Multi-sensor fusion for autonomous driving

Authors: Siavash Hosseinyalamdary, Qiao Ren, Xinran Wang

Abstract:

A safe maneuvering of an autonomous vehicle on the road requires successful accomplishment of many high level tasks. Due to the limitation of each sensor, several sensors should be efficiently integrated to cover the shortages of each sensor and provide robust solution to each task.

In this chapter, we explain the Bayes filters applied for multi-sensor integration and the shortcomings and limitations of these approaches are elaborated. Furthermore, the recent advancements of sensor fusion using deep learning are discussed and several network architectures of sensor integration are described.

The integration of multiple sensors are exemplified for several tasks in autonomous driving, including autonomous vehicle localization, nearby moving objects detection and tracking, traffic light and sign detection, and human detection.

We conclude multi-sensor integration improves the achievable performance in each autonomous driving task and we claim the use of several sensors increases the robustness of autonomous driving.

1. Introduction

Autonomous driving is one of emerging multi-disciplinary research areas, which has attracted a lot of attention. In autonomous driving, the vehicle takes full control of driving and therefore, it mimics the required tasks that a human driver should accomplish. These tasks can be categories into two classes: perception and planning. In perception, the vehicle should collect data from surrounding environment, analyze it, and extract the essential information for driving. In planning, the vehicle should decide about proper action based on the observed surrounding environment.

In perception phase, the vehicle uses several sensors to locate itself and nearby objects, understanding traffic signs and road marks. Among these sensors, Global Navigation Satellite System (GNSS) is the primary navigation sensor applied in the autonomous vehicle. The GNSS receiver collects the satellite signals and estimate its location. Unfortunately, the GNSS positioning requires direct visibility of GNSS receiver from satellites and therefore, it does not provide accurate positioning in the urban environment with tall buildings. There are a number of alternative navigation systems including Inertial Measurement Unit (IMU) and visual odometry. These sensors estimate the relative position of vehicle and their error grows over time. Therefore, they can estimate the location of vehicle for a short period. The development of an accurate and reliable alternative navigation solution has remained a challenge.

The detection and tracking of nearby vehicles, cyclists, and pedestrians is another challenging task in computer vision and deep learning. Vehicles, cyclists, and pedestrians can have various shapes and they can be seen from different poses. Usually, these objects are partially or fully occluded in crowded roads. They can appear quickly on the road and become an immediate risk of collision. Therefore, fast detection of these objects is critical. In addition, their pose and velocity should be estimated and their position is predicted in future. For instance, a stopped vehicle in the highway is dangerous, while this vehicle may not be a risk if it drives with the same speed of the autonomous platform.

An autonomous vehicle should follow traffic signs and traffic lights. Some of the traffic signs and traffic lights may be temporary and they may not be marked in the maps. An autonomous platform should be able to detect traffic signs and their information should be retrieved. Some of the informational traffic signs contain text and therefore, a text retrieval algorithm should be applied to extract proper information. In contrast, traffic lights may be confused with similar objects, such as tail light of vehicles.

At last but not least, the autonomous platform should detect lanes and road boundaries. Therefore, the lane marks and road boundary marks should be correctly detected. In urban areas, the road boundary marks are replaced with curbs and the autonomous platform should be able to detect curbs. In addition, lane marks may be erased or occluded by other vehicles and the autonomous platform should approximate these marks based on road width.

Autonomous driving is a critical task and failure in any of these tasks may cause an accident. Despite of great advancement in computer vision and machine learning, the success rate of intelligent algorithms is still lower than human. Therefore, autonomous driving based on cameras are still a challenge. However, the integration of several sensors may cause sufficient redundancy and mitigate the failure rate. In this chapter, we explain a few of multiple sensor integration approaches applied in autonomous driving.

2. Sensor fusion

In critical applications, such as autonomous driving, the use of multiple sensors improves the reliability of the system. For instance, the people in the shadow may not be detected if visual images are used, but the thermal images may be able to detect people in the shadow and at night. The use of several sensors improve the reliability of the system and reduces its failure rate. There are two sensor fusion approaches: statistical sensor fusion and deep learning based sensor fusion.

2.1. Statistical sensor fusion

In statistics, several approaches have been developed to integrate multiple sources. If the multiple sensors are complementary, the least squares can be applied to estimate the unknowns, such as the position of platform. However, the multiple sources may provide supplementary information. For instance, some sensors provide the increments and decrements of unknowns. In such a scenario, Kalman and particle filters are applied. If multiple sources of information are applied in the classification, multiple kernel learning, which is based on Expectation Maximization (EM) algorithm in statistics, is applied.

2.1.1. Multiple sensor error propagation

Let's assume there are two independent observation sources and these observations have normal distribution. Therefore, the two observation vector, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, are expressed by their mean and variance, such that:

$$\begin{aligned} \boldsymbol{z}_1 &= \mathcal{N}(\mu_1, \sigma_1^2) \\ \boldsymbol{z}_2 &= \mathcal{N}(\mu_2, \sigma_2^2) \end{aligned} \tag{1}$$

If we use these two sensors to estimate the unknown value, x, The maximum likelihood of the unknown value is estimated, such that:

$$\hat{x} = arg\max_x \Pr(\boldsymbol{z}_1, \boldsymbol{z}_2 | x) \tag{2}$$

where $\Pr(.|.)$ is the conditional probability. Since the two observation sources are independent given the unknown variable, equation (2) is decomposed into two conditional probabilities. The negative log likelihood of equation (2) is calculated, such that:

$$\mathcal{L}(\boldsymbol{x}) = -\log(\Pr(\boldsymbol{z}_1, \boldsymbol{z}_2|x)) = -\log(\Pr(\boldsymbol{z}_1|x)) - \log(\Pr(\boldsymbol{z}_2|x))$$
$$\hat{x} = \underset{\boldsymbol{x}}{arg\min}\, \mathcal{L}(x) \tag{3}$$

By replacing normal distribution of two observations, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, the maximum likelihood estimation of unknown variable is computed, such that:

$$\hat{x} = \frac{\sigma_2^2}{(\sigma_1^2 + \sigma_2^2)}\mu_1 + \frac{\sigma_1^2}{(\sigma_1^2 + \sigma_2^2)}\mu_2 \tag{4}$$

Therefore, the unknown variable is estimated based on the two observation vectors, their means and variances. This approach can be generalized when there are multiple observation sources.

