Object Tracking System using DeepSORT

Vennela Madamshetty
Computer Science
University of Central Missouri
Missouri, USA
vennelamadamshetty@gmail.com

***Abstract*—Due to their rapid growth, image detection algorithms have become widely used in security applications such as facial identification and crowd surveillance. However, real-time tracking is difficult, especially in crowded locations where the person may be partially or completely obscured for a period of time. As a result, the goal is to develop a crowd surveillance people tracking system based on the DeepSORT (Simple Online Realtime Tracking) framework. Unlike object detection frameworks such as CNN, this system not only detects a person in real-time but also uses the information it has gathered to track the person's journey until they escape the camera's frame. The system will detect people using You Only Look Once (YOLO), then process them frame by frame using Deep SORT to anticipate their movement paths. This project aims to create an object-tracking system that uses deep learning and the Deep SORT framework to track object movement in a video recording or real-time video streaming. We'd like to be able to take data from the monitoring it's done and show it to consumers. This system should recognize the trajectory of movement of humans within the frame quickly, even in congested places with heavy traffic. When it comes to crowd security, Deep SORT can help. Deep SORT's application in crowd security can result in a considerably more efficient surveillance system and, as a result, a safer society for all of us.**

***Keywords* —** YOLO, DeepSORT, Kalman filter, objects, tracking, detection

I. Introduction

Nowadays, Object tracking is now widely utilized in the camera, car, store, security, inspection, restaurant, and traffic industries. Object tracking is used to determine the position of a moving object in a single video frame. The first phase of every object tracking program is object detection. Object detection is required by computer vision applications before they can be processed further. The goal of object detection is to figure out if there are any instances of things in an image frame and where they are located. The Viola-Jones framework is a well-known object detection approach. The template matching technique was examined in a study to detect items in various scales and variants. Another method for object detection relies on object features such as contextual and geometrical knowledge. Deep learning's recent arrival has opened up a new path for object detection. The You Only Look Once (YOLO) technique, which uses a convolutional neural network (CNN)to detect objects, is one well-known approach. The rapid processing speed of the YOLO algorithm is an advantage. In addition, since the introduction of YOLOv4, the YOLO detector has improved its accuracy and speed when compared to state-of-the-art techniques for object detection [1]. The tracking procedure is utilized after the object detection to determine the locations of the detected objects in each frame. The tracking's inputs are the identified items that are placed inside bounding boxes. The tracking phase will continue to monitor the recognized items until the video stream ends. for example, When three persons are spotted, three different bounding boxes are formed, and these bounding boxes are tracked in all following video frames. The mean-shift approach has been used to track objects based on the distribution of object features, which can be expressed as scale-invariant feature transform, speeded-up robust features, binary robust independent elementary features, and histograms of oriented gradients. If the distribution changes as a result of the tracking object moving, the tracking algorithm will look for a bounding box with the most similar distribution to the tracking object. Multiple object tracking is a challenging task compared to single object tracking. Previous studies have employed the Kalman filter for multiple object tracking [2]. The particle filter algorithm was also investigated for efficient multiple object tracking with initialization and learning phases. Nowadays, a well-known object-tracking approach is the DeepSORT algorithm which combines the Kalman filter, the Hungarian algorithm, and the appearance feature vector to track the objects.

Github link: https://github.com/vennela0609/Object-Tracking-System-using-DeepSORT

II. Motivation

The sheer difficulty of tracking objects/persons in public places using regular camera surveillance is becoming a need of academic, public/industrial markets in the form of real-time monitoring of human activities in restricted environments, monitoring uncertain events, crowd counting, and pedestrian traffic management using recent advancements in information technology. As the research community has become more interested in these systems, the problems and technological issues have grown. Moving/non-moving item detection, multi-person tracking, categorization, and sophisticated human motion analysis are some of these difficulties. With the limitations and current state of the above-mentioned approaches and the increase of collisions in crowds or public places, there is a demand and need for more efficient solutions based on new technologies for object detection and tracking in large crowds [3]. Although the model trained by the original algorithm of YOLOv4 works well, there are still many missing and wrong detection phenomena. In this context, the improvement of the original YOLOv4 algorithm along with DEEPSORT to further improve the index, reduce false detections, enhance the ability of missing detections to resist the interference of objects, and the detection efficiency must be studied when even in motion.

