カスケード接続した機能の異なる CNN と 時系列フィルタによる特定物体の認識と追跡

Specific Object Recognition and Tracking
using Cascade Connection of Different types of CNNs and Time-series filter

○学　西尾 貴樹（九工大）　　学　田中 良道（九工大）
学　沈 歩偉（九工大）　　学　バルビエ　ティボ（九工大）
正　西田 健（九工大）

Takaki NISHIO, Kyushu Institute of Technology
Ryodo TANAKA, Kyushu Institute of Technology
Buwei SHEN, Kyushu Institute of Technology
Thibault BARBIE, Kyushu Institute of Technology
Takeshi NISHIDA, Kyushu Institute of Technology

Currently, convolutional neural networks (CNN) are widely used for object recognition, and it is common to use already trained network as a basis. However, the retraining cost of CNN to recognize a specific object is expensive in terms of dataset preparation and calculation cost. We propose instead to add a second CNN specialized on the recognition of the desired target. Additionally, a time-series filter is used to stabilize the detection of the target. To confirm the effectiveness of the proposed method, we used outdoor recorded movies and evaluated the accuracy of specific person recognition.

Key Words: Specific object recognition，Tracking, CNN，Time-siries filter

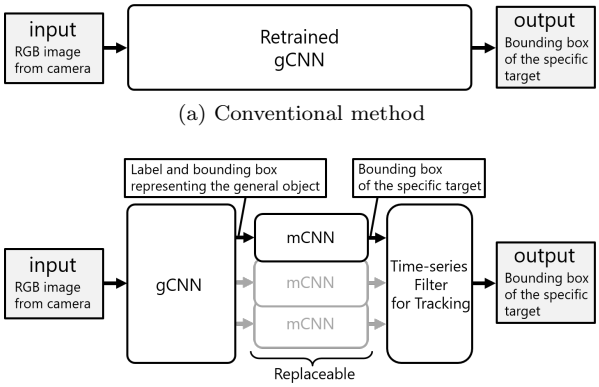
1　はじめに

近年，深層学習の一種である CNN (Convolutional Neural Networks) による一般物体認識に関する研究が盛んに行われている．一般物体認識では，画像から人や車，犬などの物体のラベル（種類）と位置（矩形等の領域）が認識される．一般物体を認識する一般物体認識 CNN は大規模なネットワーク構造を持ち，膨大なデータセットと並列演算能力の高い計算機を用いて学習される．近年発表された一般物体認識 CNN には Fast R-CNN[1] や SSD[2]，YOLOv2[3] 等がある．これらの CNN は，一般物体認識において高い性能を示しているが，特定の物体（特定の人物や車等）を認識したい場合には，画像中の対象物の位置情報を含むデータセットを作成し，大規模な CNN の学習をやり直さなければならない．そこで，本研究では，学習済みの一般物体認識 CNN に特定物体識別のための CNN をカスケード接続することで，学習をやり直すことなく特定の物体の認識を可能にする手法を提案する．また，CNN の認識結果に時系列フィルタを適用することにより，対象物を追跡する手法をあわせて提案する．さらに，提案手法の有効性を検証するため，特定の服装の人物の認識および追跡を行う実験を行った結果を示す．

2　関連研究

2.1　CNN による物体認識

物体認識のための CNN には様々なモデルが公開されている．Faster R-CNN[1] は入力画像から特徴マップを生成し，その特徴マップから推定した物体候補領域と特徴マップを合成して物体領域のラベルと矩形を推定する．SSD (Single Shot MultiBox Detector)[2] は複数のスケールのフィルタを有する CNN から特徴マップを生成し，スケールごとに物体候補領域のラベルと矩形を推定する．YOLOv2 (You Only Look Once ver.2)[3] は入力画像を $N \times N$ グリッドに分割して各グリッドごとに物体の存在確率を推定し，閾値以上の確率をもつ物体候補領域を出力する．Mask R-CNN[4] は Faster R-CNN にセグメンテーションマスクを推定するネットワークを追加し，画像中の物体領域を画素単位で識別する．これらの CNN は，Pascal VOC (Visual Object Classes)[6] や COCO (Common Objects in Context)[6] 等の大規模データセットで学習される．公開されている学習済みモデルを用いれば，80 種類以上の一般物体の認識が可能であるが，特定物体を認識する場合には対象物の座標情報を含むデータセットを用意して CNN の学習をやり直す必要がある．



(a) Conventional method

(b) Proposed method

Fig.1 Conventional method and proposed method.

