International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 03 | Mar 2021 www.irjet.net p-ISSN: 2395-0072

SURVEILLANCE TRACKING USING MULTI TASK MASK RCNN

Naimish Sharma[1], Nilesh Auradkar[2], Rahul Patel[3], Yogita Shelar[4]

[1]*Naimish Sharma, UG Student, Dept. of Information Technology, Atharva college of Engineering, Mumbai, India*
[2]*Nilesh Auradkar, UG Student, Dept. of Information Technology, Atharva college of Engineering, Mumbai, India,*
[3]*Rahul Patel, UG Student, Dept. of Information Technology, Atharva college of Engineering, Mumbai, India*
[4]*Yogita Shelar, Professor, Dept. of Information Technology, Atharva college of Engineering, Mumbai, India*

Abstract - *Data is the brand new oil in present day technological society. The effect of green facts has modified benchmarks of overall performance in phrases of pace and accuracy. The enhancement is visualizable due to the fact the processing of facts is accomplished through buzzwords in enterprise referred to as Computer Vision (CV) and Artificial Intelligence (AI). Two technologies have empowered main obligations which include item detection and monitoring for site visitors vigilance systems. As the functions in photo will increase call for green set of rules to excavate hidden functions will increase. Convolution Neural Network (CNN) version is designed for city automobile dataset for unmarried item detection and YOLOv3 for more than one item detection on KITTI and COCO dataset. Model overall performance is analyzed, evaluated and tabulated the usage of overall performance metrics which include True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Rpn_Class_Loss, Rpn_Bbox_Loss, Mrcnn_Class_Loss, Mrcnn_Bbox_Loss and Mrcnn_Mask_Loss. Objects are tracked throughout the frames the usage of YOLOv3 and Simple Online Real Time Tracking (SORT) on site visitors surveillance video. This paper upholds the distinctiveness of the present day networks like Mask-RCNN. The green detection and monitoring on custom dataset are witnessed. The algorithms deliver actual-time, accurate, specific identifications appropriate for actual time surveillance applications.*

We construct a actual-time more than one-challenge Mask RCNN for popularity and monitoring the usage of deep convolutional neural networks. To reap this, we integrate present day item detection framework, Faster R-CNN architecture. We freeze the pretrained weights for the detection community and teach the monitoring community at the custom dataset. We display that such stop- to-stop modular method for popularity and monitoring overall performance is at par with the to be had laptop imaginative and prescient techniques. We additionally use our version on actual-global situations to reveal the generality of our version.

1. INTRODUCTION

In recent years, deep learning, especially the deep convolutional neural networks (CNN), has achieved remarkable successes in various computer vision tasks, ranging from image classification to object detection and semantic segmentation, etc. In contrast to traditional computer vision approaches, deep learning methods avoid the hand-crafted design pipeline and have dominated many well-known benchmark evaluations, such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Along with the popularity of deep learning in computer vision, a surge of research attention has been emerging to explore deep learning for resolving face detection tasks.

In general, face detection can be considered as a special type of object detection task in computer vision. Researchers thus have attempted to tackle face detection by exploring some successful deep learning techniques for generic object detection tasks. One of very important and highly successful framework for generic object detection is the region-based CNN (RCNN) method, which is a kind of CNN extension for solving the object detection tasks. A range of new advances for face detection regularly comply with this line of studies with the aid of using extending the RCNN and its advanced variants.

Mask R-CNN is a simple but effective addition to the Faster R-CNN architecture that adds head for instance mask prediction. Using a small Fully Convolutional Neural Network (FCN), it can predict pixel level instance masks. Besides the mask branch, it uses a Feature Pyramid Network (FPN) backbone. This addition allows the network to make use of both high-resolution feature maps in the lower layers for accurate localization, as well as semantically more meaningful higher-level features, which are of lower resolution.

Another contribution is ROI Align which maps arbitrarily sized spatial regions of interest in the features to a fixed spatial resolution using bilinear interpolation. This modification improves the COCO Mask metrics and enables the use of instance masks which require precise localization.

Following the emerging trend of exploring deep learning for face detection, in this paper, we propose a new face detection method by extending the state of the-art Faster R-CNN algorithm. In particular, our scheme improves the prevailing Faster RCNN scheme through combining numerous critical strategies, which include characteristic concatenation [11], difficult bad mining, and multi-scale training, etc. We performed an in depth set of experiments to assess the proposed scheme at the famous Face Detection Dataset and Benchmark (FDDB), and done the modern day performance.

