# Object Tracking using CSRT Tracker and RCNN

Khurshedjon Farkhodov[1], Suk-Hwan Lee[2] and Ki-Ryong Kwon[1]
[1]*Dept. of IT Convergence and Applications Engineering, Pukyong National University, South Korea*
[2]*Dept. of Information Security, Tongmyong University, South Korea*

Keywords: Object Tracking, Object Detection, CSRT, Faster RCNN, CSR-DCF, CNN, Opencv, Deep Learning, DNN Module.

Abstract: Nowadays, Object tracking is one of the trendy and under investigation topic of Computer Vision that challenges with several issues that should be considered while creating tracking systems, such as, visual appearance, occlusions, camera motion, and so on. In several tracking algorithms Convolutional Neural Network (CNN) has been applied to take advantage of its powerfulness in feature extraction that convolutional layers can characterize the object from different perspectives and treat tracking process from misclassification. To overcome these problems, we integrated the Region based CNN (Faster RCNN) pre-trained object detection model that the OpenCV based CSRT (Channel and Spatial Reliability Tracking) tracker has a high chance to identifying objects features, classes and locations as well. Basically, CSRT tracker is C++ implementation of the CSR-DCF (Channel and Spatial Reliability of Discriminative Correlation Filter) tracking algorithm in OpenCV library. Experimental results demonstrated that CSRT tracker presents better tracking outcomes with integration of object detection model, rather than using tracking algorithm or filter itself.

## 1 INTRODUCTION

The main goal in this proposed tracking method is to focus on exact object to track, furthermore because of the real-time object tracking environment there are some parameters should be considered, such as camera movement, distance between object and camera, and so on. As an object detector Faster R-CNN have been used after comparing all object detection methods, for instance Viola-Jones algorithm: the first efficient face detector (P. Viola and M. Jones, 2001), much more efficient detection technique: Histograms of Oriented Gradients (N. Dalal and B. Triggs, 2005), and from 2012 the deep learning methods, which is called "Convolutional Neural Networks" became the gold standard for image classification after Kriszhevsky's CNN's performance during ImageNet (A. Krizhevsky, et al., 2012), While these results are impressive, image classification is far simpler than the complexity and diversity of true human visual understanding. A better approach, R-CNN has been proposed (R. Girshick, et al., 2013) after CNN realized. R-CNN creates bounding boxes, or region proposals, using a process called Selective Search. Later, where further merge of R-CNN and Fast R-CNN into a single network by sharing their convolutional features using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look (R. Girshick, 2016). In this paper, we propose an algorithm that to apply OpenCV-based CSRT tracker into person detection and tracking, which we combined with Faster R-CNN based object detector with the support of OpenCV's DNN module and obtained trained object detector model. With the great afford of deep learning-based object detection technique that we can easily avoid target misclassification and lost. Results of proposed tracking method has gained remarkable outcome, and tested in a different dataset with single and overlapping images, also the tracking system experimented with video and real-time sequence and got much better performance that rather than using tracking filter without any additional supporter like detector.

This article consists of following sections: our suggested tracking system structure explained in system overview section comes with related work. In section 3, the detailed results will be presented which obtained during the experiments and we also compared visual value with the conventional methods performance, conclusion and reference follows as well.



Farkhodov, K., Lee, S. and Kwon, K.
Object Tracking using CSRT Tracker and RCNN.
DOI: 10.5220/0009183802090212
In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 2: BIOIMAGING*, pages 209-212
ISBN: 978-989-758-398-8; ISSN: 2184-4305

# 2 RELATED WORKS

## 2.1 Overview of the Object Detection Task

Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. Variations in position, color, lighting conditions, size, etc., greatly affect the performance of the model. Object detection and tracking processes should be fine-tuned, that depending on what kind of problem is going to be solved. This is usually determined by the characteristics of the target. For instance, detecting vehicles require different parameters tuning than detecting pedestrians, animals, or faces, and so on. This feature-based technique exploits a notable differentiation character of the objects that taken from 2D pixel information in an image (Gonzalez, R. C., and Woods, R. E., 2017). While using feature points of 2D images, such as color, intensity, background information it is easy way to identify object from frames if it will not change the appearance, position, and size as well.