2.2　時系列フィルタによる物体追跡

物体追跡を行う時系列フィルタには，様々なアルゴリズムが提案されている．Median Flow[7] は追跡対象の矩形領域をグリッドに分割して各グリッド内の点をオプティカルフローにより追跡することで点の動きを求め，追跡誤差が大きいグリッドを除いて残った点の座標の中央値により矩形領域の座標を更新する．TLD (Tracking, Learning and Detection)[8] は追跡している物体領域の内部および外部からそれぞれ正例画像，負例画像を抽出し，識別器を学習することで追跡を行う．KCF (Kernelized Correlation Filters)[9] は物体検出を入力画像全体から物体位置を抽出する回帰問題とし，物体領域の画像をシフトすることで学習画像を仮想的に増やして学習する．また，近年では CNN を利用した追跡アルゴリズム GOTURN (Generic Object Tracking Using Regression Net-works)[10] も提案されている．

3　提案手法

本研究で構築した特定物体認識のためのアルゴリズムを Fig.1 (b) および Algorithm 1 に示す．以降では，一般物体認識 CNN を gCNN (general-CNN)，特定物体識別のための CNN を mCNN (module-CNN) と呼ぶ．mCNN は gCNN が認識し

Algorithm 1: Object Recognition and Tracking

Label of the specific target object: l_{target}
Load two trained CNN models :
  gCNN($\boldsymbol{Image}$), mCNN($\boldsymbol{CroppedImage}$)
while true **do**
  $\boldsymbol{I}_{\mathrm{f}} \longleftarrow$ Get an RGB color image from camera
  l_{g}, $\boldsymbol{b}_{\mathrm{g}} \longleftarrow$ gCNN($\boldsymbol{I}_{\mathrm{f}}$)
  if $l_{\mathrm{g}} = l_{\mathrm{t}}$ **then**
    $\boldsymbol{I}_{\mathrm{g}} \longleftarrow$ Crop $\boldsymbol{I}_{\mathrm{f}}$ using $\boldsymbol{b}_{\mathrm{g}}$
    $l_{\mathrm{m}} \longleftarrow$ mCNN($\boldsymbol{I}_{\mathrm{g}}$)
    if $l_{\mathrm{m}} =$ true **then**
      Run time-series filter to track the target :
       TimeSeriesFilter($\boldsymbol{I}_{\mathrm{f}}$, $\boldsymbol{b}_{\mathrm{g}}$)
  Output the result of recognition and tracking

た一般物体が特定の対象物かどうかを識別する CNN で，対象物に合わせて学習する．

ここでは，ある一般物体のラベル l_{t} に属する特定物体の認識および追跡を考える．提案手法では，カメラから得られるフレーム画像 $\boldsymbol{I}_{\mathrm{f}}$ を gCNN に入力する．gCNN の出力は一般物体のラベル l_{g} と矩形領域を表すベクトル $\boldsymbol{b}_{\mathrm{g}}$（矩形の幅，高さ，中心点の座標）である．gCNN の出力ラベル l_{g} と対象物のラベル l_{t} が一致したとき，$\boldsymbol{I}_{\mathrm{f}}$ から $\boldsymbol{b}_{\mathrm{g}}$ を用いて物体の画像 $\boldsymbol{I}_{\mathrm{g}}$ を切り出す．つぎに，$\boldsymbol{I}_{\mathrm{g}}$ の画像サイズを所定の値に変換して mCNN に入力する．mCNN の出力は対象物か否かを示す二値ラベル l_{m}（true または false）である．mCNN が対象物を識別した場合（$l_{\mathrm{m}} =$ true のとき），時系列フィルタを適用してその追跡を行う．