III. Main Contribution & Objectives

In this project, we are going to find and track multiple objects even if they are blocked by some other objects in the frame. We can either consider a pre-recorded video or real-time video as input and keep track of the detected objects. YOLOv4 can be used for the detection of the objects and the prior data that might be required is objects that are to be tracked and weights for the objects [4]. The video input is divided into frames and YOLO will find the multiple objects in each frame and evaluates the confidence score on each object to conclude what class the object belongs to [5]. Then, the DeepSORT algorithm can be used to keep track of the detected objects by following their trajectory. In this way, the object is not lost even if it is occluded in one or the other frames. The COCO dataset which has over 330K images and 80 classes for the images can be used to train the model.

The goal of this project is to use YOLOv4 detection models and DEEPSORT algorithms for real-time object recognition and tracking in public surveillance films to detect, categorize, and track object motions. To achieve the project's goal, the following objectives have been identified:

- To find appropriate and high-performance detection models for real-time object recognition and tracking in public surveillance.
- Detection models are used to classify the objects that have been selected.
- Surveillance of discovered items, including tracking and monitoring their motions.

IV. Related Work

Object detection is the process of extracting and recognizing real-world object instances from photos or videos, such as a car, bike, TV, flowers, and persons. Because it allows for the recognition, localization, and detection of many objects within an image, an object detection approach allows you to grasp the nuances of an image or video. Object detection can be accomplished in a variety of ways:

- Deep Learning Object Detection
- Viola-Jones Object Detection
- Feature-Based Object Detection
- SVM Classifications with HOG Features

In computer vision, object tracking is a crucial domain. It entails following an object across a sequence of frames, which could be a person, a ball, or an automobile [7]. We'd start with all feasible detections in a frame and assign them an ID for people monitoring. We try to carry on a person's ID in the following frames. That ID is discarded if the subject has moved away from the frame. If a new person appears, they will be given a new ID. This is a tough process since people may appear to be the same, prompting the model to switch IDs, people may get occluded, such as when a pedestrian or player is obscured by another, or things may vanish and reappear in later frames [6]. We were able to achieve outstanding tracking results because of deep learning.

A. Existing Approaches for Detecting an Object

a) Fast R-CNN: Fast R-CNN is an upgraded version of R-CNN, which has various drawbacks such as multistage pipelining, space and time consumption, and slow object recognition [10]. To get rid of them A different structure was introduced by Fast R-CNN. It accepts both the entire picture and object proposals as input. The method begins with a CNN on the input image, followed by the creation of a feature map utilizing various conv and max pooling layers. Then, for each object proposition, a Region of Interest (RoI) pooling layer develops a fixed-length feature vector and feeds it into a Fully Connected (FC) Layer. This layer then separates into two output layers: one that produces the SoftMax probability for each class as well as a "background" class, and the other that creates four real values for each class that determine the bounding box for that class.

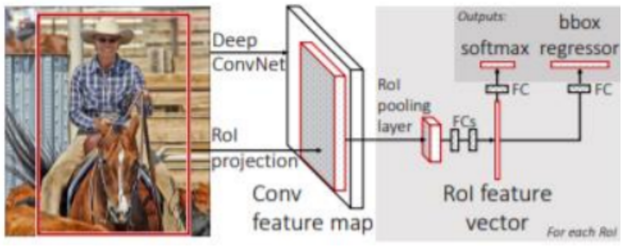

Fig. 1. Fast R-CNN Model.

b) Single Shot Multi-Box Detector (SSD): SSD uses a free-forward convolution layer to create a fixed-size collection of bounding boxes and scores for each instance of an object class discovered within those bounding boxes [10]. Non-Max Suppression is also used to make final decisions. The architecture of solid-state drives (SSDs) is simple. The model's first layers, referred to as the Base network, are the standard ConvNet layers used for Image classification. They then add some auxiliary layers to provide detections, keeping Multi-scale feature maps, default boxes, and aspect ratio in mind.

