Neural Network Based Multiple Object Tracking for Automotive FMCW Radar

Konstantinos Fatseas*
k.fatseas@utwente.nl

Marco J.G. Bekooij*†
marco.bekooij@nxp.com

*Department of Computer Architectures for Embedded Systems, University of Twente, Enschede, The Netherlands
†Department of Embedded Software and Signal Processing, NXP Semiconductors, Eindhoven, The Netherlands

2019 International Radar Conference (RADAR) 978-1-7281-2660-9/20/$31.00 ©2020 IEEE 10.1109/RADAR41533.2019.171248

***Abstract*—Tracking multiple objects from radar data poses several difficulties. In recent work, it has been shown that an algorithm consisting of a thresholding, clustering and multiple object tracking step using the Kalman filter can track multiple objects. Afterwards, features can be extracted from the range-Doppler map to classify the tracked objects. However, this method needs many heuristics on each stage and in the process, information which could be useful in subsequent steps is lost.**

To overcome these issues, in this paper we introduce a neural network based multiple object tracker. This removes the need for a separate thresholding, clustering, feature extraction and classification step because it combines those into one step which uses a neural network based on the You Only Look Once (YOLO) object detection system to classify and localize objects. The output of the neural network is fed into a Kalman filter based tracker to manage the tracks.

We show that a convolutional neural network trained as an object detector can be successfully applied in the radar domain and we show the advantages of our neural network based multiple object tracker over the clustering based method for specific scenarios. These scenarios include tracking objects that cross each other and tracking objects while the radar is non-stationary.

***Index Terms*—Deep Neural Network, Range-Doppler, FMCW**

I. Introduction

Radar systems play an important role in applications like the autonomous car, intelligent road infrastructure and property surveillance systems. There are many reasons for choosing radar sensors over image sensors. First of all, radar sensors are not affected by weather or light conditions. Furthermore, certain radar types like Fequency Modulated Continuous Wave (FMCW) radars are able to directly measure the distance to an object as well as its radial velocity.

Within the computer vision domain a lot of progress has been achieved with the introduction of Deep Convolutional Neural Networks (DCNNs) in fields like image classification and object detection. However, it is not clear whether the same approach can be applied to radar sensors. First of all, radar sensors have lower spatial resolution than image sensors making it harder to determine the shape and size of objects. Moreover, only when objects are moving, the so-called Doppler information is available in order to perform classification.

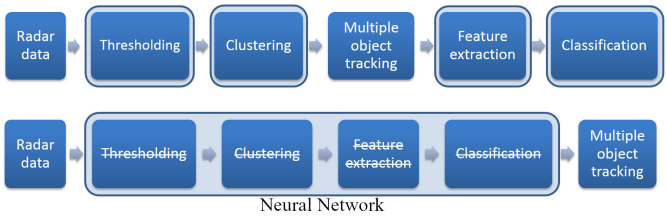


Fig. 1: Schematic illustration of the two multiple object tracking methods. Top: clustering based method. Bottom: neural network based method.

Additionally, for complex applications like autonomous driving, classification alone is not sufficient; the task of tracking these objects over time needs to be addressed as well. Wagner et al. [1] presented an algorithm to track multiple objects from a series of FMCW radar data cubes. This algorithm consists of several steps to perform the task: thresholding, clustering, tracking based on Kalman filter, feature extraction and finally classification.

This paper introduces a neural network based tracker capable of tracking multiple objects directly from FMCW range-Doppler maps. Object detection from the range-Doppler maps is possible due to the high resolution of modern FMCW radars in both velocity and range. This causes road users such as pedestrians and cyclists to have a distinctive signature in the range-Doppler map.

Compared to the clustering based method, our algorithm removes the need for separate thresholding, clustering, feature extraction and classification stages. It combines these stages into a single evaluation of the YOLO based DCNNs as can be seen in Figure 1.

Combining these steps into one, has the advantage that fewer manually designed heuristics are needed, making the solution more robust to a changing environment. Furthermore, the neural network takes all available information into account, unlike the clustering based approach which discards information along the process. More importantly, our neural network based tracker classifies the detected objects at every frame. Whereas, the clustering based method can only classify the detections after several frames, increasing in this way the total latency.

