Multiple Object Tracking for Robust Quantitative Analysis of Passenger Motion While Boarding and Alighting a Metropolitan Train

José Sebastián Gómez Meza[1], José Delpiano[1], Sergio A Velastin[2,3] , Rodrigo Fernández[1], Sebastián Seriani Awad[1]

[1]Department of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile.
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK.
[3]Department of Computer Science and Engineering, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain

Keywords: Deep learning, object detection, passenger counting, YOLO v3, Faster R-CNN.

Abstract

To achieve significant improvements in public transport it is necessary to develop an autonomous system that locates and counts passengers in real time in scenarios with a high level of occlusion, providing tools to efficiently solve problems such as reduction and stabilization in travel times, greater fluency, better control of fleets and less congestion. A deep learning method based in transfer learning is used to accomplish this: You Only Look Once (YOLO) version 3 and Faster R-CNN Inception version 2 architectures are fine tuned using PAMELA-UANDES dataset, which contains annotated images of the boarding and alighting of passengers on a subway platform from a superior perspective. The locations given by the detector are passed through a multiple object tracking system implemented based on a Markov decision process that associates subjects in consecutive frames and assigns identities considering overlaps between past detections and predicted positions using a Kalman filter.

1 Introduction

In the face of the high congestion that public transport stations present, operators have looked for different alternatives to contain and speed up the transport process.

Systems that count the number of people entering and leaving a given area are known as automatic people counters and have been of great interest to the transport industry. Early work on pedestrian detection systems used infrared sensors or radars to determine the location of a subject comparing the information from the sensors with anthropometric studies [1]. Due to the deficiencies of these systems quantifying pedestrians in crowded places, computer vision techniques like background subtraction, correlation models and local extraction methods were rapidly adopted. The wide variety of human shapes made these solutions unreliable. Considering the advances in artificial intelligence, the state of the art was revolutionized in recent years towards the detection, monitoring and counting of pedestrians with convolutional neural networks (CNNs).

One of the main approaches in the development of new CNNs is the selection of regions of interest for classification, aiming at an increase in their quality and a decrease in the processing time. Since the emergence of the region-based convolutional neural network (R-CNN) [2] a large number of improved models have emerged. Two of them have been selected for this study: Faster R-CNN [3], a two stage convolutional neural network that together with a sub-network of regions proposals (RPN) efficiently selects the zones to classify by regression in the main architecture, and YOLO v3 [4], a one stage detector that performs object detection through a regression of regions proposed by a fixed grid aiming at an increase in their quality decreasing processing time that allows a detection in real time. Convolutional neuronal networks (CNNs) ability to identify multiple objects in an image provides the possibility to implement a device that counts the number of passengers, follows their trajectories, identifies evaders and even delivers proximity alerts on the platform to avoid contagion in pandemics. In object recognition, the representation of features with the use of convolutional neural networks tends to overcome manually designed methods such as local binary patterns and the oriented gradient histogram [5, 6, 7].

Therefore, the need to implement an intelligent transport system that integrates these technologies is clear, providing the necessary information to develop strategies that deliver a predictable, fast and safe service. In this work, the count of passengers boarding and alighting a subway car is experimentally validated using artificial vision techniques with association and prediction algorithms.

2 Background on Multiple Object Detection

2.1 Main architectures

Convolutional neuronal networks capture spatial information in images and videos, reducing the number of parameters by applying filters that provide appearance characteristics needed to learn to identify a specialized dataset. Two architectures are presented below. They were pre trained in the COCO database [8], which contains information on more than two hundred thousand labeled images over eighty categories. Part of the network weights are refined using a task-specific dataset, according to the transfer learning technique.

Transfer learning consists of using trained models to solve deep learning problems. An architecture that has been trained in its own database seeking to identify a specific class can be reused for the identification of objects that share similar characteristics. The different ways to use a previously trained neural network are defined by the similarity between the objects that the architecture was intended for and the new classes that you want to identify and also taking into consideration the size of the database that is available to train the network and if the user want to start from scratch or recycle some sections of the architecture.

2.1.1 Inception v2

With the appearance of the Inception module [9], an important milestone was marked in the design of convolutional neural networks, making a transition in the design of deeper networks to wider ones, which reduce the tendency to memorize the results. This architecture brings benefits in the detection of the same object seen from different perspectives, because it uses filters of various dimensions that allow it to extract both global and local information.

