IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 4, 2021, accepted July 21, 2021, date of publication July 28, 2021, date of current version August 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101054

Automatic Object Tracking and Segmentation Using Unsupervised SiamMask

SHAHEENA NOOR[1], (Member, IEEE), MARIA WAQAS[2], MUHAMMAD IMRAN SALEEM[1], (Member, IEEE), AND HUMERA NOOR MINHAS[3]

[1]Computer Engineering Department, Sir Syed University of Engineering and Technology, Karachi 75300, Pakistan
[2]Department of Computer and Information Systems Engineering, NED University of Engineering and Technology, Karachi 75270, Pakistan
[3]Machine Learning Engineering Group, Eyeo GmbH, 50825 Cologne, Germany

Corresponding author: Shaheena Noor (shanoor@ssuet.edu.pk)

ABSTRACT In this paper we address the basic limitation of SiamMask - the state of the art single object tracking and segmentation algorithm. SiamMask requires semi-supervision in that it needs a bounding box to be drawn manually around the object that has to be tracked. This is however not always possible or feasible, and slows down the pipeline even in the best case. We overcome this limitation by using state-of-the-art object detection algorithms: Detectron2 and YOLO to automatically detect the object and then track using SiamMask. The main purpose of this study is to devise an efficient technique for an end-to-end object detection and tracking, which can then be used in other applications like self-driving cars, etc. We compared different approaches using current state-of-the-art tools for time and detection efficiency. One of the secondary aim was to test how the two approaches perform on different types of datasets. We note that YOLO gives better and more meaningful detection of objects in the scene. However, Detectron2 gives a higher detection speed than YOLO, making the overall detection and tracking process faster.

INDEX TERMS Object tracking, object detection, video segmentation, unsupervised learning, deep learning, YOLO, detectron2, SiamMask.

I. INTRODUCTION

Single Object Tracking is widely used in a number of real-world applications like surveillance and security, sports analysis, medical analysis, human-computer interaction, human activity recognition and spaceship tracking etc.

Object tracking is often regarded as a detection problem, applied repeatedly and independently across multiple frames, however this approach is neither reliable nor practical in real-time. Similarly, work done on video object segmentation is approached from the point of view of semantic segmentation and focus is placed on correct classification of pixels while losing the bigger context.

More recently, attention has been given to combine two of the three different-yet-similar problem areas of object detection, tracking and segmentation. The state-of-the-art of such a combination is SiamMask [1], where the authors based their work on Siamese Network and proposed a simplistic approach to perform single object tracking and segmentation in single-shot. They achieved realtime speed and surpassed the existing state-of-the-art benchmarks, forming the basis for further adaptations of their system. However, SiamMask suffer from its inherent dependency on semi-supervised annotation for getting started. We solve this problem by plugging the state-of-the-art detectors including YOLOv4 [2] and Detectron2 [3]. We detect the objects in the first phase and pass them on to the SiamMask in the subsequent phase for tracking and segmentation. Thus, we achieve an automated three-in-one solution for detection, tracking and segmentation. This approach allows us to identify the dominant objects in the scene automatically and track them further.

The paper is arranged as follows: In Section II we cover the background work on object detection, tracking and segmentation in detail. Next, in Sections III and IV we discuss the theoretical and mathematical foundations of SiamMask, YOLOv4 and Detectron2. We give an overview to the datasets, experiments and results in Section V and conclude the paper in Section VI.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

II. LITERATURE REVIEW

An end-to-end tracking system essentially comprises of two major components: object detection and object tracking. Thus, it is important to note the advancements in each of these categories independently.

A. OBJECT DETECTION

Traditional object detection and recognition systems have been around since 1990s. The typical approach here is to extract features using SIFT [4] or SURF [5] and then use them to train a classification model like SVM [6] or ANN [7]. However, more recent advancements in deep learning have potentially replaced the previous approaches, providing increased accuracy with low effort.

The R-CNN i.e. Regions with CNN Features [8] was a breakthrough for object detection and semantic segmentation. It comprised of three modules: search for objects, CNN for feature extraction, SVM for classification. R-CNN is slow and not very accurate in itself, however it provided a basis for future work like Fast R-CNN [9], Faster R-CNN [10] and FAIR's (Facebook Artificial Intelligence Research group) Mask R-CNN [11], all of which focussed more on improving the accuracy of the detector with a compromise on speed.

Liu *et al.* [2] presented YOLO-You Only Look Once switching the focus on speed from accuracy. YOLO works in near realtime and maintains reasonable accuracy. It is followed by further versions YOLOv2, YOLOv3, YOLOv4 and YOLOv5 which perform better (or worse) in specific areas. E.g. YOLOv3 works accurately on smaller objects, but performs worse on larger objects etc. The different versions of YOLO use CNNs of various sizes and return bounding boxes around objects in the frame. It uses a single feed forward propagation across the network to detect the objects and multiple independent logistic classifiers instead of softmax for class prediction, which are trained with binary cross-entropy loss. An alternate to YOLO is a Single Shot MultiBox Detector (SSD) [12] that also detects objects in a single forward propagation. The original SSD is based on the VGG-16 [13] backbone, however subsequent versions are based on backbones like ResNet [14], Inception [15] and MobileNet [16]. Lin *et al.* from FAIR in [17] presented RetinaNet - the solution to the reduced-accuracy problem of single feed forward-based approaches. They explored the reason behind loss of accuracy in such algorithms and noted that it is created because of the way data is generated from labeled set for training, creating class imbalance between positive and negative labels. They used a sparse dataset for training and achieved real-time speed with accuracy surpassing the state of the art. Subsequently, FAIR released Detectron2 [3] for object detection and segmentation.

B. OBJECT TRACKING

The traditional object tracking algorithms are still popular and used as the basis for current research. These include Mean Shift [18], Kalman filters [19], sparse and dense optical flow [20] etc. and are essentially regarded as following interest points over spatio-temporal dimension [21].