II. Related Work

In this work we address the problem of tracking and classifying multiple objects in FMCW radar data. Other studies related to classification from radar data can be divided into two categories: feature based and automated.

Researchers have been trying to classify human gaits based on manually extracted features [2] [3]. These studies have achieved high accuracy, but they require domain knowledge for feature extraction and thereby limit scalability [4]. Furthermore, as mentioned in [2], some of the limitations of their approach are that the algorithm is not very robust against unpredictable, non-periodic motions and the relatively low carrier frequency makes it difficult to classify objects that generate fast-changing micro-Doppler velocities.

In order to avoid manual feature extraction, recent papers experiment with neural networks to perform classification with radar data [4] [5]. Kim et al. [4] utilized a DCNN to classify between a human, dog, horse and a car. As input to the DCNN they used spectrograms with a two second time-window. Next, they classified seven human activities using a DCNN. In [5], Shao et al. use a DCNN to classify the same seven human activities as in [4], but the input to the network consists of range information as a function of time, resulting in a more robust classifier and with a better tolerance for different incident angles.

All the research mentioned thus far use either the velocity-time plot or the range-time plot as input for the classification algorithm. However, in this paper an FMCW radar is used which is capable of providing range as well as Doppler information. By using only the range or velocity information, potentially some information is lost which could be useful for tracking.

More recently Perez et al. [6] utilized a DCNN to classify range-Doppler-angle data cubes into three categories. Namely, pedestrians, cyclists and vehicles. The main limitation of this method is the inability to handle scenarios with multiple objects as each frame will eventually classified into only one of the aforementioned classes.

To the best of our knowledge, this is the first work that makes use of a DCNN to detect multiple objects directly from range-Doppler maps.

III. Radar measurements

The measurements for our work were acquired by the 77 GHz TEF810X radar transceiver from NXP. Because we aim to detect pedestrians and cyclist the settings were chosen accordingly. Radar settings have a large impact on the resulting data cube after spectrum estimation. Important considerations are the maximum range and velocity of the radar, the range and velocity resolution and the number of frames per second that are recorded.

More specifically for every measurement 128 chirps were transmitted, each consisting of 512 samples with a decimation factor of 2. The resulting range-Doppler map has 128×128 values. Furthermore, a chirp bandwidth of 1.8 GHz has been used and the dwell time, i.e. the time between chirps, has been set to 120 μs. The aforementioned settings result in a maximum velocity of 24 km/h and a maximum distance of 12.8 m. The velocity resolution is 0.375 km/h and the distance resolution is 0.1 m. The frame rate is 20 fps.

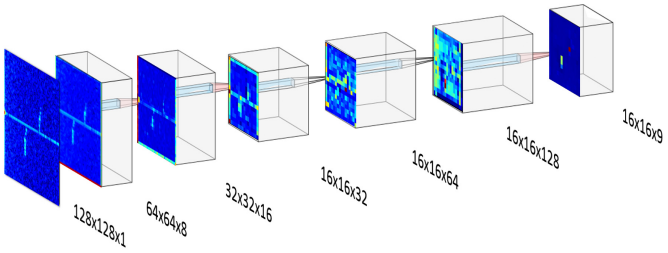


Fig. 2: Network architecture for the neural network based method

IV. Neural Network Based Method

A. Neural Network-Based Object Detector

Our tracking algorithm consists of two components. First is the object detector which follows after spectrum estimation and the second is a tracker that uses a Kalman filter to track the objects over time. The object detector of our tracking system is based on the YOLO and YOLOv2 detectors. However, the architecture of the DCNN is adapted to our dataset. First of all, the input of our network consists of 128×128 values. This dimensionality is the result of the specific radar parameters as described in III.

Moreover, the grid cell size of the detector's output is adapted as well. In general, the YOLO detectors divide the input image into a number of sub-regions. Each of the cells of this grid predicts information about several objects that lie entirely within or overlap with the grid cell. Our network has been designed to only detect one object per grid cell. This implies that the dimension of the grid cells determines the minimal distance between objects that can be detected.

If the grid cells are too large, then only few objects can be detected; if the grid cells are too small, then it becomes harder to deal with multiple grid cells detecting the same object. Therefore, a grid cell size of 8 has been empirically chosen resulting in $16 \times 16 = 256$ grid cells.