The Inception module in its second version considers two stacked filters of dimensions $1 \times n$ and $n \times 1$, which are equivalent to a filter $n \times n$ and produce a reduction in the training time of this complex architecture by 33%. Furthermore, padding layers are used to allow for the concatenation of the different filters in a matrix of depth m.

2.1.2 Darknet 53

Darknet [4] corresponds to a feature extractor which, as its name suggests, contains 53 convolutional layers, each followed by a normalization and activation layer with ReLU. In this architecture, no type of grouping layer is used and the way to decrease the dimension is achieved through the use of convolution with a filter that moves two positions (stride = 2); this allows maintaining low complexity characteristics that are usually lost with grouping layers.

2.2 Object detection

A detector is known as a convolutional neural network capable of classifying and locating multiple objects in a scene. This task is the result of transferring regions of interest of the image to a CNN where the objects to be identified could be located. Since the extraction of proposed regions from left to right and top to bottom for individual classification with a CNN is a brute force approach to object detection, the methods described below are used.

2.2.1 Two stage detector: Faster R-CNN

Given the existence of multiple scales on existing objects in an image and the large number of windows required that this means, Ren and colleagues proposed the model Faster R-CNN [3], in which a network of proposed regions called RPN receives a map of the characteristics of the convolutional layers of a network, providing an array of rectangles with proposed regions, each one with its reliability percentage.

The feature map is the result of passing the image through what can be a pre-trained convolutional neural network such as Inception v2 and Darknet 53. The RPN works by moving a window on the feature map, generating in each one k squares of fixed size, called anchor squares, of different sizes and shapes. Nine anchor boxes are located in each position adjusting their size to the proportion of the object to be detected; subsequently, the RPN predicts the probability that each one is an object, regardless of what class it belongs to, and determines whether it should be adjusted in size.

After the RPN, the group of regions of interest (ROI) without an associated class are projected onto the convolutional characteristics map and re-dimensioned to a fixed size suitable for the network through a grouping layer.

2.2.2 One stage detector: YOLO v3

Single stage detectors are designed to achieve real-time results. This can be done by eliminating the RPN used by Faster R-CNN and using a model that treats detection as a combination of regression and classification.

YOLO is a one-stage detector that develops its own convolutional architecture Darknet 53 for feature extraction. Starting from the last convolutional layers of the architecture, the prediction is made at three different scales that are the result of resizing the input image by 32, 16 and 8 respectively. Considering an input image, three scaled vectors are generated, which are used for detection.

The output matrices obtained from the detection are used to generate three anchor boxes in each position which are contrasted with the real labels in the database. This allows you to adjust the proposed regions for an image at three different scales. For each anchor box, an objectivity score is predicted that indicates whether an object is contained, the coordinates of the anchor box, and a probability that the object belongs to each class that you want to detect with the network.

The predictions of each box are contrasted and adjusted with the real information in the database using a loss function that determines the probability that the anchor box object is an object of the desired class. Unlike Faster RCNN, YOLO v3 performs regression processing for bounding boxes and classification in a single stage, skipping the network of proposed

regions. [4]

3 Multiple Object Tracking

Tracking can be understood as an estimate of the trajectory of an object while it is moving freely in the plane. This is a process that includes the identification of the subject in an image assigning him an identity that remains until he leaves the scene.

3.1 Markov decision process

For the detection of multiple objects a Markov decision process (MDP) based tracker covers the state transitions of a tracked detection. These include the registration, tracking, deletion and appearance stages. The following describes the process for the association of identities that allows to reestablish a missing subject or assign a track to an object that is in the registration stage [10, 11].

The initial state of any detection classified as pedestrian is the registration, with this the coordinates are associated with a unique identifier. In this way, a subject can change to being tracked, if he is affiliated with a new successful positive detection or to be disappeared with a new wrong positive detection. The target remains tracked, as long as there are associations in consecutive frames that meet the distance or intersection requirements, or it may become missing in situations such as occlusion, loss of detection, or leaving the scene. An object can be kept in a missing state for a predefined number of frames or it can be tracked again in case of a match, but if an identity is missing for a considerably long time it is permanently deleted.