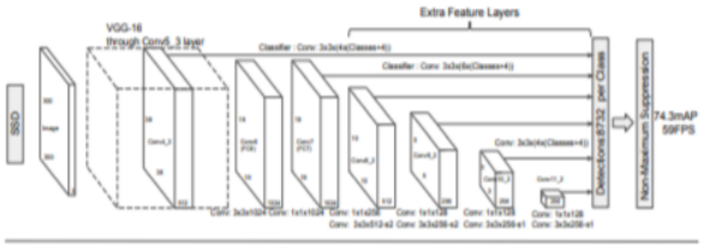

Fig. 2. Single Shot Multi-Box Detector (SSD) model.

c) Retina-Net: Retina-Net is a unified network that consists of two task-specific subnetworks and a backbone network. The backbone is a commercial convolution network that generates a conv feature map over the entire input image. Backbone output categorization is performed by the first subnet, whereas convolution bounding box regression is performed by the second subnet [9]. It uses a Feature Pyramid Network (FPN) backbone on top of a feedforward ResNet architecture to generate a rich, multi-scale convolutional feature pyramid, which is then fed to the two subnets, one of which classifies the anchor boxes and the other of which performs regression from the anchor boxes to the ground-truth anchor boxes [13].

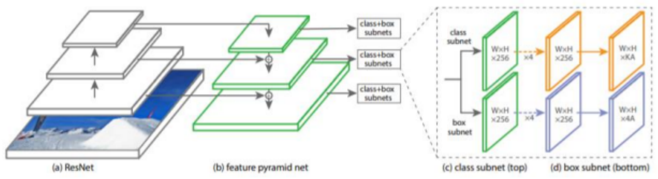

Fig. 3. Retina-Net.

B. Existing Approaches for Tracking an Object

a) Mean Shift Algorithm: Mean-shifting, often known as mode seeking, is a prominent approach for clustering and other unsupervised issues. It's similar to K-Means, but instead of using the simple centroid method to calculate cluster centers, it uses a weighted average that prioritizes points closer to the mean [20]. This algorithm's purpose is to locate all of the modes in a given data distribution. Also, unlike K-Means, this technique does not require an optimal "K" value. The mean-shift method seeks the new greatest mode and thereby tracks the object in the following frame, where the distribution has changed due to the movement of the object in the frame.

b) Optical Flow Method: The characteristics aren't always taken from the identified objects, which makes this method different from mean-shift. Rather, the object is monitored at the pixel level by varying image brightness. Four key assumptions underpin optical flow tracking:

1) Consistency of brightness: The brightness around a small zone is believed to be virtually consistent, even if the region's position changes.

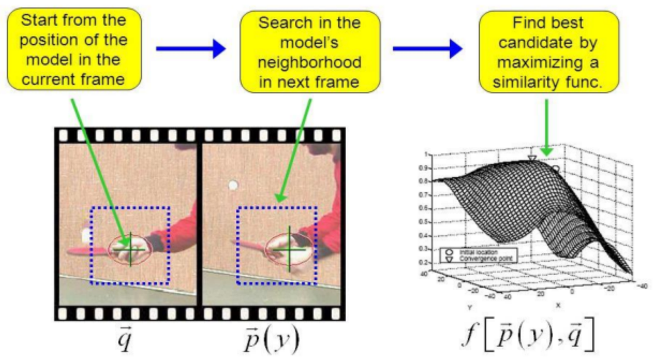
Start from the position of the model in the current frame
Search in the model's neighborhood in next frame
Find best candidate by maximizing a similarity func.
$\vec{q}$
$\vec{p}(y)$
$f[\vec{p}(y),\vec{q}]$

Fig. 4. Mean Shift

2) Spatial coherence: Neighboring points in a scene usually belong to the same surface, therefore their motions are similar.
3) Temporal persistence: A patch's motion changes gradually.
4) Points do not move very far or in a random manner.

