Parking Monitoring System*

*Neurog LLP: https://neurog.ai/

***Abstract*—Around 21,000 cars were stolen each year in Pakistan, with a staggering worth of PKR 5 billion. The number of these cars in Pakistan has been growing exponentially as the total registered vehicles reported were 6,628,063 units in Dec 2021, which is congesting the parking spaces all around Pakistan. This warrants a need to effectively track a large number of cars present in the parking spaces for informative and surveillance purposes. Taking into account the issue, the goal of this project was to design an interactive website using a model that track these cars. Additionally, the detections were visualized on a livestream by generating an alert message as well as an image of the moving car when entered the parking lot. But, this approach was in limited use due to the unavailability of data about cars in parking spaces with appropriate viewpoints. Hence, this study also presents a novel dataset for car images in a parking lot, and trains and deploys multiple models using this dataset. The staggering accuracy of 86.9 %, mAP@.5, and 92.3% MOTA is observed for object detection and tracking studies. These findings indicate that the suggested model is used to monitor private parking lots for security reasons as well as produce statistics, insights, and visualization using information about the parked cars.**

***Index Terms*—Object Tracking, Object Detection, Computer Vision, YOLO architecture**

I. Introduction

Parking lots are becoming increasingly congested in today's world. To manage traffic, it is necessary to monitor the parking spaces and the number of cars parked at any time in parking lots. Currently, parking lot systems are not that efficient and require manual effort to keep records in excel sheets and hand-written documents. Manual work increases the chances of human error and reduces the accuracy of the collected data. It also introduces inaccuracy in the statistical analysis of the manually collected data.

Therefore, an efficient monitoring system is required to keep track of parking spaces which would reduce human interference. Furthermore, not only is it necessary to monitor the number of cars parked at any given time but also to ensure the safety of the parked cars by keeping track of any unwanted car movement. Currently, vehicle detection and tracking techniques have been used in surveillance systems which are effectively being used in parking lots. In such systems, deep learning techniques have considerably improved performance. Deep learning models based on neural networks provide image identification and categorization. The You Only Look Once (Yolo) model, in particular, provides a high computation speed for image classification and determining the location of the classified object.

Our system uses Yolov5 for object detection. It is a pre-trained model which has been finetuned on manually collected data of cars parked in the G-15 Parking lot, Islamabad. The system is connected to the live video stream of the G-15 parking lot. This provides live data to the application upon which the Yolov5 model performs object detection and identifies cars in each frame. The StrongSORT algorithm has been used for object tracking to implement a security feature.

Our system comprises the following features:

- A dashboard where the user can view the total number of cars parked per hour throughout the entire day for the current date, a chart showing the number of cars parked each day in the entire week, the total number of cars parked for the current date, the maximum cars parked for the day and the average number of cars parked during the entire week.
- A live stream feature that shows the live video stream of the parking lot with object detection of vehicles. The footage shows bounding boxes drawn around the identified vehicles.
- A security feature that notifies when a car has moved. It shows the moving car as well as its coordinates.
- An About Us Page which highlights the features present in the web application.

The remaining paper is structured as follows: Section II summarizes any previous research on the topic and a possible research gap. Section III explains the data collection process along with the method of data pre-processing. Section IV will walk through the proposed methodology and related details about object detection and tracking deep learning models, integration of the best model, and website application design. Section V provides the results. Section VI walk through the discussion of results. Moreover, this section describes the evaluation measures used. Section VII present the overall conclusion with limitations and future work.

II. Literature Review

The related work section is divided into three categories: A. Vehicle Detection, B. Vehicle Tracking and C. Parking Monitoring Applications.

A. Vehicle Detection

Currently, vehicle object detection is divided into traditional machine vision methods and complex deep learning methods. Traditional machine vision methods utilize the motion of a vehicle to distinguish it from a fixed background image. This method can be sub-categorized into three categories [1]. 1) the method of using background subtraction [2], 2) the method of using continuous video frame difference [3], and 3) the method of using optical flow [4].

