Exploitation of Neural Networks for Fusion of Camera and Millimeter-Wave Radar Data

Bořek Reich*
Supervised by: Pavel Zemčík

Faculty of Information Technology
Brno University of Technology
Brno / Czechia

Abstract

This paper introduces a multiple sensor synchronization that exploits using only off-the-shelf products. Sensor synchronization is significant for sensor fusion, an important area of research in the field of computer vision. It is known that a combination of multiple sensors can improve performance of a system and overcome the disadvantages of a single sensor, assuming sensors used are synchronized. Without proper synchronization, any attempts to benefit from sensor fusion, are useless. This paper aims to explore the fusion of camera data and millimeter-wave radar data for detection and recognition purposes as well as for machine learning. This paper proposes a multiple sensor synchronization procedure that is easy to adopt for different sensors and can be used for combination of visual and non-visual sensors using only off-the-shelf products. This synchronization technique is suited for research purposes as well as some real-world applications.

Keywords: data fusion, detection, millimeter-wave radar, monocular camera

1 Introduction

Camera-based systems are considered state of the art and they work with high accuracy. The downside of camera-based methods is that they do not perform well in challenging environments. In contrast, radar data is not affected by challenging conditions such as poor lighting, mist, rain. In situations when cameras fail, millimeter-wave radar data can improve the performance of a system and provide interesting features that can be utilized for object localization and accurate speed and/or distance determination.

Computer vision techniques are getting increasingly more attention due to advancements in deep learning and other methods as well as in hardware. Almost everyone owns a smartphone that can be used for wide variaty of computer vision applications, for example QR code readers, face filters, and even Google Lens (detection-recognition system with a broad coverage). However, creation of robust, reliable object detection-recognition systems remains challenging. In this regard, complementary sensors are often used within one system to achieve desired performance. An example of complementary sensors is the combination of millimeter-wave (mmwave) radar and monocular camera [1]. Mmwave radar point cloud and image calibration example is shown in Figure 1.

Sensor fusion refers to the technique of combining data from different sensors to achieve better accuracy and performance that is not possible to attain with only one of the sensors alone. Deep learning fusion algorithms that utilize data from mmwave radar and cameras are recently getting more attention. Readers can refer to millimeter-wave radar and camera fusion review [1] for a thorough summary, including the mmwave radar processing chain and fusion methods.

The key part of every sensor fusion method is data synchronization to which many possible solutions exist. This paper proposes a synchronization procedure using low-level communication and audio channels of IP cameras that can be used for mmwave radar with camera synchronization as well as for multiple cameras synchronization.

Figure 1: Example of millimeter-wave radar and camera calibration from RVNet [5].

2 Related Work

This section introduces different millimeter-wave radar data representations, fusion algorithms, and synchronization.

*xreich06@vutbr.cz

2.1 Radar Data Signals Representations

Millimeter-wave radar signals may have to be interpreted to fit the needs of computer vision algorithms. In this regard, many radar data representations have been proposed. These can be range-(velocity)-azimuth tensors, radar point clouds, occupancy grid maps. Every representation is suitable for different methods. Majority of detection-recognition systems, however, use range-(velocity)-azimuth tensors or radar point clouds.

Point cloud is an often used simple mmwave radar data representation as point clouds are popular 3D space representations that are easily obtainable from lidar sensors. Many methods for classification, segmentation, and detection that use lidar point clouds are available. These methods can be easily adapted for millimeter-wave (mmwave) radar point clouds and one can benefit from previously designed algorithms. For example, the widely used segmentation method, primarily used for lidar point clouds, PointNet++ is adaptable for mmwave radar point clouds [3]. The disadvantage of point cloud representation is that point clouds are acquired by thresholding of multidimensional Fast Fourier Transform (FFT)[1] outputs which leads to a loss of information. An example of mmilimeter-wave radar point cloud with corresponding image is shown in Figure 2.

Range-velocity-azimuth maps are also created using multidimensional FFT and they can preserve the majority of information, compared to point clouds. Instead of thresholding and using only local maxima interpreted as points, range-(velocity)-azimuth maps use the entire output of FFTs and preserve much more information. An example of range-azimuth map representation is shown in Figure 3.