Tracking algorithms still face a number of challenges like distortion and deformation, changes in lighting, motion blur, clutter and occlusion to name a few [22]. Moreover, for practical applications, it is important that the tracking is not only accurate, but also real-time [23]. The recent trackers based on deep features achieve a higher accuracy, however they are extremely slow resulting in frames being dropped when put in real-time conditions. Also, they need huge datasets for training, because the deeper networks have more neurons, layers and weights. Hence, work is being done to increase accuracy while maintaining speed. Shin *et al.* [23] proposed a modified kernalized correlation filter (KCF) that identifies tracking failure and tries to resume tracking by searching over multiple windows. In another work, Zhang *et al.* [24] attempted to combine multi-object detection and re-identification in a single network to increase speed. The main focus of this work is detecting humans in crowded scenes. They pretrain their model (called as FairMOT) on the CrowdHuman dataset [25] using self-supervised learning and used a 4-channel approach (left, top, right, bottom) to detect bounding boxes instead of the traditional 2-channel one (width, height). They were able to achieve state of the art performance on the Multi-Object Tracking (MOT) datasets at 30 FPS.

Wojke *et al.* in [26] introduced DeepSORT - a modification of Simple Online Realtime Tracking (SORT) algorithm with deep metric learning. By using a cosine-based metric, they were able to track objects over longer durations overcoming occlusions and switches in identity.

In certain applications like robotics, it is required to grasp object, and for that purpose an object needs to be identified. To achieve this, in [27], the authors uses Detectron2 R-CNN (Regional Convolutional Neural Network) to make the mask for the object. They used the Cornell Grasping dataset to predict an optimal rectangle for grasping a particular object. Owing to the incredibly small sizes of the subjects in contrast to the background and the steep angle of inclination of the sensor, the Detectron2 effectively masks the subject (around 60% of the dataset). In another work called TrackR-CNN, Voigtlaender *et al.* [28] used semi-supervised annotation for multi-object tracking and segmentation. They performed object detection using the R-CNN mask over ResNet-101 backbone. Then created 3D convolutions to include the temporal component.

Some work is done more specifically to track fast moving objects. Such objects face additional challenges due to motion blur and changes in appearance, but are very common in scenarios like sports analysis. He *et al.* [22] proposed to use double Gaussian probability model which assumes that the velocities between adjacent frames are not correlated. Thus, they based their tracking algorithm on motion consistency.

Inspired by the bounding box regression methods for object detection, Ning *et al.* [29] proposed a combination of object detection and recurrent neural networks. They used YOLO for collecting spatial features and Long Short-Term

Memory (LSTM) [30] for handling temporal information - thus naming the approach as Recurrent YOLO (ROLO).

In [1] the authors used SiamMask in real time environment for object tracking and video object segmentation. Semi-supervised technique is used for object segmentation. Object segmentation performed on DAVIS-2016 and DAVIS-2017 dataset while object tracking was performed on the VOT2016 and VOT2018 and showed results better or comparable to state of the art algorithms. For Single Object Tracking (SOT), SiamMask is regarded as the best choice and forms the basis of a lot of future work. It uses fully convolutional Siamese Network for offline training and compensates for the losses in the previous approaches through binary segmentation. A trained SiamMask depends only on an initial bounding box and works independently generating object mask and tracking it in real-time @ 55 FPS (frames per second). Other notable work based on siamese networks is DaSiamRPN [31] and SiamDW [32].

In [33], the author proposed S-Siam framework in real time environment. In this paper they mention that the tracking of a particular object is lost when the camera jitters specially when the object is small in size and moving very fast. The experiments were conducted on VOT2016, VOT2018 and VOT2019 datasets and achieve an EAO score of 0.449 and give 10% improvement when compared with other trackers.

More recently, Zhou and Koltun *et al.* [21] took a step back from the increasingly deep learning-based approaches and applied a simplistic detection model to a pair of images and detections from previous frame. They used this input to form associations within adjacent frames and achieved real-time performance beaating the state of the art accuracy for MOT. They regard their approach as tracking objects as points and do not depend on tracking-by-detection.

In [34], the multiple object tracking is used in urban traffic environment. The detection is done using YOLOv3 and then the tracking is done using DeepSort algorithms. Urban Tracker dataset is used for experimentation and they have achieved a precision of 0.8989 and accuracy of 0.4265.

In [35] the DeepSort framework is used to track people in crowd survellience in real time environment. People detection is done using YOLOv3 and then apply Deep SORT to process frame by frame of the detected person to predict its motion path. In their work, they used 3 versions YOLOv3, YOLOv3 tiny and YOLOv3 custom with diversified in weight, file size and object class.

In [36] the author proposed Siamese Network and Optical Flow which is obtained from Kalman filter for realtime multi object tracking system. The efficiency of their method is tested on MOT13 bench mark. They also claim that the performance of their algorithm is much better than DeepSort Algorithm.

In [37], they idenified detection of pedestrian and identification of pedestrian behavior for autonomous driving. This is done to reduce road accidents. YOLOv3 TINY is used to identify pedestrian and for behvior of pedestrian they used Deepsort algorithm. Later on, Alexnet was proposed for behavior identification. They compared their result with similar algorithm and achieve better accuracy in real time environment.

In [38], the authors use multiple camera to identify activities. They uses the VIRAT V1 dataset to track and detect different activities using Detectron2 framework.

In [39], the author proposed Discriminative Single-Shot Segmentation (D3S) for object tracking and video object segmentation. there are two main advantages of using this technique. First it is invariant to wide spectrum of transformation that includes non-rigid transformation. Second, the rigid object aims to achieve high robustness and online target segmentation at the same time. They conducted their eperiments on VOT2016, VOT2018 and GOT-10k. Moreover they compare their results with TrackingNet dataset and achieve good result.

In [40], the author considered single object segmentation in a video. They used first frame and generate a bounding box over a particular object. Next they used Box2segmentation module to obtain segmentation in the corresponding frames that is based on the first predicted bounding box. Experiments were conducted on DAVIS2016 dataset and achieved an accuracy of 73.1%.