提案手法で学習を要する CNN は，二値分類を行う mCNN のみである．そのため，用意するデータセットには，矩形等の物体領域データを付加する必要がなく，「特定の対象物」および「その他」の物体の画像と，認識対象か否かの二値ラベルのみである．また，mCNN は二値のクラス分類を行う CNN であるため，一般物体認識 CNN より単純な構造で実現できる．本手法を用いれば，認識したい対象物を変更する場合に mCNN を交換することで対応が可能であり， Fig.1 (a) のように gCNN の学習をやり直す必要がない．

4 gCNN による一般物体認識

本研究では，gCNN に YOLOv2 を導入した．公開されている YOLOv2 の学習済みモデル [11] では，人や車，犬など，80 種類の物体を複数同時に認識できる．

5 mCNN による特定物体識別

5.1 ネットワーク構造

mCNN の構造を Fig.2 に示す．mCNN はクラス分類に用いられる一般的な CNN の基本構造を利用したものであり，畳み込み層，プーリング層，全結合層から構成される．YOLOv2 の認識結果によって切り出された画像は，入力時に 50×50 pixel の画像サイズに圧縮されている点に注意されたい．

5.2 データセット

gCNN で人物の認識を行い，mCNN で特定の服装の人物を識別することを考える．mCNN を学習するデータセットの作成のために，画像の合成による方法と，動画からの人物領域の切り出しによる方法を利用した．さらに，画像の回転や反転，コントラスト調整などのデータの水増しを行い，最終的に特定の対象の人物の画像を 220832 枚，その他の人物の画像を 206672 枚，合計 427504 枚の画像を用意した．なお，動画からの人物領域の切り出しには YOLOv2 を用いた．データセットの一部を Fig.3 に示す．

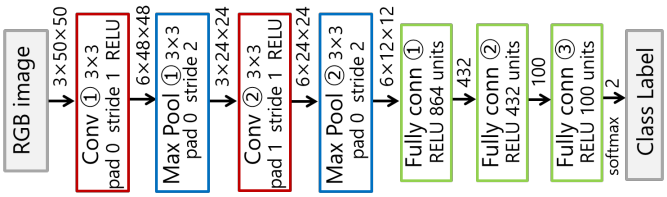


Fig.2 The structure of mCNN.

(a) Target person (b) Others

Fig.3 Examples of the dataset.

Table 1 Parameters for training mCNN.

Item	Value
Loss function	Softmax cross entropy
Optimizer	Adam ($\alpha = 0.0001$)
Training dataset	350000
Validation dataset	30000
Minibatch size	50
Iteration	10

5.3 学習

Table 1 に mCNN の学習時に設定したパラメータを示す．未学習画像 1000 枚に対する学習後の認識率は 99.4 % であった．

6 時系列フィルタによる物体追跡

本研究では，時系列フィルタに KCF を用いて対象物の追跡を行う．CNN が対象物を認識したときの矩形領域および画像フレームを用いて KCF を初期化し，追跡を行う．なお，対象物の追跡中は CNN での認識は行わず，KCF での追跡が失敗した際に CNN での認識に切り替える．

7 実験

7.1 実験環境

提案手法の精度を検証するため，屋外で撮影した動画に対して検証を行う．本研究で使用した実験環境を Table 2 に示す．

7.2 実験方法と評価指標

実験ではカメラを取り付けた台車で対象人物の周囲を走行し，動画を撮影した．つぎに，動画中の各画像フレームにおける対象人物が有無を記録して正解データとした．その後，提案手法を動画に適用して認識結果を記録し，正解データと認識結果を比較することで提案手法の精度の検証を行った．

7.3 実験結果

実験の様子を Fig.4 に示す．Fig.4 (a) は，gCNN が認識した人物に対して，mCNN が対象人物かどうか識別した結果を示している．Fig.4 (b) は，時刻 t [sec] における KCF による対象人物の追跡の様子を示している．GPU を用いるとほぼリアルタイムでの認識および追跡が可能であった．また，5 つの動画に対す

Table 2 Environment of the experiment.

