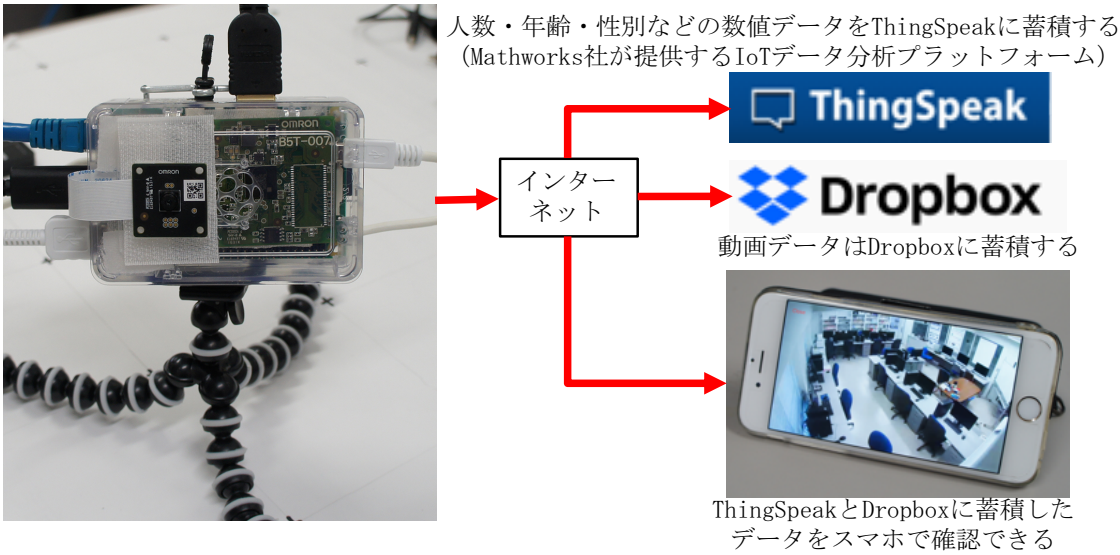

図 2.3 試作した IoT カメラの利用形態

試作した IoT カメラの動作や性能を確認したところ，顔画像の検出や年齢・性別の推定などは良い性能を示した．しかし，正面顔が観察できない条件での人体情報計測に関する性能は不十分であった．従って，一般の通路などで人数や性別など判定することは難しいと判断した．

3章　YOLO（You Only Look Once）

この章では物体認識技術と YOLO のアルゴリズム,本研究に必要な YOLO の基本知識について説明する.これにより YOLO についての理解を深める.

3.1　CNN を用いた物体検出技術

物体検出は，画像に含まれる人や車などの領域を長方形の枠（バウンディングボックス）で囲み，その物体の名称を示すことである.これを実現する技術の多くは，

① １枚の画像から，物体領域の候補を多数提案・抽出し，

② それぞれの候補領域について，その領域がどの物体なのか判断

することで物体検出を実現している.

もっとも単純な物体領域候補の提案方法として「sliding window(スライディングウィンドウ法)」がある．これは決まった大きさの領域を一定のピクセルずつずらしていくことで領域候補を提案していく手法である.

近年,候補領域から物体名を判断する処理を深層ニューラルネットの一つである畳み込みニューラルネットワーク（CNN）で実現することが一般的になっている．畳み込みニューラルネットワークを利用した物体検出として R-CNN(Regions CNN)[8]が挙げられる．R-CNN は最初に「Selective Search(選択的検索法)」で物体領域候補を提案する.領域候補の画像を，あらかじめ学習させた畳み込みニューラルネットワークに入力し，特徴を得る.この特徴を分類器に入力し，領域候補の物体クラスを予測する.スコアリングされた物体領域候補から,非最大値の抑制を行なって不要なバウンディングボックスを排除する.さらに,物体候補の CNN 特徴からバウンディングボックスのパラメータへ回帰を行い，バウディングボックスの正確な位置を推定する.

R-CNN はネットワークの順伝播計算を提案された物体領域ごとに行うため，検出速度が遅い.その問題点を解決するために，画像全体に対して計算した特徴マップをすべての提案された物体領域で再利用することで高速化させた Fast R-CNN[9]がある．また，それをさらに高速化するために，物体領域候補の提案にも CNN を用い，物体検出処理の全体を CNN 化することで高速化した Faster R-CNN[10]がある.

3.2 YOLO のアルゴリズム

多くの物体検出手法は複数のバウンディングボックスを提案し,それぞれのバウンディングボックスについて物体を分類させる.そのため処理が複雑で,処理時間も長くなる.YOLO は画像認識を回帰問題と捉えて画像の領域推定と物体分類を同時に行う.YOLO のアルゴリズムは単一の CNN で完結しているため，検出性能を直接的に最適化できる.