2.1.2. Kalman filter

In sensor fusion, not all of the observations are always homogenous. Some of the observations are absolute and gives us the value of unknowns and some are relative and provide us with the increments and decrements in the unknowns. In these cases, Bayesian estimators, such as Kalman, are applied to integrate different sources of information.

The unknown vector in Kalman filter is estimated over time. The unknown vector, also called state vector, is represented by $\mathbf{x}_t \in \mathbb{R}^n$, where t indicates that it is the state vector at time t. The observation vectors at time *t*, $\mathbf{z}_t \in \mathbb{R}^m$ is used to estimate the state vector. Sometime, all of the previous observations are also applied and we denote all observations by $\mathbf{z}_{1:t}$. The initial state of the system should be known and represented by $\mathbf{x}_0$.

The estimation of current state vector depends on the observations and initial state and therefore, probability of state vector at the current time is $\Pr(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{x}_0)$. Therefore, Maximum Likelihood (ML) is applied to estimate the current state vector, such that:

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{x}_0) \tag{5}$$

Using the Bayes rule, the conditional probability in equation (5) is expanded, such that

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \frac{\Pr(\mathbf{z}_t|\mathbf{x}_t)\Pr(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_0)}{\Pr(\mathbf{z}_{1:t})} \tag{6}$$

The denominator of equation (6) does not depend on $\mathbf{x}_t$, and therefore, it can be removed from maximization, such that

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{z}_t|\mathbf{x}_t)\Pr(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_0) \tag{7}$$

By marginalization of the equation (7) over previous state vectors, it is transformed to

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{z}_t|\mathbf{x}_t) \int \Pr(\mathbf{x}_t|\mathbf{x}_{1:t-1})\Pr(\mathbf{x}_{1:t-1}|\mathbf{z}_{1:t-1}, \mathbf{x}_0) d\mathbf{x}_{1:t-1} \tag{8}$$

Based on Markovian assumption, the current state vector is independent from previous state vectors, such that:

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{z}_t|\mathbf{x}_t) \int \Pr(\mathbf{x}_t|\mathbf{x}_{t-1}) \Pr(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{x}_0) d\mathbf{x}_{t-1} \tag{9}$$

where $\Pr(\mathbf{z}_t|\mathbf{x}_t)$ is the posterior probability estimation of the current state vector. Similar to equation (5), previous state vector is estimated, such that:

$$\hat{\mathbf{x}}_{t-1} = \underset{\mathbf{x}_{t-1}}{\operatorname{argmax}} \Pr(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{x}_0) \tag{10}$$

By replacing equation (10) into equation (9), the current state vector is calculated, such that

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{z}_t|\mathbf{x}_t) \int \Pr(\mathbf{x}_t|\mathbf{x}_{t-1}) \hat{\mathbf{x}}_{t-1} d\mathbf{x}_{t-1} \tag{11}$$

where $\Pr(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_0) = \int \Pr(\mathbf{x}_t|\mathbf{x}_{t-1}) \hat{\mathbf{x}}_{t-1} d\mathbf{x}_{t-1}$ is the prior probability estimation since the current state vector is calculated from previous observations and current observation vector, $\mathbf{z}_t \in R^m$, does not contribute in the prior estimation of current state vector.

Let's assume there are two functions that relate the state vector to the previous state vector and observation vector, f and g, such that:

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}) + \epsilon_t \\ \mathbf{z}_t &= g(\mathbf{x}_t) + \omega_t \end{aligned} \tag{12}$$

where ϵ_t and ω_t are noise of system and observation models. The Kalman filter assumes noise of system and observation follow the normal distribution with the variance matrices of Q_t and R_t, such that:

$$\begin{aligned} \epsilon_t &\sim \mathcal{N}(0, \mathrm{Q}_t) \\ \omega_t &\sim \mathcal{N}(0, \mathrm{R}_t) \end{aligned} \tag{13}$$

The Kalman filter is a linear estimator. Therefore, it can estimate the uknowns when the system and observation model, f and g, are linear. Linear functions are represented by matrices, *F* and *G*, and therefore, equation (12) is transformed into

$$\begin{aligned} \mathbf{x}_t &= \mathrm{F}\mathbf{x}_{t-1} + \epsilon_t \\ \mathbf{z}_t &= \mathrm{G}\mathbf{x}_t + \omega_t \end{aligned} \tag{14}$$

The Kalman filter estimates the optimum values of state vector in two stages: prediction and update. In the prediction, the state vector is estimated based on the system model. In this stage, the relative observations are applied to estimate the current state vector based on the previous state vector. In the update stage, the state vector is estimated based on the observation vector and it is integrated with the state vector estimated in prediction stage. The contribution of the system and observation models in the estimation of state vector is calculated based on the Kalman gain matrix, K_t. Figure 1 shows the Kalman filter: its prediction stage (left box), its update stage (right box), and its Kalman filter gain (lower box). For more details, readers are referred to Hosseinyalamdary (2018).

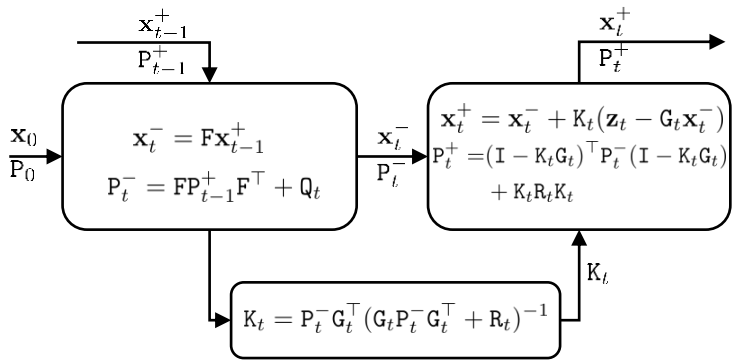

Figure 1: The Kalman filter and its prediction (left-up box), update (right-up box) stages and Kalman filter gain (lower box).