However, the use of deep convolutional networks (CNNs) has achieved amazing success in the field of vehicle object detection. CNNs have a strong ability to learn image features and can perform multiple related tasks, such as classification and bounding box regression [5]. The detection method can be sub-categorized into two categories. The two-stage method generates a candidate box of the object via various algorithms and then classifies the object by a convolutional neural network. The one-stage method does not generate a candidate box but directly converts the positioning problem of the object bounding box into a regression problem for processing. In the two-stage method, Region-CNN (R-CNN) [6] uses selective region search [7] in the image. The image input to the convolutional network must be fixed-size, and the deeper structure of the network requires a long training time and consumes a large amount of storage memory. R-FCN, FPN, and Mask RCNN have improved the feature extraction methods, feature selection, and classification capabilities of convolutional networks in different ways.

Among the one-stage methods, the most important are the Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO) [8] frameworks. The MutiBox , Region Proposal Network (RPN) and multi-scale representation methods are used in SSD, which uses a default set of anchor boxes with different aspect ratios to more accurately position the object. Unlike SSD, the YOLO [8] network divides the image into a fixed number of grids. Each grid is responsible for predicting objects whose center points are within the grid. YOLOv2 added the BN (Batch Normalization) layer, which makes the network normalize the input of each layer and accelerate the network convergence speed. YOLOv3 uses Darknet53 backbone, YOLOv4 architecture uses CSPdarknet53 as a backbone and YOLOv5 uses Focus structure with CSPdarknet53 as a backbone. The Focus layer was first introduced in YOLOv5, which is why it has the highest accuracy so far and will be used on our dataset.

B. Vehicle Tracking

Tracking movement of vehicles is an advanced and an important application of vehicle object detection known as multi-object tracking. Most multi-object tracking methods use Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT) for object initialization. The DBT method uses background modeling to detect moving objects in video frames before tracking. The DFT method needs to initialize the object of the tracking but cannot handle the addition of new objects and the departure of old objects. The Multiple Object Tracking algorithm needs to consider the similarity of intra-frame objects and the associated problem of inter-frame objects. The similarity of intra-frame objects can use normalized cross-correlation (NCC). The Bhattacharyya distance is used to calculate the distance of the color histogram between the objects [9]. When inter-frame objects are associated, it is necessary to determine that an object can only appear on one track and that one track can only correspond to one object. Currently, detection-level exclusion or trajectory-level exclusion can solve this problem. To solve the problems caused by scale changes and illumination changes of moving objects, [10] used SIFT feature points for object tracking, although this is slow. The ORB feature point detection algorithm [11] is proposed for use in this work. ORB can obtain better extraction feature points at a significantly higher speed than SIFT. However, deep learning algorithms such as DeepSORT, Tracktor, ByteTrack, TransMOT, FairMOT have started gaining popularity due to its high efficiency and state of the art performance.

C. Monitoring Applications

Parking Monitoring System [12] by i+D3 enables an individual to have real time information of all the accesses, exits and payments that occur in your car park. Smart Parking [13] monitors individual parking spaces using inground sesnors and relay occupancy status to our SmartSpot gateways, which in turn send this live status information to the SmartCloud platform, allowing real-time parking information to be viewed on multiple devices. Parklio Parking Monitoring System [14] is another example which collects comprehensive real-time data on the parking lot so that administrators can have an insight into real-time parking information. It is an intelligent AI system for parking monitoring, analyzing, and reporting vehicle parking data in both off and on-street parking, as well as providing safety and security monitoring.

However, to the best of our knowledge no such application is ready to be applied on Pakistan's parking lot due to its high level of clutter and extremely haphazard parking behaviours. Moreover, none of the technology we observed leverages deep learning principle on both detection and tracking modules with a custom dataset.

III. Data Collection

The data collection process started with the acquiring data on a daily basis to generate an initial dataset, followed by manually labelling bounding boxes around cars on the dataset. Some basic preprocessing was done to standardize and refine the dataset followed by data augmentation to increase total number of examples in dataset. At end, the dataset was divided in splits of train, test, valid in order to train a supervised detection model.