YOLO のアルゴリズムを説明する.YOLO は画像を S×S の領域に分割する．それぞれの領域から B 個のバウンディングボックス候補とその信頼度を計算する．一つのバウン

ディングボックスを定義するパラメータを表 2.1 に表す.ここで，バウンディングボックスの信頼度は,その領域に物体が含まれる確率と予測されたバウンディングボックスの正確さの指標の積によって計算される．また,それぞれの領域では上記の計算と同時に，その領域に何かの物体があるという条件のもとでの各クラスの条件つき確率「P(物体のクラス|物体が存在する)」を計算している．物体検出時には信頼度と条件つき確率の積を計算し，同一物体に対して複数のバウンディングボックスが出力されないように NMS(Non-Maximum Suppression)を行い，検出結果を出力する.

表 3.1：バウンディングボックスのパラメータ

x	バウンディングボックスの中心の x 座標
y	バウンディングボックスの中心の y 座標
w	バウンディングボックスの横幅
h	バウンディングボックスの高さ
信頼度	本文中で説明

YOLO のネットワーク構造を図 3.1 に示す．ネットワークは複数の畳み込み層と 2 つの全結合層からなり，畳み込み層にて特徴量抽出を行い，全結合層にて分類や物体領域の座標修正を行う.

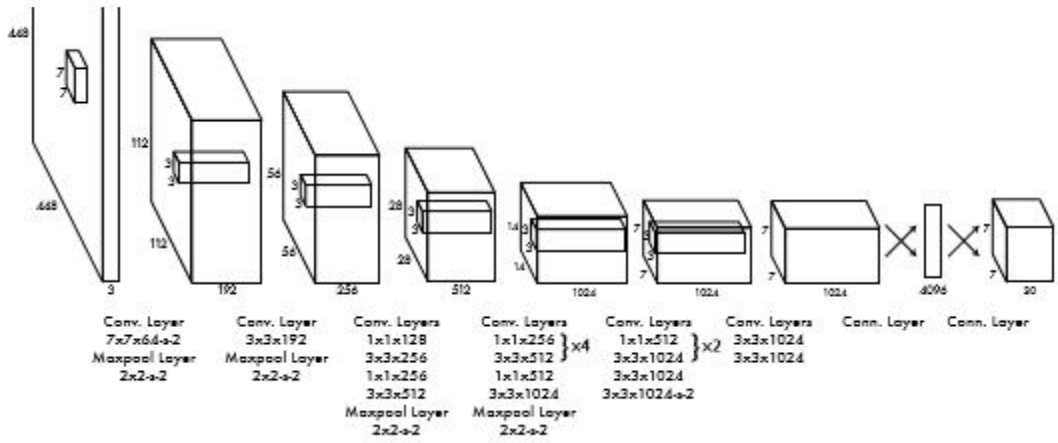

図 3.1 YOLO のネットワーク構造[1]

YOLO は一度に画像全体を見て予測することができる．従来のスライディングウィンドウ法や選択的検索法などでは,物体が存在しない背景領域において物体を誤検出する場合が見られた．YOLO では画像全体から検証,学習するため，誤検出が従来の技術よりも大幅に少なくなっている.

3.3 YOLO の実装

現在，YOLO は v3 となっており，YOLO のサイト[2]からダウンロードすることができる（但し，これは Linux か Mac OS で動作するバージョンである）．ここから学習済みのモデルをダウンロードできる．インストールも容易であり，サイトの通りに実行すると図 3.2 の結果が得られる．本研究では Windows 環境で開発を行うため，オリジナルの YOLO から派生した Windows バージョンの YOLOv3 を使用する．これは GitHub にリポジトリ[3]があるのでそこからダウンロードする．その後，Visual Studio にてビルドすることで使用可能になる．ダウンロードできる学習済み YOLOv3 は coco dataset[4]を用いて学習されたモデルである．coco を学習済みの YOLOv3 は 80 個のクラスを認識することができる．この中で人体に関するクラスは「person」という人体全体に対応したクラスだけである．