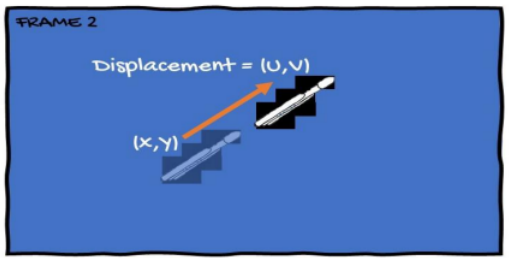

Fig. 5. Object Position Estimation using Optical flow

The Optical Flow approach examines the motion of features as a result of the scene's and camera's relative motion across frames. These motion vectors can now be used to track and even anticipate the trajectory of the object in the next frame if the object is traveling at a particular velocity.

c) Kalman Filter: The Kalman filter takes into account all available detections and prior predictions to arrive at the best estimation of the present state while minimizing the possibility of errors. We can train a YOLOv4 object detector to detect an object fairly well. However, it isn't perfect and occasionally misses detections, perhaps 10% of the time. When there are any occlusions, the Kalman filter

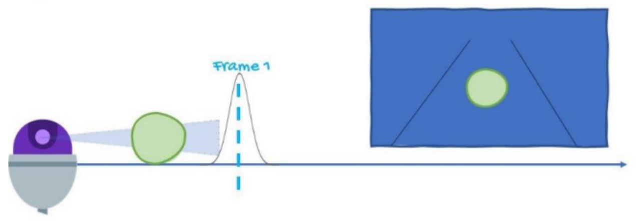

Fig. 6. Object track without occlusion

has proven to be accurate. When the object can be observed, the Kalman filter places a greater emphasis on the sensor data, giving it more weight [8]. When it is partially occluded, the filter gives both motion and sensor measurement data equal weight. The filter will rely more on motion data if it is completely occluded. The

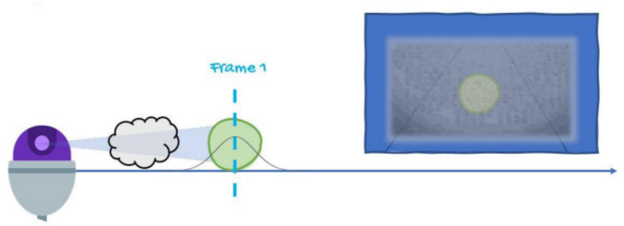

Fig. 7. Object track with occlusion

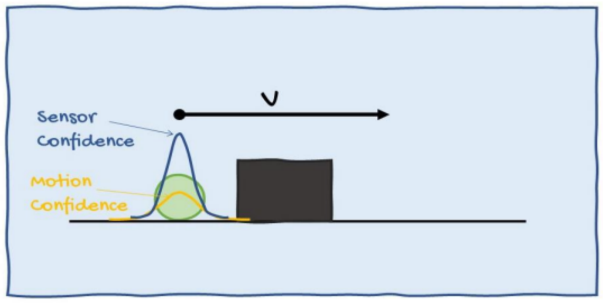

Fig. 8. Object track with sensor and motion confidence

Kalman filter is a recursive algorithm that uses current readings to anticipate the current state, then uses those measurements to update the predictions. For linear systems with Gaussian processes, the Kalman filter performs well.

V. Proposed Framework

a) YOLO: Object detection is handled differently in the YOLO framework (You Only Look Once) than in the previous methods. It predicts the bounding box coordinates and class probabilities for these boxes using the full image as a single instance. The most significant benefit of YOLO is its speed. It's lightning quick, processing 45 frames per second [8]. YOLO is also aware of the concept of generic object representation.

The YOLO technique, which employs a deep learning network to recognize objects in real-time, has two objectives. The first objective is to locate an object, and the second is to classify the thing that was discovered in the first task. The YOLO method does both tasks with a single neural network, allowing it to run at a rapid rate. The YOLO algorithm takes picture pixels as input and produces bounding boxes and class probabilities for these bounding boxes as output.

- The input image is broken down into a S x S grid of cells. By anticipating B bounding bounds for this object, each cell identifies only one object. The cell also predicts class probabilities for C conditionals, where C is the number of classes.
- There are three conditionals class probabilities P(car—object), P(bike—object), and P(bicycle—object) if the YOLO algorithm is employed to detect three classes of car, bike, and bicycle.
- Each bounding box comprises five elements (x, y, w, h, confidence), where confidence is the likelihood that the box includes an item, x, and y are the bounding box's coordinates, and width and height are the bounding box's width and height.
- To get the class-specific confidence score for each bounding box, multiply the confidence by the conditional class probabilities.

