Journal of Korea Multimedia Society Vol. 22, No. 12, December 2019(pp. 1415-1429)
https://doi.org/10.9717/kmms.2019.22.12.1415

A New CSR-DCF Tracking Algorithm based on Faster RCNN Detection Model and CSRT Tracker for Drone Data

Xurshid Farhodov[†], Oh-Heum Kwon[††], Kwang-Seok Moon[†††], Oh-Jun Kwon[††††], Suk-Hwan Lee[†††††], Ki-Ryong Kwon[††††††]

ABSTRACT

Nowadays object tracking process becoming one of the most challenging task in Computer Vision filed. A CSR-DCF (channel spatial reliability-discriminative correlation filter) tracking algorithm have been proposed on recent tracking benchmark that could achieve stat-of-the-art performance where channel spatial reliability concepts to DCF tracking and provide a novel learning algorithm for its efficient and seamless integration in the filter update and the tracking process with only two simple standard features, HoGs and Color names. However, there are some cases where this method cannot track properly, like overlapping, occlusions, motion blur, changing appearance, environmental variations and so on. To overcome that kind of complications a new modified version of CSR-DCF algorithm has been proposed by integrating deep learning based object detection and CSRT tracker which implemented in OpenCV library. As an object detection model, according to the comparable result of object detection methods and by reason of high efficiency and celerity of Faster RCNN (Region-based Convolutional Neural Network) has been used, and combined with CSRT tracker, which demonstrated outstanding real-time detection and tracking performance. The results indicate that the trained object detection model integration with tracking algorithm gives better outcomes rather than using tracking algorithm or filter itself.

Key words: Object Tracking, CSR-DCF, Object Detection, CNN, Faster R-CNN, OpenCV, DNN Module, CSRT, Drone, Deep Learning

1. INTRODUCTION

Real-time object tracking becomes much more demanding task, while applying CNN-based structure for detecting and identifying exact object from coming real-time frames, and it's also urgent to mention that during actual time tracking, position or location of the object in the real time video sequence of the frames may come with overlapping, occlusions, motion blur, changing appearance, illumination changes, cluttered background, environmental variations, and so on [1-3]. In such cases,

※ Corresponding Author : Ki-Ryong Kwon, Address: (48513) 45 Yongso-ro, Namgu, Busan, Pukyong National University, Korea, TEL : +82-51-629-6257, FAX : +82-51-629-6230, E-mail : krkwon@pknu.ac.kr

Receipt date : Oct. 7, 2019, Revision date : Dec. 16, 2019
Approval date : Dec. 20, 2019

[†] Dept. of IT Convergence and Application Engineering, Pukyong National University
(E-mail : life9940502@gmail.com)
[††] Dept. of IT Convergence and Application Engineering, Pukyong National University
(E-mail : ohkwn@pknu.ac.kr)
[†††] Dept. of Electronics Engineering, Pukyong National University (E-mail : ksmoon@pknu.ac.kr)
[††††] Dept. of Computer Software Engineering, Dongeui University (E-mail : ojkwon@deu.ac.kr)
[†††††] Dept. of Information Security, Tongmyong University (E-mail : skylee@tu.ac.kr)
[††††††] Dept. of IT Convergence and Application Engineering, Pukyong National University

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017R1A2B2012456) and the 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 contents, and MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2019-2016-0-00318) supervised by the IITP (Institute for Information & communications Technology Promotion)”.

all kind of tracking filters, algorithms, methods may fail, and may not recover object after appearing in the next frame. The proposed method can overcome problems which mentioned above and could achieve better tracking results. In this proposed method there are some tasks for creating object tracking process comes with doing some steps and general idea is shown in Fig. 1.

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 [4,5], much more efficient detection technique: Histograms of Oriented Gradients [6], 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 [7], While these results are impressive, image classification is far simpler than the complexity and diversity of true human visual understanding. In CNN based object detector there is image classification part, whereas an image with a single object as the focus and the task is to say what that image is. However when we look around us, it's far more complex task to carry out. While detecting object from coming real-time frames it may appear with complicated sights including multiple overlapping objects, and different backgrounds, and not only classify these different objects but also identify their boundaries, differences, and relations to one another.

A better approach, R-CNN has been proposed [8] after CNN realized. R-CNN creates bounding boxes, or region proposals, using a process called Selective Search. Later, improvement of R-CNN has been recommended called Fast R-CNN by the same authors [9], which Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Furthermore, the authors of Fast R-CNN tried to find out more results and high detecting accuracy in every type of classes, and one more implementation of work Faster R-CNN suggested by R. Girshick [10], where further merge 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 [10].

There are several tracking filters, methods, algorithms have been implemented and utilized in variety of task that achieved noticeable outcomes. However, this actual algorithms of tracking pedestrians shows a low performance when faced with problems like mentioned in a first paragraph of this chapter. 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 and obtained trained object detector model. With the great afford of deep learning based object detection technique that we can easily avoid target misclassification

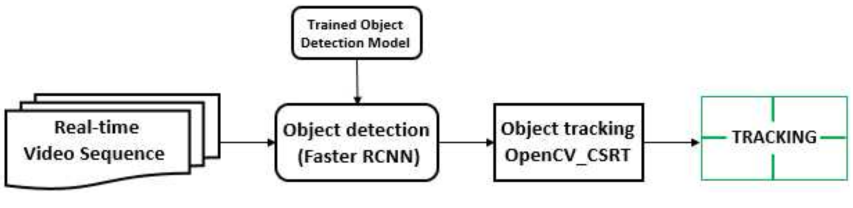
Trained Object Detection Model
Real-time Video Sequence
Object detection (Faster RCNN)
Object tracking OpenCV_CSRT
TRACKING

Fig. 1. Basic steps for tracking an object.

and lost. As an object tracker we use OpenCV-based CSRT tracker that applies trained detection model which includes all object features, candidate regions, object bounds, objectless scores at each position, and exact labeled object. Our proposed algorithm associated with detection model and tracker at the same time. Moreover, the states of object are determined by detector and tracker that can use the information respectively while in tracking process.