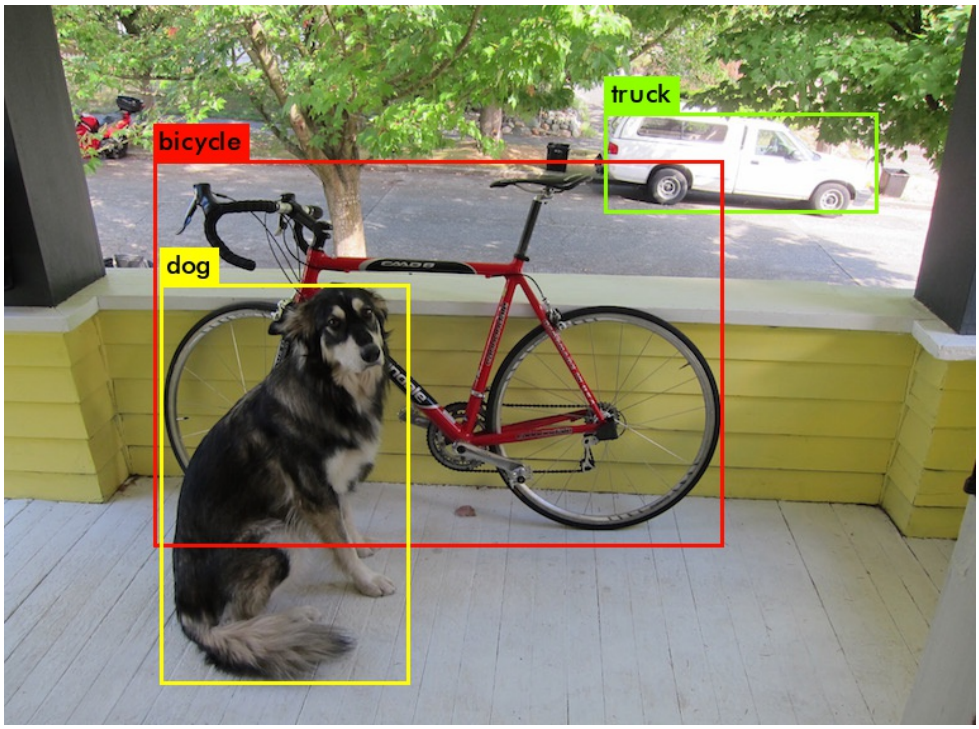

図 3.2 YOLO による検出例

YOLO は独自データを学習させることで，検出する物体を変えることが可能である．その場合，GPU を搭載した PC でなければ学習に要する時間が膨大になってしまう．PASCAL VOC2007[5]のデータを使って YOLOv3 を学習させる場合，およそ 60,000 回の学習を行うと，十分な検出性能になる．GPU GTX 1060(6GB)を使用し，10,000 回の学習を行うのにおよそ 29 時間を要した．GPU を使用せずに学習を行うと，100 回の学習に 25 時間を要した．これより，GPU を使用することで学習速度が 86 倍になっていることがわかる．

4 章 YOLO を用いた人体情報の検出による空間価値の定量化

4.1　提案手法の概要

本研究では広告やディスプレイなどの上部にカメラを設置し,撮影を行う.得られた動画から YOLO を用いて人の検出を行い,人数,ディスプレイやウィンドウを見る人の数,その空間に滞留する時間，などをデータとして記録し，その空間の価値を数値で示す.本研究では YOLO を学習させることで既存の学習済みモデルにない検出性能になるように調整する.新たに数値化する必要があると判断した 4 つのデータを表 4.1 にまとめる.

表 4.1 収集するデータ

1	各フレームで，カメラの前に存在する人数
2	各フレームで，ディスプレイやウィンドウをみる人数
3	各フレームでの男性，女性の割合
4	一人あたりの平均滞留時間

4.2　定量化のためのデータ

カメラの前に存在する人数は，単純にその場所の通行人の量の指標となるので，記録する必要がある．フレームごとに人体を検出することで数えることができる.

ディスプレイやウィンドウを見る人数は,広告やデジタルサイネージに注目していることを判断する指標になる．通行人の総数が少なくても，この人数が多ければ，広告の効果が高かったと言える．本研究では,ディスプレイやウィンドウ上部に設置したカメラ側に顔を向けていれば，それを注目していると判断した．したがって，フレームごとに正面顔を検出できれば数えることができる.

広告にはターゲットがあり,女性が多い空間であれば，その空間では化粧品などの広告やディスプレイに関する価値が高くなる.人体検出とともに男性，女性を区別できるようになれば数えることができる.

本研究で利用した動画のフレームレートは 15fps であり,仮に，1 秒ごとに人体検出を行うようにした.そして，その間のフレームをテンプレートマッチングで追跡することで同一人物の判定を行なっている.一人あたりの平均滞留時間とは，一秒ごとに写る人数の合計数を，同一人物による重複を除いた実際に通った人数で割ることで，一人あたりカメラの前に滞在した時間に換算した数値である．図 4.1 を例に挙げると状況 1 では時刻 1 に A が一人写るのみで平均滞留時間は 1 秒になる．状況 2 では 3 秒．状況 3 では 1 秒となる.平均滞留時間が長いとその空間は足を止める人が多いということがわか

