Section A-Research paper

Enhanced Supervision of Indoor Surveillance VideoUsing Deep Learning

Soumya C S
Department of CSE
Ramaiah Institute of Technology
Bangalore, India

Manjula L
Department of CSE
Ramaiah Institute of Technology
Bangalore, India

Pallavi N
Department of CSE
Ramaiah Institute of Technology
Bangalore, India

Disha D N
Department of AI and ML
NMAM Institute of Technology
Karkala, India

***Abstract*—Driven by the rapid strides in information technology, video surveillance systems have seamlessly integrated themselves as pivotal components within contemporary urban security and protection frameworks. This holds particularly true for locations like prisons, where surveillance cameras are ubiquitously deployed. However, as the surveillance network continually expands, these cameras bring not only convenience but also generate an extensive volume of monitoring data. This situation presents significant challenges related to data storage, analysis, and retrieval. Integrating intelligent video analytics technology into a smart monitoring system can effectively oversee and proactively alert for anomalous events or behaviors. This area represents a prominent avenue of research within the surveillance domain. In this study, deep learning techniques are employed, utilizing the cutting-edge instance segmentation framework known as Mask R-CNN. The methodology the authors adopt encompasses the training of a fine-tuning network using the dedicated datasets. This network showcases its adeptness in efficiently recognizing objects within video frames, all the while generating precise segmentation masks for each identified instance. Empirical findings underscore the ease of training our network and its seamless applicability to different datasets. Remarkably, the average precision of the segmentation masks approaches an impressive 98.5% on our exclusive datasets.**

***Index Terms*—Surveillance Video; Deep Learning; Mask R-CNN; Object Detection;**

I. Introduction

Conventional video surveillance systems are limited to basic functions like video recording and storage. They lack the ability to autonomously detect and alert for unusual situations. To identify abnormal behaviors in real-time monitoring, human operators must constantly monitor the video feed. However, this results in fatigue as operators need to keep track of numerous surveillance video streams. This continuous monitoring can lead to reduced concentration, potentially causing delayed responses to anomalies and overlooking crucial information in the footage.

Furthermore, the need to store a substantial volume of surveillance video over extended periods, often spanning months or years, leads to significant storage costs. Consequently, there is a pressing need for an intelligent video surveillance system that can alleviate the burden on human operators. This system should employ intelligent detection technology to process, analyze, and comprehend video signals while preserving essential information within the footage. It should also autonomously identify target categories and their locations, eliminating the need for manual intervention. Should an anomaly occur, the system should promptly issue alarms to effectively support human operators.

Conventional methods for detecting moving targets are limited to identifying frames with motion but lack comprehension of the video's semantic content. In the context of deep learning advancements, sophisticated techniques for target detection, semantic understanding, and instance segmentation have emerged. These techniques enable semantic comprehension of video content and enhance accuracy.

II. Literature Survey

A method introduced by K. He et al. [1] exemplifies target segmentation by generating bounding boxes and masks for individual objects within images. This paper garnered recognition as the best paper at ICCV2017.

The progression of video analysis technology has led to rapid developments in intelligent video surveillance systems. These systems leverage embedded video analysis algorithms to autonomously detect abnormal behavior at monitored locations. Common anomalous behaviors include entrance/exit, rapid movement, and congregation.In the work by J. Zong et al. [2], a system for motion detection and target recognition is introduced, employing frame differences and self-mapping neural networks to enhance accuracy. Meanwhile, IBM's Intelligent Surveillance System [3] leverages state-of-the-art computer vision algorithms for the automated detection of events in densely populated urban environments. Semantic segmentation-based video segmentation and object tracking are realized in the research by X. Liu et al. [4][5], offering a means to streamline video data analysis.

This study draws from the foundation laid by reference [1] to extract informative key frames from videos and construct a

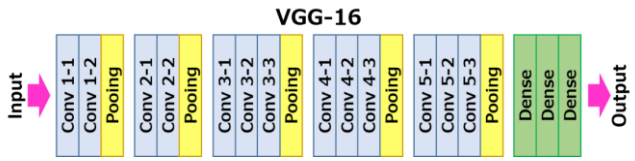

Fig. 1. VGG16 Architecture

requisite dataset. Fine-tuning is performed on the upper-layer network of Mask R-CNN to achieve optimal accuracy tailored to this dataset. Target detection and instance segmentation are executed on videos spanning six consecutive working days each week. The resulting textual output encompasses the counts and positions of prisoners and chairs across various moments in the video. This information is further utilized to chart daily people count trends over time and the distribution of individual positions and chair placements. Such visualizations enable monitors to discern differences between normal and abnormal curves, triggering automatic alarms upon detecting anomalous situations.