Figure 2 shows the architecture of the neural network. It is composed of six convolutional layers and all of them consist of 3×3 kernels. Max pooling with a window size of 2×2 is applied only after the first three convolutional layers. This is because after three pooling layers the width and height of the convolutional layers' output has reached the desired grid dimensions.

Our proposed neural network architecture contains less convolutional layers and a smaller amount of filters per layer than YOLOv2. The first reason for this disparity is the substantial difference between range-Doppler maps and images in terms of complexity. Secondly, the receptive filed of a neuron at the last convolutional layer is not required to cover the input's full extend. Receptive field is a term that describes the size of

978-1-7281-2660-9/19/$31.00 ©2019 IEEE

the area from the input which affects a single neuron at the networks' output.

In range-Doppler maps the extend of an object never occupies the entire image. This is a key difference between images and radar measurment: in range-Doppler maps the extend of the objects does not change depending on the distance, only the power varies. Whereas in images, objects look larger when they move closer to the camera. By defining the architecture as depicted in Figure 2, the output of a neuron at the last convolutional layer is affected from an area of 54×54 in the range-Doppler map.

The ability of our neural network to act as an object detector derives from the way that its output is structured. It consist of a 3-dimensional tensor where the information of the detected object per grid cell can be denoted as:

$$y_{i,j} = \begin{bmatrix} P_{i,j} \\ Y_{midpoint_{i,j}} \\ X_{midpoint_{i,j}} \\ Y_{box_{i,j}} \\ X_{box_{i,j}} \\ H_{i,j} \\ W_{i,j} \\ C_{1_{i,j}} \\ C_{2_{i,j}} \end{bmatrix} \quad i,j \in [1,16] \tag{1}$$

where $P_{i,j}$ denotes the probability that the midpoint of an object is in this grid cell, $(Y_{midpoint_{i,j}}, X_{midpoint_{i,j}})$ denotes the coordinate of the midpoint of the object, $(Y_{box_{i,j}}, X_{box_{i,j}})$ denotes the coordinate of the midpoint of the bounding box, $(H_{i,j}, W_{i,j})$ denotes the dimension of the bounding box and $(C_{1_{i,j}}, C_{2_{i,j}})$ denotes the class score for a pedestrian and cyclist respectively. All coordinate and dimension values are relative to the width of a grid cell and the coordinates are expressed with an offset that starts at that grid cell.

B. Dataset and Training

The neural network was trained in fully supervised manner with a training dataset that consists of 2008 range-Doppler maps. The recordings took place outside along different roads with buildings, trees and cars as part of the surroundings and had a total duration of 100 seconds. The number of pedestrians and cyclists that are present in the dataset is balanced and finally, in 543 frames the radar transceiver was moving. During the recordings the radar was placed 80 cm above the ground.

All 2008 r-D maps were manually labeled in order to produce bounding boxes around the objects of interest. Subsequently, the bounding boxes were encoded in the format described by equation 1. An additional 400 range-Doppler maps were produced in order to validate the results of the neural network after training.

The network was implemented in Python using Tensorflow and trained for approximately 50 minutes on a 3.4 GHz Intel Core i7-6700. The optimizer was Adam [7] with a learning rate of 0.0005 and the loss function was defined as in [8]. A batch size of 16 frames has been used for a total of 12000 steps.

C. Object Tracker

The tracking algorithm is responsible for the initiation, maintenance and termination of the tracks that correspond to the detected objects. To cope with the noisy output of the detector we use the Kalman filter [9] which is a recursive algorithm that predicts state variables over time from measurements and assumptions about the dynamics of the tracked object. The velocity and range values we extract from the object detector are somewhat inaccurate, caused by the changing shape of the detection points of the objects over time. Furthermore, an object that is being already tracked may not be detected continuously in every frame.