In [41] the authors used a quick and efficient system for instance segmentation that also performs well on bounding box identification and can be expanded to include pose estimation. They use Mask R-CNN on COCO dataset for predicting an object mask in addition to the current branch for bounding box recognition.

III. OBJECT TRACKING AND SEGMENTATION USING SiamMask

To allow online operability and fast speed, SiamMask is based on the fully convolutional Siamese framework. The approach does not rely on the specific fully convolutional method at startup, which is contrary to the previous approaches of SiamFC and SiamRPN. Figure 1 demonstrates the foundational blocks of the tracking system. It is basically an offline trained network. In the Figure, z represents the $width(w) \times height(h)$ crop with focussed on the target object, while x is a crop, larger in size, focussed on the target object's last estimated position. A common CNN processes both x and z inputs to generate respective feature maps, which are then compared to produce a dense response map. This is shown in Equation (1).

$$g_{\Phi}(z, x) = f_{\Phi}(z) * f_{\Phi}(x) \tag{1}$$

In a response map, Response Of a candidate Window (ROW) is the term used to refer each spatial element. For example, $g_{\Phi}{}^{n}$ would encode a similarity between the sample z and the n-th frame in x. SiamMask performs a depth-wise cross correlation to generate a multi-channel response map. This is an enhancement over SiamFC which used simple cross correlation. Several million video frames were used for offline training of SiamFC, producing a logistic loss

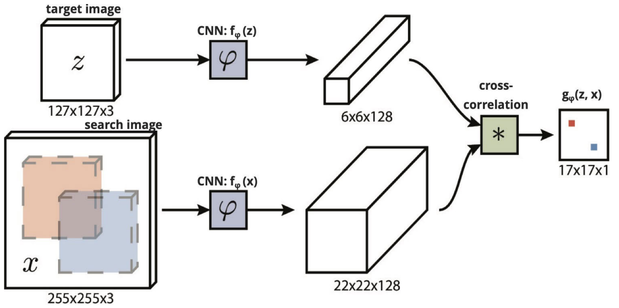


FIGURE 1. Fundamental building block of the tracking system.

as L_{sim}. Later its performance was upgraded by employing Region Proposal Network (RPN) which made it capable of drawing a bounding box of variable aspect ratio around the target location. SianRPM, on the other hand, outputs two parameters: box predictions as L_{box} and classification scores as L_{score}. In SiamMask, the authors point out that additional information can be encoded in ROW produced by a fully convolutional Siamese network, to generate a pixel-wise binary map. This is done using a simple two-layer neural network h_Θ, which produces a w × h binary mask for each ROW. Consider Eq. (2) in which the m_n denotes the predicted mask corresponding to the n-th response window. And the Loss function L_{mask} is for the prediction which is done via binary logistic regression loss over all ROWs.

$$m_n = h_\Theta(g^n_\Phi(z, x)) \tag{2}$$

Consider Figure 2a (adapted from [1]) that explains this process. On the left, we see the fully convolutional neural network that generates the Response of candidate Window, and on the right, we have the neural network for mask prediction. Based on these calculations, two variants of SiamMask are proposed: one combines the mask with RPN's paramters L_{box} and L_{score} (Eq. (3)); and the other one combines the mask with L_{sim} from SiamFC (Eq. (4)). This is shown in Fig 2a (adapted from [1]).

$$L_{3B} = \lambda_1 L_{mask} + \lambda_2 L_{score} + \lambda_3 L_{box} \tag{3}$$

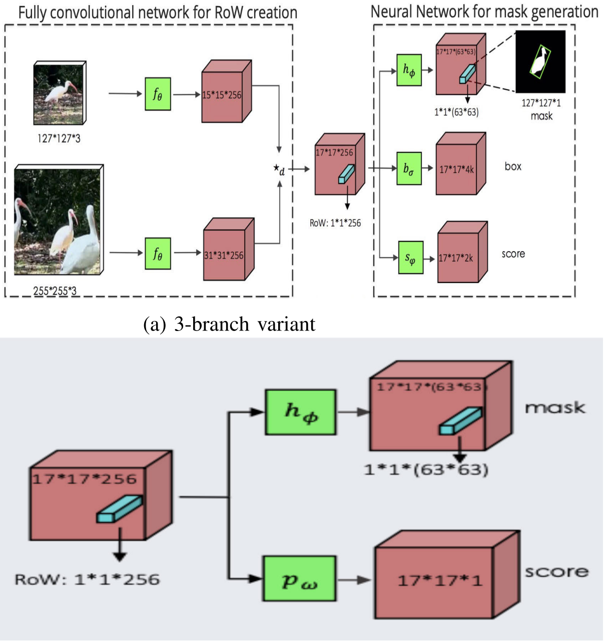
$$L_{2B} = \lambda_1 L_{mask} + \lambda_2 L_{sim} \tag{4}$$

IV. OBJECT DETECTION

A. YOLO

YOLO (You Only Look Once) is one of the first deep-learning based object detection approaches with focus on speed rather than accuracy. Hence, it has gained wide-spread use in near real-time applications with a loss of accuraacy.

YOLO is based on a single forward propagation feed to detect the objects in an image. It regards detection as a regression problem instead of the traditional classification problem. The individual components of object detection are unified into a single neural network, which uses features of the complete image to predict the bounding box. This enables high speed training and fast predictions.



(a) 3-branch variant

(b) 2-branch variant

FIGURE 2. Workflow of SiamMask.

In YOLO, the image is divided into a grid. Next, it is noted that in which grid cell a particular object falls. It becomes the responsibility of that grid cell to identify that object. For this, bounding boxes are predicted by each grid cell and their respective confidence scores. It also predicts the conditionaal class probabilities. Consider [2, Fig. 3] that explains the working process of YOLO.

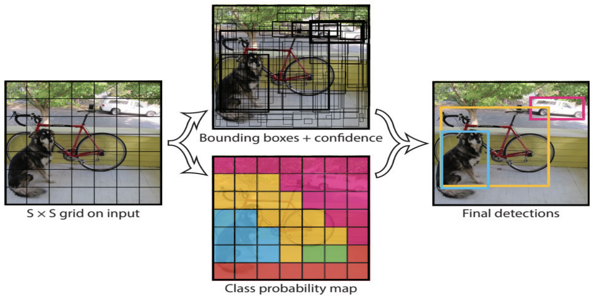


FIGURE 3. YOLO Model as explained in [2]: The image is divided into grids and for each cell bounding boxes are predicted alongwith the confidence scores and class probabilities.

