Multi-task learning for dangerous object detection in autonomous driving

Yaran Chen[a,b], Dongbin Zhao[a,b,*], Lv Le[a,b], Qichao Zhang[a,b]

[a] *The state Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[b] *the University of Chinese Academy of Sciences, China*

Abstract

Recently, autonomous driving has been extensively studied and has shown considerable promise. Vision-based dangerous object detection is a crucial technology of autonomous driving. In previous work, dangerous object detection is generally formulated as a typical object detection problem and a distance-based danger assessment problem, separately. These two problems are usually dealt with using two independent models. In fact, vision-based object detection and distance prediction present prominent visual relationship. The objects with different distance to the camera have different attributes (pose, size and definition), which are very worthy to be exploited for dangerous object detection. However, these characteristics are usually ignored in previous work. In this paper, we propose a novel multi-task learning (MTL) method to jointly model object detection and distance prediction with a Cartesian product-based multi-task combination strategy. Furthermore, we mathematically prove that the proposed Cartesian product-based combination strategy is more optimal than the linear multi-task combination strategy that is usually used in MTL models, when the multi-task itself is not independent. Systematic experiments show that the proposed approach consistently achieves better object detection and distance prediction performances compared to both the single-task and

[☆] Fully documented templates are available in the elsarticle package on CTAN.

[*] Corresponding author
Email address: dongbin.zhao@ia.ac.cn (Dongbin Zhao)
URL: www.elsevier.com (Dongbin Zhao)

multi-task dangerous object detection methods.

Keywords: Dangerous object detection, Autonomous driving, Multi-task learning, Convolutional neural network

1. Introduction

In the real-world transportation system, the host car is usually surrounded by lots of moving vehicles and pedestrians. They obstruct free driving of the host and even cause potential dangers of collision. Accurately and promptly detecting dangerous objects is extremely important for preventing traffic accidents in autonomous driving. It has been widely studied by many researchers recently.

Dangerous object detection aims to identify the potentially dangerous objects for drivers. Major dangerous objects include vehicles and pedestrians within the vehicle safety distance, which may cause a collision with the host vehicle. Based on different input signals, the common dangerous object detection methods are classified into two types: general sensor-based and vision-based dangerous object detection methods. Sensor-based dangerous object detection systems generally use a variety of sensors, such as radars [30], lasers and sonars [39], to detect surrounding obstacles. Owing to the great environmental perception capabilities of sensors, the sensor-based dangerous object detection systems achieve excellent performance and have been widely applied in autonomous driving [29, 15, 12]. For example, Google Car and Baidu Car use a rotating light detection, ranging (LIDAR) scanners [4] and several radars to obtain the surrounding environment information [9, 26]. Sheu *et al* [32] uses smart antennas to collect surrounding information and build a distance awareness system for warning drivers dangerous objects. However, laser and radar sensors are too expensive to realize large-scale applications, and they have a limited capacity to recognize object categories. Therefore, visual information is essential for practical autonomous driving systems. Inspired by human visual perception, vision-based dangerous object detection system uses images captured by an on-

board camera to directly detect dangerous objects [40]. In contrast to laser and radar sensors, cameras are not only cheap but also able to capture more traffic information including object categories, object distance, traffic signs and signals [6][8]. At present, vision-based dangerous object detection is drawing more and more attentions and has shown considerable promise in practicability. In this paper, we focus on vision-based dangerous object detection, especially on the detection and distance prediction of vehicles and pedestrians.

In previous work, vision-based dangerous object detection is usually formulated as a typical object detection problem and a distance-based danger assessment problem [41, 31, 22, 33]. They are separately dealt with using two independent models. The distance-based danger assessment problem is solved by some distance measurement sensors such as RGB-D cameras [41, 31], LIDARs [3] and radars [8]. The object detection problem is commonly solved by machine learning methods, among which a typical method is usually composed of proposing regions, extracting features of the proposed regions, and object recognition [11][38][45][7][23]. [21] and [28] propose a kind of end-to-end object detection methods to improve detection speed. It integrates the proposing regions, extracting features and recognizing objects into a model to directly detect objects. These methods, whether end-to-end or traditional methods, can be naturally used for detecting objects in autonomous driving, which have been verified on the actual autonomous driving dataset, such as KITTI [10].

In fact, vision-based object detection and distance prediction present prominent visual relationship due to the sight distance. For instance, small detected objects are generally distributed in the far field of view as shown in Fig. 1 (b), while the object closer to the camera usually occupies more visual fields, which means that it covers more pixels of an image as Fig. 1 (a) shows. In addition, the objects in an image show different poses due to the camera angle as Fig. 1 (c) and (d) show. Therefore, the object distance to the camera will affect its attributes (pose, size and definition). Obviously, these object attributes are quite valuable to recognize object. However, they are much ignored in previous work that separately deals with object detection and distance prediction tasks.

Therefore, simultaneously modeling the object detection and distance prediction tasks in one model will probably improve the performance of dangerous object detection.

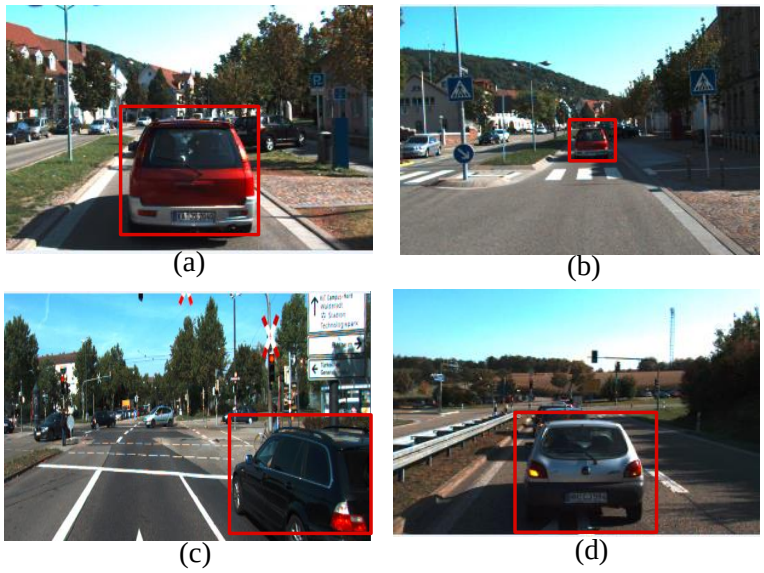

Figure 1: The cars with different distances and poses.