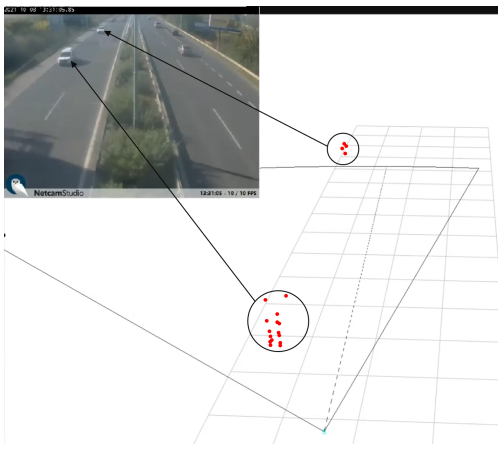

Figure 2: An example of point cloud mmwave radar data representation with corresponding camera frame.

[1]Fast Fourier Transform is applied to the received reflected signals to estimate range, velocity and azimuth. For more information, refer to [1].

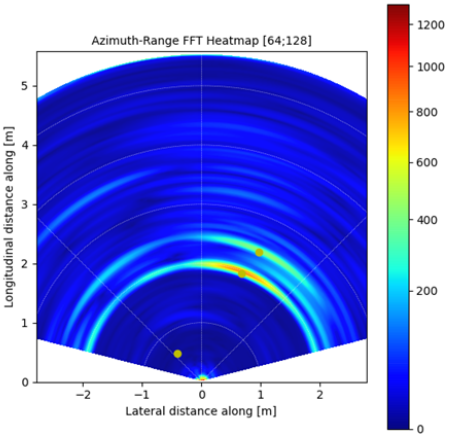

Figure 3: An example of range-azimuth map from mmwave radar data available in documentation of mmwave radar IWR6843ISK by Texas Instruments.

2.2 Overview of Mmwave Radar and Camera Fusion Algorithms

Many of mm-wave radar and camera fusion algorithms are aimed at obstacle detection and recognition. This means that the sensors are mounted on a vehicle. It is, however, possible to use these methods as traffic monitoring systems as well.

Many fusion algorithms are designed for object detection only, not object recognition. Systems proposed in Meyer and Kuschk [8] and Nobis et al. [10] belong to this category. They all combine images and radar data to achieve better obstacle detection systems for self-driven cars.

Another category of researched systems is formed by those systems that use camera data only for training. These systems, when deployed, use only mmwave radar data. Systems with this design are used only for obstacle or object detection, not recognition. The most interesting work is that of Major and Fontijne [11]. They used Range-Azimuth-Doppler Tensors representation to achieve well-performing automotive detection systems that work even for long distances.

Another category describes systems that use both camera and radar data for not only detection purposes but also recognition. RVNet [5] belongs to this category. It uses point cloud radar data representation and camera data. It consists of three parts: Image Feature Extraction Branch, Radar Feature Extraction Branch and Output Fusion Branch. The names of these parts are self-explanatory. Image feature extraction, and final convolution layers with reshape are identical to Tiny Yolo v3 [2]. Yet another example of a method using both camera and radar data is the work of Lim et al. [7]. They designed an early fusion system that uses range-velocity-azimuth maps and camera data and they achieved good results.

2.3 Synchronization Algorithms

Sensor synchronization is achievable through different procedures that usually require special equipment such as embedded cameras, cameras supporting specific protocol [9, 6]. With these specialized devices, it is possible to control the moment when a frame is taken with an outside signal and ensures precise synchronization. The problem with this approach is that mentioned equipment can be rather expensive.

It is also possible to use features such as video, audio, accelerometer, telemetry of recordings to synchronize multiple cameras [4, 12]. This approach is unfortunately not possible for a combination of non-visual data and camera data.

Another possible approach is to synchronize cameras using a clapper slate or other reference event. This procedure is, however, also not usable for non-visual data and does not guarantee synchronization over time if only one synchronization event is present.

In the next chapter, a synchronization method using the audio channel of a camera and low-level communication is proposed.

3 Proposed Synchronization Method

Sensor synchronization is a crucial problem for multi-sensor systems. Various existing methods are described in section 2.3. An example of a possible synchronization system for mmwave radar and embedded camera is shown in Figure 4. Mmwave radar can send signals to control shutter of the camera and accurately synchronize frames of the sensors. In the following section 3.1, a method using accessible off-the-shelf products is proposed.

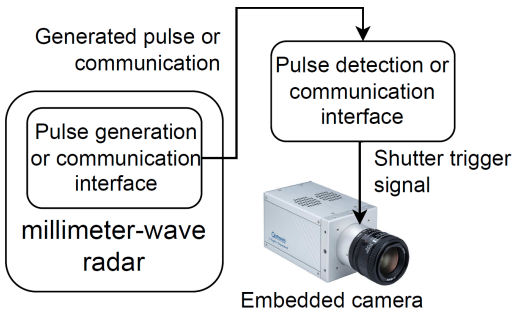

Figure 4: Mmwave radar and embedded camera synchronization diagram example. Image courtesy of Optronis GmbH.