3.2 Hungarian algorithm

Due to the cardinality that the association of a large number of targets can mean, the Hungarian algorithm was used to determine if an object in the current frame is the same as the next, providing a distribution of identities [12, 13, 14].

The Hungarian algorithm consists of efficiently allocating m available jobs to m workers, keeping one job per person and minimising cost. The association of identities is proposed in an equivalent way, replacing workers with tracks, jobs with new detections or predictions, and work costs with costs assigned according to intersection over union.

We define c_{ij} as the cost of associating the identity i with the detection j and also:

$$a_{ij} = \begin{cases} 1 & \text{If the } i\text{-th tracking is assigned to the } j\text{-th detection} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

The optimization problem to be solved for the allocation is defined in the equation 2.

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} \cdot a_{ij} \tag{2a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} a_{ij} = 1 \tag{2b}$$

$$0 \leq \sum_{j=1}^{m} a_{ij} \leq 1 \tag{2c}$$

Let $C \in \mathbb{R}^{nxn}$ be a cost matrix, its elements c_{ij} are assigned using intersection over union (IOU) defined in equation 3.

$$IOU = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{3}$$

3.3 Linear Kalman filter

The Kalman filter is an algorithm used to estimate the dynamic state of a system in the present, past or future. In general, this is a Bayesian system, where a sequence occurs between two states, prediction and update.

The prediction stage consists of estimating a state $x^{'}$ and the uncertainty $P^{'}$, in a time t, from the previous state x and uncertainty matrix P, at a time $t-1$. [15]

$$x^{'} = Fx + u \tag{4}$$

$$P^{'} = FPF^{T} + Q, \tag{5}$$

where F is the transition matrix between $t-1$ and t, u corresponds to additive noise, and Q is the process covariance matrix.

The update stage consists of using measurements from a sensor called z to correct the prediction.

$$y = z - Hx^{'} \tag{6}$$

$$S = HP^{'}H^{T} + R \tag{7}$$

$$K = P^{'}H^{T}S^{-1} \tag{8}$$

$$x = x^{'} + Ky \tag{9}$$

$$P = (I - KH)P^{'}, \tag{10}$$

where y is the difference between current measurement and prediction, S is the estimated system error, H is the measurement matrix between sensor and status, R is the covariance matrix related to sensor noise, and K corresponds to the Kalman gain that reflects the need to correct the prediction.

4 Proposed Method for Tracking of Multiple Passengers with Substitute Detections

The pedestrian location, classification, tracking and prediction system begins with taking samples from a passenger database in which bounding boxes generated manually for the passengers in each frame of the different videos of alighting and

boarding that allow training of the convolutional neural networks using deep learning. The selected convolutional architectures are Inception v2 and Darknet 53 considering the localization methods proposed by Faster R-CNN and YOLO v3 respectively. Given that some of the proposed architectures are quite deep, gradient with momentum was used, a variation that allows a faster tendency towards the global minimum of the cost function.

To fine-tune the defined models, the weights and biases of the different layers of the trained network are adjusted considering that PAMELA-UANDES dataset has enough information to model the characteristics that are important to identify pedestrians. Both general features of the network that are identified in the first layers of a convolutional neural network –those closest to the inputs– and the particular features found in the last layers are modified during training, but initialising them on with a previous set reduces considerably the training time. This process train the architecture using PAMELA-UANDES, starting the weights in all the layers with those obtained through the previous training in the *COCO* database. This means that the architecture extracts all the information from PAMELA-UANDES expecting training time to be considerably reduced by initializing the weights with values that are in the vicinity.

To match identities in adjacent frames and obtain the trajectory of an object in the entire video, the Hungarian algorithm was used in conjunction with a Markov decision process considering the intersection over union criterion. The Markov assignment process receives the bounding boxes of an object classifier or, failing that, the predictions obtained through the Kalman filter called substitute detections.

The Kalman filter was used on each identified subject to predict their position on each frame, reducing identity changes and erroneous eliminations. When an association is made, the prediction and update steps are executed. The detections delivered by the CNNs are supported by the Kalman filter that provides an estimate of the positions and the size of the bounding boxes for the identities in the missing state, decreasing the noise entered by the detector. The general system association of new identities is made through the Markov decision process in conjunction with the Hungarian algorithm according to the criteria of intersection over union. The calculation of the costs of associating a track at frame n with the detections of frame $n+1$ is by means of the intersection over union (IOU), defined in the equation 3.