The network of the intial YOLO version comprises of 24 convolutional layers, followed by 2 fully connected layers. A faster variant of YOLO used 9 layers instead of the original 24. This allowed to further increase the speed with compromised accuracy. Subsequently, more variations of YOLO were proposed with varying number of layers, each working better on one aspect, while losing on another. For example, YOLOv3 (with 53-layer CNN) works better for smaller objects, but worse on larger ones.

B. DETECTRON2

Detectron2 [3] is a multi-purpose library from Facebook that provides state-of-the-art object detection. It is based on PyTorch and aimed to perform high-speed training. It is written in a modular form to allow future work to be based on this implementation.

Detectron2 includes a number of object detection models like Faster R-CNN, DensePose, Cascade R-CNN, Mask R-CNN etc. It allows object detection and marking with bounding boxes, human pose estimation and segmentation masks. We used the default predictor, COCO-InstanceSegmentation and Mask R-CNN with score threshold value `(MODEL.ROI_HEADS.SCORE_THRESH_TEST)` set to 0.5 for our experiments. A sample of Detectron2 features is shown in [3, Fig. 4].

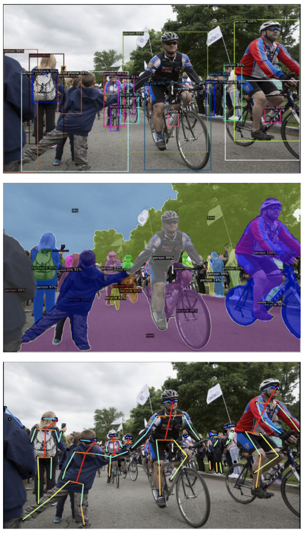


FIGURE 4. Detectron2: (Top) object detection and marking bounding boxes (Center) object segmentation (Bottom) human pose estimation.

V. EXPERIMENTS AND RESULTS

In this section, we present an overview to the dataset, experiments conducted and the results.

A. DATASET

To build and test our approach we collected a set of videos from the web. In addition, to have a comparison with on the standard datasets, we tested our approach on the YouTube-VOS [42] and Visual Object Tracking (VOT2020) [43] datasets.

YouTube-VOS is a large-scale dataset with over 340 minutes of videos for object segmentation tasks. It comprises of 4K+ YouTube videos in high resolution, over 90 semantic categories, over 7800 unique objects and almost 200K manual annotations of high quality. It was first released in 2018, and is widely used to benchmark solutions. The videos in YouTube-VOS are short with distinct objects and not much motion.

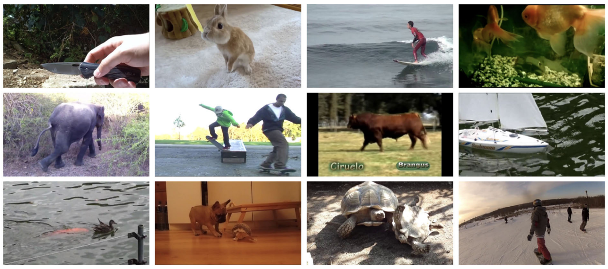


FIGURE 5. Subset of images from YouTube-VOS dataset.

The **VOT2020** benchmark set is focussed on short- and long- term tracking in addition to RGB-, RGBT- and RGBD-based tracking. It also has a section on real-time tracking. A sample of dataset images is shown in Figure 6.

FIGURE 6. Subset of images from VOT dataset.

B. EXPERIMENTS

We divide our pipeline into four major components: pre-processing, object detection, object tracking/segmentation and post-processing. We ran our experiments on Intel(R) Xeon(R) CPU @ 2.30GHz machine with 2 cores. We used Tesla K80 GPU with 3.7 computation capability and CUDA version 11.2. We had 13GB of available RAM and 30GB of disk space. We were able to automate the complete detection-tracking-segmentation pipeline end-to-end without requiring any human intervention. In this section we present the details of the pipeline automation and in the next, compare two variants i.e. Detectron2- and YOLOv4- SiamMask combination w.r.t. accuracy and speed.

We begin with dividing the video into component frames and feed the first frame into the detector. The detector returns the list of objects detected alongwith their confidence scores and coordinates. We pick up the dominant object automatically and feed it to the SiamMask, where the tracking and segmentation is triggered. This is represented in Figure 7 with logical flow explained below:

1) PRE-PROCESSING

Preprocessing comprises of:

- setting up imports
- cloning and installing the detectors (Detectron2 and YOLOv4)

- cloning and installing the tracker/segmentor (SiamMask)
- copying/loading weights from pre-trained models
- preparing input data including extracting frames from video sequences
- parameter/file initializations

2) DETECTION

Running the detector (Detectron2 and YOLOv4) entails a number of steps and activities as follows:

- apply detector on first frame
- identify objects with confidence scores
- store stats including video ID, number of objects and object coordinates
- repeat detection 5 times and note the execution speed

3) TRACKING & SEGMENTATION

The output of detector is fed to the tracker (SiamMask) as follows:

- pick the object with highest confidence score
- run tracker on frames 2 till n (where n is the last frame in the sequence and corresponds to 100 in VOT and 20 in YouTubeVOS)

4) POSTPROCESSING

Necessary cleanup is done after the functions have been executed.

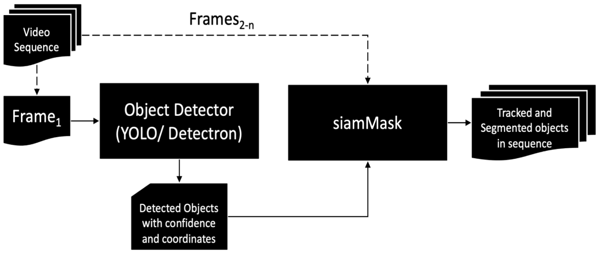

Video Sequence
Frames$_{2\text{-}n}$
Frame$_1$
Object Detector (YOLO/ Detectron)
siamMask
Tracked and Segmented objects in sequence
Detected Objects with confidence and coordinates

FIGURE 7. Workflow of pipeline automation for object detection followed by unsupervised tracking and segmentation.