3.1 Monocular Camera and Millimeter-Wave Radar Synchronization

In this work, we propose a synchronization system that is based on low-level communication protocols and audio channels of cameras. Millimeter-wave radars usually include communication interfaces such as I2C, SPI that can be utilized for pulse generation. Pulses generated can be used as an audio signal connected to the audio input of a camera. It is needed to adjust audio levels for audio voltage output levels that are lower than voltage used for mmwave radar communication. Functional synchronization diagram is shown in Figure 5.

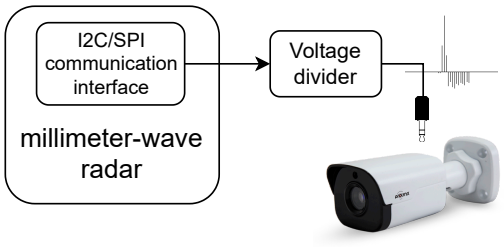

Figure 5: Millimeter-wave radar to camera synchronization diagram that uses low-level communication and connected to audio input. Image courtesy of Zhejiang Uniview Technologies Co., Ltd.

This approach is to the best of our knowledge unique and has not been proposed. It can achieve good results with accessible hardware and guarantee synchronization over time.

3.2 Multiple Camera Synchronization

A sensor synchronization system similar to one described in the previous section 3.1 can be also applied to multiple camera systems instead of camera-radar synchronization. Functional synchronization diagram is shown in Figure 6.

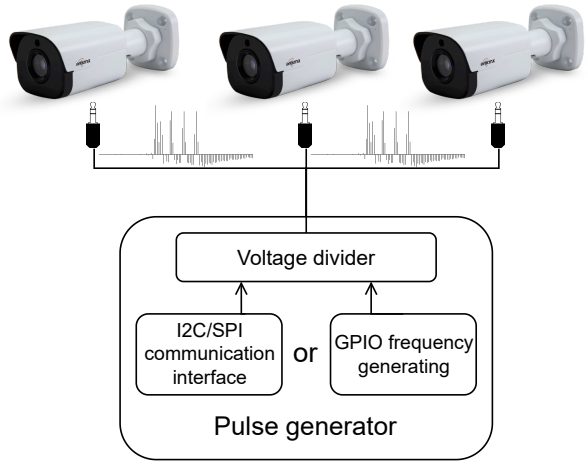

Figure 6: Multiple camera synchronization diagram that uses low-level communication and audio signals. Image courtesy of Zhejiang Uniview Technologies Co., Ltd.

4 Fusion Algorithms

This section presents previous experiments aimed at people detection and range estimation as well as the intended exploitation technique proposed.

4.1 Previous Work

An initial experiment using mmwave radar and camera data for fusion purposes has been conducted already. It uses mmwave radar data represented as point clouds and camera data for people detection and distance estimation.

In this early experiment, inter-sensor synchronization (discussed in the previous section) was not used yet. The synchronization was performed using timestamping of received data at the time of collection. Based on timestamps, the corresponding camera and radar frames were found.

Sensor calibration was based on sensor relative positions and camera extrinsic and intrinsic properties. Concatenation of camera and radar data was performed at the beginning of the processing and radar data were represented as the fourth input channel for the convolution neural network.

Experimental fusion results were acquired within the described initial experiment. It focuses on people detection and distance estimation. Examples of this system's results are shown in Figure 8.

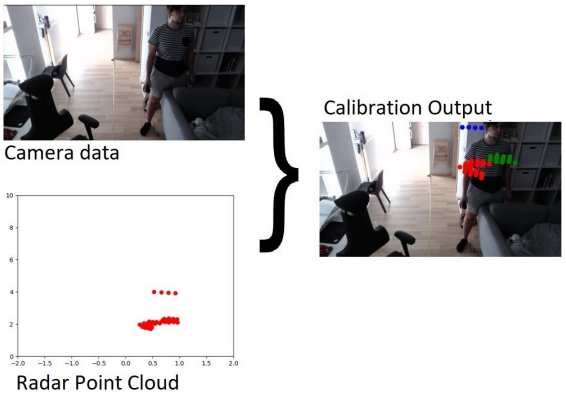

Figure 7: Diagram of sensor calibration of uncalibrated mmwave radar point cloud and image.