り，ディスプレイやウィンドウを見る機会が長いと判断できる．

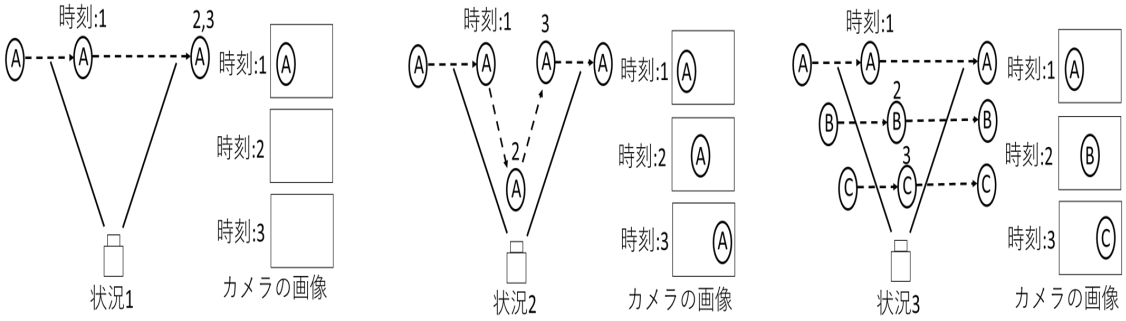

図 4.1 状況による滞留時間の差

既存の学習済み YOLO は，人体に関して「person」のみしか検出できない．上述した 4 つデータを集めるためには，YOLO を再学習させ，検出対象のクラスを変える必要がある．学習済みモデルにある「person(人)」に加え，ディスプレイやウィンドウを見た人数を数えるために「正面顔」「それ以外の方向を向いている場合の顔」を学習させる．また，男女比を記録するために「男性」と「女性」を区別して学習させる．

4.3 YOLO の学習方法

YOLO の学習に必要なものは

① 画像ファイル：シリアル番号がついた大量の画像データ．

② クラスリストファイル：クラス番号とクラス名を対応させたリスト．

③ 座標ファイル：画像ファイルと同じ番号の大量のファイルで，画像中の物体のクラス番号とバウンディングボックスの座標データを記述したもの．

④ 画像リストファイル：訓練画像をリストしたファイルと，テスト画像をリストしたファイル．

⑤ data ファイル：クラス数や訓練画像リストファイルなどへのパスを記述したファイル．

⑥ 設定（cfg）ファイル：ネットワークの構成などを記述したファイル．

である．

クラスリストファイルは，認識させたいクラスを列挙した一つのテキストファイルであり，拡張子は names である．本研究の場合，face(front)，face(other)，person，male，female の 5 つのクラスに，0,1,2,3,4 の数字を割り振る．

座標ファイルは，行ごとに「クラスを表す数字，x，y，w，h」の 5 つの数値を記述したテキストデータである．ここで x,y はバウンディングボックスの左上隅の座標，w は横幅，h は高さである．YOLO の学習に使用するこれらの座標データの値は 0 から 1

の範囲で表現する必要がある．画像ファイルと座標ファイルは1対1で対応しているので,画像の数だけ座標ファイルを用意する必要がある.実際の座標ファイルの例を表4.2に示す.

表 4.2 座標ファイルの例

クラス	x	y	W	h
1	0.245	0.579	0.074	0.058
2	0.549	0.716	0.502	0.548
3	0.239	0.639	0.086	0.194

座標ファイルを画像ごとに用意するために,画像中の認識させたいものをバウンディングボックスで囲い，座標値をテキスト出力するプログラムを python で作成した．このプログラムのユーザーインターフェースを図 4.2 に示す.このプログラムは画像ファイルを読み込み,「次へ」と「前へ」を押すことで前後の画像に移動する（画像にはシリアル番号が付与されている）．最初にクラス名をラジオボタンで選択し，画像中のそのクラスが写っている箇所を囲む．クリックを押した箇所から離した箇所までの矩形が表示される．オリジナル(VOC)を押すことで VOC データが持っていた座標データを表示させることができる.「保存」ボタンを押すことで，右下に表示されている座標データを画像と同じ名前のテキストファイルとして保存する.