## 2.2 Faster RCNN Object Detection Architecture

Faster RCNN is becoming one of the most used and popular an object detection architecture presented by R. Girshick, Sh. Ren, K. He and J. Sun in 2015 that uses Convolutional Neural Network like other famous detectors, such as YOLO (You Look Only Once), SSD (Single Shot Detector) and so on. In general, Faster RCNN composed from three main part at all that can be managed building object detection model process. They are: a) convolution layers; b) region proposal network; c) classes and bounding boxes prediction (Faster RCNN, n.d.):

## 2.3 Training Dataset

We have used our own dataset (600 images) for training (400) and testing (200) process, which has been collected throughout internet sources, such as blogs, posts, and so on. The difference between our dataset apart from other datasets is that our dataset images taken by drone camera and environment. Training process speed and time depends on what kind of CPU and GPU system we have, if our OS has last version of GPU hardware system that means we can get training results faster than using CPU system. If your computer has no supporting hardware platform or latest GPU system, we recommend you do not use your computer for other high memory or performance required processes while training, that can affect to obtain training outcomes as well as it could lead your training process to be time consuming. Training classifier should train until the loss is consistently below 0.05 or so that the law starts to plateau out. A total loss graph estimates that while learning training dataset images it can loss or misidentify objects by their features, shapes and other parameters in one average graph performance. The total loss of training process performance together with objectiveness loss, which can show us the objectiveness score (4e-3≈0.004) of the dataset's images, to indicate if this box contains an object or not while training process is given in Figure 1 below:


Figure 1: Performance of the total losses along with objectiveness loss.

## 2.4 Proposed Object Tracking Method

The basis of our tracking method taken from the Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF) (A. Lukezic, et al., 2018) tracking algorithm. Moreover, this algorithm has been implemented in a C++ and integrated into Open CV library as a DNN (Deep Neural Networks) module (OpenCV dnn module, n.d.). We proposed a tracking system that integration of Faster RCNN object detection as an object detector and OpenCV_CSRT_ tracker as a tracking algorithm for tracking method.

Implemented object tracking process contains two parts: first is already explained above, training object classifier and generating object detection model from training outcome file. From coming frame algorithm takes blobs and gives it to object detection model to predict location of object and classify as an object class. Output predictions of the detection passes to tracking algorithm to track predicted box of object class and so on, any other circumstances of object prediction changes tracker will invoke the object classifier and restarts process from the beginning.

# 3 EXPERIMENTAL RESULTS

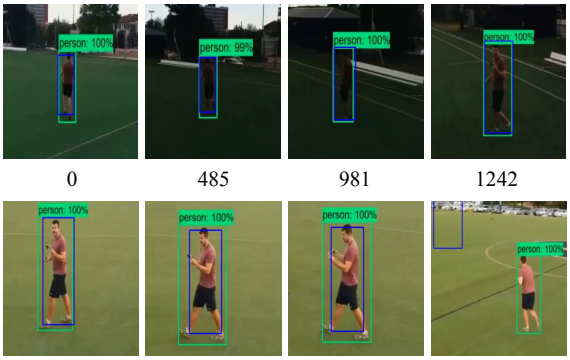
After finishing 200K times training our dataset we got

our object classifier model and we have tested our object detection (classifier) model work by applying images from different open source resources that accomplishment of our detector demonstrated perfect results, such as shown in Figure 2. We can see the remarkable results of object detection model from different sources; however, the object classifier is not detecting all objects in frame at once in a section (c) and accuracy rate is under 100%. While tracking we may face thousands of positions, location, shape as well that means it requires more quantity of images on dataset with different position and environment object located images to get perfect results. Nevertheless, with 600 labeled images dataset gave unusual result after training by Faster RCNN object classifier model.


a) b) c) d)

Figure 2: Object detection results of the pre-trained Faster RCNN object classifier: (a) image from internet sources taken by drone camera; (b, c) picture taken from drone video sequence; (d) image from VOT2018 dataset.

We have tested our tracking method with different open source datasets and got good results and compered with conventional tracker itself. The experiments show that CSR-DCF tracker algorithm cannot re-establish object once it gone from current frame or tracks object not properly and if the shape or appearance of the tracking object changes in a noticeable drape tracker can't track object properly and consequently it will lose a target. We have tested this situation on video sequence taken by drone, here is some result of tested video shown in Figure 3 below:




0 485 981 1242

1611 1696 1854 2106

Figure 3: Qualitative results of conventional (CSR-DCF, blue) and proposed (FRCNN_CSRT, green) trackers (0, 485, 981,… - frames).

