Specific Object Recognition and Tracking by Cascade Connection of Different Types of CNNs and a Time-series Filter

Takaki Nishio* and Takeshi Nishida†
Department of Mechanical and Control Engineering
Kyushu Institute of Technology, 1-1 Sensui, Kitakyushu, Fukuoka, 804-8550, Japan
*Email: nishio.takaki544@mail.kyutech.jp
†Email: nishida@cntl.kyutech.ac.jp

***Abstract*—Currently, convolutional neural networks (CNN) are widely used for object recognition, and it is common to use an already trained network as a basis. However, the retraining cost of CNN to recognize a specific object is high in terms of dataset preparation and calculation cost. Thus, we propose instead to add a second CNN specialized in the recognition of the desired target. Further, a dataset pre-filtering method is proposed to train the second CNN for specific object shape recognition. Additionally, a time-series filter is used to stabilize the detection of the target. To confirm the effectiveness of the proposed method, we use recorded videos and evaluate the accuracy of specific carrot shape recognition.**

***Index Terms*—CNN, multiple CNN structure, specific object recognition, time-series filter**

I. Introduction

In recent years, object recognition projects by convolutional neural networks (CNN) have been widely studied. Among them, CNNs for general object (e.g., person, car, and dog) recognition has drawn particular attention. We call these CNNs General-CNNs (G-CNNs). Examples of representative G-CNNs are Fast R-CNN [1], SSD [2], and YOLO [3]. In these methods, the position (rectangle region) and label of multiple general objects are detected simultaneously. G-CNN is composed of a large scale structure network and trained by a high-performance computer system with a large-scale annotated dataset. Recent G-CNNs exhibit significantly high performance and it is common to use already trained networks as a basis to recognize the specific object. However, the retraining cost of CNN to recognize specific object is high in terms of dataset preparation and calculation cost. We solve this problem by adding a second CNN called Module-CNN(M-CNN). M-CNN is connected in series to G-CNN and specialized in the recognition of the desired target. Further, we propose a dataset pre-filtering method for the specific object shape recognition. Additionally, a time-series filter is used to stabilize the detection of CNNs and track the target.

II. Related Works

A. Object Recognition using CNN

Various G-CNN methods have been proposed. Faster R-CNN [1] estimates object regions and labels by composing an estimated object candidate region and feature map from an input image. A single shot multibox detector (SSD) [2] uses multiple scale filter CNNs. SSD estimates the object regions and labels by generating a feature map for each scale. You only look once version 3 (YOLOv3) [3] splits the input image into $N \times N$ grids and estimates the object existence probability for each grid. The network outputs object regions and labels that have a greater probability value than a threshold value. Mask R-CNN [4] adds a network to Faster R-CNN for segmentation mask estimation and recognizes the object region by pixel units. These G-CNNs are trained with a large scale annotated dataset such as Pascal visual object classes (VOC) [5] and common objects in context (COCO) [6]. Although it is possible to recognize more than 80 types of general objects by using already trained G-CNNs, retraining with an annotated dataset is required for specific object recognition.

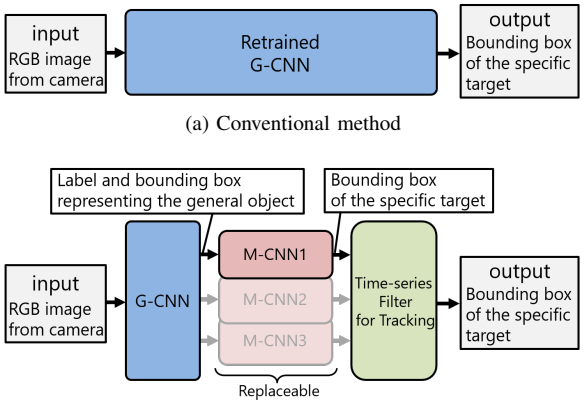
B. Object Tracking using Time-series Filter

There are object tracking methods for the target rectangle region. Median flow [7] splits the target rectangle region into grids and tracks the pixels inside each grid using optical flow. The target region is updated by the median value of the coordinates of the remaining points, with the exception of grids with large tracking errors. Tracking, learning, and detection (TLD) [8] extracts positive and negative images from inside and outside of the target region, and executes tracking using a classifier training. Kernelized correlation filters (KCF) [9] handles the object detection problem as a regression problem. KCF extracts the target position and learns the target feature with shifted regions. Generic object tracking using regression net-works (GOTURN) [10] is proposed as a CNN-based tracking algorithm.