Finally, the results demonstrate that proposed tracking method has gained remarkable outcome, and tested in a different datasets 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. We also demonstrated result of tracking that compared with detector, also without it, performance is clear an obvious.

Next section follows with the mathematical explanation of proposed approach in detail and along with some graphical visualization. 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.

2. PROPOSED TRACKING METHOD

Tracking process is quite demanding task in case of using in a real-time controlling systems. In general, the problem we are going to find a solution is starts with learning what object is going to track and collecting dataset related to our target. Suggested Object detection method in this proposed object tracking is one of the most useful and popular detection structure among detection approaches that gives good results faster and high accuracy. Tracking algorithm has been implemented to correspond each other while tracking process and get required information from trained object detection model. Fig. 1 summarizes general principal steps used in this method.

2.1 Drone Dataset (DD)

The dataset has been collected through different kind of internet resources, like post blocks, media sites, google resources, and so on. All pictures on dataset taken by drone camera in a different positions and backgrounds. After collecting all needed images, it has been labeled into two classes: person and car. Those labeled images stored into required direction to create a label map for identifying object location, bounding boxes, and object features from dataset. A differences between our dataset from other open source datasets are: in our dataset only two type of objects have been labeled and created label map after detection for identifying object's ID and name as well. Our dataset contains 600 labeled images, 400 for training and 200 for testing. All images have been labeled manually and created .csv files for training and test labeled images, which contains coordinates of bounding rectangles for every objects in every images that stored image folder as a ready dataset files to run for training process. Here is some images shown in Fig. 2 from our DD.

2.2 Overview of the object detection task

Generally, Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain classes(such as humans, buildings, or cars) in digital images and videos [11]. Well-researched domains of object detection include face detection and pedestrian detection too.

Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. Detecting objects from image whether an example of a semantic object is in a 2D image or in a real-time video sequence, one could create a classification model trained with feature extracted from positive and negative classes and use the trained model to classify a given image [12]. Variations in position, color, lighting

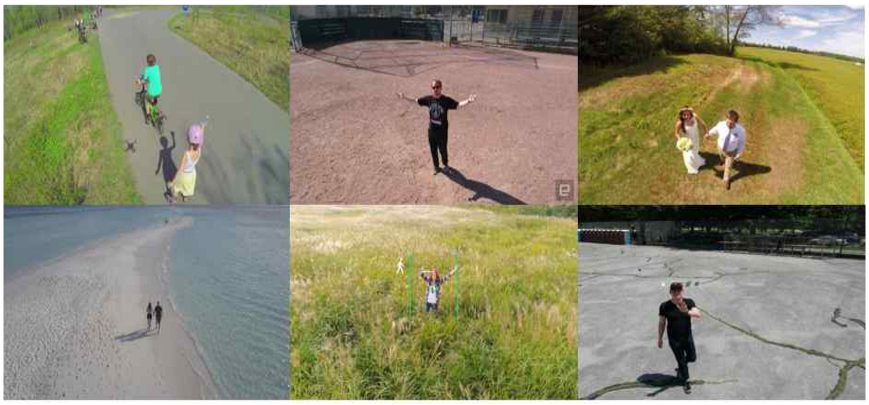
Fig. 2. Drone dataset.

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. Due to the variations of the target, the process of detection differs with internal parameters, such as environmental conditions, target size, change in appearance, etc.

This feature-based technique exploits a notable differentiation character of the objects that taken from 2D pixel information in an image [13]. 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. However this 2D pixel information of images usually cannot give us proper and exact parameters of observed objects. Deep learning approaches for object detection has achieved state-of-the-art performance and used several object detection applications. This method gives better results rather than other detection techniques and can resolve the problems mentioned in the introduction part. However, the indicated way of detection has required specific hardware and software systems for designing object detection model. There are certain high-level APIs (Application Programming Interface) like TensorFlow, Keras, and JSON that a set of functions and procedures allowing the creation of applications, that can access the features or data of an operating system, application, or other service for building a deep learning based object detection model. According to the given issue and our goal in all respects for building detection model we have used TensorFlow API model and Faster RCNN an object detector architecture presented by R. Girshick, and his colleagues in 2015 [10].

2.3 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, that shown in Fig. 3 below [14]:

a) Convolutional Layers. These layers can help to extract the appropriate features of the images from dataset via training filters that will learn

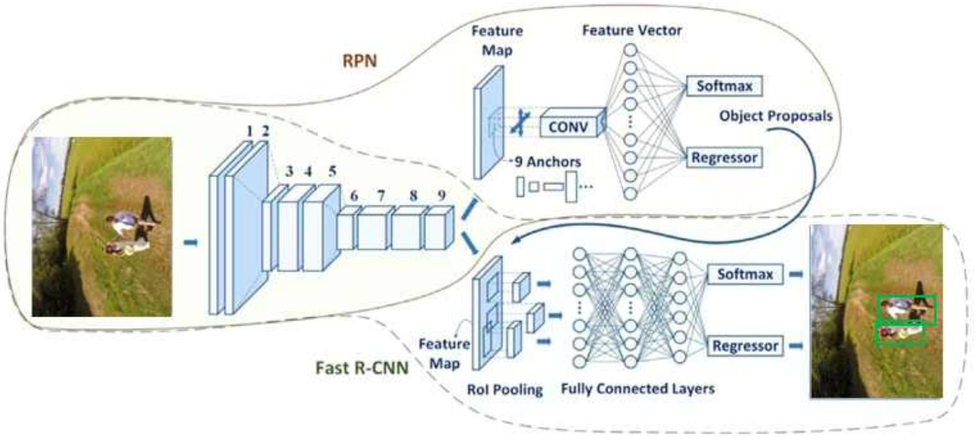

Fig. 3. Main architecture of Faster RCNN.