After forecasting the class probabilities, the algorithm proceeds to Non-max suppression, which aids in the removal of unneeded anchor boxes. By executing the IoU (Intersection over Union) with the one with the highest-class probability among them, non-max suppression

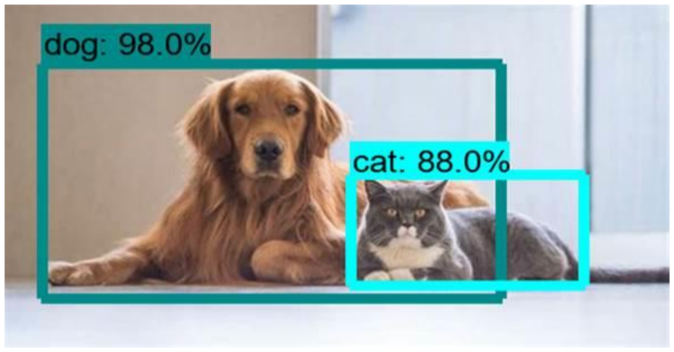

Fig. 9. Detection using YOLO

eliminates the bounding boxes that are very near. It calculates the value of IoU for all the bounding boxes that correspond to the one with the highest-class probability, then excludes those whose value of IoU exceeds a threshold. It means that those two bounding boxes are covering the same item, but one of them has a low chance of being the same, so it is discarded [19]. After the boxes have been filtered, the method identifies the bounding box with the next highest class probabilities and repeats the process until all of the bounding boxes have been found.

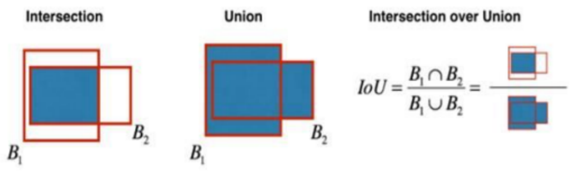

Fig. 10. Intersection Over Union

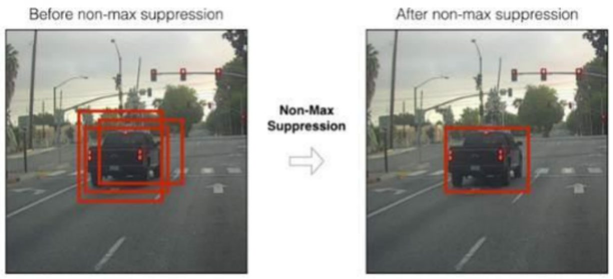

Fig. 11. Comparison Of Non-Max Suppression

This is one of the best object detection algorithms, with a performance that is comparable to the R-CNN algorithms.

b) Deep SORT: Deep SORT is a new tracking method that enhances Simple Online and Real-time Tracking and has demonstrated outstanding performance in Multiple Object Tracking (MOT) challenges.

Despite the effectiveness of the Kalman filter, SORT yields a very large number of identity switches and has a shortcoming in tracking through occlusions and multiple camera perspectives, despite its overall good performance in terms of tracking precision and accuracy. To improve the SORT algorithm, the inventors of the DeepSORT algorithm incorporated a new distance metric based on the object's "appearance." The following are the primary components of the DeepSORT algorithm:

- **The Kalman filter** is used by the track estimator, the original component of the SORT method, to estimate the locations of the object bounding boxes. Previous object velocities are used to make the prediction. The track estimator is a location metric that employs the intersection over union (IoU) distance between the detected and predicted bounding boxes to calculate location.
- **Appearance descriptor** is a new component now includes a DeepSORT. The appearance data is retrieved using CNN so that in the feature space, features from one object class are similar and features from various object classes are distinct. The appearance metric is defined by the appearance descriptor.
- **Data association:** Using the position and appearance metrics, it assigns the observed bounding boxes to an object's existing track. Every existing track has its own unique identity (ID).
- **Track Handling:** The object tracking is controlled by track handling. At frame f, if a newly detected bounding box cannot be associated with any track, it will be placed in a tentative track. In succeeding frames, the DeepSORT will attempt to connect the tentative track with other tracks. The track is updated if the association is successful. The tentative track will be withdrawn if this does not happen.