III. Proposed Method

The frameworks of the conventional and proposed methods for specific object recognition are shown in Fig.1. The algorithm of the proposed method is shown in Algorithm 1. M-CNN is trained for a specific object. After the detection of the general object by G-CNN, M-CNN classifies it into a specific target. We consider recognizing a specific object

978-1-5386-5771-3/18/$31.00



(a) Conventional method

(b) Proposed method

Fig. 1. Conventional and proposed methods.

Algorithm 1: Object Recognition and Tracking

General target label includes specific target: l_{t}
Load two trained CNN models :
  $\mathrm{gCNN}(\boldsymbol{Image})$, $\mathrm{mCNN}(\boldsymbol{CroppedImage})$
while true **do**
  $\boldsymbol{I}_{\mathrm{f}} \longleftarrow$ Get an RGB color image from camera
  l_{g}, $\boldsymbol{b}_{\mathrm{g}} \longleftarrow \mathrm{gCNN}(\boldsymbol{I}_{\mathrm{f}})$
  if $l_{\mathrm{g}} = l_{\mathrm{t}}$ **then**
    $\boldsymbol{I}_{\mathrm{g}} \longleftarrow$ Crop $\boldsymbol{I}_{\mathrm{f}}$ using $\boldsymbol{b}_{\mathrm{g}}$
    $l_{\mathrm{m}} \longleftarrow \mathrm{mCNN}(\boldsymbol{I}_{\mathrm{g}})$
    if l_{m} does not change in multiple images **then**
      Run time-series filter to track the target :
        $\mathrm{TimeSeriesFilter}(\boldsymbol{I}_{\mathrm{f}}, \boldsymbol{b}_{\mathrm{g}})$
  Output the result of recognition and tracking

belonging to the label l_{t} of general objects. In the proposed method, an image frame $\boldsymbol{I}_{\mathrm{f}}$ captured by the camera is inputted to G-CNN. G-CNN outputs the label of the general object l_{g} and the vector $\boldsymbol{b}_{\mathrm{g}}$, which represents the rectangle region (width, height, center point). If the output label l_{g} of G-CNN is equal to the target label label l_{t}, the object region is cropped out as an image $\boldsymbol{I}_{\mathrm{g}}$ from $\boldsymbol{I}_{\mathrm{f}}$. $\boldsymbol{I}_{\mathrm{g}}$ is compressed (or extended) to the specified image size and inputted to M-CNN. M-CNN outputs the label of specific object l_{m}. If l_{m} does not change in multiple images, a time-series filter is initialized and object tracking is started.

In the proposed method, training is only needed for M-CNN. Therefore, we just collect images and labels of the target and do not need to annotate the position information. M-CNN can be implemented as the simple structure of CNN compared to G-CNN because it is used only for classification. In the case of target change, the method can adapt by replacing M-CNN without retraining of the G-CNN.

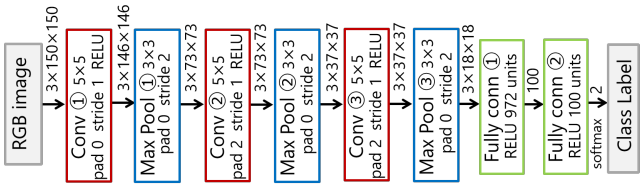


Fig. 2. Structure of M-CNN.

IV. Different Types of CNNs

A. G-CNN for general object recognition

In this study, YOLOv3 is used as G-CNN. Eighty types of multiple objects can be recognized simultaneously by an already trained YOLOv3 model [11].

B. M-CNN for object shape classification

The structure of M-CNN is shown in Fig.2. M-CNN is composed of convolution layers, pooling layers, and fully connected layers. This is a basic structure of CNN for classification. Note that the image outputted by G-CNN is changed to a 150×150 pixel image for M-CNN input.

V. Time-Series Filter

In this study, KCF is used as a time-series filter for object tracking. KCF is initialized by the object region detected by G-CNN and M-CNN. The recognition of CNNs stops during tracking and it switches when KCF fails to track the target.

VI. Training M-CNN

A. Dataset Preparations

We address a two-class sorting problem of carrots. The goal of this problem is to classify the carrots into two groups, A and B. The carrots of group A are short and thick. The carrots of group B are long and thin. Some sample images of group A and group B are shown in Fig.3. G-CNN detects region of the carrot and M-CNN classifies it into one of the two groups. A large number of images are needed to train M-CNN. G-CNN is also used to extract the carrot images from videos. The process of dataset generation is shown in Fig.4.