Multi-task learning (MTL) is one of the most well-known techniques for simultaneously dealing with multiple tasks, which can exploit the shared information of multiple related tasks and improve the performance of each other [1]. This statement has been proven both empirically and theoretically [2, 25]. For computer vision, MTL techniques have many important and realistic applications, such as pose estimation [42], action recognition [46], face detection [43], facial landmark localization [44], and achieve great successes. A typical MTL method linearly combines the objectives of multiple tasks to jointly model them. Although the linear combination way can exploit the shared information of the related tasks from the input data, it much ignores the correlations of multi-task itself.

In this paper, we propose a convolutional neural network (CNN)-based MTL method to jointly model object detection and distance prediction. In order to

facilitate the distance prediction task, we transform the distance regression problem into a classification problem by discretizing continuous distance. We further propose a joint optimization objective for the proposed model according to the Cartesian product of object categories and distance classes. A CNN similar to single shot multiBox detector (SSD) is used to optimize the proposed objective that simultaneously takes both object detection and distance classification into consideration. In addition, we mathematically prove that the proposed Cartesian product-based multi-task combination outperforms the linear multi-task combination when multi-task itself is not independent.

The main contributions of the paper are as following:

- First, we present the MTL mechanism for dangerous object detection, which can capture the visual relationship between object detection and distance prediction. To the best of our knowledge, it is the first attempt to use a MTL model to simultaneously deal with object detection and distance prediction tasks. Systematic experiments on the KITTI dataset show that the proposed MTL model achieves the improved performance of 2.27% over the representative single shot Multi-box detection (SSD) [21], which is the state-of-the-art model for fast object detections.

- We propose a novel Cartesian product-based multi-task combination strategy (CP-MTL) for MTL to jointly model the object detection and distance prediction tasks. The proposed CP-MTL further improves the detection performance (3.01% improvement over SSD).

- We mathematically prove that the Cartesian product-based multi-task combination strategy outperforms the linear multi-task combination that is usually used in MTL models. The linear multi-task combination strategy is a special example of the Cartesian product-based multi-task combination strategy under the condition that multiple tasks are independent. The Cartesian product-based multi-task combination strategy considers the dependent of multiple tasks.

The paper is organized as follows. Section 2 gives a brief review of related studies. In Section 3, we give the problem formulation and introduce two objective combination strategies of multi-task learning. Section 4 describes the structure of the proposed model. In Section 5, systematic experiments are compared and discussed. Finally, a conclusion and acknowledgments are given.

2. Related work

In autonomous driving, vision-based dangerous object detection is usually formulated as a typical object detection problem and a distance-based danger assessment problem. Multi-sensor fusion is a common method, by fusion the distance information measured by lidar sensors and visual information captured by cameras. Fusion multimodal data from different sensors is a challenge [16]. Liu *et al.* studies the fusion of tactile sense and visual sense for object recognition[20][18][19].

For dangerous object detection, the distance-based danger assessment problem can be solved by distance prediction methods. There are two kinds of approaches to predict distance: direct methods [37][35][27] and indirect methods. The direct method uses sensors to achieve distances of objects directly. For example, [37] uses lidars to directly achieve distance, stereo vision sensors to detect objects, then fuses the two kind of multimodal data form lidars and cameras. The directly achieved distance methods are simple but add cost. The indirect method predicts distance using vision-based methods. For example, Mobileye [36] uses geometric relationship of the road and vehicles in images to predict distances of vehicles. It works well for objects at front and rear, but does not work well for the cluttered road situations such as high-curvature roads [14]. [17] uses CNN and continuous conditional random field to predict depth of a single image along with large computing. In this paper, we study multi-task learning (MTL) for the vision-based distance predict method, which aims to improve the precision and increase the predict speed. MTL aims to learn related problems together by sharing information.

In computer vision community, many researchers have attempted to adopt multi-task learning methods. [42] proposes a multi-task model for face recognition, [44] also uses the multi-task learning for facial land mark detection, and [46] recognizes action by sharing information among multiple related tasks. In this paper, we propose a novel multi-task model by a Cartesian product-based multi-target combination strategy. The proposed model differs in that it can consider the dependence among related tasks.

3. Multi-task Learning

3.1. Problem Formulation

Dangerous object detection in autonomous driving consists of object detection and distance prediction. Object detection is usually expressed as a classification task, namely we can detect objects by classifying the proposed regions. In vision-based dangerous object detection, the distance prediction can be expressed as a regression problem. Due to the non-linear variation of the sight distance, it is very difficult to accurately predict continuous distance. In this paper, we transform the distance prediction problem into a classification problem, through discretizing continuous distance variables. The object detection problem and distance prediction problem are respectively denoted as c and d. For an image, the goal of dangerous object detection is to minimize the loss of all tasks, including c and d. MTL is one of the most popular techniques for dealing with related multiple tasks. MTL can exploit the shared information of the input image to jointly optimize multiple tasks, improving the performance of each other. In this paper, we propose a novel multi-task learning method to jointly model object detection and distance classification.

3.2. Linear multi-task combination

The common MTL methods generally use a linear multi-task combination (LC-MTL) strategy, which is a weighted linear combination of the multiple objective functions, to jointly model the multiple tasks [44], as follows:

$$L_{c+d} = \alpha \cdot L_c + (1 - \alpha) \cdot L_d, \tag{1}$$

where L_c and L_d are the objective functions of task c and d, respectively. And α specifies the relative importance of each task which can be experimentally chosen.

Convolutional Neural Network (CNN) is a widely-embraced representation learning technique, which has a powerful ability to exploit the shared feature representation of multiple tasks. At present, CNN has been widely used in multi-task learning and achieved satisfied performance. For the object recognition c and object distance classification d, CNN can be used to jointly model them, as shown in Fig. 2 (a). Given an input image $\mathbf{x} \in \Re_{+}^{\mathrm{m}\times\mathrm{n}}$, through the shared model parameters, CNN can simultaneously compute the probabilities of object recognition and distance classification.