through training shapes and colors that exist in training images. Our target to detect and track is person or we can say pedestrian as well. Learning rate of object detection depend on the quantity of images and structural layer combination of the CNN. As an example of learning features process we can assimilate filtration of the ready coffee. As a convolution layers to coffee filter that coffee filter does not let the coffee powder pass to the cup, so in this case also convolutional layers learn the object features and don't let anything else pass without learning, only the targeted object. In a detail, we can improve this activity by making labeling images manually as well, and as a result the ready coffee liquid will be last feature map of the CNN. Generally convolutional network composed of Convolution layers, pooling layers, activation layers and last component of the layers called fully connected layer or another extended thing that will be applied for an appropriate task like classification or detection. There is common structure of the CNN structure shown Fig. 4 [14]:

By sliding filter along throughout input image we compute convolution and result will come out as a two dimensional matrix called feature map within in anchors that shown in Fig. 5:

After computing feature map comes pooling layer which consist of decreasing quantity of features that eliminates pixels with low values from feature map shown in Fig. 6:

b) Region Proposal Networks. RPN is a small neural network to generate proposals for the region where the object located, that a small sliding anchor box over a convolutional feature map and predict whether there is an object or not and also predict the bounding box of those objects shown in Fig. 7 [15, 10]:

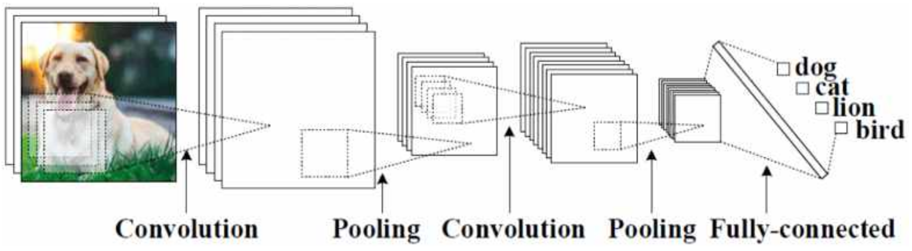

Fig. 4. Common CNN structure for object classification.

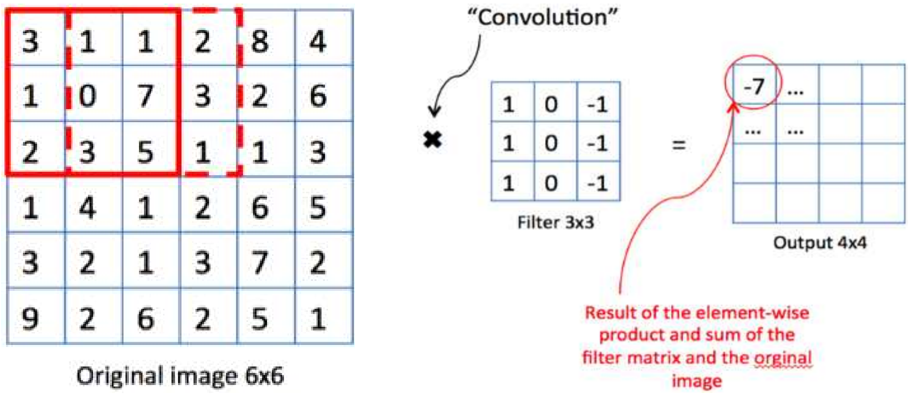

Fig. 5. Calculation of feature map.

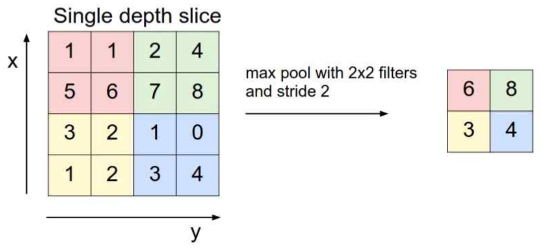

Fig. 6. Reducing the spatial size of the representation in an image by diminishing the amount of parameters and computation in the network.

RPN has a classifier that determines the probability of proposal having the target object, while regression regresses the coordinates of the proposals. In Fig. 7 shown that k anchors boxes applied for each location, which means anchors are translation invariant that uses the same ones at every location. Regression gives offsets from anchor boxes that each (regressed) anchor shows an object after classification gives the probability.

c) Classes and Bounding Boxes Prediction. This part is final step of object detection that we use fully connected neural network layer, and as an input the regions are proposed by the RPN and predicts object classes (Classification), Bounding boxes (Regression) as well.