A. Data Acquisition

The data was collected roughly around noon from 11:00 AM to 1:00 PM in sunny weather, from an aerial position i.e. the 3rd floor of the building from which the entire parking lot was visible. The camera used was an Iphone 8 camera and its specifications are: 12 MP, f/1.8, 28mm (wide), PDAF, OIS 12 MP, f/2.8, 57mm (telephoto), PDAF, 2x optical zoom and video 4K@24/30/60fps, 1080p@30/60/120/240fps. The images were captured at different times approximately after every 20 minutes. Around 10-12 images were taken at each instance of time, from different angles with different positions of the parking lot. For object detection, a total of 212 images were captured in a time-span of 3 days. A demo video was also recorded with a specific car moving out of its parking space, to check the performance of the object tracking algorithm. Figure 1 displays the original image.

B. Data Annotation and Preprocessing

The data was annotated using LabelImg. LabelImg is a free and open source tool for graphical image annotation. The tool was downloaded and run using Command Prompt, and all 212 images were annotated by drawing a bounding box on every visible car in the image. Only one class of 'car' was labelled. Each image had multiple bounding boxes and the class was kept the same for all bounding boxes i.e. car. Annotated image with bounding boxes is displayed in Figure 2. The annotations file was exported in YOLO format. After complete annotation, the labeled dataset was uploaded on Roboflow and unlabelled images were discarded (none) and preprocessed (auto orientation and resized to 640x640) which will decrease training time and increase performance.

C. Data Augmentation and Splitting

Data augmentation performs transforms on your existing images to create new variations and increase the number of images in your dataset. This ultimately makes models more accurate across a broader range of use cases. For this, the dataset of 212 images was first divided by separating 120 images for the validation and test set and keeping 92 images in the training set. Augmentations were applied only on the training set, which increased its size by 3 times i.e. 96 to 276. The augmentation techniques applied included flipping, cropping, rotation, shear, cutting parts and adjusting levels of hue, brightness, noise and exposure. As a result the total images became 396, with 276 in the training set (70%), 80 in the validation set (20%) and 40 in the test set (10%). Figure 3 shows the augmented image.

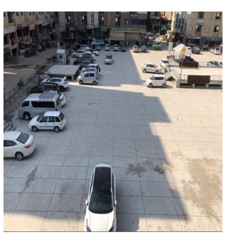

Fig. 1. The original image

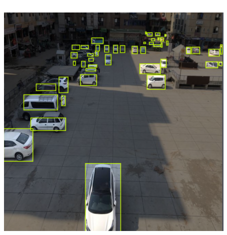

Fig. 2. The annotated image

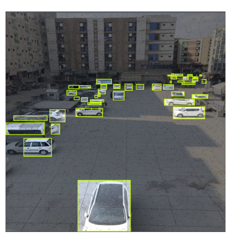

Fig. 3. The augmented image

IV. Methodology

This section will walk through the steps followed to train a working car tracking model deployed on website after data collection. It started with training multiple object detection models and selecting the best model to be used as a base detector for tracking model which was also tested with a custom dataset to compare multiple tracking models and select the best one. The last step was to deploy the model and design

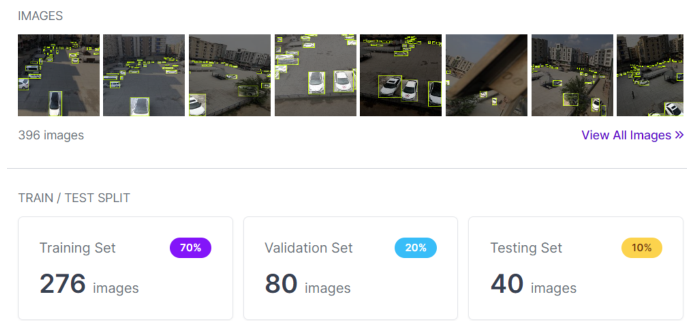

Fig. 4. Statistic of augmented data in Roboflow

the website to make it look presentable. Figure 5 gives the workflow of proposed methodology.

A. Object Detection

Object detection is a technique in computer vision that allows us to identify and locate objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene to determine and track the precise locations by accurately labeling them. An object detection model produces the output in three components: the bounding boxes — x1, y1, width, height (COCO file format), class of the bounding box and the probability score for that prediction— how certain the model is that the class is actually the predicted class. However, in our case we have only one class i.e. car. So the output will return bounding boxes of only one class and a probability score for each prediction. Hence, we trained a total of 10 models on our custom dataset, in order to compare their performance and select the best one. We used primarily five evaluation measures to compare the models. These evaluation measures are precision, recall, mAP@:.5, mAP@.5:.95, latency time.