The Kalman filter has several limitations: Kalman filter is a linear estimator and it can be applied when the system and observation models are linear. In addition, noise of system and observation models should be Gaussian and follow normal distribution. There are a number of variations to the Kalman filter, such as extended and unscented Kalman filters and particle filter.

In the Kalman filter, the observation and system models should be known. These models are modelled in different ways. For instance, Jekeli (2001) models the bias of inertial sensors as random constant while Noureldin et al. (2011) models this bias as first-order Gaussian Markov random process. Unfortunately, the observation and system models cannot be determined beforehand in many applications.

Kalman and particle filters are based on Markovian assumption and the current state vector depends on previous state vector. This property simplifies the Kalman and particle filter, but the system model cannot model long correlations in the state vectors.

2.1.3. Multiple kernel learning

In the previous section, the estimation of a continuous parameter was required. This problem is called regression since the unknowns are continuous and they are estimated in the continuous space. In contrast to regression, classification is a discrete problem and the observations should be categorized in different discrete classes. Multiple kernel learning, explained in machine learning books (Marsland 2015), uses Expectation-Maximization (EM) to estimate the probability distribution of the observations and assign the observations to their classes. The statistical classification approaches have lost their popularity when deep learning has shown great success in classification.

2.2. Sensor fusion based on deep learning

By the advancement of deep learning, we are able to fulfill the required tasks in autonomous driving. Deep learning has brought computer vision into another level and many human tasks can be replaced with machine using deep learning. Fusion of sensors can be done in different layers of a deep network. Therefore, the sensor fusion is categorized into early, halfway, and late fusion.

Early fusion is also called pixel-level fusion since it integrates images from different sensors on pixel level in image processing. In early fusion, the output of multiple sensors should be consistent. For instance, the thermal and visual images can be integrated in pixel level, if the resolution of two sensors are the same. The image manipulation can be applied to adjust two sensor data, but the image manipulation loses

information and adds artifacts in the images. In these scenarios, halfway fusion is more efficient in these cases. Figure 2a shows the early fusion when the two sensors are integrated in the very early layer of the network.

In contrast to other machine learning approaches, deep learning automatically detect the prominent properties of the input data and apply them to reach desired output. The prominent properties are called feature. Halfway fusion, also known as feature-level fusion, integrates the features of different sensors and apply the integrated features to achieve the desired output. For instance, if two imagery sensors have different resolutions, the features of their images are extracted in a deep network, these features are combined together and the integrated features are utilized to achieve the desired output. Figure 2b shows halfway fusion since the features of two sensors are integrated.

In the late fusion, also known as decision-level fusion, the two sensors are independently applied to achieve the desired output. At the end, the output of these independent network are integrated based on the output of each sensor. For instance, several cameras may look at the scene from different perspectives for scene understanding, the images of each camera are applied to classify the objects in the scene and a voting scheme is utilized to integrate the classified objects of each camera and make final decision. Figure 2c shows the late fusion and two output results of the network are integrated in the last layers.

The choice of sensor fusion depends on the type of sensor information. If the sensors are consistent, early fusion is applicable. On the other hand, the late fusion can be applied in the fusion of complementary sensors. For instance, it is not possible to integrate the audio and video for speech recognition using early fusion, since the images are represented by gray values and audio waves are represented by the amplitude and frequency of audio signals.