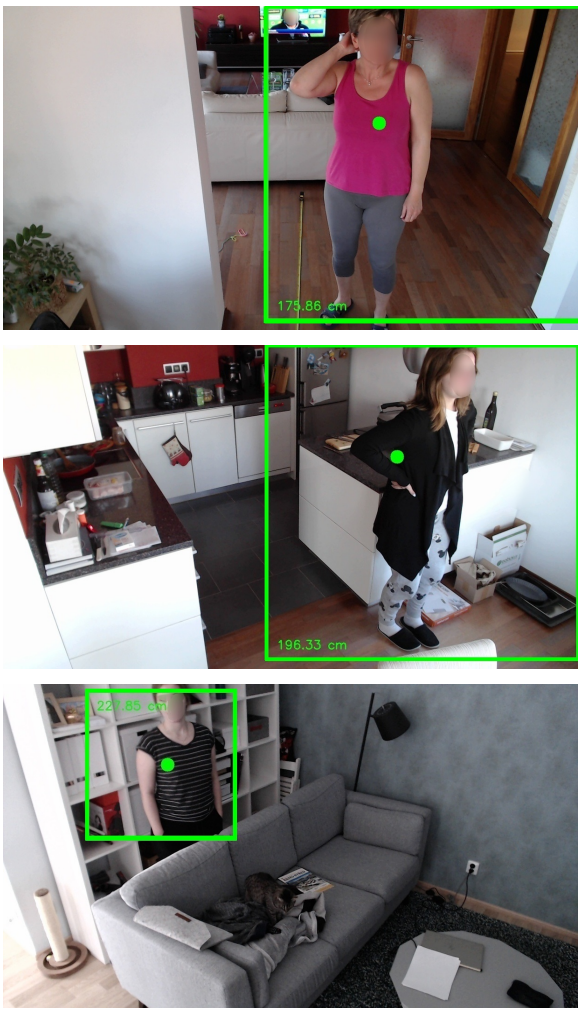

Figure 8: Fusion detection system results.

4.2 Intended Exploitation

As described in Section 2.1, range-(velocity)-azimuth maps hold more information and so they are more useful for recognition purposes. That is the reason why we are going to use this representation as an input of a neural network inspired by RVNet [5] instead of mmwave radar point clouds. A concept summary of proposed system is shown in Figure 9. The detailed architecturee of the neural network itself, for now heavily inspired by RVNet, is available in the appendix of this paper.

5 Experimental Reults

This section introduces an experiment for evaluation of the used camera's video and audio channels synchronization and the setup needed for this experiment. It is crucial to validate whether the channels of the camera are synchronized sufficiently for the method proposed in section 3.

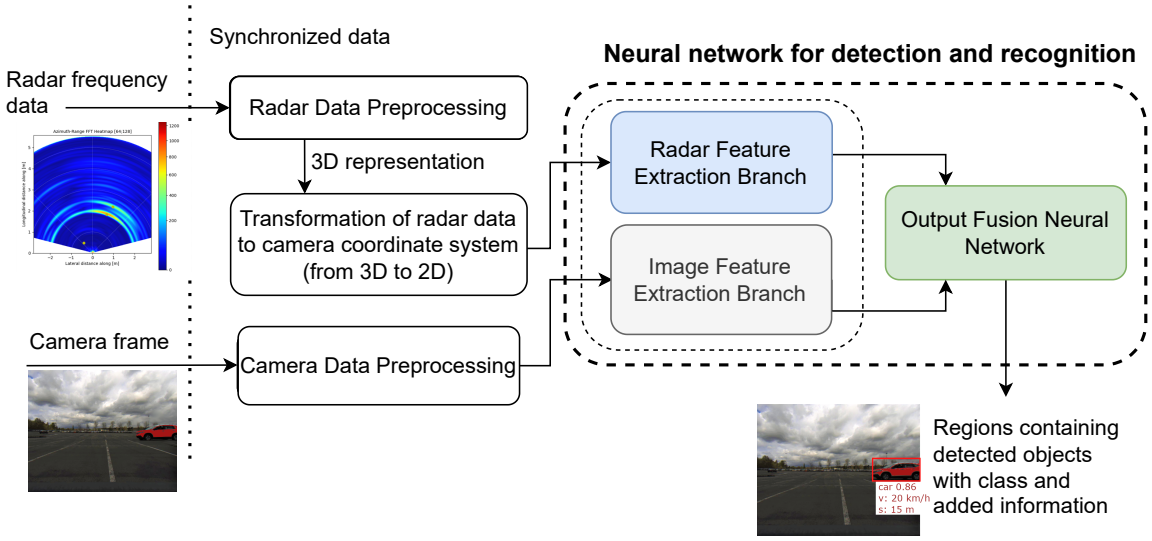

Figure 9: Fundamental diagram of the proposed system that uses radar frequency data in a form of range-(velocity)-azimuth map and camera data for detection-recognition purposes, the output is an image with detected objects with assigned class and additional information, detailed architecture of the neural network is available in the appendix of this paper.