1.1 Object Detection

Despite its long history of development since 90's, object detection has experienced major breakthrough since 2012 after AlexNet was popularized. Several pivotal works, such as Overfeat, SPPNet, Fast R-CNN, more recently Faster R-CNN and Single Shot Detector (SSD) have advanced the object detection approaches in terms of both speed and accuracy. In the following sections, the latter two works are introduced due to the fact that their designs well preserve the advantages but also compensate the shortcomings of the previous object detectors.

1.1.1 Faster R-CNN

Faster R-CNN is an object detector comprising of an object proposal generator and a detection network serving as classifiers classifying the generated object proposals as shown in Figure 2.1. Unlike Fast R-CNN relying on an external object proposal generator, e.g. Selective Search, Faster R-CNN introduces Region Proposal Network (RPN) which learns to generate the object proposals during the network training phase. The major contribution of this architecture is that it shares the convolutional features not only among the object proposals (as Fast R-CNN does) but also among the object proposals and detection networks, contributing to less wasted computation and faster inference and, in addition, higher mean Average Precision (mAP) than Fast R-CNN on PASCAL VOC 2007 and 2012 benchmark datasets. In Section we introduce the RPN architecture and the loss functions deviced to train RPN, respectively.

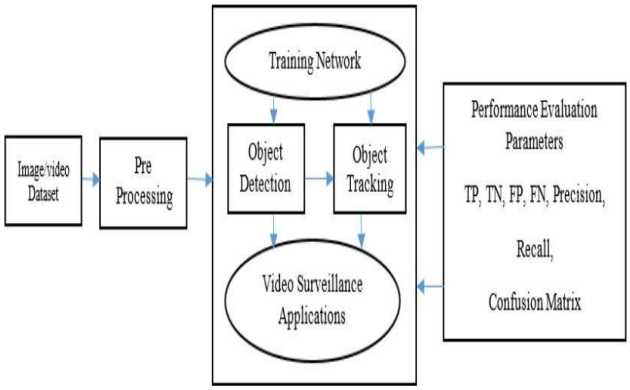
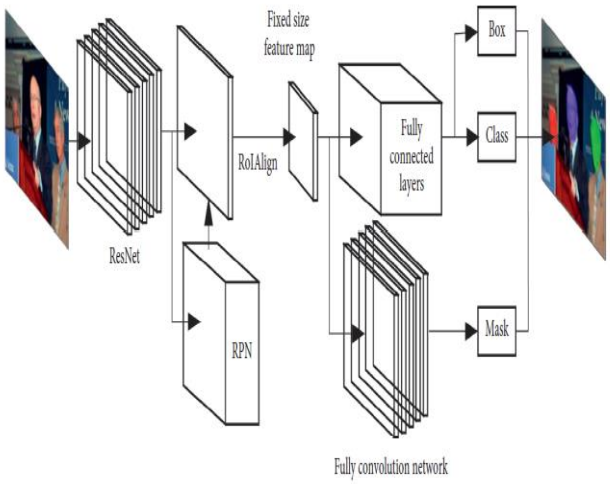


Fig 2.1 Faster R-CNN Network Structure

Network architecture of G Mask

Network Architecture proposed method is extended from the Mask R-CNN framework, which is the state of-the-art object detection scheme and demonstrated impressive performance on various object detection benchmarks.



As stated in Figure 1, the proposed G-Mask method consists of two branches, one for face detection and the other for face and background image segmentation. In this work, the ResNet-101 backbone is used to extract the facial features of the input image, and the Region of Interest (RoI) is rapidly generated on the feature map through the Region Proposal Network (RPN). We also use the Region of Interest Align (RoIAlign) to faithfully preserve exact spatial locations and output the feature map to a fixed size. At the end of the network, the bounding box is located and classified in the detection branch, and the corresponding face mask is generated on the image in the segmentation branch through the Fully Convolution Network (FCN).