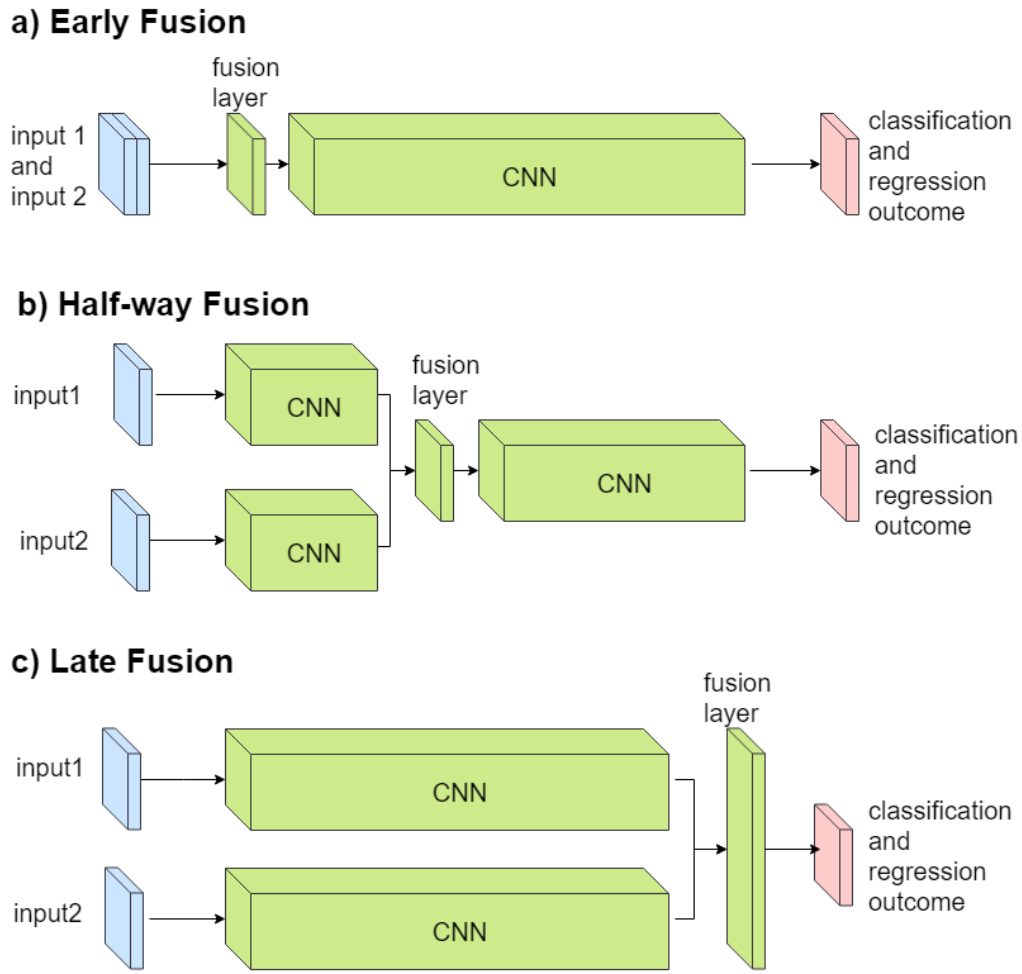

Figure 2: Sensor fusion using deep learning; (a) the input data are integrated in early fusion (b) the features are integrated in middle layers of the network in halfway fusion (c) the late fusion is applied in the last years and the decisions of several independent networks are combined together.

2.3. Deep Kalman filter

The Kalman filter is based on Markovian assumption and the state vector only depends on the previous state vector. In deep Kalman filter, the current state vector, $\mathbf{x}_t$, depends on the previous state vectors $\mathbf{x}_{t-1:t-T}$. In addition, the current state vector also depends on the latent vectors, $\mathbf{h}_{t:t-T}$. Let's assume there is a function ϕ that relates the current latent vector to the previous latent vectors and previous state vectors, such that:

$$\mathbf{h}_t = \phi(\mathbf{x}^+_{t-1:t-T}, \mathbf{h}_{t-1:t-T}) \tag{15}$$

The current state vector is directly related to the current latent vector by a function, λ, such that:

$$\mathbf{x}^{+-}_t = \lambda(\mathbf{h}_t) + \mu_t \tag{16}$$

where $\mathbf{x}^{+-}_t$ is the predicted posterior estimation of state vector. Functions ϕ and λ are a combination of linear and non-linear functions, represented by coefficient matrices, $\mathbb{W}$, and non-linear function, σ, such that:

$$\mathbf{h}_t = \sigma(\mathbb{W}_{xh}\mathbf{x}^+_{t-1:t-T} + \mathbb{W}_{hh}\mathbf{h}_{t-1:t-T}) \tag{17}$$

$$\mathbf{x}_t^{+-} = \sigma(\mathrm{W}_{xx}\mathbf{h}_t) + \mu_t$$

where the parameters of latent vector, $\mathrm{W}_h = [\mathrm{W}_{xh}, \mathrm{W}_{hh}]$, and state vector, W_{xx}. Equation (17) can be modelled using Recurrent Neural Network. This equation can be integrated with the Kalman filter. Figure 3 shows the deep Kalman filter where the lower box is given in equation (17) and can be modelled by deep learning. Interested readers are referred to Hosseinyalamdary (2018).

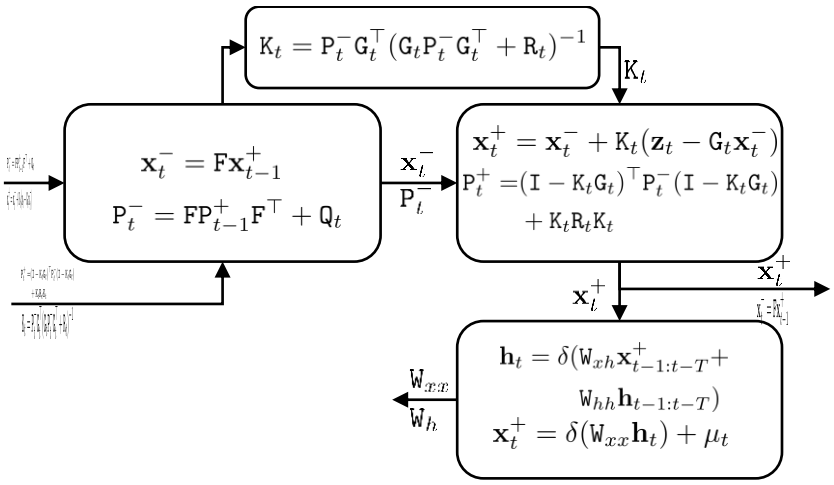

Figure 3: The deep Kalman filter. The lower part models the system using deep learning.

3. Sensor fusion in autonomous driving

3.1. Multi-sensor localization

The localization of the platform is an important part of autonomous driving. If location of the autonomous vehicle is known in a global coordinate system, the static traffic feature, such as traffic signs and lights, can be retrieved from maps. In addition, the contextual information aid to fulfill the autonomous driving tasks. For instance, the knowledge of driving in the highway, lowers the likelihood of pedestrian detection on the road.

The accurate localization of the autonomous vehicle has still remained a challenge. Global Navigation Satellite Systems (GNSS) is the primary navigational source of information in autonomous driving. In GNSS, the GNSS receiver is mount on the autonomous platform, and it localizes itself based on the signals it receives from GNSS satellites. However, it requires direct visibility of the satellites in the sky and its accuracy deteriorates in the urban environment where the satellite signals are frequently blocked by tall buildings. There are alternative navigation solutions, such as Inertial Measurement Unit (IMU) and visual odometry. However, these approaches are relative and they cannot accurately position the autonomous vehicle for a long period.