We define y_c as a category of an object and $y_c \in \{c_1, c_2, ..., c_p\}_{1\times p}$, where p is the number of object categories. Then the probability $p(y_c = c_i|\mathbf{x})$ of the image $\mathbf{x}$ belonging to the i-th category can be calculated by a softmax function:

$$p(y_c = c_i|\mathbf{x}) = \mathrm{softmax}(\mathbf{z}_\mathrm{c}) = \frac{\exp({z^i}_c)}{\sum_{j=1}^{p} \exp({z^j}_c)}, \tag{2}$$

where $\mathbf{z}_\mathrm{c}$ is the output of the last fully connected layer in CNN for the task c and its j-th element is denoted as ${z^j}_c$.

$$\mathbf{z}_\mathrm{c} = \mathrm{cnnfw}(\mathbf{x}), \qquad \mathbf{z}_\mathrm{c} \in \Re^{p\times 1}$$

where cnnfw denotes multiple convolution operations. For the classification with multiple categories, a typical objective function is the cross entropy loss:

$$L_c = y_c \cdot log(p(y_c = c_i|\mathbf{x})). \tag{3}$$

In the same way, we denote $y_d \in \{d_1, d_2, ..., d_p\}_{1\times q}$ as a category of an object distance, where q is the number of object distance categories. The probability of the image $\mathbf{x}$ belonging to the j-th category is $p(y_d = d_j|\mathbf{x}) = \mathrm{softmax}(\mathbf{z}_\mathrm{d})$. And its objective function is formulated as:

$$L_d = y_d \cdot log(p(y_d = d_j|\mathbf{x})). \tag{4}$$

In dangerous object detection, the object detection and the distance prediction tasks are all very important. We set they have the same importance: $\alpha = 0.5$. Then we rewrite Eq. (1) as:

$$L_{c+d} = 0.5(L_c + L_d), \tag{5}$$

where the coefficient 0.5 doesn't affect the optimal solution of the loss function L_{c+d}. So we can ignore it and get $L_{c+d} = L_c + L_d$. The equation can be rewritten as the following by combining Eq. (3) and Eq. (4).

$$L_{c+d} = y_c \cdot log(p(y_c|\mathbf{x})) + y_d \cdot log(p(y_d|\mathbf{x})) \tag{6}$$

With the linear multi-task combination, CNN has the capability of exploiting the shared information from the input images for the related tasks. However, it totally ignores the dependence between the multiple targets.

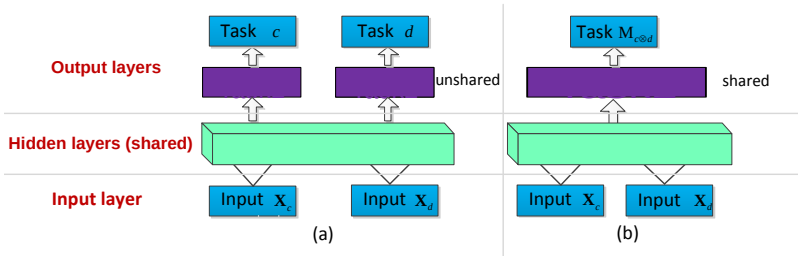

Figure 2: The structure of the traditional LC-MTL and the proposed CP-MTL. (a) is the traditional LC-MTL structure based on linear multi-task combination. (b) is the structure of the proposed CP-MTL with Cartesian product-based multi-task combination.

3.3. Cartesian product-based multi-task combination

To consider the dependence between the targets, we use a Cartesian product-based multi-task combination strategy (CP-MTL) to jointly model object detection and distance prediction of the proposed regions, shown as Fig. 2 (b). The combined task is denoted as $M = c \otimes d$, where $\otimes$ denotes the Cartesian product operator. Concretely, we define $y_{c\otimes d} = y_c \otimes y_d$ as a category of the combined task and $y_{c\otimes d} \in \{c_1d_1, c_1d_2, ...c_1d_q, \ldots, c_id_j, \ldots, c_pd_q\}_{1\times pq}$, where pq is the number

of the combined task category. Through the Cartesian product-based multi-target combination strategy, we can translate two classification tasks with q categories and p categories into one classification with pq categories.

For a given input image $\mathbf{x} \in \Re_{+}^{\mathrm{m}\times\mathrm{n}}$, CNN computes the probability of the image $\mathbf{x}$ belonging to $c_i d_j$-th category of the combined task:

$$p(y_{c\otimes d} = c_i d_j|\mathbf{x}) = \mathrm{softmax}(\mathbf{z}_{c\otimes d}) = \frac{\exp(z^{ij}_{c\otimes d})}{\sum_{k=1}^{pq}\exp(z^{k}_{c\otimes d})}. \tag{7}$$

The loss function of the combined task is formulated as:

$$L_{c\otimes d} = y_{c\otimes d} \cdot log(p(y_{c\otimes d} = c_i d_j|\mathbf{x})). \tag{8}$$

Then the loss function can be rewritten in a matrix form:

$$L_{c\otimes d} = [c_1 d_1, c_1 d_2, \ldots, c_p d_q] log(\mathbf{P}(y_{c\otimes d}))), \tag{9}$$

where $\mathbf{P}(y_{c\otimes d})$ is the probability matrix of categories:

$$\mathbf{P}(y_{c\otimes d}) = \begin{bmatrix} p(y_{c\otimes d} = c_1 d_1|\mathbf{x}) \\ p(y_{c\otimes d} = c_1 d_2|\mathbf{x}) \\ \vdots \\ p(y_{c\otimes d} = c_p d_q|\mathbf{x}) \end{bmatrix}. \tag{10}$$

Eq. (9) is a matrix multiplication, the first term is a matrix with size $1 \times pq$ and the second term is a matrix with size $pq \times 1$. pq entries of the first matrix are multiplied with the corresponding pq entries of the second matrix, and summed to produce an entry, as follows:

$$\begin{aligned} L_{c\otimes d} = c_1 d_1 \cdot log(p(y_{c\otimes d} = c_1 d_1|\mathbf{x})) + c_1 d_2 \cdot log(p(y_{c\otimes d} = c_1 d_2|\mathbf{x})) + \\ \cdots + c_p d_q \cdot log(p(y_{c\otimes d} = c_p d_q|\mathbf{x})). \end{aligned} \tag{11}$$

Eq. (11) is the sum of pq entries, and each entry contains a probability $p(y_{c\otimes d} = c_i d_j|\mathbf{x})$, which means the input image $\mathbf{x}$ is belong to c_i of task c and d_j of task d. If the task c and d are completely independent, we can get:

$$p(y_{c\otimes d} = c_i d_j|\mathbf{x}) = p(y_c = c_i|\mathbf{x}) \cdot p(y_d = d_j|\mathbf{x}). \tag{12}$$

Then bring Eq. (12) into Eq. (11) to derive:

$$\begin{aligned} L_{c\otimes d} = & c_1 d_1 \cdot log(p(y_c = c_1|\mathbf{x}) \cdot p(y_d = d_1|\mathbf{x})) \\ & + c_1 d_2 \cdot log(p(y_c = c_1|\mathbf{x}) \cdot p(y_d = d_2|\mathbf{x})) \\ & + \cdots + c_1 d_q \cdot log(p(y_c = c_1|\mathbf{x}) \cdot p(y_d = d_q|\mathbf{x})) \\ & + \cdots + c_p d_q \cdot log(p(y_c = c_p|\mathbf{x}) \cdot p(y_d = d_q|\mathbf{x})) \end{aligned} . \tag{13}$$