The Kalman filter allows predicting in real time without the need for prior information, as long as there are new sensor measurements, the filter status is quickly adjusted to the actual position of the bounding box. The main objective is the estimation of the mean x that represents the coordinates of the bounding box and the covariance that is the uncertainty of the location P.

State vector x is composed of the coordinates of two opposite corners of the bounding box and the velocity variations in both axes as shown in the equation 11.

$$x = [y_{min}, v_{y_{min}}, x_{min}, v_{x_{min}}, y_{max}, v_{y_{max}}, x_{max}, v_{x_{max}}] \tag{11}$$

Where:

- x_{min}, y_{min}: Represent the x, y coordinates of the upper left corner of a bounding box.
- $v_{x_{min}}, v_{y_{min}}$: Represent the velocity components in x, y of the coordinate (x_{min}, y_{min}).
- x_{max}, y_{max}: Represent the x, y coordinates of the lower right corner of a bounding box.
- $v_{x_{max}}, v_{y_{max}}$: Represent the velocity components in x, y of the coordinate (x_{max}, y_{max}).

Defining the state vector with the bounding box coordinates and velocities allows not only to predict the location of the centroid but also the size of the bounding box around it.

On the other hand, the estimation uncertainty matrix P is defined in the equation 12 and represents the confidence of the state with each pass of the filter.

$$P = L \cdot I, \tag{12}$$

where L is a constant that amplifies or decreases the uncertainty and I is the identity matrix.

To use the Kalman filter, the bounding boxes at $t = 0$ provided by the pedestrian detector are received and new identities are assigned by the Hungarian algorithm. Subsequently, at time $t = 1$, the new bounding boxes are associated with the previous time tracked objects, allowing the Kalman filter to be initialized. In general, the bounding boxes estimated in a time t are obtained based on those of a time $t-1$ and are updated with the detections of time t, this allows the tracking algorithm to keep the continuity for the associations even when the pedestrian is not detected correctly.

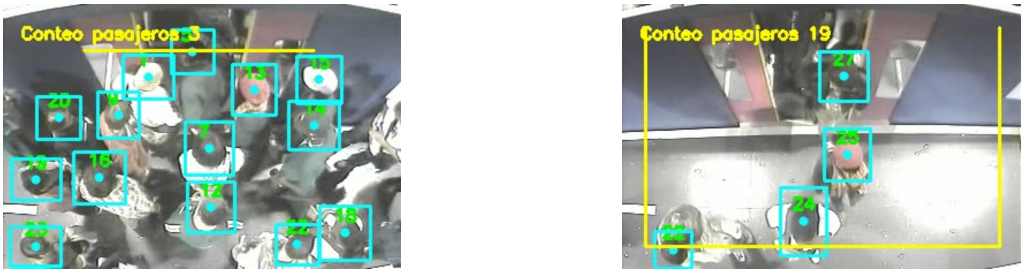
To carry out the final passenger count, limits were defined as can be seen in the illustrations 1a and 1b. When the centroid of a tracking detection from the platform exceeds the threshold of the metro door, the pedestrian counter increases by one unit and marks the tracking as "counted" since the subject is already boarded. This process is equivalent to counting the number of passengers leaving the scene, but considers that a passenger can take three directions in their descent, decreasing the counter by one unit each time a centroid is crossed with the thresholds.

5 Results

5.1 PAMELA-UANDES Dataset

The PAMELA-UANDES Dataset[1] is a database of video that was filmed at the mock-up of an underground subway station in London in 2008 with the participation of local people. It contains fifteen videos in which people's heads have been manually annotated for each frame. The videos are divided into eight

[1] http://videodatasets.org/PAMELA-UANDES/, downloaded in July 2019.



(a) Alighting limits.

(b) Boarding limits.

Figure 1: Counting limits: The line on the image is considered to count people crossing the line up (boarding) or down (alighting).

boarding and seven alighting, considering a door opening at a width of 800mm [16].

In total there are 119,397 pedestrian labels manually annotated, of which 49,906 were used during training. The reason why not all the information is used is to avoid over-fitting the model parameters because in most of the boarding and alighting videos the same people are counted, reducing the variability of the data.