The Kalman filter has been used to integrate GNSS and IMU (Jeleki 2001, Noureldin et al. 2011). In the Kalman filter, the current state vector is predicted based on the previous state vector using IMU observations. The GNSS observations are applied in the update stage of the Kalman filter. The consecutive images are also utilized to estimate the displacement and rotation of the camera mount on the platform, known as visual odometry (Scaramuzza and Fraundorfer 2011, Fraundorfer and Scaramuzza 2012). There are several studies on the integration of visual odometry and inertial navigation (Hosseinyalamdary and Yilmaz 2014, Qin et al. 2018). In addition, in holistic approaches, several sensors are mounted on autonomous driving platform, such as GNSS, IMU, visual odometry, LiDAR odometry and map, and they

are integrated to estimate the location of platform (Hosseinyalamdary et al. 2015, Balazadegan et al. 2016).

Deep learning has been applied for more efficient integration of the navigation sensors. Clark and his colleagues (2017) has introduced Visual-Inertial odometry Network (VINet) and they integrate IMU information with a sequence of images to estimate the relative pose of the platform. Wang and his colleagues (2018) have introduced Deep Visual Odometry (DeepVO) where they integrate Convolutional Neural Network (CNN) with the Recurrent Neural Network (RNN) to estimate the pose of camera over time. Wang et al. (2018) use deep learning to integrate images, GNSS/IMU, and semantic maps. Milz and his colleagues have used statistical sensor fusion to localize the autonomous vehicle, while they have accomplished several tasks using deep learning (Milz et al. 2018).

3.2. Nearby vehicle detection and tracking

Monitoring adjacent vehicles and predicting their movement is essential to prevent collision in autonomous driving. In addition to visual images, the radar, ultrasound, and laser scanner are frequently used in autonomous driving for collision avoidance. The integration of these sensors improve the detection of adjacent vehicles and it significantly reduces the accidents in autonomous vehicles.

Chellappa et al. (2004) integrate visual and ultrasound sensors to detect vehicles. Kim and his colleagues (2005) propose to detect and track vehicles using visual and sonar sensors. They claim their method is not influenced by lighting conditions and distance. Liu and his colleagues (2008) integrate radar and visual images to detect nearby objects. They report this approach effective for vehicle detection. Liu et al. (2011) also propose the integration of radar and visual images. The results of visual and radar sensors are applied to verify the detection results and improve its accuracy. In this way, the deficiencies of individual sensors are compensated by their integration.

In addition, the deep learning based sensor fusion has been applied to detect adjacent vehicles and track them. Chen and his colleagues (2016) integrate data of laser scanner and camera for vehicle detection. In their work, they use laser scanner information and generate object proposals. The object proposals are applied in visual images to detect the nearby vehicles. Asvadi et al. (2017) also propose the fusion of LiDAR and color camera for vehicle detection. They apply decision-level sensor fusion to integrate the results of the reflectance and range of LiDAR with the result of visual images. Selbes and Sert (2017) apply visual and acoustic sensor fusion to detect different classes of vehicles. Since acoustic and visual data are very different, they used Convolutional Neural Network (CNN) and the Mel Frequency Cepstral Coefficient (MFCC) features and integrated the sensors in the feature level.

3.3. Pedestrian detection

Pedestrian detection is one of the critical tasks in autonomous driving. The autonomous vehicle should be able to detect people on the streets, track them, estimate their location and motion, and avoid any risk of accident. Human detection has been studied for more than three decades, but it has remained a challenge. Human body can have different shape, color, and posture; people may wear different cloths and costumes; and they may be partially or fully occluded by vehicles. Therefore, a robust human detection algorithm is required for autonomous driving.

Visual camera is able to perceive the environment if the lighting condition is suitable. The visual images have poor quality at nights and under-shadow regions. In contrast, thermal images sense the human body

temperature at night. Therefore, thermal camera is applied to detect human body at night. Thermal images have lower resolution and thermal information is not as detailed as visual images. As a result, the integration of thermal and visual images is benefited from the advantages of both sensors. The thermal and visual images are frequently integrated for pedestrian detection in autonomous driving. Figure 4 shows the human detection in visual and thermal images.

Figure 4: The pedestrian detection in visual and thermal images. Courtesy of KAIST dataset (Hwang 2015).

The most popular human detection algorithm is developed by Dalal and Triggs (2005). In their seminal work, they apply the Histogram of Oriented Gradient (HOG) as a feature for human body detection and they learn this feature using labeled human body in images. Dalal and Triggs have applied this approach to detect human body in visual images, but this approach has been effectively applied for thermal images. The visual and thermal images are integrated to create thermal-visual images and the HOG signature of these integrated images are applied to detect human bodies (Xu 2017).

In deep learning, different tasks are divided into two categories: classification and regression. Detection tasks belong to regression. In contrast to classification, the results of regression is the estimation of continuous values rather than categorizing images. There have been extensive research for object detection in deep learning. The most popular detection approaches are Region-based Convolutional Neural Network (R-CNN) (Girshick et al. 2014), Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015), Single Shot Detector (SSD).(Liu et al. 2016),You Only Look Once (YOLO) (Redman et al. 2016), and RetinaNet (Lin et al. 2017). These detection networks are originally developed for objection detection in visual images. However, these approaches have been generalized to work on two or more imagery sensors.

Li and his colleagues (2018) propose illumination aware Faster R-CNN for robust multispectral pedestrian detection. They apply various sensor fusion architectures including early, halfway, and late fusion to integrate thermal and visual images and detect pedestrians on the road. They conclude halfway and score fusion outperform other approaches and the early fusion performs the worst. They argue color images create confusion and it is better to convert them to monochromatic images.

König and his colleagues (2017) apply a sequence of multispectral images to detect pedestrians and therfore, they enforce temporal consisitency in human detection through the image sequence. They also

conclude that the halfway sensor fusion architecture is the best among different architectures. They show middle features are the best to integrate.

Cai and his colleagues (2017) apply thermal and visual images and they conclude the pixel-level fusion outperforms the other architectures. They claim the late fusion combined with joint bilateral fusion have a great potential in pedestrian detection at night. They also claim pixel-level sensor fusion may sometimes result in partial information loss and it may degrade the performance.