Each entry of Eq. (13) contains a $log(p(y_c = c_i|\mathbf{x}) \cdot p(y_d = d_j|\mathbf{x}))$, which can be rewritten as $log(p(y_c = c_i|\mathbf{x}) + log(p(y_d = d_j|\mathbf{x}))$. Then the loss function of the combined task can be rewritten as:

$$\begin{aligned} L_{c\otimes d} = & c_1 d_1 \cdot log(p(y_c = c_1|\mathbf{x}) + c_1 d_1 \cdot log(p(y_d = d_1|\mathbf{x})) \\ & + c_1 d_2 \cdot log(p(y_{\mathrm{c}} = c_1|\mathbf{x}) + c_1 d_2 \cdot log(p(y_d = d_2|\mathbf{x})) \\ & + \cdots + c_1 d_q \cdot log(p(y_{\mathrm{c}} = c_1|\mathbf{x}) + c_1 d_q \cdot log(p(y_{\mathrm{d}} = d_q|\mathbf{x})) \\ & + \cdots + c_p d_q \cdot log(p(y_{\mathrm{c}} = c_p|\mathbf{x}) + c_p d_q \cdot log(p(y_{\mathrm{d}} = d_q|\mathbf{x})) \end{aligned} . \tag{14}$$

We find like terms which contain the same variable $log(p(y))$. For example, there are p terms containing $log(p(y_{\mathrm{c}} = c_i|\mathbf{x})$ and the coefficients of these like terms are $c_i d_1, c_i d_2, \cdots, c_i d_p$. The sum of these coefficients is equal to c_i. We find all the like terms, and sum these coefficients of the like terms, shown as:

$$\begin{aligned} L_{c\otimes d} = & (c_1 d_1 + c_1 d_2 + \cdots + c_1 d_q) \cdot log(p(y_c = c_1|\mathbf{x})) \\ & + (c_2 d_1 + c_2 d_2 + \cdots + c_2 d_q) \cdot log(p(y_c = c_2|\mathbf{x})) + \\ & \qquad \vdots \\ & + (c_p d_1 + c_p d_2 + \cdots + c_p d_q) \cdot log(p(y_c = c_p|\mathbf{x})) \\ & + (c_1 d_1 + c_2 d_1 + \cdots + c_p d_1) \cdot log(p(y_d = d_1|\mathbf{x})) \\ & + (c_1 d_2 + c_2 d_2 + \cdots + c_p d_2) \cdot log(p(y_d = d_2|\mathbf{x})) \\ & \qquad \vdots \\ & + (c_1 d_q + c_2 d_q + \cdots + c_p d_q) \cdot log(p(y_d = d_q|\mathbf{x})) \end{aligned} . \tag{15}$$

$$\begin{aligned} = & c_1 \cdot log(p(y_c = c_1|\mathbf{x})) + c_2 \cdot log(p(y_c = c_2|x_i)) + \cdots \\ & + c_p \cdot log(p(y_c = c_p|\mathbf{x})) + \cdots + d_q \cdot log(p(y_d = d_q|\mathbf{x})) \end{aligned}$$

Then we find that the sum of the first p terms is the objective function of task c and the remaining q terms constitute the objective function of task d. Therefore,

we can obtain the loss function of the combined task:

$$L_{c\otimes d} = L_c + L_d = L_{c+d}. \tag{16}$$

Through formula derivation, we prove that the linear multi-task combination is equivalent to the Cartesian product-based multi-task combination if the two tasks are independent. Otherwise, the proposed Cartesian product-based multi-task combination outperforms the linear multi-task combination since the proposed method takes the dependency between multiple tasks into account. For dangerous object detection, the object detection task and object distance classification task are probably not independent, which may be more suitable for being modeled by the proposed model.

4. CP-MTL SSD Method

Dangerous object detection contains object detection and assessing of danger with the object distance. Owing to CNN having the strong capability of learning feature representation, CNN-based object detection methods have achieved satisfied performance. Among these methods, single shot multiBox detection (SSD) is one of the state-of-the-art object detection methods. It directly predicts object bounding boxes and classifies object categories, avoiding complex object detection pipeline. Compared with other methods, SSD achieves faster detection speed and higher detection accurate. In this paper, we furthermore improve SSD by incorporating the proposed CP-MTL (Cartesian product-based combination multi-target) into the optimization objective. The CP-MTL SSD is capable of simultaneously dealing with the object detection and distance classification tasks.

4.1. Model architecture

Fig. 3 gives an overview of the proposed CP-MTL SSD model. It is composed of multiple hierarchical convolutional layers, a number of default bounding boxes with different sizes and aspect ratios, and a lot of detections. By

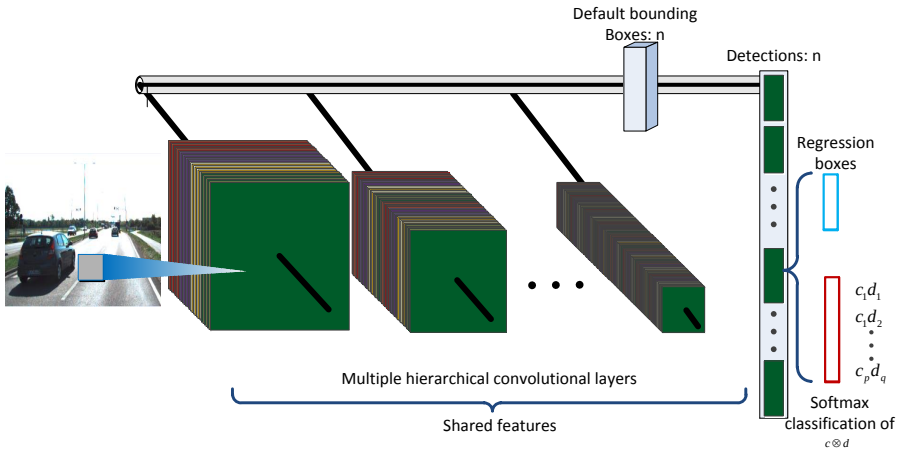

Figure 3: The architecture of CP-MTL SSD.

convolution operations, the hierarchical convolution layers can produce a lot of feature maps of different scales and resolutions for an input image. Each feature map has some default bounding boxes. For a default bounding box, the model regresses the bounding box and classifies object categories simultaneously, through a detection which contains a full-connected classification layer and regression layer.

Multiple hierarchical convolutional layers: Owing to the ability of CNN in transfer learning, especially the transfer of learning feature, the early convolutional layers usually inherit the trained CNN model in a large scale image dataset, such as VGG [34]. That means the parameters of the early convolutional layers are initialized with those of a trained high-performance CNN in advance. The following convolutional feature layers gradually reduce their size for producing different scale feature maps. For an input image, the multiple hierarchical convolutional layers can produce a lot of feature maps with different scales. A detection region of a feature map is responsive to a specific area of the input image. Different detection regions in multi-scale feature maps are corresponding to different scale areas of the input image. Combining all the detections in the multi-scale feature maps can cover various object sizes of input

images.