Precision quantifies the number of positive class predictions that actually belong to the positive class.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The general definition for the Average Precision (AP) is finding the area under the precision-recall curve above and mAP (mean average precision) is the average of AP. The mAP score is calculated by taking the mean AP over all classes and/or overall IoU thresholds, depending on different detection challenges that exist. mAP@.5 means mAP is calculated over a fixed IoU (Intersection over Union) threshold of 0.5 whereas mAP@[.5:.95] means average mAP over different IoU thresholds, from 0.5 to 0.95, step 0.05

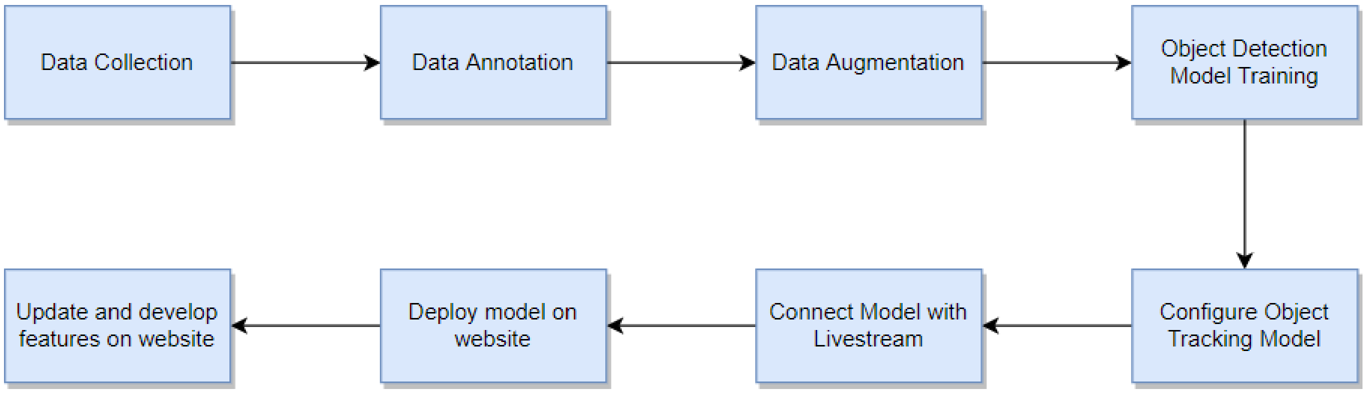

Fig. 5. Flowchart of methodology in developing the application for the Parking Monitoring System. Data collection was done using Iphone 8, annotation using LabelImg, augmentation using Roboflow, and model training and website devolopment was done on Visual Studio Code.

(0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95).

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{3}$$

Latency time here specifically refers to the time taken for each prediction. Its general formula is given as below, where L is average latency, P is average preprocess time, I is average inference time and NMS is average NMS (Non Maximum Suppression) time:

$$L = P + I + NMS \tag{4}$$

However, for consistency purposes only mAPs and latency time is used for the comparison of models. Initially, the model architectures were trained from scratch but this approach was discarded due to low performance. So, it was finalized that there is a need to fine tune all the models using its pretrained weights and compare the latency time and mAPs using the test set of our custom dataset.

B. *Object Tracking*

Next, phase is object tracking which is basically a deep learning process where the algorithm tracks the movement of an object. In other words, it is the task of estimating or predicting the positions and other relevant information of moving objects in a video. Object tracking usually involves the process of object detection for which we will use the weights of the best model selected in the previous stage. Object tracking involves three basic steps:

- Object detection, where the algorithm classifies and detects the object by creating a bounding box around it.
- Assigning unique identification for each object (ID).
- Tracking the detected object as it moves through frames while storing the relevant information.

For object tracking, three main architectures were analyzed and compared using MOTA (multiple object tracking accuracy) on a pre-recorded test video. These architectures were Tracktor++, StrongSORT and DeepSORT. MOTA measures the overall accuracy of both the tracker and detection. It deals with both tracker output and detection output.
An additional security feature is also implemented with object tracking. This feature works by calculating and storing the centers of all detected bounding boxes in the initial frame of the stream. Whenever, the center of the bounding boxes changes, our algorithm will check if the euclidean distance between two points in time, of the center of bounding box has exceeded a certain threshold (set at 100 pixels). The algorithm will generate an alert. As it stores the unique id number of the car and extracts the image inside the detected bounding box of the moved car, the details of the moved cars will be sent and displayed on the front end.

