Vehicle Detection System on Dynamic Camera Sources using CNN and SORT Tracker

Fae Nicole C. Serrano*1, Elmer P. Dadios*2, Argel A. Bandala*1, Robert Kerwin C. Billones*2, John Carlo V. Puno*2, Allysa Kate M. Brillantes*1 and Jay Robert B. del Rosario*1

*1Electronics and Communications Engineering Department
2Manufacturing Engineering and Management Department
Email: fae_nicole_serrano@dlsu.edu.ph
Gokongwei College of Engineering, De La Salle University
Manila, Philippines

Abstract. With the increasing popularity of dashcam usage, there are other applications of these devices that can contribute to Intelligent Transport Systems such as road surveillance, law violation detection, and autonomous driving. This study mainly tackles detecting vehicles in a dynamic background, tracking vehicles despite varying angle views (front, side and, back views) and extracting vehicular information (vehicle type or license plate presence). Several CNN models were compared based on accuracy and speed in order to identify the architecture best suited for the set-up. The result shows that FRCNN Inception, is the choice for accuracy-based implementation with 85% mAP and SSD MobileNet for speed-based implementation with a frame rate of 7.81fps on a GPU.

Keywords: Intelligent Transport System, Vehicle Detection, Image Processing, Convolutional Neural Network, SORT Tracker

1. INTRODUCTION

Intelligent Transport System (ITS) is the integration of computers, sensors, cameras, and other communication devices designed to work together in order to give services that are innovated for transport and traffic management. It utilizes information and communication technologies to safely and efficiently use transportation infrastructure and networks [1][2]. ITS caters to people, cars and infrastructures. The concept of vehicle detection and tracking is an important role in the field of traffic surveillance system where effective traffic management and safety is the primary focus [3]. Vehicle detection plays is the first step in video processing. The efficiency & accuracy of vehicle detection is of great importance for vehicle tracking, vehicle movement expression, and behavior understanding and is the basis for subsequent processing. Appearance based techniques utilizes the vehicles physical characteristics such as color, shape, and texture while motion-based techniques utilizes the moving characteristic to differentiate a moving vehicle from the stationary background. Vehicle tracking is a significant and challenging research field in image processing which is widely utilized in computer vision and video image [4]. Deep learning and Convolutional Neural Network (CNN) revolutionize image, video, speech and audio processing. CNN is an artificial neural network that uses perceptron, a machine learning unit algorithm, for supervised learning, to analyze data. In 2012, Kriszhevsky [5] introduced CNN in ImageNet that changed the standard for image classification. Deep learning approach is a discriminative method or tracking-by-detection that has a more robust approach in tracking [6]. Thus, more projects adopted deep learning and CNN. In the scenario of road video surveillance, vehicle detection and classification play a significant role as it can be used in traffic control and gathering of relevant road statistics that can be utilized in intelligent transportation systems [7]. Vehicle classification is a crucial component in traffic management software. Learning and tracking the type of vehicle is beneficial since it allows queries when a particular vehicle is inspected. Possible queries that it can accommodate is tracking when did vehicle x passed this way or where did vehicle x go. Due to this, vehicle classification has a wide range of applications such as road monitoring (vehicle count, vehicle type detection, vehicle trajectory), intelligent parking systems, law enforcement, and emergency vehicle prioritization [8][9].

There is an increasing popularity in dashboard cameras due to consistent drop in price in the market, emergence of driving assistance systems, and influence of insurance companies [10]. Dashboard camera or dashcam is a type of surveillance camera that is mounted in the vehicle. It starts operating as soon as the engine is started and continuously records during the drive. There are several types of dashcams such as front-view cameras or front and rear cameras, and cabin view dash cams for taxis. Aside from its personal safety usage, the application of dashcam can be extended in the domain of Intelligent Transport System.