The carrot image extracted by G-CNN is transformed to a 150×150 pixel square image. By compressing the rectangle region of the G-CNN detection output, the shape feature is lost, as shown in Fig.5. To avoid this problem, three dataset pre-filtering methods are proposed.

(a) Group A (b) Group B

Fig. 3. Examples of two groups of carrots.

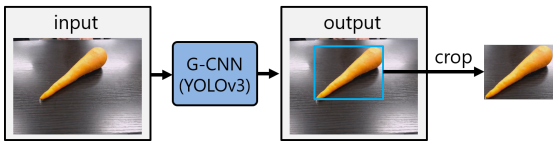


Fig. 4. Dataset generation.

(a) Group A (b) Group B

Fig. 5. Shape feature loss by image compression.

1) **Method1: Square Cropping***:* This involves the cropping of a square carrot image using the longer side of the carrot rectangle region detected by G-CNN. DATA1 is generated by this method. Sample images are shown in Fig.6.

2) **Method2: Square Cropping with Black Background***:* This involves the application of Method1 with black background samples. DATA2 is generated by this method. Sample images are shown in Fig.7.

3) **Method3: Superposition with Black Square***:* This involves the superposing a carrot rectangle region detected by G-CNN on a black square image generated using the longer side of the region. DATA3 is generated by this method. Sample images are shown in Fig.8.

(a) Group A

(b) Group B

Fig. 6. Examples of DATA1.

(a) Group A

(b) Group B

Fig. 7. Examples of DATA2.

(a) Group A

(b) Group B

Fig. 8. Examples of DATA3.

TABLE I
THE NUMBER OF IMAGES.

	DATA1	DATA2	DATA3
A	10178	12355	9065
B	11039	12768	9899
Total	21217	25123	19054

TABLE II
PARAMETERS FOR M-CNN TRAINING.

Item	Value
Loss function	Softmax cross entropy
Optimizer	Adam ($\alpha = 0.0001$)
Training dataset	15000
Minibatch size	50
Iteration	15

TABLE III
TEST RESULTS FOR 1000 IMAGES.

	M-CNN1	M-CNN2	M-CNN3
Acc.	97.6	95.8	98.9

The number of images is increased by rotation, reversal, contrast adjustment, noise application, and smoothing. The numbers of images in datasets DATA1, DATA2, and DATA3 are shown in Table I.

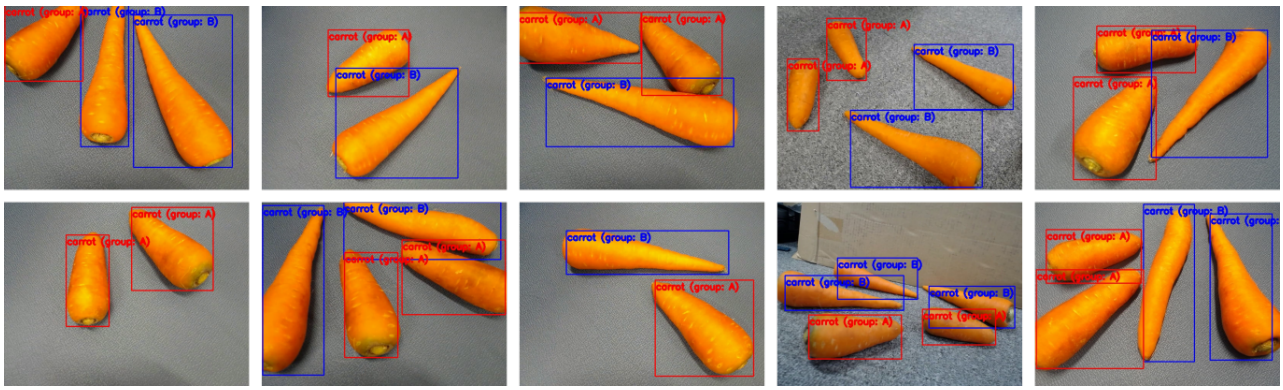
B. Training Results

Let M-CNN learned by DATA1, DATA2, and DATA3 be M-CNN1, M-CNN2, and M-CNN3, respectively. The parameters for M-CNN training are shown in Table II. All M-CNNs are trained with same parameters shown in the table. The classification accuracy of each M-CNN with 1000 unseen images is shown in Table III.