C. *Model Integration and Website Design*

Once, the best model for object tracking is selected, next there is a need to save the model and serve it by using a Flask, a micro web framework written in Python (it's referred to as a "micro" framework because it doesn't require particular tools or libraries). For this purpose, a web application is designed that tracks the position of multiple cars on a live video stream using object tracking. To obtain outputs statistics for hourly detections of the car using object detection, two routes were added on the flask app for each feature. The front end design of the website is built using HTML, CSS and Bootstrap (an open source CSS framework), and a website with 4 pages is created:

- Dashboard: will include multiple statistics about the cars parked in the parking lot over a period of time, such as maximum cars parked on a single instance today and yesterday, and average numbers of cars parked in a week.

- Livestream: will include an embedded lievstream directly from our parking lot, with the bounding boxes from the tracking model directly visualized on the video.
- Security system: will display a list of all the moved vehicles with its unique identification number from tracking model and its image extracted from the detected bounding box.
- About us: will show a short summary about what features the website offers and its importance.

V. Results

To deduce the best model for object detection, we leveraged a balance between performance and efficiency. Table I. shows the results after training or fine tuning the each model. In terms of performance, YOLOX outperforms all models with a mAP@.5 of 86.9% accuracy followed by YOLOV7 with an accuracy of 80.3%, whereas in terms of speed YOLOv5 proved to be the fastest with an average latency of 5.6ms and a decent performance followed by RetinaNet (8.1) and CenterNet (9.5) at second and third but with a poor performance respectively. It was later observed that due to an additional implemented security feature, object detection prediction time proved to be a bottleneck in the website speed. Hence, more importance was given to the latency speed, due to which YOLOv5 was selected as the best model with the fastest prediction time and above average performance i.e. ranked 4th best out of 10 models.

TABLE I
Object Detection Model Statistics

Model	mAP@.5	mAP@ 5:.95	Latency (ms)	Epochs
YOLOv5	73.1	35.5	5.6	150
YOLOv6	55.4	22.6	13.1	150
YOLOv7	80.3	40.3	19.2	80
YOLOX	86.9	48.4	15.5	100
YOLOR	78.7	38.3	15.4	100
YOLOS	40.2	14.4	15.1	75
RF Train v2	77	37.5	-	200
RetinaNet	56	19	8.1	80
CenterNet	64.2	21.2	9.5	50
EfficientDet	49.2	16.4	23.1	80

The YOLOv5 weights were stored using .pt extension and then were used to test the three architectures of object tracking. The clear winner was StrongSORT, which is an upgraded version of DeepSORT and gave a MOTA of 83% and an average prediction time of 0.27 sec/frame. Similarly as observed in Table II, the StrongSORT tracking algorithm was selected and the security feature was implemented in it, which exceeded the average prediction time from 0.27 sec/frame to 0.35 sec/frame, which was still better than the rest of the algorithms. In the end, these two models were integrated into the website and deployed using Flask and all the four pages on the front end were created successfully.

VI. Discussion

YOLOv5's speed in identifying objects made implementing a web application that received live data efficient. The web application consists of a dashboard where statistical data can be viewed. Figure 6 shows a view of the dashboard.

TABLE II
Object Tracking Model Statistics

Model	MOTA	IDF1
StrongSORT	92.3	12
DeepSORT	76.9	10
CSRT	69.2	9
Tracktor++	46.2	6