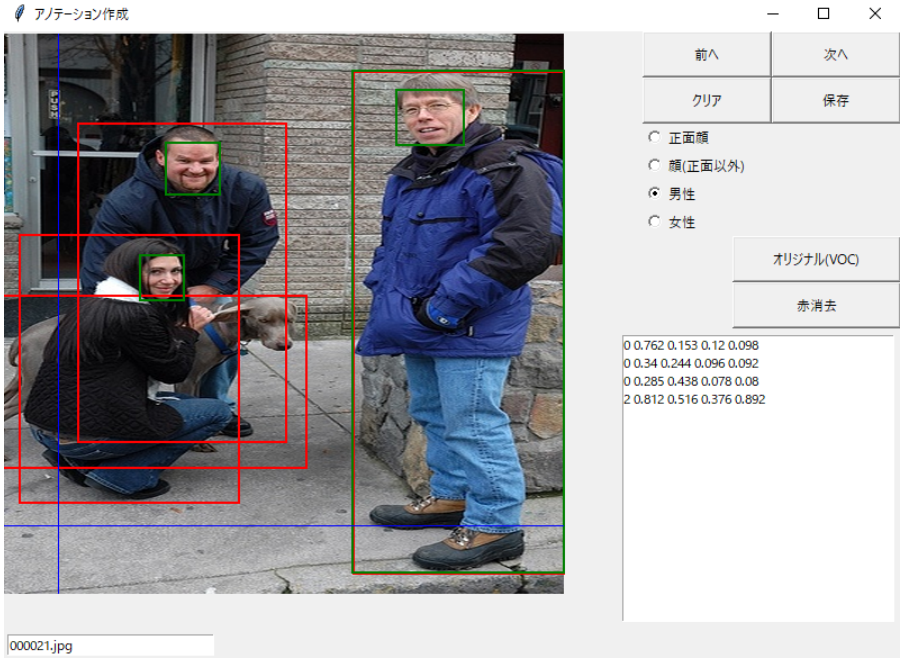

図 4.2 座標データ作成用プログラム

今回の学習では PASCAL VOC2007[5]と ImageNet[6]からの画像ファイルを使用した. 予備的な実験では VOC2007 のデータのみを使用し，人が写っている画像を 4010 枚抜き出した．VOC2007 の人物が写っている画像は，男性の比率が多いので，バランスをとるために ImageNet から女性の画像を 1368 枚加え,本研究では,合計で 5378 枚の画像を学習に使用する．画像の男女比を表 4.3 に示す.

表 4.3 画像中の男女比(複数人，両方写っている場合も含む)

	予備実験	改良版
男性	3072 枚	3072 枚
女性	1878 枚	3246 枚
合計	4010 枚	5378 枚

次いで，画像リストファイルを作成する.これは学習させる画像ファイル名のリストである.このリストはテストデータ用と訓練データ用の 2 つに分ける必要がある．「How to train YOLOv2 to detect custom objects」[5]を参考にし，「process.py」というプログラムを作成し,これを使用してテストデータ用の「test.txt」と訓練データ用の「train.txt」を生成した.

これらのデータが用意できたら,学習パラメータを設定するための data ファイルと設定(cfg)ファイルを，今回の学習用に調整する（この２つは，YOLO をインストールした際に一緒にインストールされているので，それらを修正する）.data ファイルはクラス数やデータのパスなどを指定するファイルであり，内容は図 4.3 となる．「classes」は学習させたいクラス数である．本研究では「人」，「正面顔」，「顔(それ以外)」，「男性」，「女性」の 5 つにあたる．「train」は train.txt へのパスであり．「valid」は test.txt へのパスである．「names」は names ファイルへのパスであり，「backup」は学習によって発生する weights ファイルを保存する場所へのパスである.設定(cfg)ファイルはニューラルネットワークの構成のファイルであり，学習条件やネットワーク構造について書かれている．学習させる際に変更する必要があるのは「classes」，「filters」である．「classes」は検出させたいクラス数，「filters」は(クラス数+5)*3 の値に変更する.

```
classes= 5
train  = C:/Users/g1544601/Desktop/darknet-master/darknet-
master/build/darknet/x64/data/train.txt
valid  = C:/Users/g1544601/Desktop/darknet-master/darknet-
master/build/darknet/x64/data/test.txt
names = data/images/personlist2.names
backup = C:/Users/g1544601/Desktop/darknet-master/darknet-
master/build/darknet/x64/backup2/
```

図 4.3 学習に使用する data ファイル

上記のファイルを揃えて学習を開始させる.この時,「darknet53.conv.74」というウエイトファイルをウエイトの初期値とし，以下のコマンドを実行する.

「darknet detector train

cfg/voc-person2.data cfg/yolov3-voc-3rd.cfg darknet53.conv.74」

学習を開始すると学習回数が進むにつれ誤差が 0 に近づいていく．その様子がグラフで表示され，状態を確認できる.学習回数を重ねてこれ以上,誤差が小さくならないと判断すれば学習を終了させる.