Default bounding boxes: Considering the different shapes of objects, there are a set of default bounding boxes with different aspect ratios in each position of a feature map. For a feature map with the size of $w \times h$, if there are k default bounding boxes in each position, the feature map has $w \times h \times k$ default bounding boxes in all. A default bounding box with a specific aspect ratio can be responsive to the area with the same specific aspect ratio of the input image. All the detections for $w \times h \times k$ default bounding boxes can cover various shape objects of input images.

Detections: In the model, each default bounding box is followed by a detection, which consists of a full-connected classification layer and a regression layer. The classification layer computes probabilities of object categories, namely scores. While the regression layer predicts offsets between default bounding boxes and ground truth boxes, simultaneously. The full-connected classification layer and the regression layer share input representations (regions of feature maps surrounded by default boxes). Due to the larger number of default bounding boxes, the model can produce many detection boxes. Through non-maximum suppression [24], the model will predict the final boxes.

CP-MTL SSD is a variant of SSD. Although they seem to be similar, they have an essential difference. The key difference is that CP-MTL optimizes the combination targets of object recognition and object distance classification based on the Cartesian product, while SSD just only optimizes the target of object recognition.

4.2. Cartesian product-based combination targets

In order to optimize object detection and object distance prediction simultaneously, we propose a Cartesian product-based combination of object detection task and distance classification task. In autonomous driving, we focus on the detection of vehicles and pedestrians. Based on the sizes and shapes of objects, we classify objects into three categories: cars, vans and pedestrians, denoted as $\{c_1, c_2, c_3\}$. Due to the relationship between the distance and the object at-

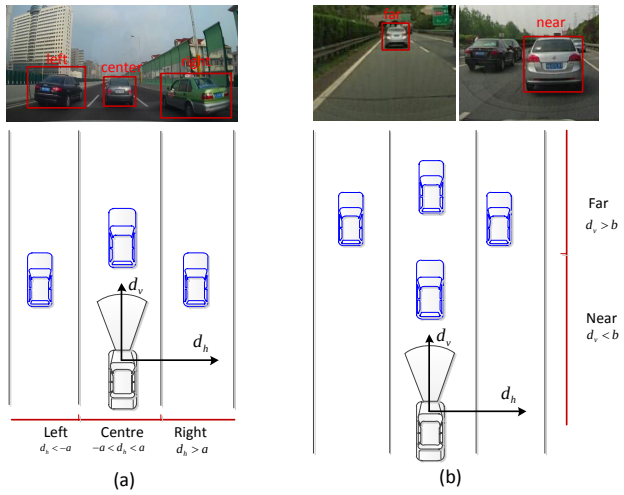

Figure 4: The change of attributes (size, definition and pose) with the distance. (a) shows the change with the horizontal distance, and (b) shows the change with the vertical distance. We build the coordinate system relative to the host vehicle. The origin is the camera on the host, the horizontal axis shows the horizontal distance to the host vehicle d_h, and the vertical axis stands the vertical distance d_v. And we divide the space into three regions according to d_h coordinate and also divide the space into 2 regions according to d_v coordinate: the center region ($-a < d_h < a$ meters) where vehicles locate in the front of the host), the left region ($d_h < -a$ meters) where vehicles locate in the left of the host and show the right profiles; the right region ($a < d_h$ meters) where vehicles locate in the right of the host and show the left profiles, the far region ($d_v > b$ meters), where the vehicle is far from the host and not clear with a small size, and the near region ($d_v < b$ meters), where the one is close to the host and clear with a big size.

tributes (pose, size and definition), we consider the distance category task from two dimensions: the vertical distance and the horizontal distance.

In driving, vehicles in the same lane or the adjacent lane with the host car have the opportunity to become potential dangerous objects. So the vertical distance d_v from the host car is the main factor used to assess the danger of objects. At the same time, the vertical distance can affect the sizes and definitions of vehicles and the horizontal distance d_h can affect the poses of vehicles, as shown in Fig. 4. The variety of object attributes (size, definition

and pose) increases the difficulty of object recognition. So it benefits object recognition to divide a category of objects into different categories according to their sizes, definitions and poses.

In this paper, we discuss the distance classification task from two dimensions. We part the image into four regions according to the vertical distance and part the space into three regions according to the horizontal distance shown in Fig. 5 (a). Due to the symmetry of vehicles, the vehicles in the left region are similar to the right ones. So the space can be classified into 8 categories denoted as $\{d_1, d_2, \cdots, d_8\}$. And the red one denotes the shortest vertical distance and the most dangerous category, followed by the yellow one, the green one, and the blue one.

Through the Cartesian product-based combination targets, we transform c with 3 categories and d with 8 categories into one task $M_{c\otimes d}$, which has 24 categories shown in Fig. 5 (b). Fig. 5 (b) is a two dimensional plane mapped from (a). In Fig. 5 (b), each divided region is a distance category, and contains all the categories c_1, c_2, c_3 of object recognition c. Recognizing the objects during a given distance category is much easier than recognizing them at all the range of distance. So the new combined task $M_{c\otimes d}$ means to classify the two difficult classification tasks into a simple classification task.

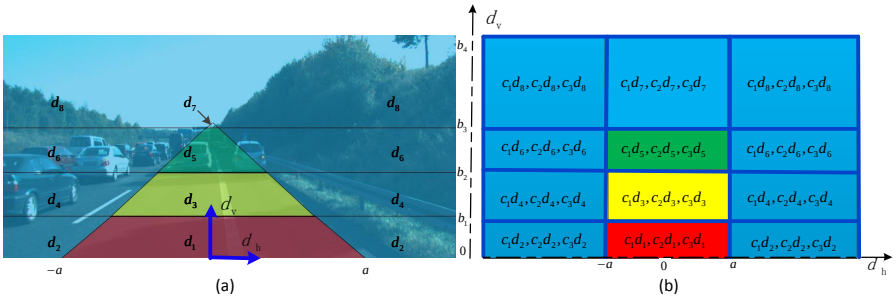

Figure 5: The object categories of the dangerous object detection task based on the Cartesian product combination . (a) the image geographic division according to the distance and the visual angle. (b) the categories of the Cartesian product-based combination target, where (a) is mapped to the two dimensional plane(b).