The main problem considered in this study is the limited capability of the existing vehicle classification systems implemented in static cameras. Current system utilizes a single static camera with a designated area to focus on. However, this kind of set-up can be considered disadvantageous in a larger scale coverage in terms of resources since additional single static camera systems are needed in proportion with the area coverage thus resulting in a significant resource consumption.

Here are some of the problem points of the current vehicle classification system:

1. Implemented in a fixed area or limited view.
2. Lower video quality, dynamic background, camera positioning and angles are less taken into consideration.
3. Limited viewpoints in vehicle classification.
    a. Current static system mainly focuses on one to two views (front, side or back)
    b. Dashcam system focuses on front, side and back view of the vehicles.
4. Minimal utilization given the increasing dashboard camera resources.

Moving object detection is based on describing objects in motion by binary mask per frame. It is a significant issue and vital in many vision-based applications. For fixed camera scenarios, it is the difference between successive frames that are caused by moving objects. For moving camera scenarios, the method of moving object detection also considers the difficulties due to compensation of camera motion. Thus, background subtraction with simple motion compensation model is not applicable in this scenario. There are different methods for moving objects for dynamic background namely: background modeling, trajectory classification, low rank and sparse representation, and object tracking. Table 1 shows the difference between these four approaches.

Table. 1 Summary of Different Object Detection Methods

Category	**Negative Points**	**Positive Points**	**References**
Background modelling	- Performance is not suitable for camera motion - Performance is highly dependent on background model	- Moderately Complex - Suitable for real time	[12] [13] [14] [15] [16] [17] [18]
Trajectory classification	- Performance is highly dependent on motion tracker model - Highly sensible to noise	- Moderately complex - Gives proper trajectory over time	[19] [20] [21]
Low rank and sparse representation	- Highly sensible to noise	- Shows good performance accuracy	[22] [23] [24] [25]
Object tracking	- Highly sensible to noise	- Good performance with regards to camera motion scenario - Moderate complexity	[26] [27] [28] [29]

Based on Table 1, the approach of object tracking is the most appropriate in this study. Object tracking relates different regions from the same object in succeeding video frames. Tracking is done through localization of a moving objects through frames thus it is regarded as process of moving object detection [11]. Figure 1 shows the general concept of object tracking.

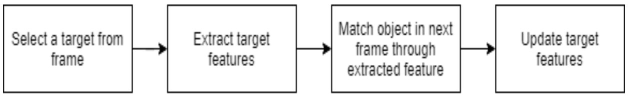

Select a target from frame
Extract target features
Match object in next frame through extracted feature
Update target features

Fig. 1 General Concept of Object Tracking for Moving Camera

Convolutional Neural Network (CNN) is used to extract appearance features. It can be trained to learn descriptors encoding local spatial-temporal features [30]. CNN learns features in an adaptive and distributed way that is suitable when considering appearance alteration [31].

In object tracking, Gan et al [32] proposed another end-to-end trainable model that combines CNN and RNN. It differs from traditional approaches since full pipeline of visual tracking is jointly tuned to maximize the tracking quality. However, this model is yet to be tested on the natural environment. The training is also offline and is not yet adaptable on an online pre-trained model.

Aside from single object tracking, study of Agarwal [33] combined Faster R-CNN with modified GOTURN (Generic Object Tracking Using Regression Networks) architecture for a real-time multiple object tracker (MOT). Result shows that deep learning techniques are at par with conventional multi-object tracking techniques. Milan et al [34] proposed the first an end-to-end learning for online multi-target tracking based on RNN. This was followed by [35] that applied RNN with multiple cues over a temporal window. It is an online method that encodes long-term temporal dependencies across multiple cues. It allows to correct data association errors and recover observations from occluded states.