Another Distance metric used in DeepSORT:

Appearance feature vector: Deep SORT uses it as a distance metric. A classifier is created using the dataset, which is methodically trained until it achieves a reasonable level of accuracy. The final classification layer is then stripped away, leaving a dense layer that creates a single feature vector that is ready to be classified. The appearance descriptor is the name given to this feature vector.

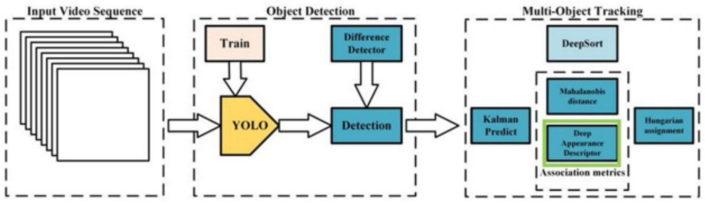

Fig. 12. DeepSORT Architecture

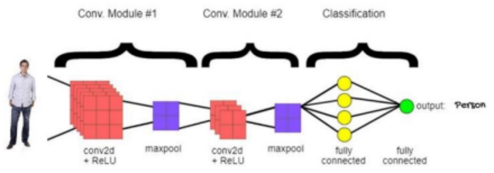

Fig. 13. Appearance metric Network

To build the measurement-to-track correlation, the appearance descriptor performs nearest neighbor queries in the visual appearance. The process of finding the relationship between a measurement and an existing track is known as measurement-to-track association, or MTA. For MTA, we now use the Mahalanobis distance rather than the Euclidean distance.

Deep Sort's most important feature is the ability to connect Appearance data, using the ReID domain model to extract features and reduce the number of ID switches. The following is the overall flow chart:

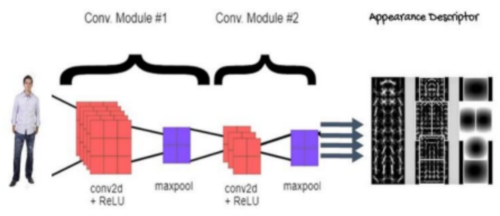

Fig. 14. Feature extraction using Appearance metric network

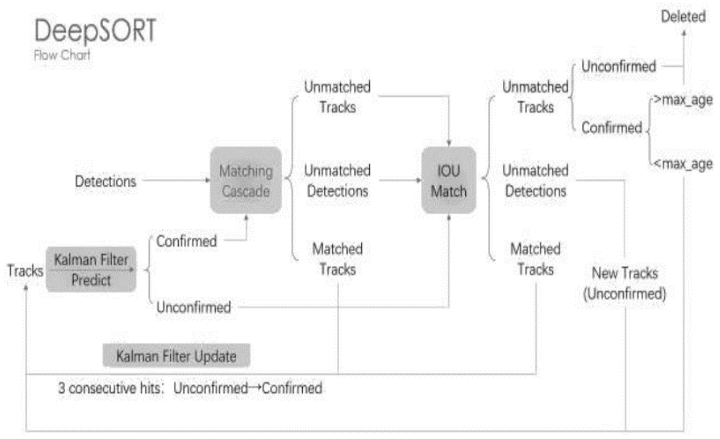

Fig. 15. DeepSORT FLOWCHART

VI. Data Description

COCO is a dataset with a great amount of object detection, categorization, key-point detection, and labeling. The dataset's name, Common Things in Contextual (COCO), literally means "common objects recorded from everyday scenarios". This gives the objects collected in the scenes additional context. The COCO dataset contains challenging, high-quality visual datasets for computer vision, mostly state-of-the-art neural networks. Many cutting-edge object detection and segmentation models have been trained and evaluated using the COCO dataset, which is frequently utilized in computer vision research. There are 328K images in the dataset. It has 164K images divided into three sets: training (83K), validation (41K), and test (41K). COCO consists of 91 classes, but only 80 of them are used in the data. Each record of the dataset has annotations for

- *object detection:* bounding boxes and per-instance segmentation masks with 80 object categories.
- *captioning:* natural language descriptions of the images (see MS COCO Captions).
- *key points detection:* containing more than 200,000 images and 250,000 person instances labeled with key points (17 possible key points, such as left eye, nose, right hip, right ankle).
- *stuff image segmentation:* per-pixel segmentation masks with 91 stuff categories, such as grass, wall, and sky (see MS COCO Stuff).
- *panoptic:* full scene segmentation, with 80 thing categories (such as a person, bicycle, elephant) and a subset of 91 stuff categories (grass, sky, road).
- *dense pose:* more than 39,000 images and 56,000 person instances labeled with DensePose annotations – each labeled person is annotated with an instance id and a mapping between image pixels that belong to that person's body and a template 3D model. The annotations are publicly available only for training and validation images.