C. RESULTS

We conducted tests on YouTube-VOS [42] and VOT [43] and show that we were able to execute the SiamMask segmentation and tracking autonomously without compromising its accuracy or speed. We compared the performance of Detectron2 and YOLO-v4 by applying them end-to-end automatedly without requiring any human intervention. Given below are details on the experimental outcomes.

1) AUTOMATED PIPELINE

Consider Figure 8, where we show the output of the detection and tracking/segmentation stages applied on video sequences. In each case the first frame indicates all the objects identified, and the rest of the frames indicate the tracking of the object with maximum confidence.

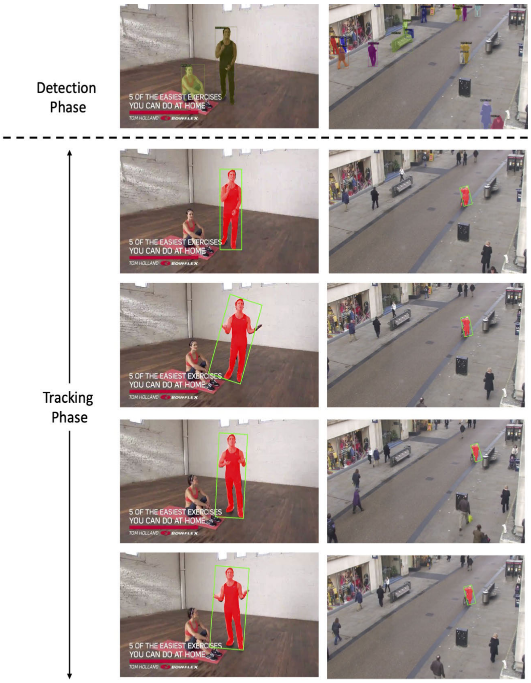


FIGURE 8. Automated detection and tracking applied to video sequences.

2) ACCURACY

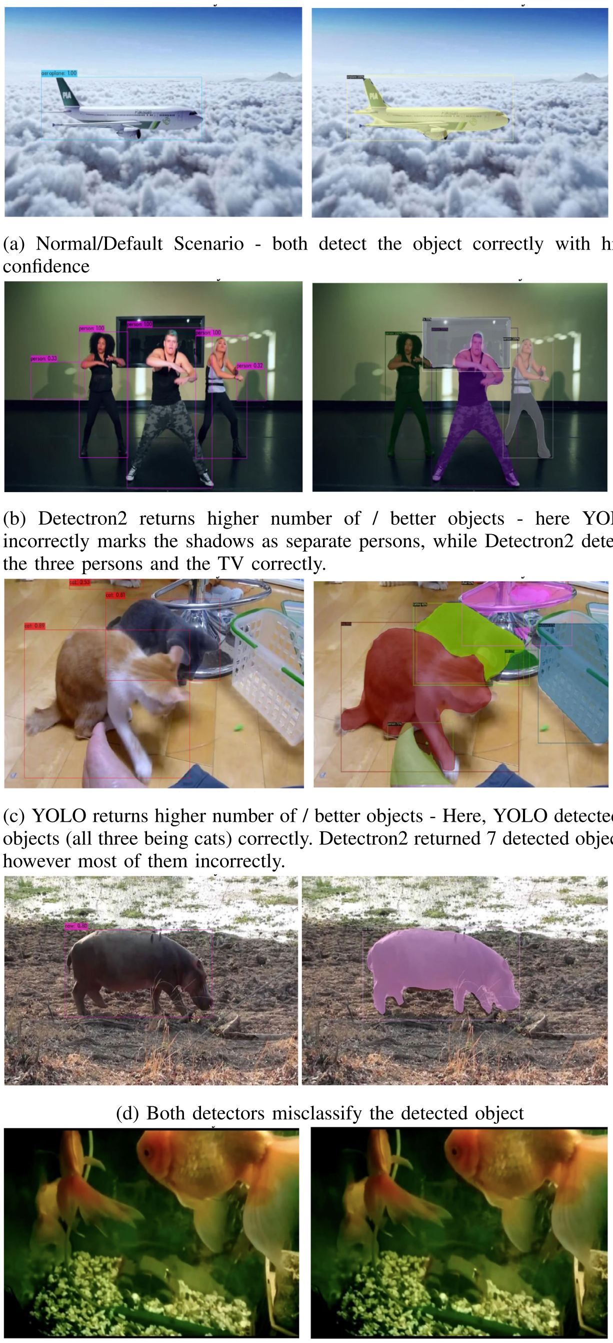
We compared YOLO and Detectron2 by applying both the detectors on our own collection of videos as well as standard YouTubeVOS and VOT dataasets. We note that the number of objects detected by the two detectors is highly dependent on the type of objects in the scene. Each of Detectron2 and YOLO have their strengths and weaknesses, however overall we noticed that YOLO returned more semantically meaningful results compared to Detectron2. Consider Figure 9 where we show object detection results for the following cases:

a: NORMAL/DEFAULT SCENARIO

Where both detectors perform equally well and detect the same or equally meaningful objects (Figure 9a)

b: DETECTRON RETURNS MORE/BETTER OBJECTS

Where either Detectron2 detects higher number of objects compared to YOLO or more meaningful ones (Figure 9b). Note that returning higher number of objects is not always an indication of better performance. However, in many cases even with overlapping objects, Detectron2 detects them separately, while YOLO identifies them as single. This distinction allows us to track each object independently, and increases accuracy.



(a) Normal/Default Scenario - both detect the object correctly with high confidence

(b) Detectron2 returns higher number of / better objects - here YOLO incorrectly marks the shadows as separate persons, while Detectron2 detects the three persons and the TV correctly.