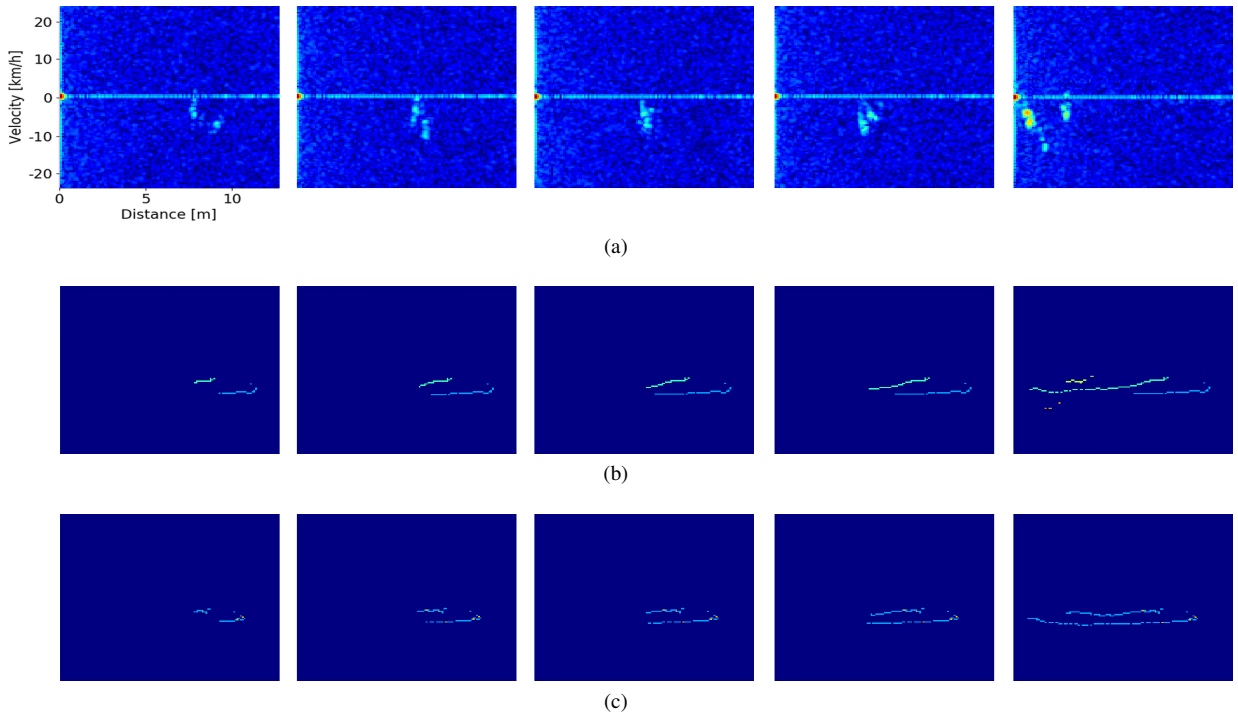
The Kalman filter helps by smoothing these inaccuracies and trying to predict the real values for the velocity and range of an object, before updating its track. The state vector of the Kalman filter contains the velocity and range of the object, where a constant velocity is assumed.

As in [1], a new track and a new instance of the Kalman filter is created for every new detection. A track is removed if it has not been tracked for 0.2-0.8 s depending on the time the object has been tracked already. This way a false detection will start a track which will be removed after 0.2 s and a longer tracked object will receive track updates up to 0.8 s. This longer time before an untracked object will be removed, makes the algorithm more robust against occlusions or crossings of objects.

The assignment problem, which refers to the choice of the correct measurement to update an existing track, is solved by using the nearest neighbor approach. By doing so we recursively assign detections to tracks that have the lowest Euclidean distance. If the object has been tracked already for more than 0.2 s, then this distance should be below a certain threshold. This prevents a wrong assignment that could occur in the case that the detection for a tracked object is absent for that frame and there is a false detection at a different place. This threshold does not hold for objects that are only being tracked for less than 0.2 s. This is because the distance between the track estimate and detection can still be large, since the Kalman filter did not have enough time to converge yet.

After training, the detection network is able to predict the presence of an object in each of the grid's cell by giving a probability witch is denoted in equation 1 as $P_{i,j}$. Because an object will be detected in multiple neighboring cells we need to define a threshold which will dictate whether the detection is valid or not. The threshold value has been set to 0.5. In some cases, even after thresholding the detected bounding boxes, there are still double detections. To overcome the problem, the intersection over union (IOU) value can be calculated and if the IOU metric is above a threshold the non-maximum detection will be removed recursively.