5.1 Experimental Setup for Synchronization Evaluation

Unique synchronization approach is proposed in this work. To confirm its accuracy, tests were needed. The potentially problematic part of this approach is the camera's video to audio synchronization stability and also the relative shift of audio and video frames that needed to be determined and are not necessarily guaranteed. We used an Arduino prototyping platform to control a LED at the same time as generating pulse for audio camera input.

After an evaluation of audio and video camera channels synchronization of camera recording in a video editor, it is, in given circumstances, e.g. light conditions and camera exposure time, possible to establish the delay between the video and audio camera channels. For this purpose a simple circuit was designed as presented in Figure 10.

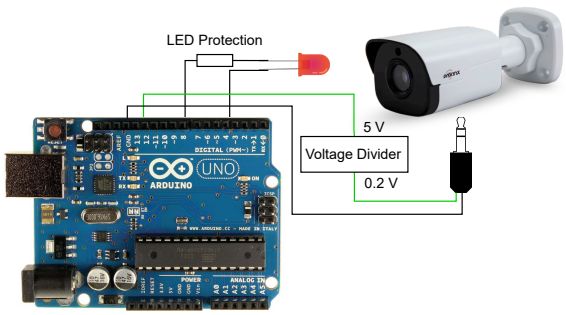

Figure 10: Circuit designed for camera video with audio channels synchronization tests. Image courtesy of Zhejiang Uniview Technologies Co., Ltd. and Arduino S.r.l.

5.2 Results of Synchronization Evaluation

The setup described in the previous chapter was used to verify the accuracy of an audio and video channels synchronization of IP camera Hikvision DS-2CD2686G2-IZS in the given conditions. For this purpose, an Arduino onboard LED was used as shown in Figure 11.

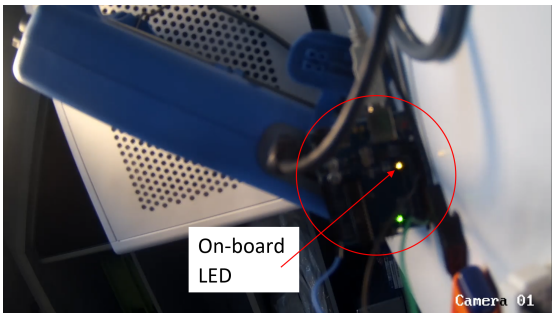

Figure 11: Footage from testing with marked LED.

This setup was then used to record test results that consist of video and audio channels. Thereafter, the difference between pulse generated by Arduino and LED state change was possible to acquire. The LED state change was based on the camera frame where the diode is turned off and the pulse timestamp was determined through a video editor with the millisecond resolution.

However, a problem with the camera frame frequency exists. The state of the diode is changed before it appears on a frame because of the low framerate of the camera. The difference is, therefore, not accurate. It is possible to tackle this problem by recording multiple independent video frames if we assume that the difference between

channels of the camera is constant in the given conditions for the camera used in each recording. The lowest difference is then going to be the real delay of the video channel in relation to the audio channel. The following table shows the obtained delays in series of independent measurements.

If we follow the assumption described previously, the real delay, based on Table 1, is 10 *ms* from test: $id = 13$. It is then possible to use this information to synchronize sensors properly. This approach could be improved with more LEDs usage, e.g. Gray code counter to measure the delay. With Gray code LED counter, it would be possible to acquire the delay between channels as well as the delay variability using only one test recording.

Test ID	1	2	3	4	5
LED Delay	28 ms	30 ms	20 ms	37 ms	24 ms
Test ID	6	7	8	9	10
LED Delay	29 ms	31 ms	35 ms	37 ms	48 ms
Test ID	11	12	13	14	15
LED Delay	45 ms	36 ms	10 ms	31 ms	29 ms
Test ID	16	17	18	19	20
LED Delay	16 ms	13 ms	35 ms	42 ms	17 ms

Table 1: Video, audio channels synchronization test results, each test is a camera recording and the "LED Delay" is the difference between channels in the given test recording.