III. METHODOLOGY

A. CNN

Convolutional Neural Networks (CNNs) represent a type of artificial neural network initially introduced by LeCun et al. as a pivotal deep learning algorithm. This paradigm has achieved remarkable success particularly in the domain of computer vision, notably within image recognition and speech analysis, thus becoming a central focus of research. A distinguishing feature of CNNs is weight sharing, which mitigates the challenges of processing high-dimensional data. They offer the unique capability to directly employ images as network inputs, showcasing strong generalization prowess.

Illustrating the effectiveness of CNNs, the VGG16 network [8] serves as an exemplar to elucidate network structure, the utilization of the Backpropagation (BP) algorithm during training, and the overall training procedure. Convolutional neural networks exhibit several archetypal configurations that have gained prominence over the years. The trajectory commenced with LeNet in 1998 and culminated in the emergence of AlexNet in 2012. Subsequently, they gained prominence in diverse image-related domains, boasting iterations such as ZF-Net, GoogleNet, VGG-Net, and ResNet. The subsequent exposition highlights VGG16 as an illustrative case to elucidate the construction of a CNN. The network's architecture is visually depicted in Figure 1.

The approach known as Region-based Convolutional Neural Network (R-CNN) [10] employs the selective search method [11] to generate potential target regions, allowing individual regions of interest (RoIs) to undergo processing. This convolutional neural network utilizes an SVM classifier to ascertain the target category and utilizes border regression to refine the border position, ultimately achieving the detection of target bounding boxes.

Keras, a framework developed by Google engineer Francois Chollet, serves as a deep learning modeling environment in Python, leveraging backend computing frameworks like TensorFlow, CNTK, or Theano [9]. In comparison to other frequently utilized deep learning frameworks, Keras offers a significant advantage due to its user-friendliness. It liberates users from intricate mathematical formula commands, providing a direct approach to constructing specific architectures for deep learning neural networks.

Keras presents notable benefits in practical applications [9]: Keras provides highly efficient Python APIs, adaptable to various deep learning frameworks. It supports prevalent structures like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), enabling rapid construction and training of personalized deep learning models.

The framework is characterized by its lightweight and modular nature. Users can flexibly combine modules to compose the desired model. In Keras, a neural network model can be depicted as either a sequential or a graph model. The components, including neural network layers, activation functions, loss functions, regularization methods, initialization techniques, and optimization engines, are organized into modules. Users can arrange these modules within a sequential or graph model to construct the desired architecture, reducing the need for extensive code writing, enhancing efficiency, and minimizing error-prone situations.

Keras, built on the Python language, offers exceptional ease of use and scalability.

The framework seamlessly facilitates switching between Central Processing Units (CPU) and Graphics Processing Units (GPU), accommodating diverse application environments.

B. Mask R-CNN

The configuration of the network is displayed in Figure 2. In a broad sense, the Mask R-CNN architecture can be segmented into two principal components: the lower layer and the upper layer network. The lower layer network takes the form of a ResNet-FPN convolutional neural network, primarily tasked with extracting distinctive features from the input image. Conversely, the upper layer network, integrated into the Faster R-CNN model, incorporates an Fully Convolutional Network (FCN) that encompasses classification, border regression, and mask prediction elements.

(1) Training: The training process centers on utilizing images as the foundation. The initial step involves resizing the images, with each Graphics Processing Unit (GPU) simultaneously processing two images. Following the Region Proposal Network (RPN) phase, every image yields N Region of Interest (RoI) samples. The positive-to-negative sample ratio is set at 1:3, with N being 64 for the C4 backbone and 512 for FPN. Training involves 160,000 iterations on 8 GPUs, with a learning rate of 0.02. Learning rate attenuation is applied, reducing the rate by a factor of 10 at the 120th iteration. Additional parameters include a weight decay of 0.0001 and

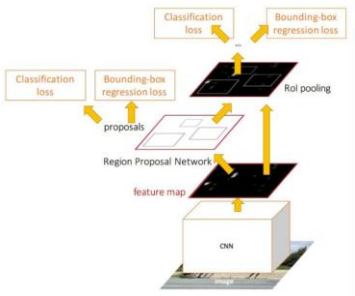

Fig. 2. Mask R-CNN architecture

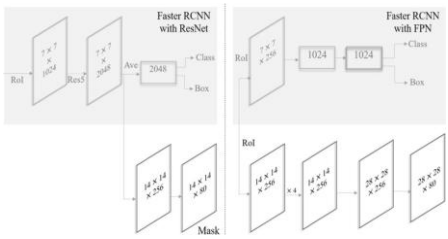

Fig. 3. Upper Layer Network of Mask R-CNN

momentum of 0.9. The RPN anchor spans 5 scales and 3 aspect ratios.