A problem arises when there are in fact two objects very close together. One of the two detections will be removed, because the IOU value of the two detections is above the threshold. This problem can be prevented by using information from the tracker itself. When two objects are approaching each other, the IOU threshold is increased accordingly. Once the

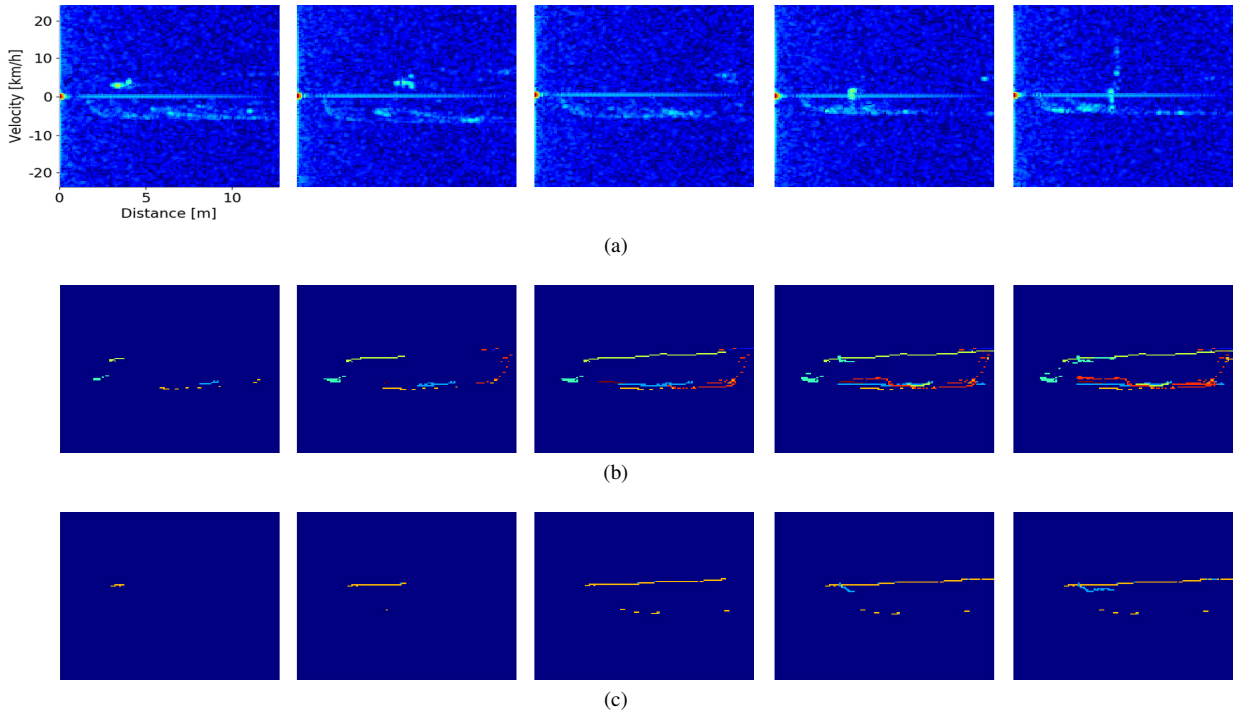


(a)

(b)

(c)

Fig. 3: Sample range-Doppler maps (a) and tracking results of the clustering based (b) and our YOLO based (c) approach for a recording with 2 overtaking pedestrians.



(a)

(b)

(c)

Fig. 4: Sample range-Doppler maps (a) and tracking results of the clustering based (b) and our YOLO based (c) approach for a recording where the radar was non-stationary.

978-1-7281-2660-9/19/$31.00 ©2019 IEEE

objects are further separated the IOU threshold is set back to the original value.

V. Results

Our initial assumption has been that a neural network based tracker will alleviate the shortcomings of the clustering method that derive from the heuristics that are involved. We evaluate the performance of our algorithm by comparing the two methods on a set of recordings that were not part of the training dataset.

Figure 3 shows the tracks generated by the two methods when the input was a recording where two pedestrians are crossing each other. The two pedestrians are both moving towards the radar but one of them is walking faster. Subfigure 3a contains five sample range-Doppler maps from the recording in chronological order, from left to right.