VII. Results & Analysis

The system takes the type of video input specified by the user as it can process both pre-recorded and real-time videos. Then, the machine will transform the incoming video into video frames, frame by frame, in order to extract color information and recognize objects. YOLO was chosen since the photos only need to transit across the network once. YOLO predicts multiples based on a unique property of the photos, each containing an object. S x S regions are used to partition images. If the object's center falls within one of the regions of Interest (ROI), that ROI will be in charge of detecting the object. Each discovered item will have a confidence score that will be used to classify the object's class name. Only the class name, however, will be handled.

In order to track the detected objects, the Kalman filter algorithm is used in conjunction with Deep SORT. The Kalman filter is used to predict the target's state vector. It made use of a set of measurements taken over a period of time. This method can accurately follow the object during occlusion by predicting the likely movement path. The tracking component identified the person, assigned it a unique id, and began following its movement trajectory. During this stage, the system can track each person's motion while keeping their unique identifier. After the tracking portion is completed, the system will store the result as an Audio Video Interleave (AVI) video file. The produced video file will be in the user's preferred format.

The Kalman filter is an important part of the Deep Sort tracker. The Kalman traceability model is specified on an 8-dimensional state space (u, v, a, h, u', v', a', h') that features the bounding box center position (u, v), 'a' is the aspect ratio, and 'h' is the picture height. Other variables are the variables' corresponding velocities. Because a simple mathematical linear velocity model is used, the variables only have actual position and velocity indicators. The Kalman filter provides to account for noise in detection and predicts a perfect match for bounding boxes based on the prior state. We generate a "Track" with each detection that contains all of the essential state data. It also provides a feature to track and erase tracks that had their previous successful detection a long time ago, indicating that such items had left the scene. For the first few frames, there is an average amount of detection criteria to exclude duplication tracks. The next difficulty is linking new detections with new predictions once we receive the new bounding boxes obtained from the Kalman filter. The 'assignment problem' arises since we have no notion of how to attach which track over which inbound detection since they are processed autonomously.

To remedy this, the Kalman filter's uncertainties are incorporated using a distance metric called Mahalanobis distance (an effective metric for dealing with distributions). This distance has provided a pretty excellent indication of the actual linkages. This measure, which effectively quantifies the gap between two distributions, has proven to be more precise than Euclidean distance (Underneath the Kalman filter, everything has to be a distribution.). The typical Hungarian approach for solving the assignment issue has shown to be a very successful and easy probabilistic optimization algorithm.

The integration of YOLO object recognition and Deep SORT object tracking has revolutionized research opportunities. Using the YOLO and Deep SORT algorithms, this project introduces an actual object/human tracking system. Even if a blockage occurs, this approach seeks to keep track of the individual or object. This technique can be useful for maintaining security in areas where there are a lot of people coming in and out. The results suggest that by supplying a consistent and precise dataset, the tracking might be improved.

As of now the system only accepts input only from one camera/live feed. If multiple cameras cover a broad view of the area and give different perspectives of the same object it would be difficult for this system to track the object throughout. So, these factors will be addressed in the future of the project. The proposed system achieves

9-10 FPS on average on the input video footage and successfully 13-14 FPS on live webcam which is closer to real-time surveillance (24 FPS). This can be improved by running the system on a faster GPU machine.