2.4 Training Drone Dataset

Most open source datasets have been applied for classifying objects and identifying them as well. There are several datasets which contains more than 1000 classes and each classes have training and tasting images, that most common datasets: ImageNet, MNIST, LSUN, MS COCO – Common Objects in Context, Youtube-8M, Yelp Reviews, CIFAR-10, Open Image Dataset and so on. We collected our own dataset which has 600 images at all, of these 400 for training and 200 for testing images. As we mentioned before, we labeled all images manually and created label map for classifying them into two classes: pedestrian and vehicle. There is no exact number of labeled person or car classes, that most of images have more than one object and car or person class. These images have been stored into needed direction and created train_label and test_label .csv files which holds coordinates of the bounding rectangles for every ob-

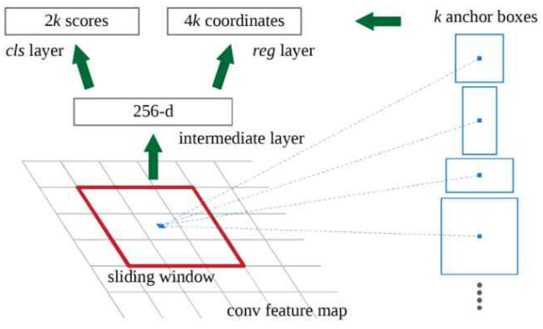

Fig. 7. Example of RPN structure.

ject in every image. After that by using created .csv files training and testing tfrecords have been generated where we can use to read data efficiently and it can be helpful to serialize our data and store it in a set of files that can each be read linearly.

Now we need to configure object detection pipeline that defines which models and what parameters will be used for training and points through the training images and data. Our training model is called Faster RCNN Inception V2 which we will train our dataset by editing some parameters and passing it to training folder. While editing this model we have to change number of classes which we want to classify and detect. Moreover, there is some file paths like checkpoint, tfrecords, label map has been changed, also number of training and testing images quantity changed to exact number in images folder. After setting up all parameters and needed configurations our selected model ready to training process. If everything has been set up correctly tensorflow will initialize the training and initialization can take 30 seconds before the actual training begins.

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 results faster than using CPU system, while training time we recommend not doing anything on your computer that requires lots of processing power, because training classifier uses 100% of your CPU and GPU. The training process begins with stepping through training batches and reporting the loss at each step and the law sloth starts off high and then gets lower and

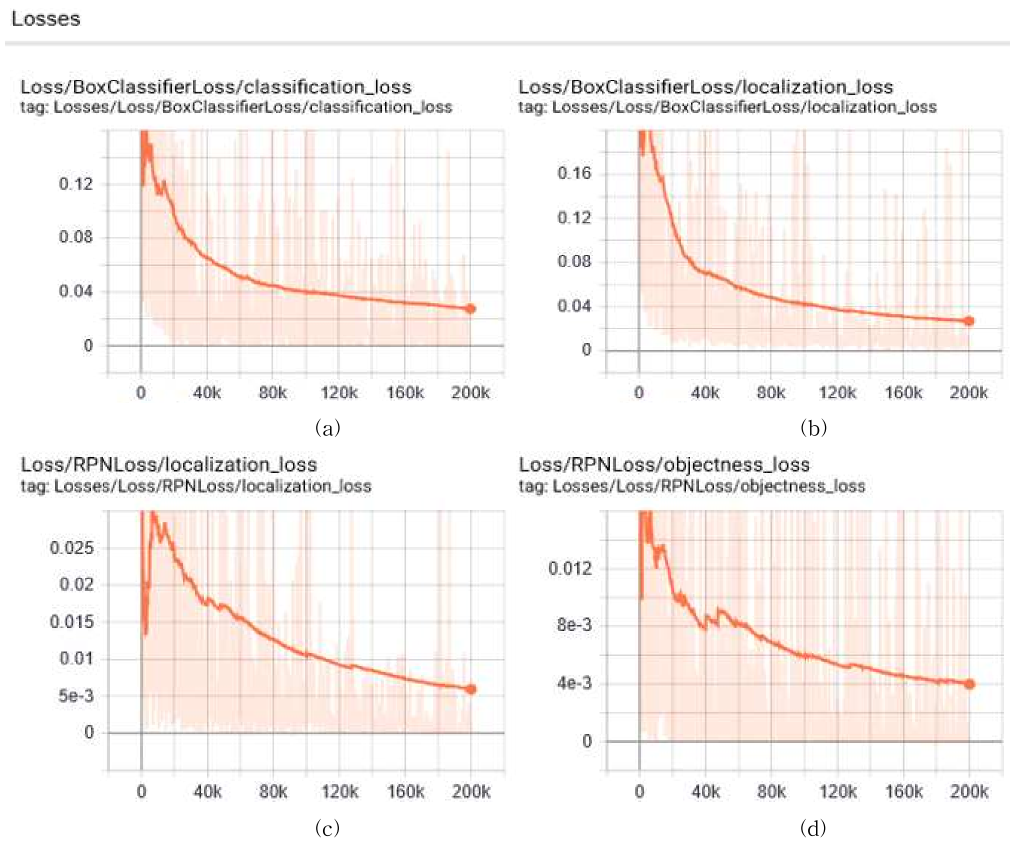

Fig. 8. Tensor board losses results after 200K times training: a) in a box classifier / classification loss; b) in a box classifier / localization loss; c) in the RPN / localization loss; d) in the RPN / objectless loss

lower as the object detection classifier trains. Training classifier should train until the loss is consistently below 0.05 or so that the law starts to plateau out. We can view the process of training job by using tensor board to set up it on our local-host and view through a web browser.