4.3. Optimization objectives

The objective of the combined classification task based on the Cartesian product is shown in Eq. (8). We extend the objective to adapt traditional objective of SSD. The loss function of the CP-MTL SSD is a sum of the object localization loss L_{loc} and the combined classification loss L_{conf}:

$$L = L_{loc} + L_{conf} = L_{loc} + L_{c \otimes d}, \tag{17}$$

where the L_{loc} is the loss function of predicting bounding boxes. L_{conf} is equal to $L_{c \otimes d}$, which is the objective combination of the object recognition and distance classification based on the Cartesian product. The overall objective function contains the objective function of c and d. Therefore CP-MTL can learning object recognition and distance prediction through optimizing the overall objective function L.

5. Experiments

In this section, we comprehensively evaluate the proposed CP-MTL method on dangerous object detection task in autonomous driving. First, we systematically analyze and compare the performances of the multi-task learning and single task learning methods on dangerous object detection. In addition, we validate the effectiveness of the proposed Cartesian product-based multi-task combination strategy for MTL through comparison with the linear multi-task combination strategy.

5.1. Dataset

For an objective evaluation, we use a publicly available dataset, KITTI [10]. It contains more than 40,000 real transportation images captured by a car driving in European cities. In these images, there are about 16,000 images containing categories and position information of objects, which can be used for dangerous object detection. Objects categories cover 8 daily common objects, including cars, vans, trucks, pedestrians, person_sittings, trams, cyclists and misc. But

for dangerous object detection in autonomous driving, we only focus on the detections of cars, vans and pedestrians, and regard other objects in images as background. Among these objects, there are 56,028 cars, 15,957 pedestrians, and 6,214 vans, but the objects with the size smaller than 25×25 pixels will be ignored in our experiments. Obviously, there is a serious imbalance between these object categories, which increases the difficulty of the detection.

In experiments, we randomly divide the 16,000 images into 3 parts: training set, testing set and validating set. Among them, the training set contains 12,000 images, the testing sets contains 3000 images and the validating set contains 1000 images. All experimental configures are experimentally chosen according to the performances on the validating set.

5.2. Evaluation Metrics

We take the average precision (AP) of object detection as the evaluation metric [13]. AP measures the comprehensive performance, including the recall rate and precision rate of object detection. It is calculated by the area under the precision-recall curve, and the higher value means the better object detection performance. In addition, we calculate the mean average precision (mAP) to evaluate the overall performances of different methods. The mAP is the mean value of the APs of different object categories.

5.3. Experimental Setup

In this study, we take SSD as the baseline model. At present, SSD is one of the state-of-the-art models for object detection. It is composed of 18 convolutional layers and 5 detectors. The early 13 convolutional layers are initialized using the Oxford VGG network [34] trained on a large-scale image dataset, and the following 5 convolutional layers are randomly initialized. The 5 detectors perform detections on the feature maps from the 10-th, 15-th, 16-th, 17-th, and 18-th convolutional layers, respectively. And five bounding boxes with different aspect ratios ($\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$) are detected at each position of the feature maps. Due to the detection with the multiple shapes, resolutions and scales, detectors

can deal with various objects with different shapes and sizes. The proposed models, whether the CP-MTL or the LC-MTL, are variants of SSD. They have the same network architectures and configures with SSD. But the key difference is the output target of the detector. In addition, for the object detection, the proposed CP-MTL and the LC-MTL also take the object distance prediction into account. Specifically, the CP-MTL uses the Cartesian product-based multi-task combination of object detection and distance prediction, and the LC-MTL uses the linear multi-task combination.

For the distance division, first, we divide the whole image into 12 regions as shown in Fig. 5. Then, we give a distance label d_q to each object in each region according to the horizontal distance and the vertical distance to the camera. In experiments, we add 8 distance labels (denoted as $\{d_1, d_2, \cdots, d_8\}$) to each object category, but for pedestrians we merge the horizontal divisions since pedestrians have no distinct differences of poses in regions with different horizontal distances. Therefore, there are only 4 distance labels for pedestrians according to the vertical distance. Finally, we combine the object categories and the distance labels with the Cartesian product, and obtain the Cartesian product-based multi-task for the proposed CP-MTL model.

For each model, we use the mini-batch gradient descent of 32 samples to optimize the networks, and the maximum epoch is set to 100. A fixed learning rate of 0.001 is used for the first 50 epochs, after which the learning rate decreases with a scale factor of 0.9. In order to speed up the training, the computation of network is paralleled on a graphics processing unit.

5.4. Comparison with different divisions

This scenario aims to evaluate how the distance division affects the performance of the proposed CP-MTL. In experiments, we divide the whole image into 12 regions as shown in Fig. 5 which contains 4 hyper-parameters (a, b_1, b_2, and b_3). a is related to the width of the center region and a larger a means a wider center region which affects poses of vehicles. b_1, b_2 and b_3 specify the division according to the vertical distance which affect sizes and definitions of

vehicles. Therefore, a set of appropriate hyper-parameters (a, b_1, b_2, and b_3) in a distance division is very important to the dangerous object detection. In this paper, we choose two strategies of the distance division: a division based on hand-designed and a division based on data analysis.

In the hand-designed division, we set $a = 2$m, $b_1 = 10$m, $b_2 = 20$m, and $b_3 = 40$m by observing. Table 1 shows the detection results of each object category in each divided region using the first strategy: hand-designed based division. We find that the proposed CP-MTL has achieved satisfactory performances in most regions, especially in the regions near the camera. However, in the regions far away from the camera, such as d_7 and d_8, the performances of object detection are severely degraded, especially for pedestrians. On one hand, objects far away from the camera cover a few pixels and have a low resolution. It is very difficult to detect small objects. On the other hand, there are only a few objects in the regions far away from the camera. It is very difficult to train a model using insufficient data. Moreover, the imbalance of data also increases the difficulty of object detection.

Table 1: The performances of each object category in each divided regions with the proposed CP-MTL by using hand-designed based division and data based division.