Guan and his colleagues (2018) propose a network to distiguish between daylight and nighttime and suggest different networks for daylight and at nighttime. The features extracted from two imagery sensors, thermal and visual cameras, are integrated to estimate the illumination. The weights of these networks are estimated for daylight and nightime based on the estimated illumination.

3.4. Traffic sign and light detection

Accurate traffic sign and light detection is one of the critical tasks in autonomous driving. An autonomous vehicle should properly detect the traffic signs, classify them, and take proper actions to follow them. The traffic lights should also be detected and their signals should be tracked over time.

Dalaff and his colleagues (2003) integrate color and thermal images to detect traffic lights. Peker and his colleagues (2014) propose the integration of map matching with image based traffic sign detection and they claim they can detect traffic signs in different lighting conditions. The integration of LiDAR data and camera images are utilized by Zhou and Deng (2014) to detect and classify the traffic signs.

In their seminal work, Sermanet and Lecun (2011) applied deep learning to classify various traffic signs using visual images. John and his colleagues (2014) have also applied deep learning for the traffic light detection. Meng and his colleagues (2017) use SSD to detct traffic signs in the visual images. They claim they can detect small traffic signs in the large images. Behrendt and his colleagues (2017) propose a traffic light monitoring and recognition system based on a sequence of stereo images.

3.5. Multi-task learning

In multi-task learning, the network is trained to perform several tasks simultaneously. Multi-task learning improves the performance of single-task learning: firstly, a network with single task learning can do overfitting to this single task while multi-task learning can avoid overfitting since it is more adaptive to different tasks; secondly, multi-task learning reduces computation complexity since several tasks share early features in the network. Figure 5 shows a schematic architecture of a multi-task learning network.

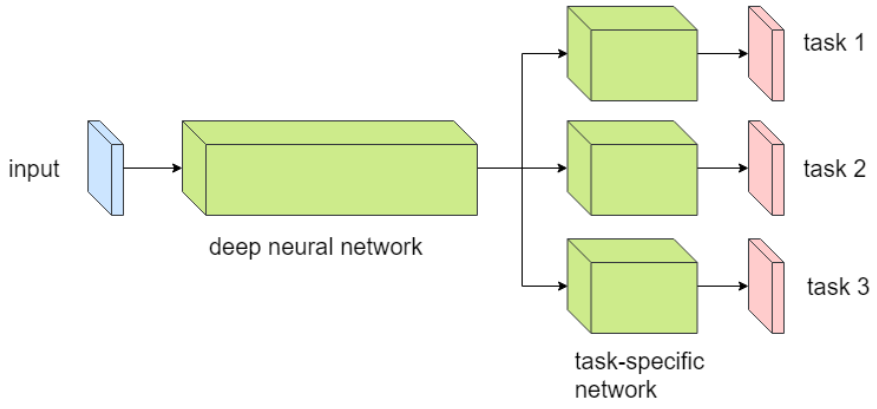

Figure 5: Multi-task learning; various tasks share the early features of the netwok.

Teichmann and his colleagues (2016) introduce MultiNet that performs classification, detection and semantic segmentation simultaneously. Siam et al. (2017) propose a two-stream architecture that combines motion and appearance cues and jointly learns object detection, road segmentation and motion segmentation. Chowdhuri and his colleagues propose a multi-sensor multi-task learning network for autonomous driving. They claim this approach outperforms multiple individual networks trained separately.

In their seminal work, Kendall and his colleagues (2018) integrate several tasks, such as pedestrian detection, depth estimation, and semantic segmentation, and it outperforms the individual tasks (Kendall et al. 2018). Chen et al. (2018) solve object detection and danger assessment in a multi-task learning network. In other words, they detect objects and predict their distance, simultaneously. They claim Cartesian product-based is better than other multi-task combination strategies.

3.6. End-to-end learning

End-to-end learning network is a network that it transforms the data into proper actions without performing intermediate tasks. For a multi-task autonomous driving, the several perception tasks should be accomplished and the proper routes should be planned. Based on the sensed environment and planned route, the proper command are sent to the actuators of the autonomous vehicle. In contrast, end-to-end learning attempts to transfer the input data, such as visual images, into proper commands. In other words, an end-to-end autonomous vehicle collects visual images and other information sources and generates the required steering and throttle commands to actuators. Figure 6 shows an end-to-end network of aoutnomous vehicle.

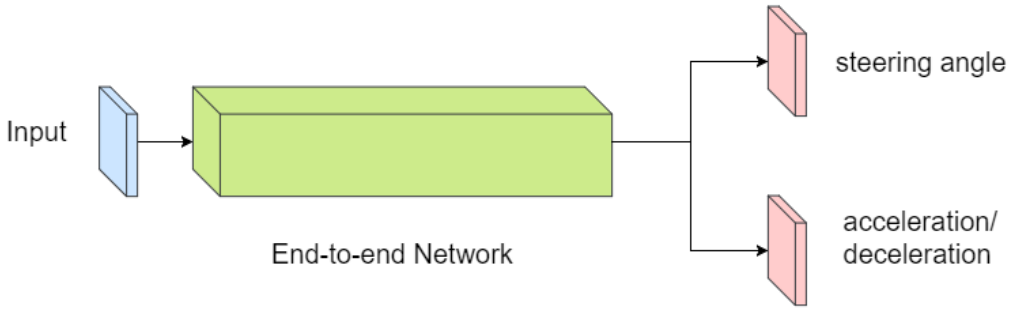

Figure 6: The deep Kalman filter. The lower part models the system using deep learning.

Pan et al. (2017) have built an end-to-end autopilot system for agile off-road autonomous driving. The system relies more on the theory of imitation learning. The system successfully demonstrated high-speed off-road autonomous driving in tests. Yang et al. (2018) use the prior speed and visual images to predict the proper steering angle and speed of autonomous vehicle.