(c) YOLO returns higher number of / better objects - Here, YOLO detected 3 objects (all three being cats) correctly. Detectron2 returned 7 detected objects, however most of them incorrectly.

(d) Both detectors misclassify the detected object

(e) Both detectors fail to detect any object

FIGURE 9. Object detection via YOLOv4 (Left) vs Detectron2 (Right).

TABLE 1. Comparison of YOLOv4 and Detectron2 in terms of number of objects detected.

		Number of objects detected	
Video #	Dataset	YOLOv4	Detectron2
V1	General	2	2
V2	General	2	4
V3	General	1	1
V4	General	17	17
V5	VOT	9	12
V6	VOT	15	12
V7	VOT	1	7
V8	VOT	8	7
V9	VOT	1	3
V10	VOT	1	1
V11	VOT	13	20
V12	VOT	11	12
V13	VOT	26	21
V14	VOT	5	4
V15	ytb_vos	3	7
V16	ytb_vos	1	1
V17	ytb_vos	2	1
V18	ytb_vos	3	3
V19	ytb_vos	1	2
V20	ytb_vos	2	2
V21	ytb_vos	0	3
V22	ytb_vos	0	0
V23	ytb_vos	1	1
V24	ytb_vos	4	2
V25	ytb_vos	1	1
V26	ytb_vos	1	1
V27	ytb_vos	2	1
V28	ytb_vos	1	3
V29	ytb_vos	2	4
V30	ytb_vos	2	1
V31	ytb_vos	3	6
V32	ytb_vos	14	11
V33	ytb_vos	0	2
V34	ytb_vos	2	3

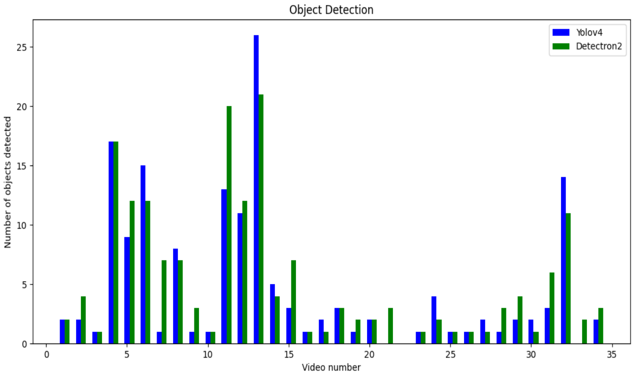


FIGURE 10. Comparison of objects detected by Detectron2 and YOLOv4.

c: YOLO RETURNS MORE/BETTER OBJECTS

Where YOLO performs better in terms of number and quality of detected objects (Figure 9c). Again, it is important to note that a higher number of detected objects does not mean a better result, as can be seen in this Figure.

d: BOTH DETECTORS FAIL

Where either both detectors fail to recognize any object or return false detections (Figures 9d and 9e). For example in Figure 9d, both detected one object (hippo), but classified incorrectly (cow). In Figure 9e, both failed to detect any object although fish is easily identifiable.

The comparative results are shown in Table 1 and Figure 10. Comparing the number of objects detected via the two detectors, in 32.35% cases, both detected same number of objects; in 41.67% cases Detectron2 detected more objects than YOLO and in 26.47% cases YOLO returns

TABLE 2. Comparison of YOLOv4 and Detectron2 in terms of time taken to run the combination with SiamMask.

Video #	Dataset	Best run time (sec) YOLOv4-SiamMask	 Detectron2-SiamMask	Speedup	
V1	General	12.53	6.43	1.95	
V2	General	16.45	9.19	1.79	average speedup 1.75
V3	General	16.6	9.66	1.7	
V4	General	16.17	9.94	1.63	
V5	VOT	9.27	3.26	2.84	
V6	VOT	8.99	3.01	2.99	
V7	VOT	9.87	3.6	2.74	
V8	VOT	10.05	3.62	2.78	
V9	VOT	10.7	4.43	2.42	average speedup 2.43
V10	VOT	10.51	4.23	2.48	
V11	VOT	10.23	4.15	2.47	
V12	VOT	11.26	5.29	2.13	
V13	VOT	13.19	7.05	1.87	
V14	VOT	19.63	12.11	1.62	
V15	ytb_vos	7.17	2.73	2.63	
V16	ytb_vos	6.85	3.29	2.08	
V17	ytb_vos	6.48	3.03	2.14	
V18	ytb_vos	6.06	2.66	2.28	
V19	ytb_vos	6.42	2.72	2.36	
V20	ytb_vos	6.11	2.63	2.32	
V21	ytb_vos	4.29	2.87	1.49	
V22	ytb_vos	4.3	1.31	3.28	
V23	ytb_vos	6.68	2.8	2.39	
V24	ytb_vos	6.13	2.47	2.48	average speedup 2.28
V25	ytb_vos	7.08	3.22	2.2	
V26	ytb_vos	6.56	2.94	2.23	
V27	ytb_vos	6.67	3.02	2.21	
V28	ytb_vos	6.24	2.59	2.41	
V29	ytb_vos	6.18	2.59	2.41	
V30	ytb_vos	6.8	2.78	2.39	
V31	ytb_vos	6.55	2.72	2.45	
V32	ytb_vos	6.21	2.67	2.33	
V33	ytb_vos	4.41	2.66	1.66	
V34	ytb_vos	6.27	3.46	1.81	

higher number of detected objects. However, as noted above, YOLO returns more meaningful detections compared to Detectron.

3) SPEED

Next, we compare YOLO and Detectron2 w.r.t. the execution times. Here, we note that Detectron2 is overall faster than YOLO. Our experiments show that on average, Detectron2 is 2.15 times faster comparatively. This makes it more plausible to be used in conjunction with SiamMask, which works in real-time as well. Consider Table 2, where we show the best runtimes out of 5 runs for each of Detectron2- and YOLOv4- SiamMask combination. We note that for all three datasets of random/general video sequences, VOT and YouTube VOS the speedup of Detectron2 over YOLO is more than 1.