Strategy	Categories	d_1	d_3	d_2	d_4	d_5	d_7	d_6	d_8
Hand-designed division	Cars	0.900	0.901	0.890	0.891	0.893	0.876	0.711	0.721
	Vans	0.878	0.812	0.772	0.885	0.858	0.899	0.766	0.784
	Pedestrians	0.790		0.703		0.491		0.20	
Data based division	Cars	0.901	0.900	0.895	0.893	0.872		0.794	
	Vans	0.883		0.900		0.768		0.865	
	Pedestrians	0.793		0.622					

In order to alleviate the problem of unbalanced and insufficient data for further improving the performance of CP-TML, we propose the second strategy:

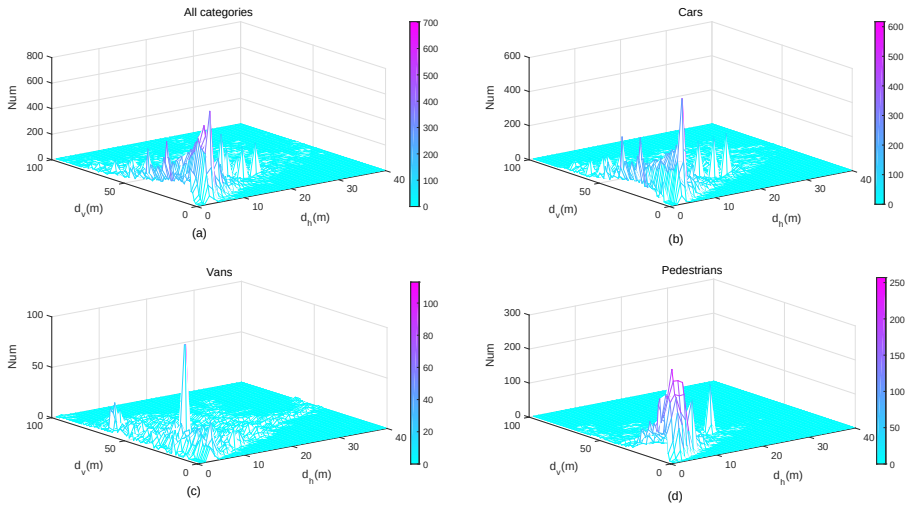

Figure 6: The distribution of objects. The uneven data distributions show objects are mainly located in the regions close to the host car.

data-based division. We re-divide the input images according to the spatial distribution of objects. Fig. 6. (b), (c) and (d) show the spatial distributions of cars, vans and pedestrians, respectively, and Fig. 6. (a) shows the spatial distributions of all objects. In Fig. 6, the axis d_h indicates the horizontal distance to the camera, the axis d_v indicates the vertical distance to the camera, and the Num indicates to the number of objects in the corresponding region. We observe that most of objects are concentrated in the regions within 40 meters from the camera, and there is a sparse distribution of objects in the regions far away from the camera. An over-division to the far regions would lead to the unbalanced and insufficient object data, which probably increases the difficulty of object detection. Therefore, we merge some far regions for obtaining a relatively uniform distribution of data. Specifically, for car detection, we merge the regions d_5 and d_7, and merge the regions d_6 and d_8. After the merger, we reduce the distance labels of cars to 6. For van detection, we merge the regions d_1 and d_3, d_2 and d_4, d_5 and d_7 as well as d_6 and d_8, respectively. Finally, we obtain 4 distance labels for vans. For pedestrian detection, the regions d_1 and d_3 are merged, and the regions d_2, d_4, d_5, d_7, d_6 and d_8 are merged, which finally produce 2 merged regions. So there are 2 distance labels for pedestrians.

Under the scheme of division, the number of object categories to be detected by the CP-MTL becomes 12 from the previous 20.

Table 1 also shows the detection results of each object category in each re-divided region by using the data based division. Compared with the hand-designed based division strategy, the performances of object detection for the merged regions are significantly improved with data based division. Table 2 shows the performances of each object category in the whole image by using the hand-designed based division and the data based division. The data-based division outperforms the hand-designed division. It mainly owes to the more objects and more uniform distribution of data in the merged regions. This suggests that the proposed CP-MTL needs more data to be trained and more data would further improve its performance.

Table 2: The detection performances of each object category in the whole image with the proposed CP-MTL by using hand-designed based division and data based division.

Strategy	mAP	AP(Cars)	AP(Pedestrians)	AP(Vans)
Hand-designed division	0.8261	0.8841	0.7069	0.8873
Data based division	0.8405	0.8945	0.7292	0.8980

5.5. Comparison with SSD

In this section, we compare the proposed MTL models (LC-MTL and CP-MTL) with SSD [21] to evaluate how object attributes associated with distance affect the object detection performance. We choose the data-based division for MTL models. Compared with the SSD, the proposed CP-MTL and LC-MTL not only deal with the object detection task but also take the object distance prediction task into account. Table 3 reports the performances of the CP-MTL, LC-MTL and SSD on KITTI dataset. It can be seen that the proposed MTL models of object detection and distance prediction consistently and significantly

outperforms SSD that only deals with object detection task. The improvement could be attributed to the MTL being able to exploit the visual relationship between object detection and object distance.

Although both CP-MTL and LC-MTL optimize the object detection task together with object distance prediction task, they have a significant difference in the objective of multi-task learning. The LC-MTL uses the linear multi-task combination strategy to simultaneously handle two classification tasks and the CP-MTL uses the Cartesian product-based multi-task combination strategy to integrate two tasks into a classification task. From Table 3 we observe a further performance improvement (mAP=0.8331 vs 0.8405) when using the proposed Cartesian product-based multi-task combination strategy for MTL. It suggests that the Cartesian product-based strategy is more suitable for the joint modeling of object detection and object distance.

Table 3: The detection performances of CP-MTL, LC-MTL and SSD with the data-based division strategy.

Method	mAP	Cars	Vans	Pedestrians
SSD	0.8104	0.8779	0.6741	0.8790
LC-MTL	0.8331	0.8933	0.8945	0.7113
CP-MTL	0.8405	0.8945	0.8980	0.7292

Finally, we exhibit an example of real-time dangerous object detection using a video. Fig. 7 shows four snapshots of the video at $t = 1s$, $t = 10s$, $t = 20s$ and $t = 30s$, respectively. Compared with other object detection systems, the proposed CP-MTL not only bounds the object in an image but also gives its danger level according to the predicted object distance, as the a bounding boxes with different colors, shown in Fig. 7. The detection system is implemented by MXNET [5] and runs on a Lenovo computer with a Nvidia Titan X GPU, an Intel i76700k CPU (4.0GHz) and 32GB RAM. The detection speed reaches 33

fps (frames per second), which is enough fast to deal with general videos with 24 fps. The demo is publicly available at `https://youtu.be/G5tf4O16Jf4`.

Figure 7: Snapshots from video detection with CP-MTL model.

6. Conclusion

We propose the CP-MTL algorithm for dangerous object detection in autonomous driving. Through Cartesian product-based multiple-task combination, CP-MTL can simultaneously optimize object detection and object distance prediction to exploit the relationship between them, namely the object attributes (pose, size and definition). We mathematically prove that the proposed Cartesian product-based combination strategy is more optimal than the linear multi-task combination strategy used in traditional MTL, when the two tasks are not independent. Also, we carry out systematic experiments to verify that the proposed method outperforms the state-of-the-art SSD object detection method and the traditional MTL method.

7. Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under Grants 61273136, 61573353 and 61533017, and the National Key Research and Development Plan under Grant No. 2016YFB0101000.