Mehta et al. (2018) propose a multi-task learning end-to-end network for autonomous driving. They claim the end-to-end network performs better if additional tasks are given to the network. In other words, they claim that the joint learning of axillary and main tasks, help the system to learn quickly and it improves the transparency of the network.

Jaritz et al. (2018) apply reinforcement learning algorithms to end-to-end networks and they introduce reward and learning strategies to this system. The system only gets information from a front-facing color camera and generates proper actions for autonomous vehicle. They demonstrate that the vehicle learn the road structure, such as turns, and follow the road.

4. Summary

Multi-sensor integration is crucial in autonomous driving, where the failure rate should be less than human failure rate. The use of multiple sensors improve the accuracy of different task and provide a reliable solution to autonomous driving problems.

In this chapter, multi-sensor integration has been explained using statistical filters and deep learning. Statistical filters include least squears, Kalman filter, particle filter, and multiple kernel learning. The deep learning approaches include early, halfway, and late fusion. The studies have diverging idea on the performance of the early, halfway, and late fusion. It it suggested that proper sensor fusion benchmarks are developed and these approaches are systematically tested.

Several tasks in autonomous driving have been expalined in this chapter and it has been shown that multi-sensor integration can help to improve the results of these tasks. The multi-task learning and end-to-end learning have also been described in this chapter and different architectures of multi-sensor, multi-task, and end-to-end learning are elaborated.

Reference

S. Hosseinyalamdary, Deep Kalman Filter: Simultaneous Multi-Sensor Integration and Modelling; A GNSS/IMU Case Study, Sensors, 18, 5, 1316, 2018.

S. Hosseinyalamdary, A. Yilmaz, A Bayesian Approach to Traffic Light Detection and Mapping, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 125, 2017, p. 184-192.

Y. Balazadegan, S. Hosseinyalamdary, Y. Gao, Visual-LiDAR Odometry Aided by Reduced IMU, ISPRS International Journal of Geo-Information, vol. 5(1), 2016, p. 3-24.

S. Hosseinyalamdary}, Y. Balazadegan, C. Toth, Tracking 3D Moving Objects Based on GPS/IMU Navigation Solution, Laser Scanner Point Cloud and GIS Data, ISPRS International Journal of Geo-Information, 4(3), 2015, p. 1301-1316.

S. Hosseinyalamdary, A. Yilmaz, Motion Vector Field Estimation Using Brightness Constancy Assumption and Epipolar Geometry Constraint , Pecora 19/ ISPRS TC 1 and IAG Commission 4 Symposium, 2014, Denver, CO.

Chen, Y., Zhao, D., Lv, L., & Zhang, Q. (2018). Multi-task learning for dangerous object detection in autonomous driving. Information Sciences Journal, 432, 559–571. https://doi.org/10.1016/j.ins.2017.08.035

Chowdhuri, S., Pankaj, T., & Zipser, K. (2017). MultiNet : Multi-Modal Multi-Task Learning for Autonomous Driving. ARXIV. Retrieved from arxiv.org/abs/1709.05581

Daniel, K., Adam, M., Jarvers, C., Layher, G., Neumann, H., Teutsch, M., & Gmbh, H. O. (2017). Fully Convolutional Region Proposal Networks for Multispectral Person Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 243–250. https://doi.org/10.1109/CVPRW.2017.36

Guan, D., Cao, Y., Yang, J., Cao, Y., & Ying Yang, M. (2018). Fusion of Multispectral Data Through Illumination-aware Deep Neural Networks for Pedestrian Detection. ARXIV. Retrieved from https://arxiv.org/pdf/1802.09972.pdf

Li, C., Song, D., Tong, R., & Tang, M. (2019). Illumination-aware faster R-CNN for robust multispectral pedestrian detection. Pattern Recognition, 85, 161–171. https://doi.org/10.1016/j.patcog.2018.08.005

Milz, S., Arbeiter, G., Witt, C., Abdallah, B., Yogamani, S.: Visual slam for auto-mated driving: Exploring the applications of deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 247–257 (2018

Alex Kendall, Yarin Gal, Roberto Cipolla:
Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. CoRR abs/1705.07115 (2017)

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, Amar Shah:
Learning to Drive in a Day. CoRR abs/1807.00412 (2018)

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. ARXIV. Retrieved from arxiv.org/abs/1706.05098

Siam, M., Mahgoub, H., Zahran, M., Yogamani, S., Jagersand, M., & El-Sallab, A. (2017). Motion and Appearance Based Multi-Task Learning Network for Autonomous Driving, 1–8. Retrieved from arxiv.org/pdf/1709.04821.pdf

Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., & Urtasun, R. (2016). MultiNet : Real-time Joint Semantic Reasoning for Autonomous Driving. ARXIV. Retrieved from arxiv.org/abs/1612.07695

Zhang, T., Ghanem, B., Liu, S., & Ahuja, N. (2012). Robust Visual Tracking via Multi-Task Sparse Learning. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2042–2049. https://doi.org/10.1109/CVPR.2012.6247908

Berger C., Rumpe B. (2012) Autonomous Driving - 5 Years after the Urban Challenge: The Anticipatory Vehicle as a Cyber-Physical System, Proceedings of the 10th Workshop on Automotive Software Engineering (ASE 2012), pp. 789-798.

Asvadi, A., Garrote, L., Premebida, C., Peixoto, P., & J. Nunes, U. (2017). Multimodal vehicle detection: Fusing 3D- LiDAR and color camera data. *Pattern Recognition Letters*, (May 2018). https://doi.org/10.1016/j.patrec.2017.09.038

Behrendt, K., Novak, L., & Botros, R. (2017). A deep learning approach to traffic lights: Detection, tracking, and classification. *IEEE International Conference on Robotics and Automation (ICRA)*, 1370–1377. https://doi.org/10.1109/ICRA.2017.7989163

Chellappa, R., Gang Qian, & Qinfen Zheng. (2004). Vehicle detection and tracking using acoustic and video sensors. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, *3*(4), iii-793-6. https://doi.org/10.1109/ICASSP.2004.1326664

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2016). Multi-View 3D Object Detection Network for Autonomous Driving. https://doi.org/10.1109/CVPR.2017.691

Dalaff, C., Reulke, R., Kroen, A., Kahl, T., Ruhe, M., Schischmanow, A., … Tuchscheerer, W. (n.d.). A Traffic

Object Detection System for Road Traffic. *Image (Rochester, N.Y.)*, 78–83.