Item	Version
Operating system	Ubuntu 14.04 LTS
CPU	Intel core i7 6700
System memory	8 GB
GPU	NVIDIA GTX 1070
GPU driver	CUDA 8.0 V8.0.44
Programming language	Python 3.5.5
Neural network framework	Chainer 1.18.0
Image processing library	OpenCV 3.4.0
Camera	Logicool c930e

Table 3 Recognition accuracy for 4 movies.

Movie No.	Accuracy [%]
1	94.54
2	91.07
3	97.81
4	92.47
5	96.82

る認識率を Table 3 に示す．全ての動画に対する認識率が 90 % を越えているため，本手法の有効性が確認できる．

8 まとめ

本研究では，屋外で特定物体を認識するために，機能の異なる CNN をカスケード接続し，時系列フィルタにより対象物の追跡が可能なアルゴリズムを提案した．特定人物の認識および追跡を行う実験により，提案手法の有用性を検証した．今後は提案手法を移動ロボットに搭載し，特定物を探索するシステムを構築する．

参考文献

[1] Ren, Shaoqing, et al. “Faster R-CNN: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems, 2015.

[2] Liu, Wei, et al. “SSD: Single Shot MultiBox Detector.” arXiv preprint arXiv:1512.02325, 2015.

[3] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, arXiv preprint arXiv:1612.08242, 2016.

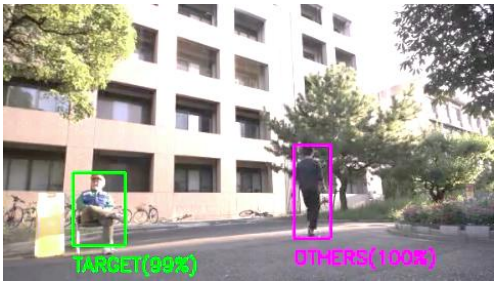
[4] Kaiming He, Georgia Gkioxari, Piotr Dollr and Ross Girshick. “Mask R-CNN”, arXiv preprint arXiv:arXiv:1703.06870, 2018.

[5] Mottaghi, Roozbeh, et al. “The role of context for object detection and semantic segmentation in the wild.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

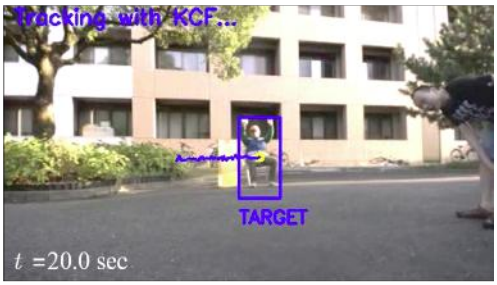
[6] Lin, Tsung-Yi, et al. “Microsoft coco: Common objects in context.” European conference on computer vision. Springer, Cham, 2014.

[7] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. “Forward-backward error: Automatic detection of tracking failures.” Pattern recognition (ICPR), 2010 20th international conference on. IEEE, 2010.

[8] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. “Tracking-learning-detection.” IEEE transactions on pattern analysis and machine intelligence 34.7: 1409-1422, 2012.



(a) Recognition result of CNNs



(b) Tracking result of KCF

Fig.4 Results of experiment.

[9] Li Y. and Zhu J. “A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration” In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops: Lecture Notes in Computer Science, vol 8926. Springer, Cham, 2014.

[10] Held, David, Sebastian Thrun, and Silvio Savarese. “Learning to track at 100 fps with deep regression networks.” European Conference on Computer Vision. Springer, Cham, 2016.

[11] Joseph Redmon. YOLO: Real-Time Object Detection. `https://pjreddie.com/darknet/yolo/` (Mar. 2018)