5 章　研究結果と考察

5.1 学習結果と検証

60,000 回でグラフの様子を確認し，十分に収束していると判断したので学習を終了させた．グラフを図 5.1 に示す．

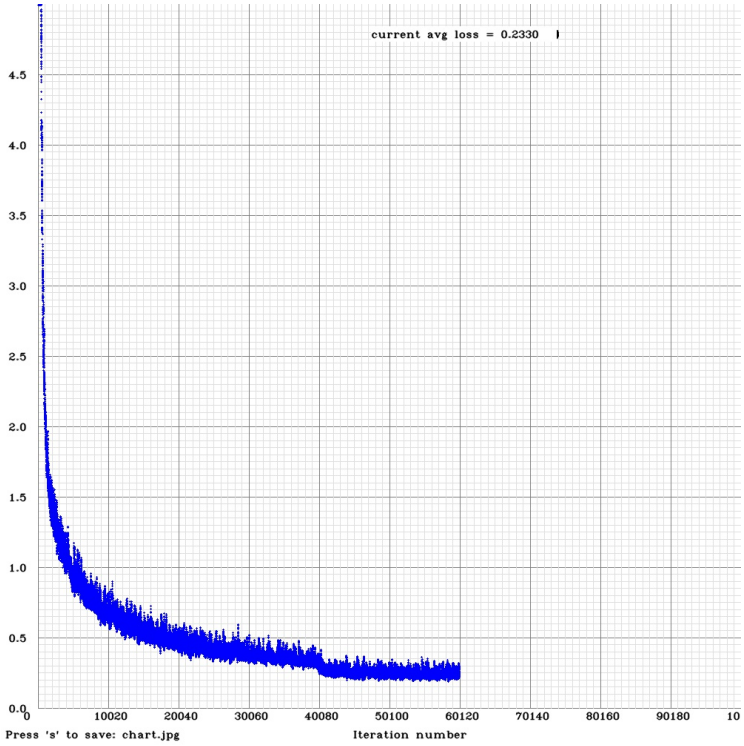

図 5.1 試行回数による学習状態を表すグラフ

物体検出の評価に，クラスごとの平均適合率(AP，Average Precision)が利用されている．クラスごとの AP を全クラスで平均したものが mAP(mean Average Precision)である．この mAP が深層ニューラルネットの検出性能の定量的指標としてよく用いられている．YOLO には mAP を出力するコマンドがあり，実行結果を AP についてまとめたグラフを図 5.2 に示し， mAP について既存の COCO dataset で学習済みのモデルと比較したグラフを図 5.3 に示す．図 5.2 からクラスごとの AP について確認すると正面顔の精度は高く，男性の精度は低い．この結果から男女の区別があまりできていないと言える．図 5.3 から mAP は 54.97%になっている．COCO dataset で学習済みの YOLOv3 の mAP は 57.9%である．mAP だけでは定量化するための人に関するデータを集めることに適しているか判断できないので検証を行う．