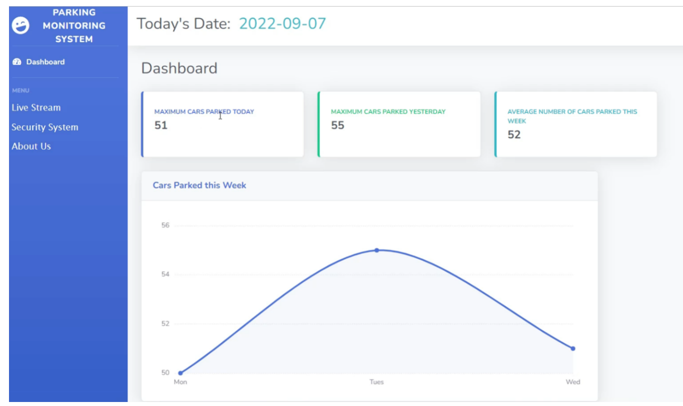

Fig. 6. Dashboard of Parking Monitoring System

We used the Yolov5 model to identify cars in each frame of the received live-stream. This was used to count the total number of cars parked on the current date. Then the maximum number cars parked that day were shown. Figure 7 shows that feature.

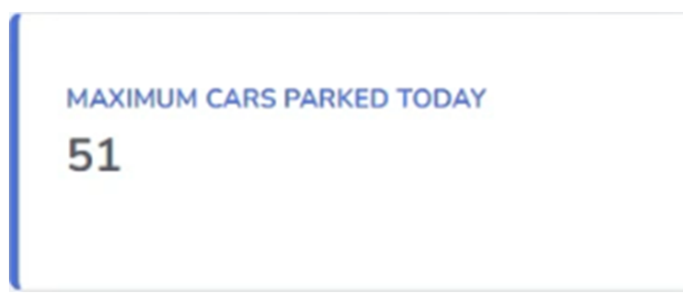

Fig. 7. Maximum Cars Parked for the current date

The same data was used to find the maximum number of cars a day before the current date and was displayed as shown in the Figure 8.

The Average number of cars parked the entire week were also displayed as shown in Figure 9.

The total number of cars parked was also displayed as a chart on the dashboard. Figure 10 shows that chart. One can hover over the chart to view the total count for that day.

A table displays the hourly count of cars by counting the cars parked per hour in the parking lot. Figure 11 shows that table.

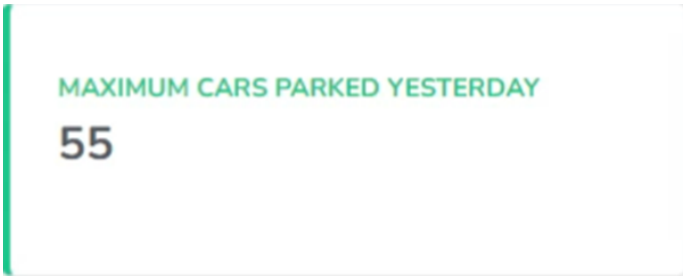

Fig. 8. Maximum Cars Parked yesterday

Fig. 9. Average number of cars parked the entire week

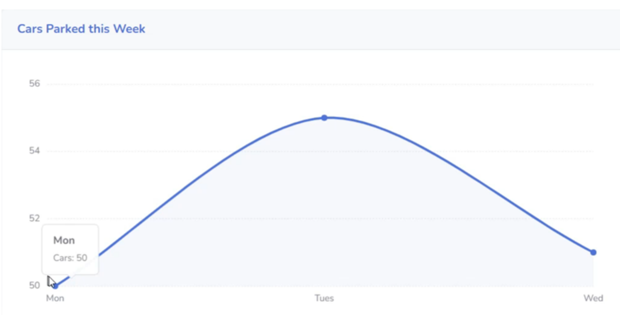

Fig. 10. Total number of cars parked each day for the entire week

Cars Parked per hour — See Maximum

Time (24h)	Cars Parked
01	19
02	23
03	33
04	40
05	39
06	42
07	45
08	51

Fig. 11. 24 hours data showing cars parked per hour for the current date

Another feature of the parking monitoring system is the live stream view which encloses identified cars in bounding boxes and the accuracy associated with correct identification. Figure 12 shows the live stream video view.