2. METHODS AND MATERIALS

2.1. System Setup and Design

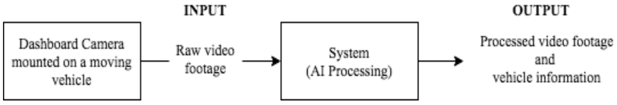

INPUT
OUTPUT
Dashboard Camera mounted on a moving vehicle
Raw video footage
System (AI Processing)
Processed video footage and vehicle information

Fig. 2 System Set-up Block Diagram

The source of input in the set-up is through a dashboard camera mounted in a moving vehicle for capturing road footage that will be used as the input video to the system. Once a road footage has been collected and saved, it will be fed in the system for post processing. The system is responsible for all the processing in the study and the implementation of the algorithm. The detailed discussion of the system will be discussed in the following section. The system will output the final processed video appended with the relevant vehicle information such as the vehicle location with regards to the video frame and other distinguishable information.

2.2. Dataset Preparation

Dashcam road footages were gathered and compiled in order to collect images of different vehicular types with different viewpoints from dashcams which will be used for the training and testing datasets. Since the setup of the proposed system is a moving background with vehicles moving in and out of the frame, the viewpoint coverage of the vehicles includes all angles of the front view, side view, and back view of vehicles. The vehicles were classified into 8 classes: Car (sedan, SUV, and van), Truck, Jeepney, Tricycle, Motorcycle, and Bus. The figure below shows the sample images of the data set.

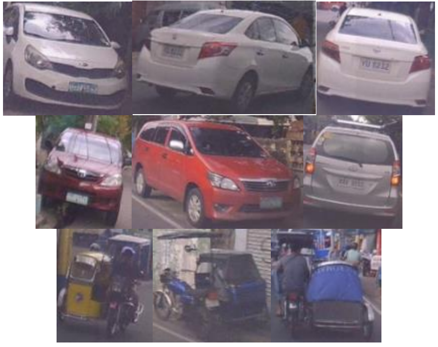


Fig. 3 Sample Vehicle Dataset of Sedan, SUV, and Tricycle with Multiple View

The dataset consists of videos with a total of 36 minutes of road footage with 5040 frames consisting of 8603 images of different vehicles. The complete annotated dataset was converted into a tfrecord which will be used in the training process of the model.

2.3. Program Design and Construction

The program design follows the ideas and development of the system that will comply with the research's objectives. Figure 4 below shows the logic overview of the system in terms of tracking vehicles.

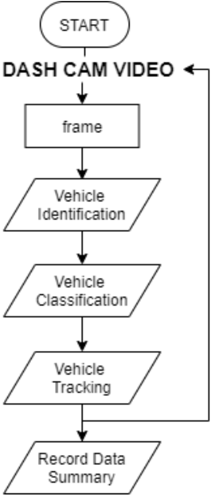


Fig. 4 Algorithm Flowchart Overview

The algorithm as shown in figure 5 will apply as a post processing technique in the dash cam videos. Once the vehicle stops recording, the video footage will be extracted and will be fed in the system. The video's frame by frame will pass through the trained model. The model will locate all the vehicles inside the frame and classify the type of the vehicles. The algorithm will return the vehicle type, bounding box coordinates, and confidence value that will be fed in the SORT (Simple Online and Real Time) [36] tracker to track the vehicles across the video. It will append a unique id tag per vehicle tracked. CNN models in Table 2 will be trained with the researcher's dataset to recognize vehicles from a dashcam's point of view. The annotated data will be used to retrain the object classification models. Transfer learning will be applied in retraining the models to take advantage of the learned feature maps of the models.

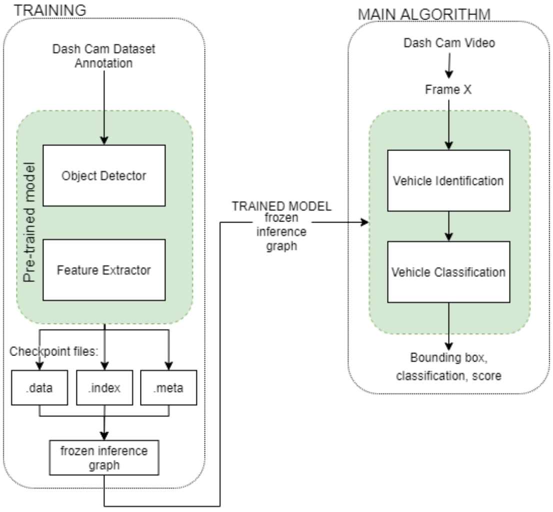


Fig. 5 Vehicle Training and Module Integration Overview

The table below shows the list of model architectures trained. These models were chosen based on two factors, speed and accuracy.