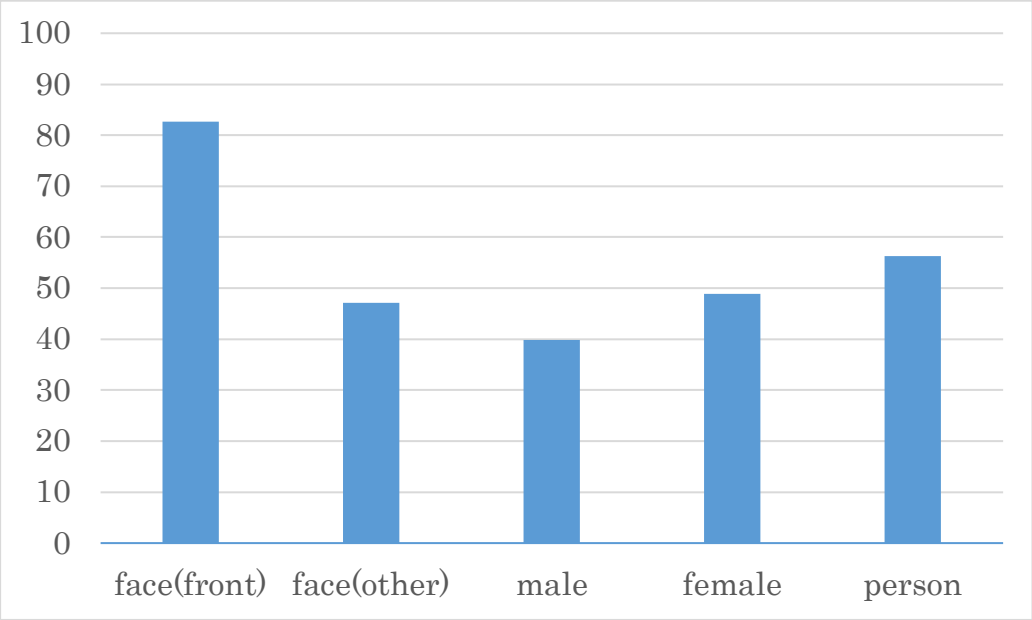

図 5.2 学習データの各クラスの AP

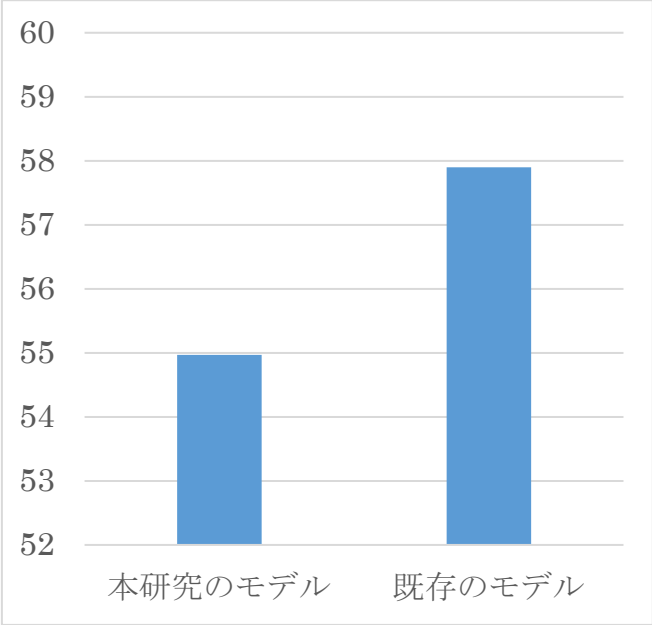

図 5.3 mAP の比較

次に画像を 300 枚使用して検証を行う．まず，VOC2007 と ImageNet から，体全身が写っている画像を 100 枚用意した．これは，男性と女性を同じぐらい検証できるように男性，女性ともに写っている画像を 20 枚，男性の画像 40 枚，女性の画像 40 枚という

内訳にした．次に学習に使用した 5378 枚のうちから 100 枚を用意した．そして，ImageNet から道路や道を歩いている通行人の画像を 100 枚用意した．この通行人の画像では，人が比較的小さく写っている．画像一覧を表 5.1 に表す．

表 5.1 検証に使用する画像

体全身が写っている画像(男性のみ 40 枚, 女性のみ 40 枚, 両方 20 枚)	100 枚
学習に使用した訓練画像	100 枚
通行人の画像（人が比較的小さく写っている）	100 枚

検出できている／できていないは，①正面顔が検出できている／できていない，②正面以外の顔が検出できている／できていない，③男性が検出できている／できていない，④女性が検出できている／できていない，⑤人が検出できている／できていない，の５つの場合について検証した．

正面顔が検出できていると判断するのは，正面顔とラベル付けした領域を正面顔と出力する場合のみであり，それ以外のクラスも同様にラベル付けされた領域を正しく出力する場合のみとする．人に関するクラスを増やした今回の学習データが有効であるか判断するために人に関してのクラスが「person」のみの学習済みモデルも検証する．検証結果を表5.2に示す．表5.2を見ると体全身画像と訓練画像に関しては良い精度であるが通行人に関しては精度が著しく低い．

表 5.2 画像 300 枚の検出精度比較

	face(front)	face(other)	male	female	person	person(coco)
全身画像	96.3	85.2	82.8	83.5	88.5	99.3
訓練画像	90.7	76.9	70.0	82.7	73.6	98.1
通行人画像	54.8	41.8	30.2	20.2	29.4	97.4
クラス平均	80.6	68.0	61.0	62.1	63.8	98.2

また，学習させたモデルによる検出例を図 5.4 に示し，既存の学習済みモデルによ

る検出例を図 5.5 に示す.