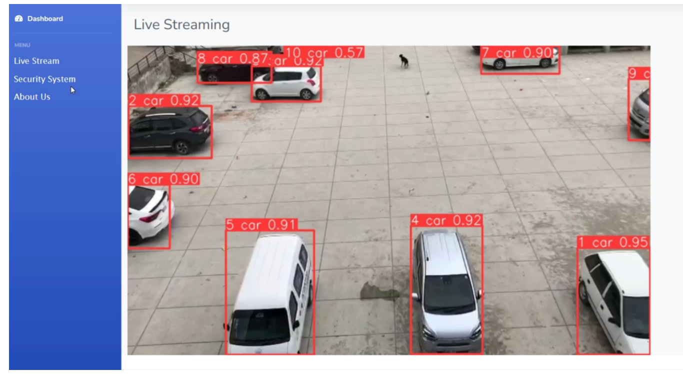

Fig. 12. Live-stream of the parking lot

A Security feature has been implemented that tracks any movement by a car. Object tracking was done by the StrongSORT algorithm. An alert is generated in table which shows the moved car image and vehicle number. Figure 13 shows the implemented security feature.

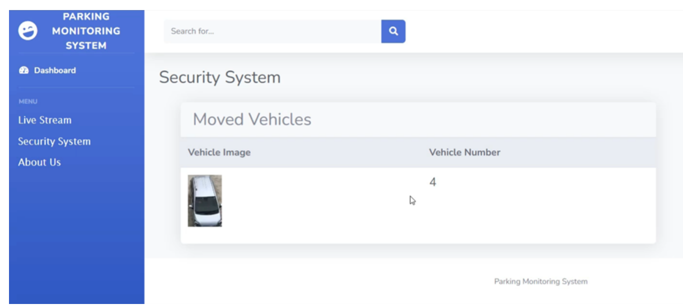

Fig. 13. Security Feature for tracking car movements

VII. Conclusion

The role of computer vision-assisted technologies in monitoring and surveillance is by now very common. But countries like Pakistan continually fail to utilize this technology for monitoring cars in parking spaces, where it is most vulnerable to be stolen. Besides the security reason, the analysis of parking spaces provides us with insightful information the peak hours, customer trends and efficient resource management of the parking space, etc. Computer vision technology also has not yet been applied to such areas with extreme clutter and overlap where accuracy of the model holds immense value. Hence, this study proposed a YOLOv5 object detection model with 73.1% accuracy of mAP@.5 and object tracking model for StrongSORT with MOTA of 92.3%. However, a current limitation is inability to identify car owner based on car image instead of unique identification numbers, For this, we intend to train a classification model using the employees of a specific company (Neurog in our case), and only detect and monitor those cars, which will eventually reduce model computation and memory overhead. In the whole study, we have demonstrated how to automate the task of car parking monitoring with tracking models and achieving insightful statistics on an interactive dashboard through a user-friendly website.

Transitioning to car parking space monitoring will ultimately reduce the manual labour involved in the process and setting an example for other developing countries. Security firms, private companies, and commercial areas can benefit hugely from this technology, giving state of the art mAP and MOTA model accuracies, with continuous improvements promised in the future.

REFERENCES

[1] K. Abdulrahim and R. A. Salam, "Traffic surveillance: A review of vision based vehicle detection, recognition and tracking," *International journal of applied engineering research*, vol. 11, no. 1, pp. 713–726, 2016.

[2] M. Radhakrishnan, "Video object extraction by using background subtraction techniques for sports applications," *Digital Image Processing*, vol. 5, no. 9, pp. 91–97, 2013.

[3] Q.-L. Li and J.-F. He, "Vehicles detection based on three-frame-difference method and cross-entropy threshold method," *Jisuanji Gongcheng/ Computer Engineering*, vol. 37, no. 4, 2011.

[4] Y. Liu, Y. Lu, Q. Shi, and J. Ding, "Optical flow based urban road vehicle tracking," in *2013 Ninth International Conference on Computational Intelligence and Security*, 2013, pp. 391–395.

[5] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *CoRR*, vol. abs/1807.05511, 2018. [Online]. Available: http://arxiv.org/abs/1807.05511

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[9] J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1200–1207.

[10] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009, special Issue on Video Analysis. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314208001331

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[12] "Parking monitoring system," Oct 2017. [Online]. Available: https://imasdetres.com/en/parking-monitoring-system/

[13] S. parking, "Vehicle detection sensors," Apr 2022. [Online]. Available: https://www.smartparking.com/smartpark-system/smart-sensors

[14] "Parklio smart parking pms - innovative parking management system." [Online]. Available: https://parklio.com/en/parking-solutions/parking-management-system