Initially, conventional tracking method performance was good, it tracked a target properly, but after some movement of target frame by frame and changing a foreground/background color of frame the tracker could not predict properly (given in 1611, 1696, 1854 frames) a new location of target and position of the maximum in correlation between backgrounds and image patch features extraction on position and weight by the channel reliability scores.

Our tracker performance precision (accuracy) has been calculated by the ability of our tracking model that to identify only the relevant objects. It is the percentage of correct positive predictions. Precision calculation equation given below:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detections}$$

Here, TP is a *True Positive* – a correct detection, detection with intersection over union (IOU ≥ threshold); FP is a *False Positive* – a wrong detection, detection with IOU < threshold.

Furthermore, a proposed tracking method has been tested with several well-known open source datasets, like OTB (Object Tracking Benchmark), VOT (Visual Object Tracking), and others. However, we have compared results in Table 1 with some open source datasets which conventional method has been tested and posted as a final approach performance in several internet blogs. We can see the comparison results of conventional and proposed tracking methods in Table 1, where has been tested with three different open source datasets that the main difference between two methods is deep learning-based approach and real time frames per second as well. As we can see below in the first column of TABLE 1 results of CVRP2013 dataset while conventional method presents 80 percent accuracy, proposed method gained 89 percent tracking accuracy respectively. Furthermore, outcomes of the tested technique performance can be seen in next two column with some of systems feature.

Table 1: Comparsion Results.

| Tracker | Precision-CVPR2013 | Precision-OTB100 | Precision -OTB50 | Deep Learning (Yes/No) | Real Time |
|---|---|---|---|---|---|
| CSR-DCF | 0.8 | 0.733 | | N | Y(13) |
| Proposed | 0.89 | 0.829 | 0.81 | Y | Y(30) |

# 4 CONCLUSIONS

In conclusion, we can say that our proposed tracking method has gained good performance as well as comparable results. Our trained object classifier model has showed excellent work by training 200K times only 400 images for training and 200 images for testing. In this method we have integrated deep learning-based object classifier model with CSRT tracker OpenCV implementation version of CSR-DCF algorithm with the support of OpenCV DNN module. We have compared performance of CSR-DCF tracking algorithm with our integrated tracking method. Comparison results showed a much better performance in proposed tracking method that in tracking process an object classifier gives exact location of the target in frame with the benefit of pre-trained object classifier. In the future work we will try to create CNN tracker that can be work without pre-trained model.

# ACKNOWLEDGEMENTS

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2020-2016-0-00318) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2016R1D1A3B03931003, No. 2017R1A2B2012456), and Ministry of Trade, Industry and Energy for its financial support of the project titled “the establishment of advanced marine industry open laboratory and development of realistic convergence content”

# REFERENCES

P. Viola and M. Jones, July 13, 2001. “Robust Real-time Object Detection,” *Second international workshop on statistical and computational theories of vision – modeling, learning, computing, and sampling,* Vancouver, Canada, IJCV 2001 See pp 1-3.

N. Dalal and B. Triggs, Jun, 2005. “Histograms of Oriented Gradients for Human Detection,” *International Conference on Computer Vision & Pattern Recognition (CVPR '05)*, San Diego, United States. pp.886-893.

A. Krizhevsky, I. Sutskever, Geoffrey E. Hinton, 2012. “ImageNet Classification with Deep Convolutional Neural Networks” - Part of: Advances in Neural Information Processing Systems 25 (NIPS), DOI: 10.1145/3065386.

R. Girshick, J. Donahue, T. Darrell, and J. Malik, November 2013. “Rich feature hierarchies for accurate object detection and semantic segmentation”, DOI: 10.1109/CVPR.2014.81.

R. Girshick, 6 Jan. 2016. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” - Computer Vision and Pattern Recognition (cs.CV), arXiv:1506.01497 [cs.CV].

Gonzalez, R.C., and Woods, R.E., 2017: *Digital Image Processing. Pearson*, 4th edition.

Faster RCNN, https://towardsdatascience.com/faster-rcnn-object-detection- f865e5ed7fc4

A. Lukezic, T. Vojır, L. C. Zajc, J. Matas, and M. Kristan, July 2018. “Discriminative Correlation Filter Tracker with Channel and Spatial Reliability,” *International Journal of Computer Vision:* Volume 126, Issue 7, pp 671–688.

OpenCV dnn module, cv::dnn::Net Class Reference, https://docs.opencv.org/master/db/d30/classcv_1_1dnn_1_1Net.html