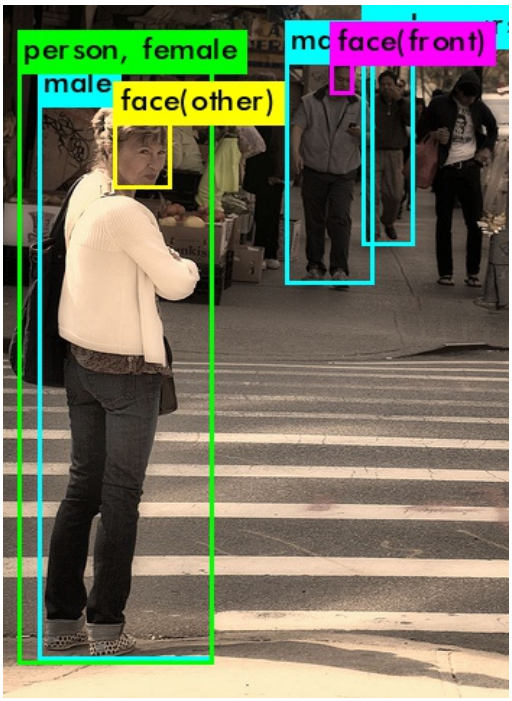

図 5.4 独自に学習させたモデルによる検出例

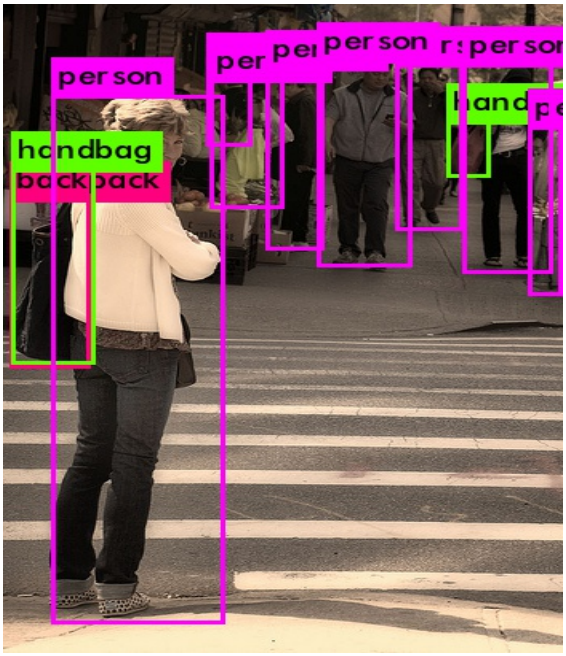

図 5.5 学習済みモデルによる検出例

図 5.4 と図 5.5 を比べてみると検出されている「人(person)」の数が違う．独自の学

習データでは「person」に加え，「male」，「female」として囲われることが理想であるが，遠い場所にいる人や一部が隠れている人の検出精度が低いことがわかる.この傾向は例として示したこの画像に限らず全体的に見て取れる.人の検出精度に関しては学習済みモデルに比べると低下しているが，男性，女性がこの空間に存在し,どこを向いているのかは判別できるという利点が増えている.

5.2 考察

独自にデータを集め，YOLOv3 の学習を行った結果，男性，女性，どこを向いているのかを検出することはできたが，学習済みモデルより人の検出に関してより良いモデルではない．今後の課題として

(1) 男性と女性を多重に検出している場合や，誤検出する場合が多い．人が判断できたとしても学習させるときに条件や基準を定めて，それに従い，学習データを作るべきだったと考えられる.

(2) 夜などの暗い空間での検出や体の一部が隠れる人，遠くに写る人の検出の精度が低い．そのため，人混みなどではあまり機能しない．これは使用した学習データには人が大きくはっきりと写っている理想的なデータが多く含まれ，似たような状況の画像が少なかったことが原因である.

6 章　結論

YOLOv3 を独自データで学習させるには，学習に必要な画像を収集し，それぞれの画像に対応する座標データを手作業で作成する必要がある．しかし,これらの全てをフリーで入手することができ，大きな修正を加えることなく，学習を行うことができるので非常に扱いやすいと言える．座標データ作成などは単純作業ではあるので,人手による負荷の分散も可能である.YOLOは精度も良く，処理速度が速い．欠点をあげるならばGPU がなければ学習を行うことは困難であることである.

課題として以下のことが残っている.

(1)様々な状況での人の追加や男女の基準設定での学習データの改良

(2)実際に通りを人が歩いている映像を用いて学習させたデータの検証

(3)得られたデータから空間価値の比較

参考文献

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” IEEE CVPR 2016, pp.779-788, 2016.

[2] YOLO: Real-Time Object Detection, https://pjreddie.com/darknet/yolo/

[3] AlexeyAB/darknet, https://github.com/AlexeyAB/darknet

[4] COCO Common Objects in Context, http://cocodataset.org/#overview

[5] PASCAL VOC2007 , http://host.robots.ox.ac.uk/pascal/VOC/voc2007

[6] ImageNet, http://www.image-net.org/index

[7] How to train YOLOv2 to detect custom objects, https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/

[8] R. Girshick, et al., “Rich feature hierarchies for accurate object detection and semantic segmentation, ” IEEE CVPR 2014, pp.580-587, 2014.

[9] R. Girshick, “Fast R-CNN, ” IEEE ICCV 2015, pp.1440-1448, 2015.

[10] S. Ren, et al, “Faster R-CNN: towards real-time object detection with region proposal networks,” NIPS’15, pp.91-99, 2015.

謝辞

本論文を作成にあたり,丁寧な御指導を賜りました蚊野浩教授に感謝いたします.

付録　開発したプログラムの説明

[20180320Opencv340YoloTracking(Source.cpp)]

Visual Studio2015 にて C++によって作成した. OpenCV から YOLO を用いている. 動画ファイルを複数読み込み, 15fps ごとに画像として保存し, 1 フレームだけ YOLO による検出を行う. このとき検出した人数をカウントしていく. 残りの 14 フレームを用いてテンプレートマッチングを行う. 次の YOLO の検出時にテンプレートマッチングによって同一人物だとされた人物はカウントされないようになる.

[imgshow.py]

PyCharm という環境を利用して python で座標データを作成するために作られたプログラムである. 画像ファイルを読み込み, GUI 上にて表示する. クラス名をラジオボタンで指定した後, クリックした位置から離した位置までをバウンディングボックスで囲む. バウンディングボックスの座標値を 0~1 に変換し, 画像データと同名のテキストファイルに出力される.