Backbone

For Backbone of the model, any convolutional Neural Network which serves the purpose of extracting features from the image can be used. For the following analysis, Resnet50 has been used. Other models like Resnet101 which is the successor of the Resnet50 can also be used. The main function of this segment is to identify the low-level features (edges and corners) and further detect high level features. These extracted features are then feed to FPN (Feature Pyramid Network). The main functionality of FPN is to improve the standard and performance of high-level feature extraction, which is done by Resnet50 or any other respective model.

Region Proposal Network (RPN)

The main objective of RPN is to implement the functionality of Region Proposal. This is done by scanning the image and identifying the regions which may contain the object. The speed at which RPN operated is very high in case when done with GPU. One of the major contributing factors to this speed is the usage of weights stored in the FPN. This allows RPN to reuse the extracted features efficiently and avoid duplicate calculations.

There are two outputs obtained from RPN. One is called Anchor Class which acts as the actual bounding box for the object in the image. The second output is Bounding Box Refinement, which is not exactly encloses the object in the frame, but RPN uses it to calculate the delta value (% change in x, y, width and height) which is then further used to refine the anchor box to fit in the object

ROI Classifier & Bounding Box Regressors:

The operations of ROI Classifier and Bounding Box Regressor depends upon the output obtained from RPN in the form of ROIs (Region of Interest). The two outputs obtained from this stage are Class and Bounding Box Refinement. Identification of class is basically the part Multi-Class Identification in where the detected object is assigned to a particular class to which it belongs. Bounding Box Refinement is further refinement of the results obtained from RPN where the location and the bounding box size is redefined to capture the object in the image.

ROIPooling:

One of the major challenges faced in object detection and classification is that the Classifier do not handle the variable input size properly. This issue arises due to the fact that the classifier can handle only fixed input size. However, in Mask R-CNN, due to the involvement of RPN, different Regions of Interest of different sizes are proposed. This result of RPN is then used by a special function called ROIPooling. ROIPooling is implemented using Bilinear Interpolation, which allows the model to select the size of the box.

Segmentation Masks:

One of the biggest changes introduced by Mask R-CNN is the Classification of Images using Masks through Instance Segmentation. In this approach, pixel level comparison is done in order to get the exact layout of the object present in the image

2. Working

In this chapter, we describe the proposed framework for multiple object tracking in detail. Our primary object of interest in this work is *pedestrian*, and we do not impose any assumptions on the target object's size, aspect ratio, or appearance. Thereby, it is possible that the proposed framework can be extended to different object classes. In general, we follow the framework proposed but with few major adaptions. First, we do not train an object detector specifically, but we employ the object detector trained on MS COCO dataset where it provides multiple detectors of different base networks (i.e. MobileNet V1, InceptionV2, RFCN, Faster RCNN) that tradeoff the speed and accuracy. Second, in order to monitor if a tracker starts to drift or has drifted, we measure the similarity between the patches of the tracked target in any two consecutive frames. Third, to enable the tracker recover from tracking failure, we employ a simple person re-identification method that is as well based on the same deep features. In the pursuit of a more efficient implementation, the similarity is measured based on the deep features extracted from the network that has been served as the base network in the object detector in use. These modifications enable the proposed framework to detect and track the object.

Table 3.1: Base network structure of Restnet101. The classification layers have been removed as we adopt the network as a generic feature ex-traction, hence the classification layers are not needed. Please note that the input size and the structure are different from what is described in. Our implementation follows the one provided in. Figure 3.3a to 3.3j.

Table -3.1: Base network structure of Restnet101

FasterRCNN(
(Transform): Generalized RCNN Transform(
Normalize (mean= [0.485, 0.456, 0.406], std= [0.229, 0.224, 0.225])

Resize (min_size= (800,), max_size=1024, mode='inference')
)
(backbone): Backbone With FPN(
(body): Intermediate Layer Getter(
(conv1): Conv2d(1, 64, kernel_size=(7, 7), stride=(4,8,16,32,64), padding=(3, 3), bias=False)
(bn1): FrozenBatchNorm2d(2)
(relu): ReLU(inplace=True)
(maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
(layer1): Sequential(
(0): Bottleneck(
(conv1): Conv2d(3, 128, kernel_size=(1, 1), stride=(4,8,16,32,64), bias=False)
(bn1): Frozen Batch Norm2d(64)
(conv2): Conv2d(1, 256, kernel_size=(3, 3), stride=(4,8,16,32,64), padding=(1, 1), bias=False)
(bn2): Frozen Batch Norm2d(64)
(conv3): Conv2d(1, 256, kernel_size=(1, 1), stride=(4,8,16,32,64), bias=False)
(bn3): Frozen Batch Norm2d(256)
(relu): ReLU(inplace=True)
(downsample): Sequential(
(0): Conv2d(1, 256, kernel_size=(1, 1), stride=(4,8,16,32,64), bias=False)
(1): Frozen Batch Norm2d(256)
)
)