Table. 2 List of Models Trained

Model Architecture	Last trained from
SSD-MobileNet	COCO dataset
FasterRCNN-ResNet	Static CCTV vehicle dataset
FasterRCNN-Inception	Static CCTV vehicle dataset
FasterRCNN-Inception	COCO dataset
FasterRCNN-ResNet	COCO dataset

Based on the comparison study made [9], Faster RCNN, R-FCN and SSD are the modern 'meta-architectures' in object detection. Thus, FRCNN models are used to observe the accuracy-oriented model and SSD for speed-oriented models in order to compare performance in this study's set-up. Most of the models used were pretrained from COCO dataset. In addition, two models gathered, Faster RCNN-Inception and Faster RCNN-Resnet, were last trained from the dataset consisting of different types of vehicles from a static CCTV viewpoint from an open source video. The static vehicle dataset consists of a 45-minute long video footage capturing top front view vehicles as shown in Figure 6 with 10 vehicle classes (car, suv, van, truck, taxi, motorcycle, bicycle, tricycle, bus, jeep). Utilizing several model architectures pretrained from different sources with tailored configuration will help assess what model is best suited for this set-up by comparing its evaluation results and observing its performance on the test videos.

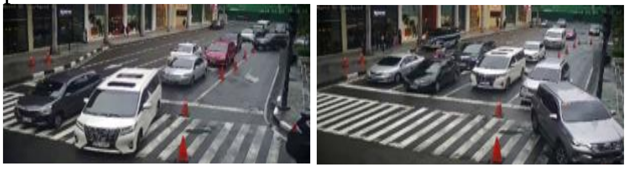


Fig. 6 Sample Images from CCTV Vehicle Dataset

Retraining these models will result in updated weights and new checkpoint files: model.ckpt.data, model.ckpt.index, and model.ckpt.meta. Meta file stores the graph structure. Data file stores the values for all the variables of the graph. Index files identifies the checkpoint. These files will be converted to a frozen inference graph through a protobuf file that will be utilized in the system implementation as shown in Figure 5. Once the training is finished, checking its performance can be determined by running its evaluation code.

2.4. Testing and Deployment

The algorithm will be tested in order to assess its compliance to the design. For vehicle detection and classification, the average precision (AP), mean average precision (mAP) and speed were used in order to show the accuracy of each model in classifying vehicles. Precision determines the accuracy of prediction or the percentage of the results that are relevant. The average precision will be used in order to compare the precision per vehicle types per model. The mAP will be used to compare the overall precision per model. The mAP is computed as the average of all the AP per class. The processing time per video is also recorded in order to compare the overall processing speed of the different models used.

3. Results and Discussion

3.1. Performance Comparison of Trained Models

Five trained models (see Table 2) were evaluated in terms of their average precision. The figure below shows the comparison of the performance per model per vehicle type. Same testing dataset was fed into these models.