The resulting track of the clustering based method which is depicted in subfigure 3b, is not correct. Initially it detects the two pedestrians but as they approach each other they are falsely detected as a single object for many consecutive frames. As a result, the Kalman filter based tracker discontinues one of the tracks and later when the two pedestrians are further apart, it initiates a new one.

Our method on the other hand, is able to correctly track the two objects for the entire duration of the recording. This can be seen in subfigure 3c. The reason is that the neural network based object detector can separately detect the two pedestrians in almost all the frames. That allows the Kalman filter to continue the tracks of both pedestrians uninterrupted.

Another scenario that our method generated better results is when the radar was moving. In this case objects that are standing still, have a negative velocity which equals the velocity of the radar. Similarly to figure 3, subfigure 4a shows five sample frames from the recording while subfigures 4b and 4c depict the tracks generated from the clustering based method and our algorithm respectively.

Our object tracker has been trained with recordings where the radar was moving and can successfully detect only the cyclists and pedestrians due to their unique signature in the range-Doppler map. On the contrary, the clustering based algorithm cannot handle the moving environment and initiates a lot of false tracks

Note that the color of the tracks in subfigures 3c and 4c shows the predicted class for each detected object. Orange color indicates a cyclist, whereas light blue is a pedestrian. On the other hand, in subfigures 3b and 4b the colors of the tracks are chosen arbitrarily and do not indicate any class because the classification stage of the algorithm presented in [1] was not implemented.

Table I provides more detailed results over the comparison of the two methods. We show separate results for the case of the moving radar because the performance of the clustering method drops sharply. More specifically, the recall, precision and the False Alarms per Frame (FAF) values are a lot worse than the results of the neural network based tracker.

When the radar is stationary our method perform slightly better with an exception on the recall metric. This is because when the objects of interest move further away from the sensor, the reflection of separate components like the arms, legs and wheels, is getting weaker. Subsequently, the neural network based object detector fails to detect the objects more often.

TABLE I: Quantitative results comparing clustering based method to neural network based method

Method	Moving radar	Recall	Precision	FAF
Clustering	Yes	0.65	0.21	2.38
	No	0.85	0.94	0.08
Neural network	Yes	0.85	0.96	0.04
	No	0.69	1.00	0.00

Finally, our method can directly classify the detected objects with an accuracy of 97%. In contrast to the clustering approach that needs some time to extract the micro-Doppler signature of the detected object and then classify them based on manually selected features.

VI. Conclusion

This paper introduces a neural network based multiple object tracker. This method removes the need for a separate thresholding, clustering, feature extraction and classification step, as applied in the state-of-the-art clustering based method described in [1]. Instead, we incorporate those stages into a single evaluation of a DCNN.

The results show that the proposed algorithm is more robust for difficult scenarios in which one pedestrian overtakes another person as well as for the case that the radar is non-stationary. Furthermore, the total latency until an object is classified is drastically reduced as our method performs classification on every frame.

References

[1] T. Wagner, R. Feger, and A. Stelzer, "Radar Signal Processing for Jointly Estimating Tracks and Micro-Doppler Signatures," *IEEE Access*, vol. 5, pp. 1220–1238, 2017.

[2] Youngwook Kim, Sungjae Ha, and Jihoon Kwon, "Human Detection Using Doppler Radar Based on Physical Characteristics of Targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 289–293, 2015.

[3] Z. Zhang, R. Zhang, W. Sheng, Y. Han, and X. Ma, "Feature extraction and classification of human motions with LFMCW radar," *IEEE International Workshop on Electromagnetics, iWEM 2016 - Proceeding*, 2016.

[4] Y. Kim and T. Moon, "Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.

[5] Y. Shao, S. Guo, L. Sun, and W. Chen, "Human Motion Classification Based on Range Information with Deep Convolutional Neural Network," *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1519–1523, 2017.

[6] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, "Single-frame vulnerable road users classification with a 77 ghz fmcw radar sensor and a convolutional neural network," in *2018 19th International Radar Symposium (IRS)*, June 2018, pp. 1–10.

[7] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2015.

[9] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

978-1-7281-2660-9/19/$31.00 ©2019 IEEE