(2) Prediction: In the prediction stage, the C4 backbone network generates 300 candidate regions, whereas the FPN produces 1000. These regions then undergo bounding box prediction, followed by non-maximum suppression (NMS) [22]. Subsequently, the mask branch is applied to the top 100 detection boxes with the highest scores. For each Region of Interest (RoI), predictions for K masks are generated, but only the mask corresponding to the kth category of the classification branch prediction is chosen.

C. Intelligent Video Detection Utilizing the Mask R-CNN Approach

In contrast to Faster R-CNN, Mask R-CNN introduces only a minimal increase in computational overhead while offering a training process that is both uncomplicated and adaptable. This study employs the Mask R-CNN methodology for intelligent detection within indoor surveillance videos. Thanks to the pre-trained weight model derived from the COCO dataset, capable of recognizing 80 diverse object classes and backgrounds, the recognition capability is extensive. Consequently, the AP values achieved on the datasets discussed in this paper are not particularly high, as presented in Table 6.

However, the specific objective of this study revolves around identifying two specific categories of targets along with the background, concentrating on quantifying and establishing location distribution. To address this objective, a transfer learning approach is embraced, wherein the Mask R-CNN network undergoes fine-tuning using the pre-trained COCO weight model. Subsequently, the upper layer network is retrained to attain the optimal model for accurately detecting individuals and chairs, as well as determining their numbers and positions

Fig. 4. Labelling the images with labelme

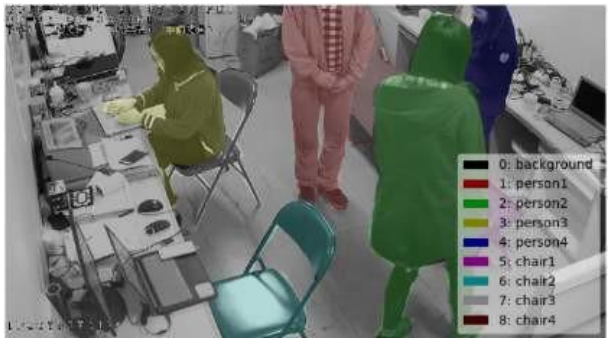

Fig. 5. Labels during dataset production

across various instances in the video. Visualizations in the form of count and distribution curves are generated, facilitating the automatic detection of abnormal events.

The experimental dataset for this research is derived from video recordings captured by a surveillance camera in a school laboratory. Videos were recorded during the working days of Monday to Saturday, spanning a week's routine. The process involves extracting key frames from the video, annotating target outlines and categories within the images, saving this information in JSON files, which conform to the standard data format compatible with Mask R-CNN's processing.

The process unfolds as follows: Members of the laboratory captured a video spanning six days, adhering to daily routines. The surveillance video's frame rate was 25 frames per second (fps), yielding closely spaced frames. To mitigate redundancy, a frame was selected every 12 seconds, equating to one frame every 288 frames, resulting in approximately 7,500 images from one day's video.

However, initial frame extraction yielded numerous duplicates, largely from scenes such as dormitory sleep or leaving for work. These duplicates had minimal impact on neural network training. To mitigate this, frames were chosen from sequences with significant target variations. After these two steps, over 600 useful images were culled from the original, voluminous dataset.

Subsequently, under the ubuntu16.04 system, the annotation software "labelme" was employed to annotate images. This involved delineating masks for individuals and chairs within

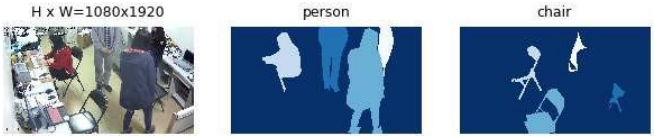

Fig. 6. Mask visualization of images

the images, along with assigning corresponding categories. Naming conventions followed the patterns ”person1,” ”person2,” ”chair1,” ”chair2,” and so forth, depicted in Figure 4. These annotations were saved as JSON files.