Table. 3 Precision Summary (7 Classifications)

	FRCNN Inception COCO	FRCNN Resnet COCO	FRCNN Inception CCTV	FRCNN Resnet CCTV	SSD Mobile Net COCO
Sedan	0.9470	0.8771	0.8774	0.9004	0.5926
Jeep	0.9412	0.9313	0.883	0.9494	0.8589
Motor	0.9512	0.9409	0.9207	0.9313	0.7711
Bus	0.9597	0.9502	0.9211	0.9662	0.6176
Tricycle	0.9013	0.8777	0.8899	0.8936	0.577
SUV	0.7090	0.5732	0.7900	0.6087	0.6917
Van	0.7099	0.3974	0.4899	0.4999	0.4021
Truck	0.6807	0.4186	0.4911	0.4287	0.2391
Ave. mAP	**0.8500**	**0.7458**	**0.7829**	**0.7723**	**0.5938**

Comparing per class, result showed that the models can easily distinguish sedans, jeeps, motorcycles, tricycles, and buses since those types of vehicles exhibit distinct features. However, SUV, vans and trucks have lower precision score. This is due to the factor that the models have difficulty distinguishing these classes due to either some common shared features to other vehicle types or complexity of the vehicle type.

Comparing per model, the FRCNN Inception (COCO and CCTV) gave the best precision scores, followed by FRCNN ResNet (CCTV and COCO) and lastly, SSD MobileNet.

3.2. Reduced Class

As mentioned, there were classes where models have difficulty in differentiating due to some vehicle type similarities and vehicle type. Thus, in the next section, sedan, SUV and van were merged in one class under car. Below are the observations made in merging of the said classes.

Table. 4 Precision Summary (Reduced Class)

	FRCNN Inception COCO	FRCNN Resnet COCO	FRCNN Inception CCTV	FRCNN Resnet CCTV	SSD Mobile Net COCO
Jeep	0.9441	0.9031	0.9029	0.9348	0.7525
Tricycle	0.9509	0.9216	0.9108	0.8202	0.5544
Motor	0.9558	0.9391	0.9244	0.9384	0.8513
Bus	0.9320	0.9532	0.9397	0.9544	0.9276
Car	0.7320	0.6380	0.7214	0.6881	0.7278
Truck	0.5038	0.5113	0.5337	0.5010	0.4947
Ave. mAP	**0.8365**	**0.8111**	**0.8222**	**0.8062**	**0.7181**

Comparing per class, results showed that models can easily distinguish jeeps, motorcycles, tricycles, and buses. Cars can be relatively distinguished. Possible reasons for the truck's low precision is due to the need of more truck training dataset to capture truck's unique feature. Some type of trucks, such as pick-up trucks, share similar features to SUV or sedan from its front view. The moment the program views the side of the vehicle showing tailgate portion, then it will be able to recognize that it is a truck.

Comparing per model, the COCO-pretrained FRCNN models (FRCNN Inception and ResNet) gave the best precision, followed by the CCTV-pretrained models (FRCNN Inception and ResNet) and lastly, by SSD MobileNet. Due to the merging of class types, the overall precision of the models increased compared to previous performance on the 8 vehicle types.

3.3. Processing Time

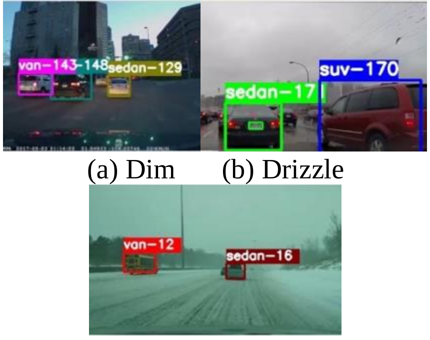
For the processing time, Table 5 shows the summary of the different models run on the same video consisting of 1521 frames. As observed the SSD model performed the fastest compared to the FRCNN models.

Table. 5 Models' Processing Time Summary

Model	Total Time (sec)	(fps)
SSD Mobilenet COCO	194.742	7.810
FRCNN Inception COCO	359.656	4.229
FRCNN Inception CCTV	365.027	4.167
FRCNN ResNet COCO	805.653	1.888
FRCNN ResNet CCTV	599.638	2.537

3.4. Performance on Weather Variation

Given that FRCNN Inception COCO is the best performing model in terms of accuracy, this model was tested in dimmed, drizzling, and snowing dashcam videos demonstrating change in weather condition/surrounding.