In Fig. 8 has shown the result of training process after 200k times at all, where we can see the classifier has reached lower than 0.05 in a first (a) graph that means our chosen model for training has showed a good performance. 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 clone loss given in Fig. 9 below:

One more essential part of graphical result of training process are learning rate of trained and tested dataset and step of learning, which shows how quickly the model is adapted to the problem in the range of -1 and 1, but exact learning rate is 2e-4 shown in Fig. 10. Specifically, the learning rate is a configurable hyper parameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0 [16]. Next graph named Global step is shows that time where the Faster RCNN object detection model took time for learning dataset images one by one. In this graph we can see average step for learning one step took 0.28 second, also it depends

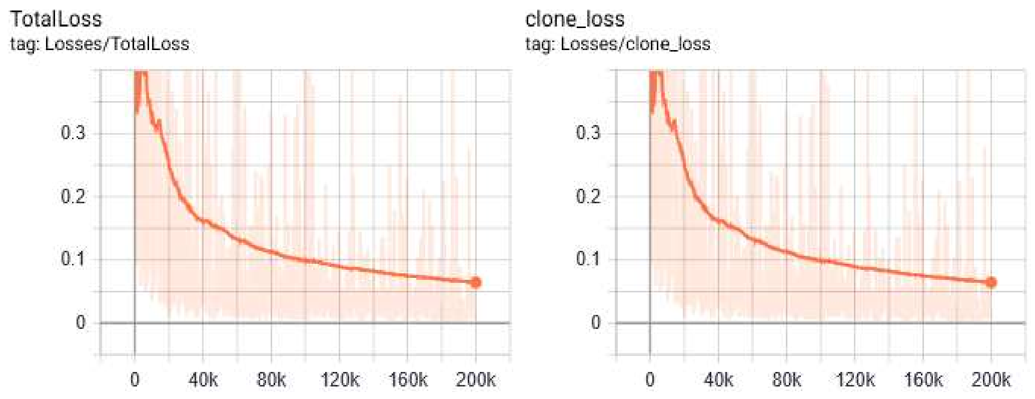

Fig. 9. Performance of the total losses along with clone loss.

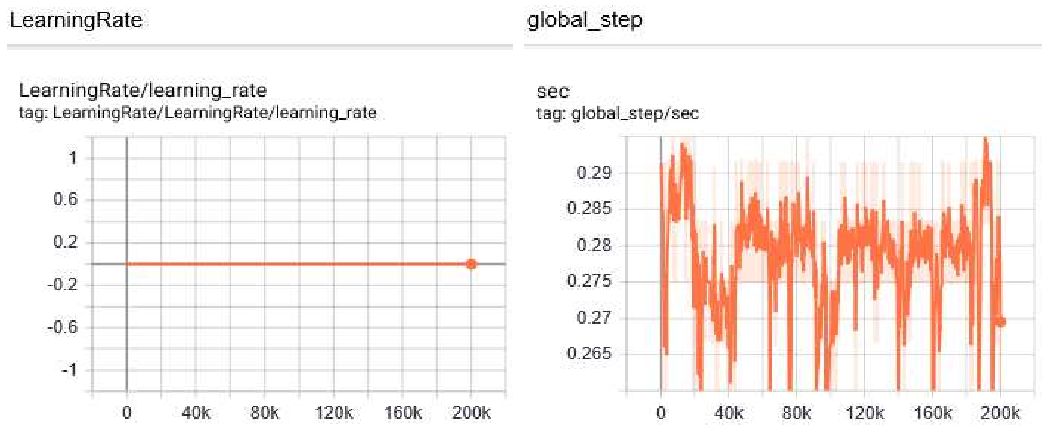

Fig. 10. Learning Rate and Global Step of training process.

on how many images in dataset as well. Tensorflow saves training checkpoints every five minutes or so and stores them into the training folder, where the checkpoint of highest step count will be used to generate a classifier. Finally, after training 200k times finished in a training folder there are some generated training result files like checkpoint, model.ckpt.meta, model.ckpt.index, pipline.config, and so on, which we can use to export inference frozen graph as an object detection model or object classifier.

The last step is to generate a frozen inference graph by using the step count of the highest model checkpoint in the training folder that in our case it's two hundred thousand (200k). By typing needed command line code we will create a frozen inference graph .pb file into the given output path. This .pb files contains the object detection classifier that will be used as an object detection model while object tracking process.

2.5 Proposed Object Tracking Method

Object Tracking is one of the most challenging task in Computer Vision and it has an important domain in this sphere. It involves the process of tracking an object which could be a person, car or any type of movable objects across a series of frames. Our proposed tracking system related to tracking people which we would start with all possible detection in a frame by using Faster RCNN object detection model. Even in subsequent frames person will move away from the frame our detection model will identify the object with any position of it across a series of frames. The basis of our tracking method taken from the Discriminative Correlation Filter Tracking with Channel and Spatial Reliability (CSR-DCF) [17] tracking algorithm. Moreover, this algorithm has been implemented in a C++ and integrated into Open CV library as a DNN (Deep Neural Networks) module [18]. 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 that shown in Fig. 11.

We have integrated CNN based object detector with OpenCV tracker where we will apply Faster RCNN based object detector model to support tracking algorithm with stable object identifier from coming real time frames. As an Object detection model have been used output of the Faster RCNN object classifier file that will support to identify objects from coming frames and gives