References

[1] Azhar, M. I., Zaman, F. H., Tahir, N. M., &; Hashim, H. (2020). People tracking system using DeepSORT. 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). https://doi.org/10.1109/iccsce50387.2020.9204956

[2] Raghunandan, A., Mohana, Raghav, P., &; Aradhya, H. V. (2018). Object detection algorithms for video surveillance applications. 2018 International Conference on Communication and Signal Processing (ICCSP). https://doi.org/10.1109/iccsp.2018.8524461

[3] Bin Zuraimi, M. A., &; Kamaru Zaman, F. H. (2021). Vehicle detection and tracking using Yolo and DeepSORT. 2021 IEEE 11th IEEE Symposium on Computer Applications &; Industrial Electronics (ISCAIE). https://doi.org/10.1109/iscaie51753.2021.9431784

[4] Boyuan, W., &; Muqing, W. (2020). Study on pedestrian detection based on an improved Yolov4 algorithm. 2020 IEEE 6th International Conference on Computer and Communications (ICCC). https://doi.org/10.1109/iccc51575.2020.9344983

[5] Hu, X., Wei, Z., &; Zhou, W. (2021). A video streaming vehicle detection algorithm based on yolov4. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). https://doi.org/10.1109/iaeac50856.2021.9390613

[6] Zhao, Y., Chang, M.-C., &; Tu, P. (2019). Deep Intelligent Network for device-free people tracking. Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems. https://doi.org/10.1145/3302509.3313312

[7] Luo, H.-W., Zhang, C.-S., Pan, F.-C., &; Ju, X.-M. (2019). Contextual-YOLOV3: Implement better small object detection based deep learning. 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). https://doi.org/10.1109/mlbdbi48998.2019.00032

[8] Kumar, S., Vishal, Sharma, P., &; Pal, N. (2021). Object tracking and counting in a zone using Yolov4, DeepSORT and tensorflow. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). https://doi.org/10.1109/icais50930.2021.9395971

[9] Mandal, V., &; Adu-Gyamfi, Y. (2020). Object detection and tracking algorithms for vehicle counting: A comparative analysis. Journal of Big Data Analytics in Transportation, 2(3), 251–261. https://doi.org/10.1007/s42421-020-00025-w

[10] Cheng, G., & Han, J. (2016). A Survey on Object Detection in Optical Remote Sensing Images. ArXiv. https://doi.org/10.1016/j.isprsjprs.2016.03.014

[11] Wang, Y., Xie, Z., Xu, K., Dou, Y., & Lei, Y. (2016). An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning. Neurocomputing, 174, 988-998.

[12] Susskind, J., Hinton, G., Memisevic, R., &; Pollefeys, M. (2011). Modeling the joint density of two images under a variety of transformations. CVPR 2011. https://doi.org/10.1109/cvpr.2011.5995541

[13] Hu, Y., Liao, S., Lei, Z., Yi, D., & Li, S. (2013). Exploring Structural Information and Fusing Multiple Features for Person Re-identification. 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 794-799.

[14] Vasuhi, S., Vijayakumar, M., &; Vaidehi, V. (2015). Real time multiple human tracking using Kalman filter. 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN). https://doi.org/10.1109/icscn.2015.7219902

[15] Tang, S., Andriluka, M., Milan, A., Schindler, K., Roth, S., &; Schiele, B. (2013). Learning people detectors for tracking in crowded scenes. 2013 IEEE International Conference on Computer Vision. https://doi.org/10.1109/iccv.2013.134

[16] Costin, A. (2016). Security of CCTV and Video Surveillance Systems. Proceedings of the 6th International Workshop on Trustworthy Embedded Devices. https://doi.org/10.1145/2995289.2995290

[17] Ranzato, M., Susskind, J.M., Mnih, V., & Hinton, G.E. (2011). On deep generative models with applications to recognition. CVPR 2011, 2857-2864.

[18] Ouyang, W., & Wang, X. (2013). Joint Deep Learning for Pedestrian Detection. 2013 IEEE International Conference on Computer Vision, 2056-2063.

[19] Scheidegger, S., Benjaminsson, J., Rosenberg, E., Krishnan, A., &; Granstrom, K. (2018). Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering. 2018 IEEE Intelligent Vehicles Symposium (IV). https://doi.org/10.1109/ivs.2018.8500454

[20] Meimetis, D., Daramouskas, I., Perikos, I., &; Hatzilygeroudis, I. (2021). Real-time multiple object tracking using deep learning methods. Neural Computing and Applications, 35(1), 89–118. https://doi.org/10.1007/s00521-021-06391-y