IV. Experimental results

There are two methodologies for pre-training models: Training solely the upper layer network component: This approach is generally suited for scenarios where the new dataset closely resembles the original dataset and is relatively small in size.In order to preserve the feature extraction capabilities of the underlying backbone network, all layers of ResNet-101-FPN are locked, maintaining the current weight values of these lower-level backbone networks. The sole layer subjected to training is the upper layer network’s newly initialized component.

Training all layers: In scenarios necessitating better adaptation to the new dataset, all layers must be retrained. This involves using pre-training weights as initial values and conducting training across the entire network, from its first layer to the last.

Given that this study’s dataset shares similarities with the MS COCO dataset and is comparatively compact, to ensure swift convergence and efficiency without compromising accuracy, the first method is employed. Training is conducted on a single GPU with an effective small batch size of 2, spanning 3,000 iterations, and a learning rate of 0.001.

1) Process of Training: Throughout the training process, an image is selected randomly from the training set, visualizing the masks for distinct objects like individuals and chairs, as illustrated in Figure 6. Concurrently, the ”fit_generate()” function systematically records the training log within a designated logs file. Within this study, the TensorFlow framework’s tensorboard tool is harnessed to dynamically monitor the evolution of each loss function. By scrutinizing the loss function’s alterations in response to iteration counts, continuous fine-tuning of network parameters is executed. Optimality is achieved when the loss function exhibits a descending trend and approaches convergence. For the training set comprising 300 images and the validation set encompassing 40 images, the network parameters adhere to the specifications outlined in Table 4. The progression curves depicting the loss function’s dynamics over iterations for both the training and validation sets are showcased in Figure 7 and Figure 8, respectively.

When dealing with a training set size of 300, this study delves into the influence of the anchor size within the RPN network on network accuracy. The results, showcased in Table 2, underscore that when the anchor’s aspect ratio remains consistent at (0.5, 1, 2), the mAP (mean Average Precision) attains its zenith at an anchor size of (32, 64, 128, 256, 512). This specific anchor size selection aligns optimally with the original image’s dimensions, as the dimensions of these five scales correspond most effectively to the sizes of the distinct targets within the image. This strategic choice ensures that larger targets aren’t overlooked due to overly diminutive anchors, nor are smaller targets missed due to excessively large anchors.

When the training set size comprises 100 samples, this study examines the impact of employing distinct batch sizes on both the speed of network training and the resultant model accuracy. In this scenario, a single GPU is utilized, and the epoch’s steps are determined by the formula steps = 100 / batch_size. As indicated in Table 3, the overall training duration is nearly equivalent when the batch size is either 1 or 2. The training duration for each image is approximately 10 seconds, and the loss function converges to a value of 1 for a batch size of 1, oscillating during the process. In contrast, a batch size of 2 leads to a reduction in the required iterations for a single cycle. However, with a batch size of 4, the process fails to execute due to insufficient memory.

Initially, this study utilized a training set comprising 100 images, coupled with a validation set containing 40 images for training purposes. Acknowledging the inherent limitation of a small training set, which can lead to overfitting, adjustments were made to expand the training set size to 200 and 300 images, as detailed in Table 4. This variation aimed to evaluate the influence of training set size on network accuracy. With network parameters kept constant and an epoch duration of 18, the mAP value exhibited a steady rise as the training set size escalated. This phenomenon can be attributed to the larger training sets yielding a greater extraction of features, thus amplifying the network’s generalization capabilities, and correspondingly augmenting accuracy in both detection and segmentation tasks. As a result, future endeavors can continue to amplify the training set size, further bolstering the network’s generalization aptitude while mitigating overfitting tendencies. Nevertheless, such augmentation comes at the cost of increased training time, subsequently elevating the time investment required for the process.

Among the top 20 epochs, the Mask R-CNN model demonstrating the highest accuracy on the validation set was selected as the definitive model. The average accuracy scores for both the COCO weight model and the model assessed on the Lab426 validation set are provided in Table 4. Building upon the COCO weight model’s foundation, the upper layer network within the Mask R-CNN architecture is subsequently retrained using the dataset introduced in this paper. Through a process of successive enhancements, the mask Average Precision (AP) of the ultimate model advances remarkably to an impressive 98.5%. Comparative results of the two models applied to identical images are illustrated in Figure 9.

The outcomes of the fine-tuned Mask R-CNN are visually depicted in Figure 10. Favorable results have been attained in both target detection and instance segmentation tasks.