Each individual has been assigned a unique identifier allowing quantification with tracking metrics. The average length of the videos is between one and two minutes with a resolution of 352x288 pixels at twenty-five frames per second.

The coordinates of each annotation are represented by a rectangle of which the coordinates (x_{min}, y_{min}) of the upper left corner, the width w and the height h are known. The illustrations 2a and 2b show bounding boxes extracted for a frame of the dataset and considering that there are no annotations over the entrance door, where the heads of boarded passengers can be seen, to improve training and reduce processing time, a section of the images are eliminated. This contemplate a change in the original axes O_1, which were in the upper left corner, to new coordinates O_2 located fifty pixels down and five pixels to the right.

5.2 Metrics for multiple object tracking

Tracking performance evaluation for multiple objects is based on a set of metrics proposed by the multiple object tracking (MOT) challenge which provides a platform for the evaluation of the different algorithms [17, 18].

Between a new detection of a CNN and an existing object in the database, the following is defined:

- True positive (TP): Correct detection defined by an intersection over union (IOU) threshold of 50%.
- False positive (FP): Detection with an IOU lower than 50% to any database label.
- False negative (FN): Represents a label in the database that is not associated with any detection by intersection over union.
- Fragmentation: Corresponds to a trajectory of the database that is tracked and is temporarily interrupted.
- Identity switch: Generated when a tracked trajectory of the database changes from one identity to another due to the proximity between the traces.
- Mostly tracked (MT): A subject is mostly tracked if he/she has a follow-up of at least 80 % of the known trajectory.
- Mostly lost (ML): Corresponds to a subject whose real trajectory is recovered for less than 20 % of its total length.

It should be noted that the 'mostly tracked' and 'lost' trajectories do not consider whether the identity is the same throughout the known path. In general, a tracking algorithm is expected to yield an improved detection performance due to temporal context from position history. Therefore, a successful algorithm will deliver as few 'mostly lost' trajectories, identity switches, and fragmentations as possible, but will have also a reduced number of false positives and false negatives. To assess the quality of a tracking algorithm, it is sought to measure precision, recall, MOTP, and MOTA defined below.

- Precision: It measures the ability to find true positives over the number of detections.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

- Recall: It measures the ability to find true positives over the number of labels in the database.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

- MOTP: Multiple object tracking precision. It corresponds to the sum of the intersection over union $d_{t,i}$ between real bounding boxes t with detections i for all the frames of a video, divided by the total number of matches made c_t.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{15}$$

It demonstrates the ability of the tracking algorithm to estimate positions of objects independent of their ability to maintain a trajectory.

(a) Bounding box annotations drawn over a full frame

(b) Annotations over cropped frame

Figure 2: Definition of a region of interest and visualisation of bounding box annotations for a frame of a video of boarding passengers. The working region of interest excludes the interior of the mock-up train.

- MOTA: Multiple object tracking accuracy. It corresponds to a measure of the precision of the tracking, considering the number of false positive detections FP_t , labels not detected $\overline{FN_t}$ and identity changes $IDSW_t$ over the number of actual tags GT_t in all frames of the video.

$$MOTA = 1 - \frac{\sum_t FN_t + \overline{FP_t} + IDSW_t}{\sum_t GT_t} \quad (16)$$

5.3 Results for multiple pedestrian detection and tracking

The results presented below are computed using a Python implementation[2] of metrics in the MOT challenge toolkit and the method considers an association by intersection over union with a threshold of 30 % and a permanence of the missing identities of 10 consecutive frames. A too demanding IOU threshold(Over 50%) prevents the association of rapidly moving pedestrians given a twenty-five frames per second video causing a considerable increase in identity changes, usually during alighting the passengers move at a higher speed since they do not have any obstacles. This is an occurring problem related to the stabilisation of the Kalman gain since once there are enough measurements, when a particular object is under tracking a spontaneous change in velocity is diminished as it is an anomaly compared with the previously known states, so the estimated position could be not as far ahead as it is needed.

The operation of the detection and the tracking algorithm is verified by evaluating 6 videos, never seen before by the CNN, equally distributed between boarding and alighting (6 other videos were used for training and 3 for validation).