2.1 Implementation

The image is resized with its shorter side randomly sampled in [256,480] for scale augmentation. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted. The learning rate starts from 0.1 and is divided by 10 when the error plateaus and the models are trained for up to 60×10000 iterations. They use a weight decay of 0.0001 and a momentum of 0.9.

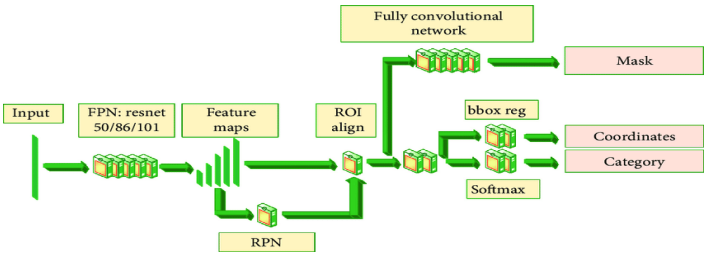


Fig -1: Mask RCNN

2.2 Update of Trackers

Three major steps are involved in updating the trackers, (1) update the DSST tracker, (2) update the auxiliary templates of the targets to track, and (3) update the auxiliary templates with adaptive learning rate. As described in 2.2.2.1, DSST seeks the location with maximal response as the final prediction of the target location. The maximal response (or regression score) can, in the one hand, be interpreted as how confident the tracker is, but in the other hand, the scores are positive numbers which are not strictly bounded within a range, e.g. [0, 100] or [0, 1]. This makes the response difficult to be interpreted and served as a reliable measurement.

Fig-2 Image Auxilary.

Fig-3 Target Drift

3. Problem Analysis

In real-world situations, we can generally classify face occlusion problems into three categories: facial landmark occlusion, occluded by faces and occluded by objects. Facial landmark occlusion includes conditions like wearing glasses and gauze masks. Occluded by faces is a complicated situation because a detector easily mis-recognize several faces into one or only detect a part of the faces. The segmentation method is proposed in order to mitigate this problem. When occluded by an object, usually more than half of a face will be directly masked. An original masking strategy is used to mimic these in-the-wild situations. We also visualized features of occluded faces, finding that occluded areas rarely respond. For some heavily occluded faces, useful information in feature maps is too scarce for a detector to identify. To tackle this problem, we may need to enhance representation ability of exposed area. Meanwhile, recognition of occluded area can also bespeak that "there is a face" on the condition that sufficient context information is provided. For the most complex problem where a face is occluded by another face, the context area should cover at least the nearby faces and a larger receptive field is required so that the integrity of the background information can be ensured. This idea is enlightened by human vision, that is, human need a large context to define a small or incomplete object. Besides, as the quality of features directly determines the results, segmentation is better conducted on image features.

4. Conclusions and Future Work

4.1 Conlucion

We presented in this thesis an on-line detect-and-track framework which aims to be operated in real-time with minimal system support. Unlike most of the multiple object tracking systems, the proposed framework does not assume the detector's availability in every frame as object detection is usually the largest computational burden in such systems. Overall, the proposed system works as follows. The system localizes the targets itself on the trained faces or with the information (e.g. image provided to the system) provided by the system administrator. During the course of tracking, a track of a target is constructed based on continually receiving above-the-threshold similarity measure between the target's auxiliary templates in the two consecutive frames. In other words, a track is interrupted if the auxiliary templates are dissimilar to some extent. However, the frames are moved to history tracks in which the frames and saved and displayed over the web frame. The proposed system devises Mask RCNN with RestNet 101 as Backbone architecture and correlation filter as the object detector and tracker, respectively. All relevant similarity measurements are based on the distance between the features in the Euclidean space extracted from the deep neural net.