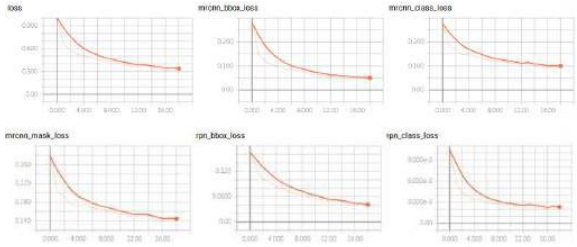

Fig. 7. Training set loss function curve

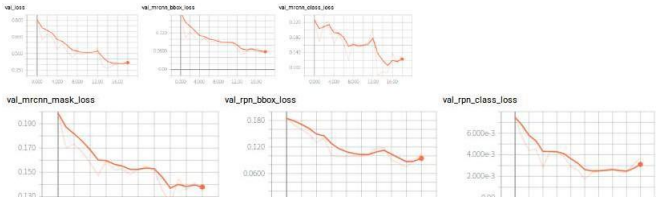

Fig. 8. validation set loss function

Notably, even instances involving partial occlusions and overlaps between targets yield improved segmentation outcomes. However, within the 40 images constituting the validation set, instances of detection errors still exist, as showcased in Figure 4.10. Analysis of these images highlights that substantial overlaps between distinct targets, or challenges arising from occlusion (as exemplified in Figure 4.10(a), where a person seated on a chair obstructs the central part of the chair, rendering only the backrest and a portion of the chair leg visible, leading to Mask R-CNN mistaking it for two chairs), result in errors. Given the constraint imposed by the limited dataset size, the occurrence of fitting phenomena is inevitable. Therefore, for the purpose of optimizing training velocity, the strategy of augmenting the training dataset size stands as a viable approach, contributing to the enhancement of the network's generalization capabilities.

V. CONCLUSION

This study presents an approach centered on Mask R-CNN to enable intelligent monitoring of indoor surveillance videos. By harnessing cutting-edge target detection and instance segmentation techniques within the domain of deep learning, the method achieves semantic comprehension of video content. This not only preserves crucial details embedded within the original video but also alleviates the storage and computational burden posed by extensive surveillance footage. Additionally, the incorporation of an automated alarm mechanism for detecting anomalous events serves to alleviate the workload of

Fig. 10. Test results of lab426 model

Fig. 11. Test results on fine tuned Mask R-CNN

weights	batch size	epoch	validation steps
0.0001	2	19	20
0.001	2	25	25

TABLE I
TRAINING PARAMETERS WITH LEARNING RATE OF 0.001

anchor ratio	anchor sclaes	anchors per image
0.5,1,2	32, 64, 128, 256, 512	256

TABLE II
RPN PARAMETERS WITH NMS THRESHOLD OF 0.7 AND POSITIVE ANCHOR RATIO 0.33

weights	batch size	epoch	validation steps
0.0001	2	19	20
0.001	2	25	25

TABLE III
TRAINING PARAMETERS WITH LEARNING RATE OF 0.001

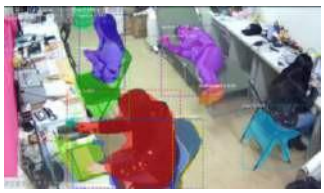

Fig. 9. Test results of COCO model

Fig. 12. Few Misclassified error samples

monitoring personnel.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[2] Yun, Deng Xiao-Hui, C.. (2012). Research and design of intelligent video surveillance system. 4. 378-388.

[3] Tian, Yl., Brown, L., Hampapur, A. et al. IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework. Machine Vision and Applications 19, 315–327 (2008). https://doi.org/10.1007/s00138-008-0153-z

[4] Liu, X., Song, M., Tao, D., Bu, J., Chen, C. (2015). Random geometric prior forest for multiclass object segmentation. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 24(10), 3060–3070. https://doi.org/10.1109/TIP.2015.2432711

[5] Liu, X., Tao, D., Song, M., Ruan, Y., Chen, C., Bu, J. (2014). Weakly Supervised Multiclass Video Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 57-64.

[6] Lin, TY. et al. (2014). Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

[7] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[8] McLeod, Clay. (2015). A Framework for Distributed Deep Learning Layer Design in Python.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[10] Uijlings J R R, Sande K E A V D, Gevers T, et al. Selective Search for Object Recognition[J]. International Journal of Computer Vision, 2013, 104(2):154-171.