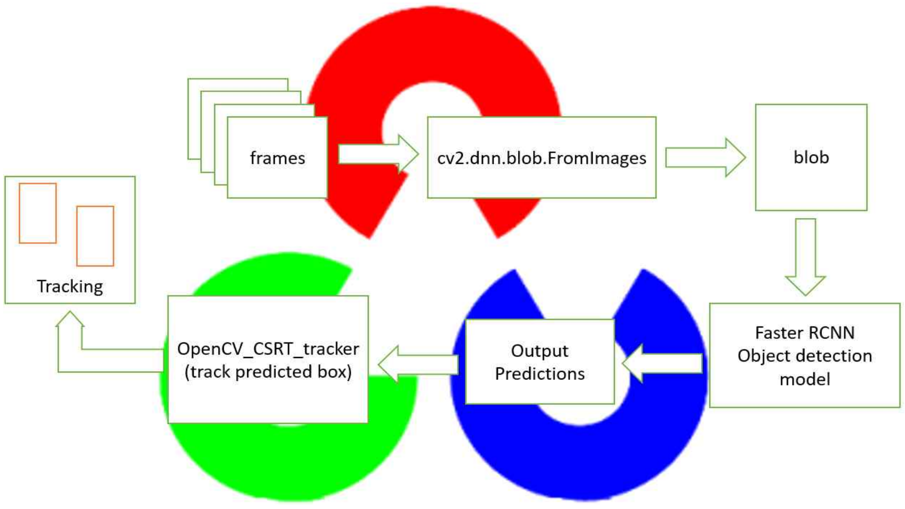
frames
cv2.dnn.blob.FromImages
blob
Faster RCNN Object detection model
Output Predictions
OpenCV_CSRT_tracker (track predicted box)
Tracking

Fig. 11. Proposed tracking method structure.

Algorithm 1: The FRCNN_CSRT integrated tracking algorithm

```
1.  Loop over frames of the sequence, and store corresponding information from each frame;
2.  Resize a frame and compute the absolute difference between the current frame and first frame;
3.  Loop over the contours identified and compute the bounding box for the contour, draw it on
      the frame, (x, y, w, h), and use tensorflow object detection model files to detect object;
4.  Extract the index of the class label from the 'detections', then compute the (x, y)-coordinates
      of the bounding box for the object;
5.  Draw the prediction on the frame and start tracker with differing initial frame and initial bounding
      box;
6.  Check to see if we are currently tracking an object, if so, ignore other boxes this code is relevant
      if we want to identify particular persons and grab the new bb coordinates of the object;
7.  Check to see if the tracking was a success and update the FPS counter;
8.  loop over the info tuples and draw them on our frame and draw the text and timestamp on the
      frame;
9.  Show the frame and record if the user presses a key.
```

output predictions to the CSRT tracker.

Our proposed integrated algorithm from taking real-time frames till tracking process pseudocode given in Algorithm 1.

To support object detection model with OpenCV tracker has been used OpenCV dnn module in the library that implements forward pass (inferencing) with deep networks, pre-trained object classifier that used one of the most popular deep learning framework TensorFlow. DNN module contains two main functions that can be used for pre-processing images and preparing them for classification via pre-trained deep learning model.

Structure of the dnn module allows to create and manipulate comprehensive artificial neural networks [18]. Neural network is presented as directed acyclic graph (DAG), where vertices are Layer instances, and edges specify relationships between layers inputs and outputs. In dnn library in each network layer given unique integer id and unique string name as well inside of the network. Moreover, layer ID can store either layer name or layer ID, also this class supports reference counting of its instances, i. e. copies point to the same instance. Inference diagram for cv::dnn::Net shown in Fig. 12 below:

The given diagram shows main structure of the dnn net where we have been applied Classification Model and Detection Model only to track objects wherewith our pre-trained object classifier. DNN module supports all type of CNN (33 different) layers. Our object detection method Faster RCNN also built on with the help of some dnn layers where OpenCV dnn module can support easily.

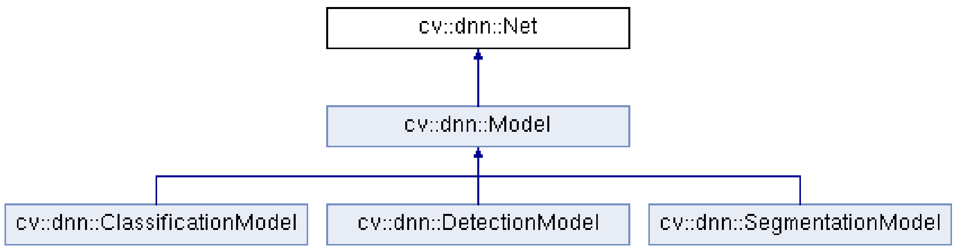

Fig. 12. Inference diagram of DNN module in OpenCV library.

Detection model of OpenCV represents high-level API for object detection networks, also allows to set parameters for preprocessing input image. Furthermore, it creates net from with trained weights and config, sets preprocessing input, runs forward pass and return result detections, also supports SSD, Faster RCNN, and YOLO topologies. Also Classification part the same with detection model, it also uses trained weights and config files, sets preprocessing input, runs forward pass and return top - 1 prediction.

OpenCV CSRT object tracker implementation is based on Discriminative Correlation Filter with Channel and Spatial Reliability [17]. The inherence diagram for CSRT tracker given in Fig. 13 below:

OpenCV library is enough big to use different kind of purposes with the support of several programming environment in an Image Processing. This tracker base abstract class for the long-term tracking and it gives stats-of-the-art performance with the integration of CNN object detection. Parameters for initialize the tracker with a known bounding box that surrounded the target image returns with true if initialization went successfully, false otherwise. While tracking process reads algorithm parameters from a file storage, where implemented in the Algorithm part of the OpenCV library. Our integrated tracking method updates the tracker with the help of pre-trained object classifier, which includes object features, bounding boxes, object shape information that to find object location in image and class as well and find new most likely bounding box for target from the current frame, which represent the new target location, if true was returned, not modified otherwise. True means that target was located and false means that tracker cannot locate target in current frame. Note, that latter does not imply that tracker has failed, maybe target is indeed missing from the frame, we may say out of sight too.