We conducted experiments on the MOT'17 challenge dataset and custom dataset to demonstrate how the framework performs under full and partial availability of the detector, i.e. the detector is triggered in every frame. The main findings in the experiments are: (1) in static sequences (where the camera is not moving) the case with partial availability of the detector achieves comparable or slightly better performance than the other case, however, (2) in dynamic sequences with a moving camera or when the dynamics in the video are high (i.e. when people are moving faster), the case with full availability of the detector tends to outperform that with partial availability of the detector.

4.2 Future Work

Besides the improvements already suggested, it is also important to investigate how to share the features among the detector, tracker, and data association stages. Currently, in the proposed framework, the detector uses its own network model to do the inference while the tracker utilizes pixel values and a histogram of oriented gradients as the features. Sharing the features in similar tasks may bring several benefits, such as a higher level of generalization and less wasted computations, as suggested in [26, 28].

A recent publication proposed a two-way Siamese-like networks (i.e. two network streams fed with the frames at time t and (t + 1) as inputs respectively) to allow the system learn object representation and localization end-to-end [6]. Hence, it would be interesting to extend their work to one that learns object representation, detection, and localization given two consecutive frames. In addition, while the state-of-the-art object detectors pre-dominantly consider only spatial information from a single frame, in the context of object tracking, temporal information can be considered and possibly used to enhance the detection accuracy and consistency over frames if the detection and tracking are performed within a unified network. We leave the aforementioned as some thoughts for the future development of the project.

REFERENCES

1. J Yuji Chen, Adversarial Occlusion-aware Face Detection, arXiv: 1709.05188v6 [cs.CV] 29 Sep 2018
2. Song-Hai Zhang, Pose2Seg: Detection Free Human Instance Segmentation, arXiv:1803.03683v2 [cv.CV] 15 Nov 2018
3. Roland S. Zimmermann, Faster Training of Mask R-CNN by Focusing on Instance Boundaries, arXiv: 1809.07069v2 [cv.CV] 3 Oct 2018
4. V. Hoang, V. Hoang and K. Jo, "Realtime Multi-Person Pose Estimation with RCNN and Depthwise Separable Convolution," 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh, Vietnam, 2020, pp. 1-5.
5. T. Wang, Y. Hsieh, F. Wong and Y. Chen, "Mask-RCNN Based People Detection Using A Top-View Fisheye Camera," 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Kaohsiung, Taiwan, 2019, pp. 1-4.
6. S. Paste and S. Chickerur, "Analysis of Instance Segmentation using Mask-RCNN," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur,Kerala, India, 2019, pp. 191-196.
7. Kaigan Lin et al., “Face Detection and Segmentation based on Improved Mask R-CNN” in Discrete Dynamics in Nature and Society Volume 2020 , May 2020
8. Xiaoping Sun, Xiangfeng Luo, Jin Liu, Xiaorui Jiang, Junsheng Zhang, Semantics in Deep Neural-Network Computing in 2015 11th International Conference on Semantics, Knowledge and Grids.
9. Aniket Satishchandra Paste and Dr. Satyadhyan Chickerur ,”Analysis of Instance Segmentation using Mask-RCNN” in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT).
10. ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVEN- BERG, J., MANE´, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TAL-WAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIEGAS´, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
11. BABENKO, B., YANG, M.-H., AND BELONGIE, S. Visual tracking with online multiple instance learning. In Computer Vision and Pat-tern Recognition, 2009. CVPR 2009. IEEE Conference on (2009), IEEE, pp. 983–990.
12. BEWLEY, A., GE, Z., OTT, L., RAMOS, F., AND UPCROFT, B. Simple online and real time tracking. In Image Processing (ICIP), 2016 IEEE International Conference on (2016), IEEE, pp. 3464–3468.
13. BISHOP, G., AND WELCH, G. An introduction to the kalman filter. Proc of SIGGRAPH, Course 8, 27599-23175 (2001), 41.
14. BOLME, D. S., BEVERIDGE, J. R., DRAPER, B. A., AND LUI, Y. M. Visual object tracking using adaptive correlation filters. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (2010), IEEE, pp. 2544–2550.
15. GIRSHICK, R. Fast r-cnn. In Proceedings of the IEEE international conference oncomputer vision (2015), pp. 1440–1448.