(a) Dim (b) Drizzle

(c) Snowy

Fig. 7 Sample Screenshot of Different Weather Conditions

For the dimmed videos, the model was able to detect 72.09% of the vehicles within 1,800 frames and out of all the detected vehicles, the model was able to correctly classify 69.73% of the detected vehicles. For the drizzling video, the model was able to detect 64.65% of the vehicles within 3,378 frames and out of all the detected vehicles, the model was able to correctly classify 81.00% of the detected vehicles. For the snowing video, the model was able to detect 72.67% of the vehicles within 2,064 frames and out of all the detected vehicles, the model was able to correctly classify 88.53% of the detected vehicles. Challenges faced are due to the foggy environment that clouds or distorts the image of a vehicle and to the weather of the environment where vehicles utilized windshield wipers that obstructs the view of the dashcam. Another challenge faced is the distance of the vehicles since most of the vehicles encountered in the footage are distant. The model was trained in larger images or near vehicles thus the model cannot easily detect some vehicles in the footage. These obstructions caused misdetections to some vehicles.

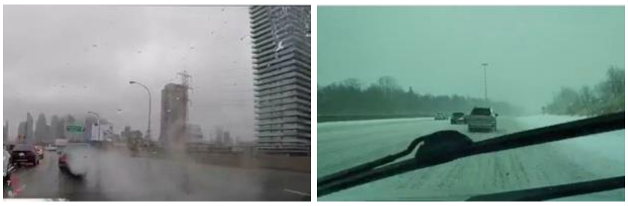
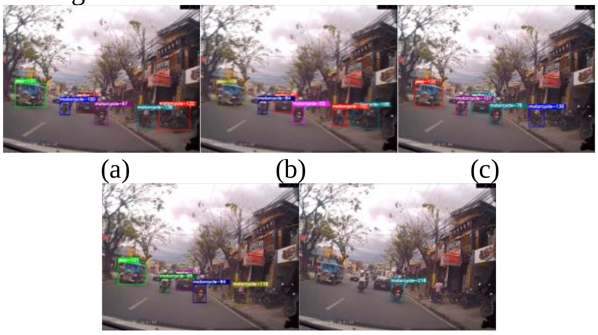


Fig. 8. Sample Obstructions

3.5. Actual Road Footage

The following figures below show sample screenshots comparison of the models on an actual dashcam footage within the same frame. FRCNN Inception models detects more vehicles in a frame, FRCNN ResNet models gives a fair performance, SSD MobileNet detects fewer images compared to other models. FRCNN Inception COCO detected and classified all the vehicles correct. Other models got 1 or 2 misclassifications.



(a) (b) (c)

(d) (e)

Fig. 9. (a) FRCNN Inception COCO (b) FRCNN Inception CCTV (c) FRCNN ResNet COCO (d) FRCNN ResNet CCTV (e) SSD MobileNet COCO

4. CONCLUSION

This paper showed vehicle detection in a dynamic background by utilizing neural network models for detection and SORT for tracking. Several neural networks were tested and implemented in order to assess which model will best fit the scenario. Among the five, it is shown that Faster RCNN models, specifically FRCNN Inception COCO, is the choice for accuracy-based implementation and SSD for speed-based implementation. SORT tracker was able to track the detected vehicles within the given region of interest handling minimal occlusion or small re-identification scenarios. For the challenge encountered in this study, a factor to consider is the changes in the environment. Change in weather may affect the performance of the system. The algorithm also handles minimal occlusion however re-entering of vehicles is not within the scope of the algorithm. It is hoped that these can be addressed in the succeeding studies.

Acknowledgements

The authors would like to thank ERDT-DOST and Dr. Jonathan Dungca, Dean of the Gokongwei College of Engineering and Dr. Edwin Sybingco, Faculty member from the Electronics and Communications Engineering Department of De La Salle University-Taft, Manila.