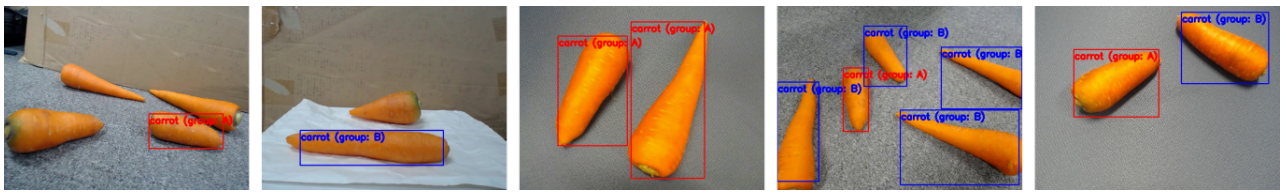
VII. EXPERIMENTS

A. Method

To confirm the effectiveness of the proposed method, recorded videos are used for validation. The correct labels for each group (A or B) are recorded by analyzing the image



(a) Succeeded recognition results.



(b) Failed recognition results.

Fig. 9. Recognition results of YOLOv3 + M-CNN3.

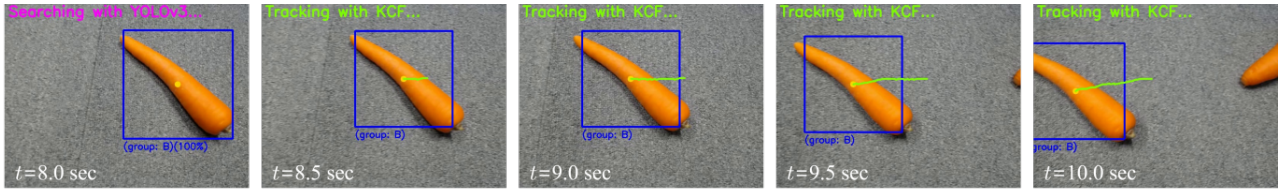


Fig. 10. Tracking result of YOLOv3 + M-CNN3 + KCF.

TABLE IV
EXPERIMENTAL ENVIRONMENT.

Item	Version
Operating system	Ubuntu 16.04 LTS
CPU	Intel Core i7 6700
GPU	NVIDIA GTX 1070
Programming language	Python 3.5.5
Neural network framework	Darknet, Chainer 4.2.0
Image processing library	OpenCV 3.4.0
Camera	Logicool c930e (image size: 480 × 640)

TABLE V
RECOGNITION ACCURACY FOR 300 FRAMES.

	M-CNN1	M-CNN2	M-CNN3
A	59.00	65.00	91.33
B	66.67	85.33	82.67
Mixed	41.00	56.67	57.33
Average	55.56	69.00	77.11

frames in the video. This recording process is completed manually. For each group, 300 image frames are collected. A further 300 image frames are collected for the case where the two groups are mixed. The accuracy of the proposed method is validated by comparing the outputted labels and correct labels. The experiment environment is shown in Table IV.

B. Results

Recognition results of YOLOv3 + M-CNN3 are shown in Fig.9. The carrots of groups A and B are represented by red and blue rectangles, respectively. Successful samples are shown Fig.9 (a) and failed samples are shown Fig.9 (b). The tracking results of YOLOv3 + M-CNN3 + KCF are shown in Fig.10. The green line in Fig.10 is the trajectory of the center point. KCF is effective for complementing the recognition flickers of CNN. Recognition accuracy of the proposed method for each M-CNN is shown in Table V. Only the perfect label answers are counted as successful in these experiments. The accuracy of the method with M-CNN3 is the highest on average. Recognition and tracking worked in real time with a GPU system.

VIII. CONCLUSIONS

In this study, we connected two different types of CNNs to recognize specific objects and applied a time-series filter to track the target. In addition, to make it possible to distinguish according to the shape of the object, we proposed a pre-processing method for the input image. The effectiveness of the proposed method is confirmed by using recorded videos and evaluating the accuracy of specific carrot shape recognition. Additional study is needed to improve the effectiveness because the accuracy does yet have sufficient quality. In the future, we will install the proposed method in real robot systems, such as the pick-and-place system with an industrial robot.

REFERENCES

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems, 2015.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C. Berg, "SSD: Single Shot MultiBox Detector." arXiv preprint arXiv:1512.02325, 2015.

[3] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement", arXiv:1804.02767v1 [cs.CV] 8 Apr 2018.

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

[5] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun and Alan Yuille, "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] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick and Piotr Dollar, "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.

[9] Yang Li and Jianke Zhu, "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] David Held, 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/ (Jun. 2018).