Guan, D., Cao, Y., Yang, J., & Yang, M. Y. (2018). Fusion of Multispectral Data Through Illumination-aware Deep Neural Networks for Pedestrian Detection. *ARXIV*.

Jaritz, M., de Charette, R., Toromanoff, M., Perot, E., & Nashashibi, F. (2018). End-to-End Race Driving with Deep Reinforcement Learning, 2070–2075. https://doi.org/arXiv:1807.02371v2

Kim, S., Oh, S., Kang, J., Kim, K., Park, S., & Park, K. (2005). Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2173–2178. https://doi.org/10.1109/IROS.2005.1545321

König, D., Adam, M., Jarvers, C., Layher, G., Neumann, H., & Teutsch, M. (2017). *Fully Convolutional Region Proposal Networks for Multispectral Person Detection, In CVPR 2017.*

Li, C., Song, D., Tong, R., & Tang, M. (2018) *Illumination-aware Faster R-CNN for Robust Multispectral Pedestrian Detection*.

Liu, F., Sparbert, J., & Stiller, C. (2008). 04621161, 168–173.

Liu, X., Sun, Z., & He, H. (2011). On-road vehicle detection fusing radar and vision. *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2011*, 150–154. https://doi.org/10.1109/ICVES.2011.5983805

Ashish Mehta, Adithya Subramanian, Anbumani Subramanian, (2018). Learning End-to-end Autonomous Driving using Guided Auxiliary Supervision, arXiv:1808.10393

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., & Boots, B. (2017). Agile Off-Road Autonomous Driving Using End-to-End Deep Imitation Learning. https://doi.org/10.15607/RSS.2018.XIV.056

Peker, A. U., Tosun, O., Akin, H. L., & Acarman, T. (2014). Fusion of map matching and traffic sign recognition. *2014 IEEE Intelligent Vehicles Symposium Proceedings*, (Iv), 867–872. https://doi.org/10.1109/IVS.2014.6856536

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks ∗ arXiv : 1706 . 05098v1 [cs . LG] 15 Jun 2017, (May).

Selbes, B., & Sert, M. (2017). Multimodal vehicle type classification using convolutional neural network and statistical representations of MFCC. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, (August), 1–6. https://doi.org/10.1109/AVSS.2017.8078514

Yang, Z., Zhang, Y., Yu, J., Cai, J., & Luo, J. (n.d.). End-to-end Multi-Modal Multi-Task Vehicle Control for Self-Driving Cars with Visual Perceptions.

Zhang, T., Ghanem, B., Liu, S., & Ahuja, N. (2012). Robust Visual Tracking via Multi-Task Sparse Learning, 2042–2049.

Zhou, L., & Deng, Z. (2014). LiDAR and Vision-Based Real-Time Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicle. *17th International Conference on Intelligent Transportation Systems (ITSC)*, 578–583. https://doi.org/10.1109/ITSC.2014.6957752

Scaramuzza, D.,Fraundorfer, F., Visual Odometry: Part I - The First 30 Years and Fundamentals, IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.

Fraundorfer, F.,Scaramuzza, D., Visual Odometry: Part II - Matching, Robustness, and Applications, IEEE Robotics and Automation Magazine, Volume 19, issue 2, 2012

S. Wang, R. Clark, H. Wen and N. Trigoni, DeepVO: Towards End−to−End Visual Odometry with Deep Recurrent Convolutional Neural Networks, In International Conference on Robotics and Automation. 2017.

R. Clark, S. Wang, H. Wen, A. Markham and N. Trigoni, INet: Visual Inertial Odometry as a Sequence to Sequence Learning Problem, In AAAI Conference on Artificial Intelligence (AAAI). 2017.

S. Wang, R. Clark, H. Wen and N. Trigoni, DeepVO: Towards End−to−End Visual Odometry with Deep Recurrent Convolutional Neural Networks, In International Conference on Robotics *and Automation*. 2017.

Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi and In So Kweon, Multispectral Pedestrian Detection: Benchmark Dataset and Baseline, CVPR, 2015.

Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe. Learning cross-modal deep representations for robust pedestrian detection. In CVPR , 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 1. MIT Press, Cambridge, MA, USA, 91-99.

Ross Girshick. 2015. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15). IEEE Computer Society, Washington, DC, USA, 1440-1448.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR '14). IEEE Computer Society, Washington, DC, USA, 580-587

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37. Springer, 2016.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.

Lin, Tsung-Yi, Priya Goyal, Ross B. Girshick, Kaiming He and Piotr Dollár. Focal Loss for Dense Object Detection.” 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 2999-3007.

H. Cai, G. Chen, Z. Liu , and Z. Geng , Fusion algorithm of infrared and visible images based on joint bilateral filter, 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017.

John, V., Yoneda, K., Qi, B., Liu, Z., & Mita, S. (2014). Traffic light recognition in varying illumination using deep learning and saliency map. In Proceedings intelligent transportation systems conference (pp. 2286–2291), Qingdao, China.

Z. Meng, X. Fan, X. Chen, M. Chen, and Y. Tong, Detecting small signs from large images,CoRR, vol.

abs/1706.08574, Jun. 2017. [Online]. Available: http://arxiv.org/abs/1706.08574 Accessed on: May 12, 2018.

M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. J agersand, and A. E. Sallab. Modnet: Moving object detection network with motion and appearance for autonomous driving. CoRR , abs/1709.04821, 2017.