REFERENCES:

[1] E. Jonkers and T. Gorris, “Intelligent Transport Systems and traffic management in urban areas,” 2015.

[2] C. Draganescu, C. Popa, and A. C. Tundrea, “Context-Aware Adaptive System for Intelligent Transport Management,” *in Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, 2017, doi: 10.1109/CSCS.2017.59.

[3] P. K. Bhaskar and S. P. Yong, “Image processing based vehicle detection and tracking method,” in *2014 International Conference on Computer and Information Sciences, ICCOINS 2014 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2014 - Proceedings*, 2014, doi: 10.1109/ICCOINS.2014.6868357.

[4] P. A. Kandalkar and G. P. Dhok, “Review on Image Processing Based Vehicle Detection & Tracking System,” vol. 3, no. 8, pp. 566–569, 2017.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM,* 2017, doi: 10.1145/3065386.

[6] X. Feng, W. Mei, and D. Hu, “A Review of Visual Tracking with Deep Learning,” 2016, doi: 10.2991/aiie-16.2016.54.

[7] A. Ambardekar, M. Nicolescu, G. Bebis, and M. Nicolescu, “Vehicle classification framework: a comparative study,” *Eurasip J. Image Video Process.,* 2014, doi: 10.1186/1687-5281-2014-29.

[8] S. Kul, S. Eken, and A. Sayar, “A concise review on vehicle detection and classification,” in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017,* 2018, doi: 10.1109/ICEngTechnol.2017.8308199.

[9] R. N. Celso, Z. B. Ting, D. J. R. Del Carmen, and R. D. Cajote, “Two-Step Vehicle Classification System for Traffic Monitoring in the Philippines,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2019, doi: 10.1109/TENCON.2018.8650420.

[10] Mordor Intelligence, “DASHBOARD CAMERA MARKET - GROWTH, TRENDS, AND FORECAST (2020 - 2025),” 2020.

[11] M. Yazdi and T. Bouwmans, “New trends on moving object detection in video images captured by a moving camera: A survey,” *Computer Science Review.* 2018, doi: 10.1016/j.cosrev.2018.03.001.

[12] C. Cuevas, R. Mohedano, and N. García, “Statistical moving object detection for mobile devices with camera,” *2015 IEEE Int. Conf. Consum. Electron. ICCE 2015*, pp. 15–16, 2015, doi: 10.1109/ICCE.2015.7066301.

[13] A. Viswanath, R. K. Behera, V. Senthamilarasu, and K. Kutty, “Background Modelling from a Moving Camera*,” Procedia Comput. Sci.,* vol. 58, pp. 289–296, 2015, doi: 10.1016/j.procs.2015.08.023.

[14] S. Minaeian, J. Liu, and Y. J. Son, “Effective and Efficient Detection of Moving Targets from a UAV’s Camera*,” IEEE Trans. Intell. Transp. Syst.,* vol. 19, no. 2, pp. 497–506, 2018, doi: 10.1109/TITS.2017.2782790.

[15] M. Babaee, D. T. Dinh, and G. Rigoll, “A deep convolutional neural network for video sequence background subtraction*,” Pattern Recognit.*, vol. 76, pp. 635–649, 2018, doi: 10.1016/j.patcog.2017.09.040.

[16] L. Gong, M. Yu, and T. Gordon, “Online codebook modeling based background subtraction with a moving camera*,” 2017 3rd Int. Conf. Front. Signal Process. ICFSP 2017,* pp. 136–140, 2017, doi: 10.1109/ICFSP.2017.8097157.

[17] Y. Wu, X. He, and T. Q. Nguyen, “Moving Object Detection with a Freely Moving Camera via Background Motion Subtraction,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 27, no. 2, pp. 236–248, 2017, doi: 10.1109/TCSVT.2015.2493499.

[18] Y. Zhu and A. Elgammal, “A Multilayer-Based Framework for Online Background Subtraction with Freely Moving Cameras,” *Proc. IEEE Int. Conf. Comput. Vis.,* vol. 2017-Octob, pp. 5142–5151, 2017, doi: 10.1109/ICCV.2017.549.