6 Conclusions

In this paper, we propose a synchronization method using off-the-shelf products that are synchronized by audio signals passed between sensors in the context of sensor fusion. This method proved to be applicable for sensor synchronization. Millimeter-wave (mmwave) radar and the monocular camera are used to demonstrate sensor fusion. A summary of fusion methods using these sensors is provided as well as a summary of synchronization methods. The initial experiment with detection using mmwave radar and camera is presented with example results. Finally, a future exploitation fusion algorithm for the sensors mentioned is provided.

References

[1] Fahad Jibrin Abdu, Yixiong Zhang, Maozhong Fu, Yuhan Li, and Zhenmiao Deng. Application of deep learning on millimeter-wave radar signals: A review. *Sensors*, 21(6), 2021.

[2] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 687–694, 2020.

[3] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2d car detection in radar data with pointnets. *CoRR*, abs/1904.08414, 2019.

[4] Lex Fridman, Daniel E Brown, William Angell, Irman Abdić, Bryan Reimer, and Hae Young Noh. Automated synchronization of driving data using vibration and steering events. *Pattern Recognition Letters*, 75:9–15, 2016.

[5] Vijay John and Seiichi Mita. Rvnet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments. In Chilwoo Lee, Zhixun Su, and Akihiro Sugimoto, editors, *Image and Video Technology*, pages 351–364, Cham, 2019. Springer International Publishing.

[6] Sami M. Lasassmeh and James M. Conrad. Time synchronization in wireless sensor networks: A survey. In *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, pages 242–245, 2010.

[7] Teck-Yian Lim, Amin Ansari, Bence Major, Daniel Fontijne, Michael Hamilton, Radhika Gowaikar, and Sundar Subramanian. Radar and camera early fusion for vehicle detection in advanced driver assistance systems. *NeurIPS Machine Learning for Autonomous Driving Workshop*, 2019.

[8] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. *2019 16th European Radar Conference (EuRAD)*, pages 133–136, 2019.

[9] D.L. Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.

[10] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. *CoRR*, abs/2005.07431, 2020.

[11] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Object detection under severe conditions using vision-radio cross-modal supervision. *CoRR*, abs/2003.01816, 2020.

[12] Zhe Zhang, Chunyu Wang, and Wenhu Qin. Semantically synchronizing multiple-camera systems with human pose estimation. *Sensors*, 21(7), 2021.

7 Appendix

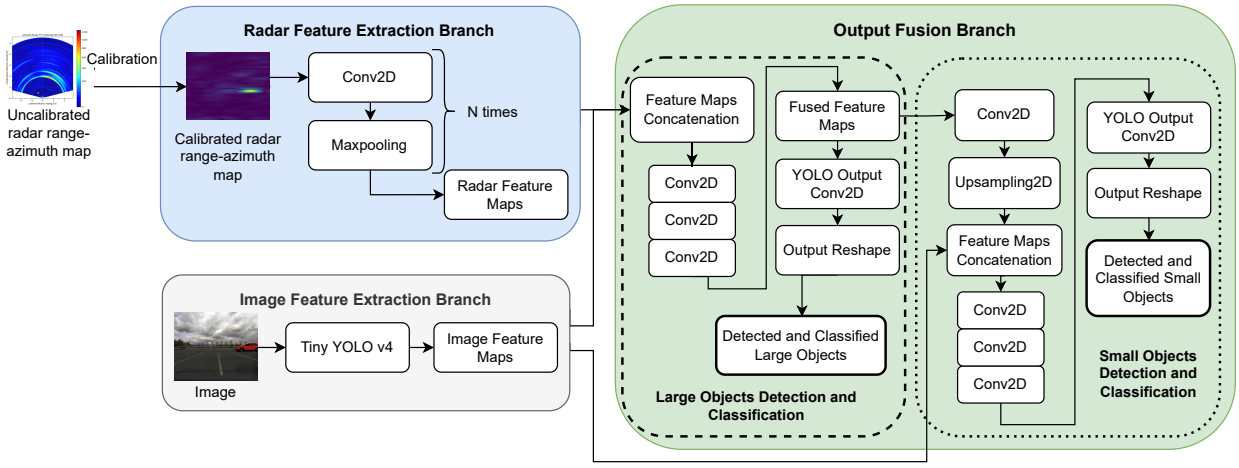

Figure 12: Proposed fusion detection-recognition system neural network architecture.