The speed comparative results are shown in Figure 11. As can be seen, the execution time of Detectron2 is less than that of YOLO. We found that it is on average 2.15 times faster.

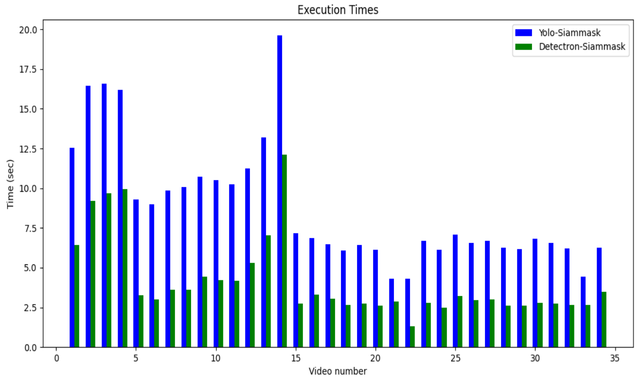


FIGURE 11. Comparison of execution times (in sec) of Detectron2- and YOLOv4- SiamMask combination.

Consider Figure 12, where we show the speedup of Detectron2 over YOLO for the three datasets separately.

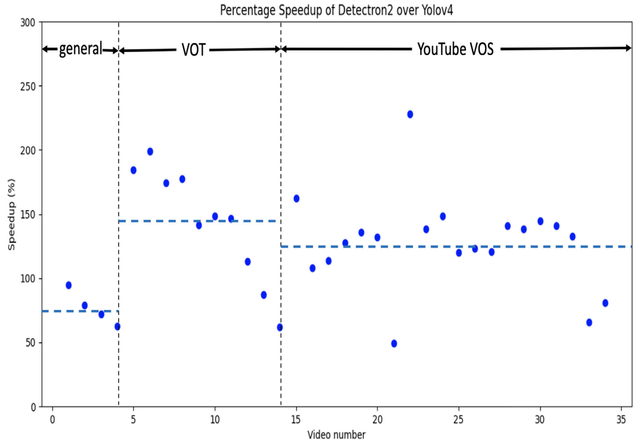


FIGURE 12. Speedup comparison of Detectron2- and YOLOv4- SiamMask combination.

VI. CONCLUSION

SiamMask is a state-of-the-art single object tracking and video segmentation algorithm that achieves real-time speed and performance gains. However, SiamMask suffers from the basic limitation of requiring a manually marked bounding box around the object to be tracked. We overcome this limitation by pre-appending the state-of-the-art object detection algorithms to SiamMask. We used Detectron2 and YOLOv4, and showed that the approach is detector-agnostic and works well with both combinations. We noticed that YOLO detects more meaningful objects compared Detectron2, however Detectron2 is 2.15 times faster. In terms of tightened bounding boxes, both algorithms work fine and based on input video give interchangeably good results. A tighter box works better with SiamMask, because in this case the object is clearly separate from the background. With this work, we lay foundation for a fully autonomous detector-tracking-segmentation pipeline using current state-of-the-art approaches and remove the requirement of semi-supervision.

REFERENCES

[1] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, ''Fast online object tracking and segmentation: A unifying approach,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338, doi: 10.1109/CVPR.2019.00142.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[3] W. Y. Yuxin, A. Kirillov, F. Massa, L. Wan-Yen, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[4] D. G. Lowe, ''Distinctive image features from scale-invariant keypoints,'' *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, ''SURF: Speeded up robust features,'' in *Proc. 9th Eur. Conf. Comput. Vis.*, May 2006, pp. 404–417.

[6] N. Cristianini and E. Ricci, ''Support vector machines,'' in *Encyclopedia of Algorithms*, M. Y. Kao, Ed. Boston, MA, USA: Springer, 2008, doi: 10.1007/978-0-387-30162-4_415.

[7] S. Kustrin and R. Beresford, ''Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research,'' *J. Pharmaceutical Biomed. Anal.*, vol. 22, no. 5, pp. 717–727, 2000, doi: 10.1016/S0731-7085(99)00272-1.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[9] R. Girshick, ''Fast R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[10] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot MultiBox detector,'' in *Computer Vision—ECCV* (Lecture Notes in Computer Science). Springer, 2016, pp. 21–37.

[13] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. Comput. Vis. Pattern Recognit.*, Sep. 2015, pp. 1–14.

[14] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ''Focal loss for dense object detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[18] D. Comaniciu and P. Meer, ''Mean shift: A robust approach toward feature space analysis,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[19] R. E. Kalman, ''A new approach to linear filtering and prediction problems,'' *Trans. ASME, D, J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960, doi: 10.1115/1.3662552.

[20] J. J. Gibson, *The Perception of the Visual World*. Boston, MA, USA: Houghton Mifflin, 1950.

[21] X. Zhou, V. Koltun, and P. Krähenbühl, ''Tracking objects as points,'' 2020, *arXiv:2004.01177*. [Online]. Available: http://arxiv.org/abs/2004.01177

[22] L. He, X. Qiao, S. Wen, and F. Li, ''Robust object tracking based on motion consistency,'' *Sensors*, vol. 18, no. 2, p. 572, 2018, doi: 10.3390/s18020572.

[23] J. Shin, H. Kim, D. Kim, and J. Paik, ''Fast and robust object tracking using tracking failure detection in kernelized correlation filter,'' *Appl. Sci.*, vol. 10, no. 2, p. 713, 2020, doi: 10.3390/app10020713.

[24] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, ''FairMOT: On the fairness of detection and re-identification in multiple object tracking,'' 2020, *arXiv:2004.01888*. [Online]. Available: http://arxiv.org/abs/2004.01888

[25] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, ''CrowdHuman: A benchmark for detecting human in a crowd,'' in *Proc. Comput. Vis. Pattern Recognit.*, Apr. 2018, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1805.00123

[26] N. Wojke, A. Bewley, and D. Paulus, ''Simple online and realtime tracking with a deep association metric,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649, doi: 10.1109/ICIP.2017.8296962.