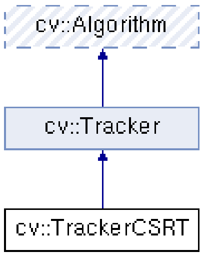

Fig. 13. Inference diagram of CSRT tracker in OpenCV library.

3. EXPERIMENT RESULTS AND DISCUSSION

We have collected our own dataset that contains images taken by drones to make an object detection model to apply for tracking process. Insofar as a main goal was to create a tracker for drone and track only two class of object. For this reason we collected our own dataset which includes two image classes: pedestrian and vehicle, where we trained our object detection model with the help of Faster RCNN object detection approach to make an object classifier. This step was urgent to create an individual object tracker rather than using open source datasets which has more classes. Furthermore, our trained object classifier dataset images only taken by drone cameras, if we apply our proposed tracking method to drones it works as expected, because of the dataset which contains drone camera based images that helps while training object classifier to learn objects location, position, shape, and other parameters of the objects. This difference of dataset's image from ordinary one may influence tracking performance massively. On account of this fact we may see the accomplishment of object classifier even with a small amount of images on dataset can show a noticeable outcome while tracking by drone.

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 Fig. 14. We can see the remarkable results of object de-

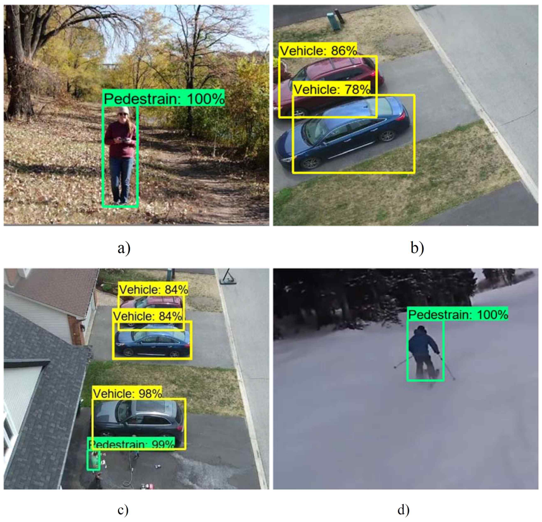

Fig. 14. Object detection results of the pre-trained Faster RCNN object classifier: (a) image from internet sources taken by drone camera; (b, c) picture taken from drone video sequence; (d) image from VOT2018 dataset.

tection 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 position, 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.

Additionally, object detection model have been tested with different environment, for instance, video sequence that based on drone camera, with real time webcam, and with mobile camera as well, result was good according to the different video environment and situation.

The main goal is in this paper to create a new and simple object tracker for drones and to be simple to set up it to hardware platform. However we suggested new tracking method by using one of the most popular object detection model integration with a tracker algorithm. Our suggested tracking algorithm to use for a new tracker is one of the most commonly used algorithm in this sphere and we have integrated detector and tracker. 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 Fig. 15 below:

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. This video sequence taken by drone after some video footage the target position on video sequence has changed totally where CSR-DCF tracker could not estimate reliability map and channel reliability (frame-2106,2480,2965,3275), even failed at all in some frame (frame-2273). In order to solve that kind of problems we have proposed solution by apply deep learning based object detector to recovering tracking process in every step of tracking.

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

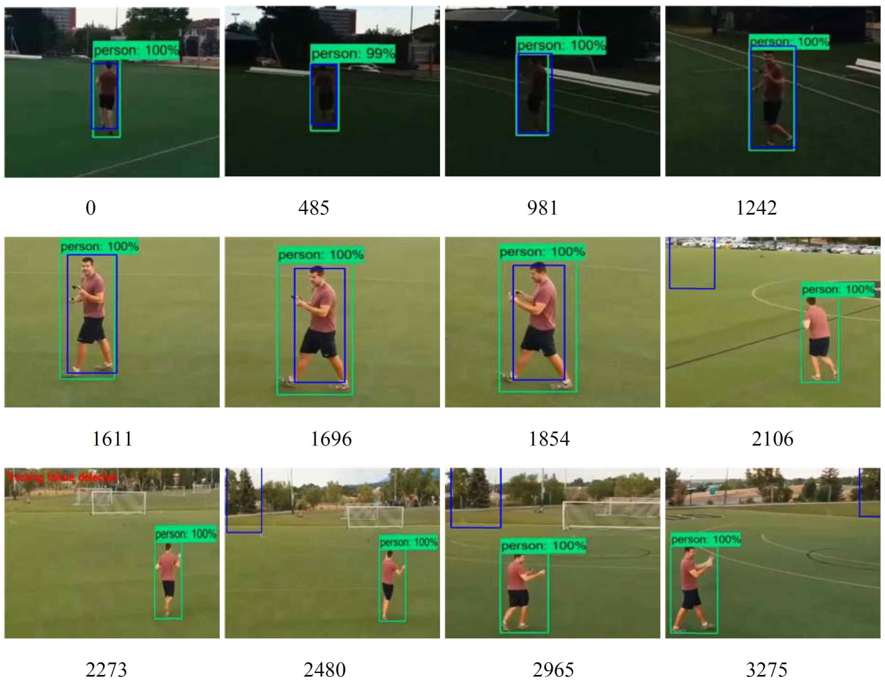

Fig. 15. Qualitative results of conventional (CSR-DCF, blue) and proposed (FRCNN_CSRT, green) trackers (0, 485, 981,⋯ – frames) .

can see the comparison result of conventional and proposed tracking methods in Table 1, where has been tested with three different open source data-sets that the main difference between two methods is deep learning based approach and real time frames per second as well.