[19] S. Zhang et al., “Tracking Persons-of-Interest via Unsupervised Representation Adaptation,” *Int. J. Comput. Vis.,* 2019, doi: 10.1007/s11263-019-01212-1.

[20] S. Singh, C. Arora, and C. V. Jawahar, “Trajectory aligned features for first person action recognition,” *Pattern Recognit.,* vol. 62, pp. 45–55, 2017, doi: 10.1016/j.patcog.2016.07.031.

[21] X. Yin, B. Wang, W. Li, Y. Liu, and M. Zhang, “Background subtraction for moving cameras based on trajectory-controlled segmentation and label inference,” *KSII Trans. Internet Inf. Syst.,* 2015, doi: 10.3837/tiis.2015.10.018.

[22] C. Chen, S. Li, H. Qin, and A. Hao, “Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis*," Pattern Recognit., 2016*, doi: 10.1016/j.patcog.2015.09.033.

[23] G. Chau and P. Rodríguez, “Panning and Jitter Invariant Incremental Principal Component Pursuit for Video Background Modeling,” *J. Electr. Comput. Eng.,* vol. 2019, 2019, doi: 10.1155/2019/7675805.

[24] C. Gao, B. E. Moore, and R. R. Nadakuditi, “Augmented robust PCA for foreground-background separation on noisy, moving camera video,” *2017 IEEE Glob. Conf. Signal Inf. Process. Glob. 2017 - Proc.,* vol. 2018-Janua, no. 3, pp. 1240–1244, 2018, doi: 10.1109/GlobalSIP.2017.8309159.

[25] S. E. Ebadi, V. G. Ones, and E. Izquierdo, “EFFICIENT BACKGROUND SUBTRACTION WITH LOW-RANK AND SPARSE MATRIX DECOMPOSITION Queen Mary University of London , † Delft University of Technology,” *Int. Conf. Image Process.,* 2015.

[26] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, “Enhancing Detection Model for Multiple Hypothesis Tracking,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.,* vol. 2017-July, pp. 2143–2152, 2017, doi: 10.1109/CVPRW.2017.266.

[27] M. Zhai, L. Chen, G. Mori, and M. J. Roshtkhari, “Deep learning of appearance models for online object tracking*," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),* vol. 11132 LNCS, pp. 681–686, 2019, doi: 10.1007/978-3-030-11018-5_57.

[28] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, “Transferring Rich Feature Hierarchies for Robust Visual Tracking,” 2015.

[29] M. A. Bagherzadeh and M. Yazdi, “Fast object tracking with long-term occlusions handling in dynamic scenes,” *2014 2nd RSI/ISM Int. Conf. Robot. Mechatronics, ICRoM 2014,* pp. 823–827, 2014, doi: 10.1109/ICRoM.2014.6991006.

[30] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, “Learning by Tracking: Siamese CNN for Robust Target Association*," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 418–425, 2016, doi: 10.1109/CVPRW.2016.59.

[31] J. Huang, X. Yang, and M. Yang, “Hierarchical Convolutional Features for Visual Tracking - Jia-Bin Huang’s Homepage*," Int. Conf. Comput. Vis.,* pp. 3074–3082, 2015, doi: 10.1109/ICCV.2015.352.

[32] Q. Gan, Q. Guo, Z. Zhang, and K. Cho, “First Step toward Model-Free, Anonymous Object Tracking with Recurrent Neural Networks,” pp. 1–13, 2015.

[33] A. Agarwal and S. Suryavanshi, “Real-Time* Multiple Object Tracking (MOT) for Autonomous Navigation,” *Report*, 2017.

[34] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

[35] A. Sadeghian, A. Alahi, and S. Savarese, “Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, doi: 10.1109/ICCV.2017.41.

[36] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)* , Phoenix, AZ, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003.