[2]https://github.com/cheind/py-motmetrics

Metrics	Yolov3	Faster R-CNN
TP	**39.744**	37.232
FP	**890**	4.012
FN	**7.104**	9.616
GT labels	46.848	46.848
IDSW	**35**	86
Frag	**152**	173
MT	**172**	124
ML	**0**	2
Recall	**84.47%**	78.70%
Precision	**96.96%**	89.02%
MOTP	73.32%	**77.03%**
MOTA	**81.70%**	68.49%

Table 1: Tracking results using Kalman filter.

The results obtained for the detection and counting of pedestrians both ascending and descending for the different metrics are considered the most satisfactory using YOLO v3, in each table bold values indicate the best results for each row. MOTA Multi-Object Tracking Accuracy averages 81.7% and represents the algorithm's ability to track targets on their path independent of identity switches, false positives, and undetected labels. In addition, the average of the intersection over union of the associations between the labels obtained with the algorithm versus the real ones (MOTP) achieves a 73.32% supporting the accuracy of the detection algorithm and the work of the Kalman filter.

On average, 84.47% of the database labels were found, and of the 39,744 positive detections, 96.96% of the assumptions were correct. Also, no trajectory was followed by less than 20% of its path and, on the contrary, 172 of 234 identities were followed by at least 80% of their path. In this way, the system manages to count 99% of the 234 passengers for the six test videos of the PAMELA-UANDES database.

6 Conclusions

An exhaustive review of computer vision technologies for the detection of pedestrians from a higher perspective was carried out, which were applied considering the information of the labels and locations of passengers boarding and alighting in the videos. In addition, the Hungarian algorithm was implemented for the association of detections in consecutive frames of a video according to the criteria of distance and intersection over union, establishing an adequate Markov decision process for the assignment of identities and elimination of erroneous or missing tracks. Identity changes and noise added by the detector were also reduced using a Kalman filter for the forecast of passenger locations, establishing the tracking system and as detection of a new standard for the database *PAMELA-UANDES*.

The best results are obtained by combining YOLO v3 with the tracking algorithm, giving an average precision of 97% that considers the number of true positives over the number of hypotheses of the system. On the other hand, the precision of multiple objects $MOTA$ supports that in 80.7% of the cases the system did not have errors of FP, FN and changes of identity. Furthermore, the bounding boxes detected had an accuracy in their estimate $MOTP$ of 84.5% considering the intersection over union with the real bounding boxes. The tests carried out on six videos show that out of a total of 234 people, 100% of passengers were counted on the boarding and the 99% on the alighting.

The precision of the convolutional models depends on extensive databases with labels for training and it is considered that the different architectures have successfully learned to identify the subjects of PAMELA-UANDES since the characteristics extracted from a superior perspective consider less complexity than a view of whole body. It is expected that future versions will consider a database that allows learning in different situations of light and perspective.

References

[1] J. Sallay, *Automatic People Counting and Matching*. PhD thesis, 2009.

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

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] J. Redmon and A. Farhadi, "YOLO v.3," *Tech report*, pp. 1–6, 2018.

[5] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Handbook of Approximation Algorithms and Metaheuristics*, pp. 1–1432, 2012.

[8] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016.

[10] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 4705–4713, 2015.

[11] J. E. Espinosa, S. A. Velastin, and J. W. Branch, "Detection and tracking of motorcycles in congested urban environments using deep learning and markov decision processes," in *Pattern Recognition* (J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-López, and J. Salas, eds.), (Cham), pp. 139–148, Springer International Publishing, 2019.

[12] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 52, no. 1, pp. 7–21, 2005.

[13] J. Munkres, "Algorithms for the assignment and transportation problems," 1957.

[14] G. A. Mills-tettey, A. Stentz, and M. B. Dias, "The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs," *Naval Research Logistics Quarterly*, no. July, pp. 83–87, 2007.

[15] A. Kelly, *Mobile robotics: Mathematics, models, and methods*, vol. 9781107031159. Cambridge University Press, 1 2013.

[16] S. A. Velastin, R. Fernández, J. E. Espinosa, and A. Bay, "Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–20, 2020.

[17] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *arXiv e-prints*, p. arXiv:1603.00831, Mar. 2016.

[18] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *Eurasip Journal on Image and Video Processing*, vol. 2008, 2008.