[27] F. H. Zunjani, S. Sen, H. Shekhar, A. Powale, D. Godnaik, and G. C. Nandi, ''Intent-based object grasping by a robot using deep learning,'' in *Proc. IEEE 8th Int. Advance Comput. Conf. (IACC)*, Dec. 2018, pp. 246–251, doi: 10.1109/IADCC.2018.8692134.

[28] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, ''MOTS: Multi-object tracking and segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7942–7951.

[29] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, ''Spatially supervised recurrent convolutional neural networks for visual object tracking,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[30] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, ''Distractor-aware Siamese networks for visual object tracking,'' in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 101–117.

[32] Z. Zhang and H. Peng, ''Deeper and wider Siamese networks for real-time visual tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[33] Z. Wang, Z. Zhao, and F. Su, ''Real-time tracking with stabilized frame,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 4431–4438, doi: 10.1109/CVPRW50498.2020.00522.

[34] Z. Y. Chan and S. A. Suandi, ''City tracker: Multiple object tracking in urban mixed traffic scenes,'' in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Sep. 2019, pp. 335–339, doi: 10.1109/ICSIPA45851.2019.8977783.

[35] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir, and H. Hashim, ''People tracking system using DeepSORT,'' in *Proc. 10th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Aug. 2020, pp. 137–141, doi: 10.1109/ICCSCE50387.2020.9204956.

[36] J. Jin, X. Li, X. Li, and S. Guan, ''Online multi-object tracking with Siamese network and optical flow,'' in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2020, pp. 193–198, doi: 10.1109/ICIVC50857.2020.9177480.

[37] H. Zhan, Y. Liu, Z. Cui, and H. Cheng, ''Pedestrian detection and behavior recognition based on vision,'' in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 771–776.

[38] R. Thomanek, C. Roschke, B. Platte, R. Manthey, T. Rolletschke, M. Heinzig, M. Vodel, F. Zimmer, M. Eibl, and M. Ritter, ''A scalable system architecture for activity detection with simple heuristics,'' in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2019, pp. 27–34, doi: 10.1109/WACVW.2019.00012.

[39] A. Lukezic, J. Matas, and M. Kristan, ''D3S—A discriminative single shot segmentation tracker,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7131–7140, doi: 10.1109/CVPR42600.2020.00716.

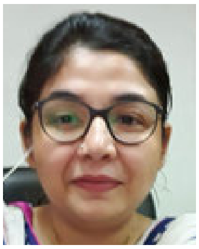
[40] S. Xiong, S. Li, L. Kou, W. Guo, Z. Zhou, and Z. Zhao, ''Td-VOS: Tracking-driven single-object video object segmentation,'' in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2020, pp. 102–107, doi: 10.1109/ICIVC50857.2020.9177471.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

[42] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, ''YouTube-VOS: Sequence-to-sequence video object segmentation,'' in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 585–601.

[43] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, ''A novel performance evaluation methodology for single-target trackers,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016, doi: 10.1109/TPAMI.2016.2516982.

SHAHEENA NOOR (Member, IEEE) received the bachelor's degree in computer engineering from Usman Institute of Technology Hamdard University, the master's degree in computer systems from the NED University of Engineering and Technology, and the Ph.D. degree in computer engineering with specialization in computer vision and image processing from Hamdard University, Karachi, Pakistan. She is currently working as an Assistant Professor with the Department of Computer Engineering, Sir Syed University of Engineering and Technology (SSUET), Karachi. She has an experience in the areas of research and academics. She has written more than 15 research articles in different journals, conferences and a book chapter. Her research interests include object recognition and activity recognition. She is also serving as a reviewer for local and well reputed international journals. She is a member of IEEE Computer Society (Karachi section). She has been serving as an advisor for IEEE SSUET Computer Society, since February 2018.



MARIA WAQAS received the B.E. and M.Eng. degrees in computer systems engineering and the Ph.D. degree in the field of computer systems engineering with specialization in computational biology from the Department of Computer and Information Systems Engineering, NED University of Engineering and Technology, Karachi, Pakistan, in 2001, 2008, and 2018, respectively. She has been working with the Department of Computer and Information Systems Engineering, NED University of Engineering and Technology, since 2001, and currently holds the position of an Assistant Professor. She is also the Co-Principle Investigator of National Center in Big Data and Cloud Computing, established research center with the NED University of Engineering and Technology. Her research interests include computational biology, neuromorphic and cytomorphic circuit design, integrated circuits design, machine learning, big data, and cloud computing.

MUHAMMAD IMRAN SALEEM (Member, IEEE) received the bachelor's degree in electronic engineering and the master's degree in computer engineering with specialization in computer network from Sir Syed University of Engineering and Technology (SSUET), Karachi, Pakistan. He is currently pursuing the Ph.D. degree in telecommunication engineering with the University of Malaga, Spain. His thesis topic was differentiated and integrated services of IP packet. He is currently working as an Assistant Professor with the Department of Computer Engineering, (SSUET). He is associated with the University, since January 2001. He has an experience of 19 years in SSUET. SSUET Management assigned him different tasks and positions during his 19 years of service. He was also Network and Internship in-charge beside to his teaching duties. He has an experience in the areas of research and academics. He wrote number of research papers in different conferences.

HUMERA NOOR MINHAS received the Ph.D. degree in computer vision and machine learning from the University of Central Florida. She worked with the Computer Vision Laboratory, University of Central Florida, in this connection. She worked as the Co-chair Person and an Associate Professor with the Department of Computer and Information Systems and Telecommunications Engineering, NED University of Engineering and Technology (NEDUET), Pakistan, for over 12 years. As a Postdoctoral Scholar with the Technische Universität München, Germany, she remained involved with robotic vision and using gaze-based cameras and eye tracking devices for context-aware environmental perception. She led the Quality Analysis Team at Cliqz GmbH, Munich Germany, and worked closely with big data for implementing privacy-based search solutions. She is currently the Engineering Lead with the Machine Learning Engineering Team, Eyeo GmbH, Cologne, Germany, and working towards automated ad-blocking.