4. CONCLUSION

The method proposed in this study proves that it can reach perfect performance in tracking process using deep learning based object detection approach that could achieve better results in several tested open source datasets. We have presented Faster RCNN training procedure only with two class object classification includes 600 images. Our proposed additional object classifier part to CSR-DCF algorithm could help to overcome problems that mentioned in abstract and through this manual script. 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. 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. Moreover, because of the drone dataset which all training and testing images taken from drone camera could hand even

Table 1. Comparison 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)

with not much pre-trained images.

The method presented here can be used by modifying for any kind of intended controlling systems or tracking purposes that can be managed easily. In the future work we would try to improve tracking accuracy and performance as well.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, Vol. 38, No. 4, pp.1-45, 2006.

[2] A. Ali, A. Jalil, J.W. Niu, X.K. Zhao, S. Rathore, J. Ahmed, et al., "Visual Object Tracking-classical and Contemporary Approaches," *Frontiers of Computer Science,* Vol. 10, No. 1, pp. 167-188, 2016.

[3] A.H.A. El-Shafie and S.E.D. Habib, "A Survey on Hardware Implementations of Visual Object Trackers," *Electrical Engineering and Systems Science, Image and Video Processing, arXiv:1711.02441,* 2017.

[4] P. Viola and M. Jones, "Robust Real-time Object Detection," *Proceeding of Second International Workshop on Statistical and Computational Theories of Vision-modeling, Learning, Computing, and Sampling,* pp.1-45, 2001.

[5] J.B. Lim and I.K. Ha, "Analysis of Drone Target Search Performance According to Environment Change," *Journal of Korea Multimedia Society*, Vol. 22, No. 10, pp. 1178-1186, 2019.

[6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 20-25, 2005.

[7] A. Krizhevsky, I. Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceeding of the 25th International Conference on Neural Information Processing Systems*, Vol. 1, pp. 1097-1105, 2012.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *Computer Vision and Pattern Recognition,* Vol. 1, No. 1, pp. 1-8, 2014.

[9] R. Girshick, "Fast R-CNN," *Computer Vision and Pattern Recognition,* Vol. 1, No. 1, pp.1-9, 2015.

[10] R. Girshick, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Computer Vision and Pattern Recognition,* Vol. 1, No. 1, pp. 1-14, 2016.

[11] Object Detection, https://en.m.wikipedia.org/wiki/Object_detection, Oct. 2018.

[12] S. Phon-Amnuaisuk, K.T. Murata, P. Pavarangkoon, K. Yamamoto, and T. Mizuhara, "Exploring the Applications of Faster R-CNN and Single-shot Multi-box Detection in a Smart Nursery Domain," *Computer Vision and Pattern Recognition, arXiv:1808.08675v1,* 2018.

[13] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Pearson(4 edition), 2017.

[14] Faster RCNN, https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4, Nov. 2018.

[15] Region Proposal Network (RPN)-Backbone of Faster RCNN, https://medium.com/egen/ region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9, Aug. 2018.

[16] Understanding the Impact of Learning Rate on Neural Network Performance, https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/, Jan. 2019.

[17] A. Lukezic, T. Vojır, L.C. Zajc, J. Matas, and M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability," *International Journal of Computer Vision:* Vol. 126, Issue 7, pp. 671-688, 2018.

[18] OpenCV Dnn Module, Net Class Reference, https://docs.opencv.org/master/db/ d30/classcv _1_1dnn_1_1Net.html, Nov. 2018.

Farkhodov Khurshedjon Furqat o'g'li

He received the B.S. degree Computer Engineering Tashkent University of Information Technologies, Uzbekistan in 2017. He is currently a Master student in the department of IT Convertgence and Application Engineering in Pukyong National University. His research interests include Digital Image Processing and Machine Learning.

Oh-Heum Kwon

received the B.S. degree in computer engineering from Seoul National University in 1988, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology in 1991 and 1996 respectively. He is currently a professor in the Dept. of IT Convergence and Application Engineering at Pukyong National University, Busan, and Republic of Korea. His research interests include design and analysis of algorithms and distributed computing.

Oh-Jun Kwon

He received a B.S. in Electronic Engineering from Kyungpook National University, in 1986, an M.S. in Computer Science from Chungnam National University in 1992 and a Ph.D. in Computer Science from Pohang University of Science and Technology (POSTECH), Korea in 1998. From Jan. 1986 to Feb. 2000, he worked for the Electronic and Telecommunication Research Institute (ETRI). Since 2000, he has been a Professor of Computer and Software Engineering Department at Dongeui University. His research interests include a computer network, computer security, wireless internet, pattern recognition and machine learning/deep learning.

Kwang-Seok Moon

received the B.S., M.S., and Ph. D degrees in Electronics Engineering in Kyungpook National University, Korea in 1979, 1981, and 1989 respectively. He is currently a professor in department of Electronic Engineering at Pukyong National University. His research interests include digital image processing, video watermarking, and multimedia communication.

Suk-Hwan Lee

He received a B.S., a M.S., and a Ph. D. degrees in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively . He is currently an associate professor in Department of Information Security at Tongmyong University. His research interests include multimedia security, digital image processing, and computer graphics.

Ki-Ryong Kwon

He received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986-1988 and at Pusan University of Foreign Language from 1996-2006. He is currently a professor in Department of IT Convertgence and Application Engineering at the Pukyong National University. He has researched University of Minnesota in USA in 2000-2002 with Post-Doc. and Colorado State University on 2011-2012 with visiting professor. He was the President of Korea Multimedia Society in 2015-2016. His research interests are in the area of digital image processing, multimedia security and watermarking, bioinformatics, weather radar information processing, and machine learning.