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[3] R. Aufrère, J. Gowdy, C. Mertz, C. Thorpe, C.-C. Wang, and T. Yata. Perception for collision avoidance and autonomous driving. *Mechatronics*, 13(10):1149–1161, 2003.

[4] M. Bruch. Velodyne HDL-64E LIDAR for unmanned surface vehicle obstacle detection. In *Proceedings of SPIE - The International Society for Optical Engineering*, pages 76920D1–76920D8, 2010.

[5] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[6] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3D object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[7] Y. Chen, D. Zhao, L. Lv, and C. Li. A visual attention based convolutional neural network for image classification. In *12th World Congress on Intelligent Control and Automation*, pages 764–769, June 2016.

[8] H. Cho, Y. W. Seo, B. V. K. Vijaya Kumar, and R. R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1836–1843, 2014.

[9] G. Erico. How google's self-driving car works. *IEEE Spectrum Online*, October 18, 2011. `http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works/`.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[11] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE Conference on Computer Vision*, pages 1440–1448, 2015.

[12] D. Halan. Self-driving cars: The next revolution. *Electronics for You*, 3(6):24–59, 2014.

[13] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *Proceedings of the 12th European conference on Computer Vision-Volume Part III*, pages 340–353, 2012.

[14] L. John, H. Jonathan, T. Seth, B. Mitch, C. Stefan, F. Gaston, F. Luke, F. Emilio, A. Huang, and K. Sertac. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008.

[15] J. Kim, H. Kim, K. Lakshmanan, and R. Rajkumar. Parallel scheduling for cyber-physical systems: analysis and case study on a self-driving car. In *ACM/IEEE International Conference on Cyber-Physical Systems*, pages 463–471, 2013.

[16] D. Lahat, T. Adali, and C. Jutten. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[17] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[18] H. Liu, F. Sun, D. Guo, B. Fang, and Z. Peng. Structured output-associated dictionary learning for haptic understanding. *IEEE Transactions on Systems Man & Cybernetics Systems*, 47(7):1564–1574, 2017.

[19] H. Liu, Y. Wu, F. Sun, B. Fang, and D. Guo. Weakly paired multimodal fusion for object recognition. *IEEE Transactions on Automation Science & Engineering*, PP(99):1–12, 2017. DOI: 10.1109/TASE.2017.2692271.

[20] H. Liu, Y. Yu, F. Sun, and J. Gu. Visualctactile fusion for object recognition. *IEEE Transactions on Automation Science & Engineering*, 14(2):996–1008, 2017.

[21] 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, 2016.

[22] H. Luo, S. C. Chu, X. Wu, Z. Wang, and F. Xu. Traffic collisions early warning aided by small unmanned aerial vehicle companion. *Telecommunication Systems*, pages 1–12, 2016. `.https://ideas.repec.org/a/spr/telsys/vyid10.1007_s11235-015-0131-5.html`.

[23] L. Lv, D. Zhao, and Q. Deng. A semi-supervised predictive sparse decomposition based on task-driven dictionary learning. *Cognitive Computation*, 9(1):115–124, 2017.

[24] A. Neubeck and L. V. Gool. Efficient non-maximum suppression. In *International Conference on Pattern Recognition*, pages 850–855, 2006.

[25] S. Nie, S. Liang, W. Xue, X. Zhang, W. Liu, L. Dong, and H. Yang. Two-stage multi-target joint learning for monaural speech separation. In *Sixteenth Annual Conference of the International Speech Communication Association*, pages 1503–1507, 2015.

[26] S. L. Poczter and L. M. Jankovic. The google car: driving toward a better future? *Journal of Business Case Studies*, 10(1):7–14, 2014.

[27] C. Premebida, O. Ludwig, and U. Nunes. LIDAR and visionbased pedestrian detection system. *Field Robotics*, 26(9):696–711, 2009.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[29] S. Roth, B. Hamner, S. Singh, and M. Hwangbo. Results in combined route traversal and collision avoidance. In *Field and Service Robotics*, pages 491–504, 2006.

[30] J. P. Sang, T. Y. Kim, S. M. Kang, and K. H. Koo. A novel signal processing technique for vehicle detection radar. In *IEEE MTT-S International Microwave Symposium Digest*, pages 607 – 610, June 2003.

[31] Z. Shan, Q. Zhu, and D. Zhao. Vehicle collision risk estimation based on rgb-d camera for urban road. *Multimedia Systems*, 23(1):119–127, 2017.

[32] S. T. Sheu, J. S. Wu, C. H. Huang, Y. C. Cheng, and L. W. Chen. Ddas: Distance and direction awareness system for intelligent vehicles. *Journal of Informationence & Engineering*, 23(3):709–722, 2007.

[33] W. Shin, J. Yu, and H. Seo. Rear warning control method and system for vehicle, US, US20140203924 Al, 2016, 2016.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] L. Spinello and R. Siegwart. Human detection using multimodal and multidimensional features. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 3264–3269, 2008.

[36] G. P. Stein, O. Mano, and A. Shashua. Vision-based ACC with a single camera: bounds on range and range rate accuracy. In *Proceedings of the IEEE Conference on Intelligent vehicles symposium*, pages 120–125, 2003.

[37] C. Stiller, J. Hippb, and S. Ewaldb. Multisensor obstacle detection and tracking. *Image & Vision Computing*, 18(5):389–396, 2000.

[38] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. *Advances in Neural Information Processing Systems*, 26:2553–2561, 2013.

[39] C. C. Wang, C. Thorpe, and A. Suppe. LADAR-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 416–421, 2003.

[40] B. F. Wu, Y. H. Chen, C. C. Kao, Y. F. Li, and C. J. Chen. A vision-based collision warning system by surrounding vehicles detection. *Ksii Transactions on Internet & Information Systems*, 6(4):1203–1222, 2012.

[41] Y. Xia, C. Wang, X. Shi, and L. Zhang. Vehicles overtaking detection using RGB-D data. *Signal Processing*, 112:98–109, 2015.

[42] J. Yim, H. Jung, B. I. Yoo, and C. Choi. Rotating your face using multi-task deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 676–684, 2015.

[43] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1036–1041, 2014.

[44] Z. Zhang, P. Luo, C. L. Chen, and X. Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108, 2014.

[45] D. Zhao, Y. Chen, and L. Lv. Deep reinforcement learning with visual attention for vehicle classification. *IEEE Transactions on Cognitive and Developmental Systems,*, 2016. DOI: 10.1109/TCDS. 2016.2614675.

[46] Q. Zhou, G. Wang, K. Jia, and Q. Zhao. Learning to share latent tasks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2264–2271, 2013.