IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING TECHNOLOGIES FOR INTERNET OF VIDEO THINGS

Received February 2, 2021, accepted March 15, 2021, date of publication March 26, 2021, date of current version April 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068991

MTSAN: Multi-Task Semantic Attention Network for ADAS Applications

CHUN-YU LAI[1], BO-XUN WU[1], VINAY MALLIGERE SHIVANNA[1], AND JIUN-IN GUO[1,2,3]

[1]Department of Electronics Engineering, Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan
[2]Pervasive Artificial Intelligence Research (PAIR) Labs, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan
[3]Wistron-NCTU Embedded Artificial Intelligence Research Center, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Vinay Malligere Shivanna (vinay.ms23@gmail.com)

This work was supported in part by the Center for mmWave Smart Radar Systems and Technologies through the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan, in part by the Ministry of Science and Technology (MOST), Taiwan Projects through the Pervasive Artificial Intelligence Research Labs (PAIR Labs) in Taiwan, under Grant MOST 108-3017-F-009-001 and Grant MOST 110-2634-F-009-020, and in part by the Qualcomm Technologies through the Qualcomm Taiwan University Research Program under Grant 408929.

ABSTRACT This paper presents a lightweight Multi-task Semantic Attention Network (MTSAN) to collectively deal with object detection as well as semantic segmentation aiding real-time applications of the Advanced Driver Assistance Systems (ADAS). This paper proposes a Semantic Attention Module (SAM) that introduces the semantic contextual clues from a segmentation subnet to guide a detection subnet. The SAM significantly boosts up the detection performance and computational cost by considerably decreasing the false alarm rate and it is completely independent of any other parameters. The experimental results show the effectiveness of each component of the network and demonstrate that the proposed MTSAN yields a better balance between accuracy and speed. Following the post-processing methods, the proposed module is tested and proved for its accuracy in the Lane Departure Warning System (LDWS) and Forward Collision Warning System (FCWS). In addition, the proposed lightweight network is deployable on low-power embedded devices to meet the requirements of the real-time applications yielding 10FPS @ 512 X 256 on NVIDIA Jetson Xavier and 15FPS @ 512 X 256 on Texas Instrument's TDA2x.

INDEX TERMS Advanced Driver Assistance System (ADAS), detection subnet, image segmentation, multi-task learning network, object detection, segmentation subnet, semantic attention module (SAM).

I. INTRODUCTION

Due to swift developments of deep learning and vision-based technologies, autonomous driving has developed into an extremely popular field of discussion in the recent years. The autonomous driving system is a vast and complicated system consisting of numerous modules and sensors with different functions. The key for a reliable autonomous driving system is its ability to recognize and understand the surrounding environment such as the behavior of the vehicles nearby, pedestrians and motorcyclists and their corresponding behaviors and more as shown in Fig. 1.

Deep Convolution Neural Networks (DCNN) exhibit a tremendous ability to tackle numerous vision based challenges such as image classification, object detection, semantic segmentation and so on. Many DCNNs demonstrate substantial accuracy on a variety of benchmarking tasks at the expense of a plenty of different parameters and incurring high computation costs. However, in order to meet some of the salient requirements of real-time applications of the advanced driver assistance system (ADAS) such as, lane departure warning system (LDWS), forward collision warning system (FCWS), adaptive cruise control (ACC), autonomous emergency braking (AEB), blind-spot detection (BSB) and so on, the algorithms should be capable of processing at an adequate frame rate and higher accuracy so that the implementation on resource-limited embedded platforms for ADAS real-time applications become feasible.

Most of the networks now aim to solve one specific task. In real applications, the process of integrating multiple individual algorithms into a single-unified learning framework is more efficient. Multi-task learning networks combine multiple tasks into a single unified task by exploiting the relationship between these different tasks, making it more efficient

The associate editor coordinating the review of this manuscript and approving it for publication was Jiu Xu.

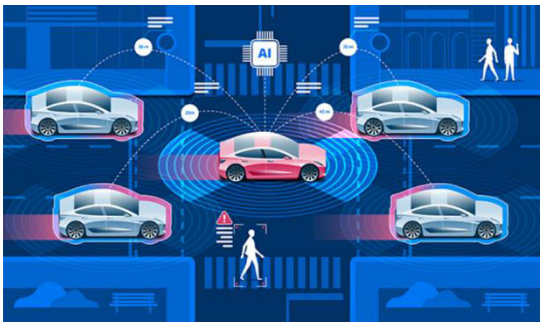

FIGURE 1. Schematic diagram of an autonomous vehicle.

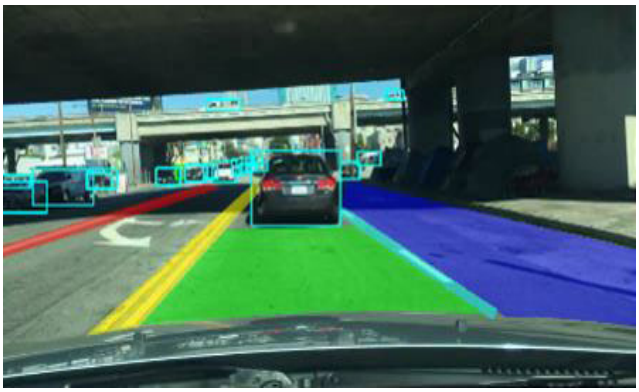

FIGURE 2. Multiple objects detection and semantic segmentation of the road for ADAS applications.

for the real-time applications. The networks can generalize a more accurate representation of targets by sharing the features between each tasks and thus may improve prediction accuracies and increase learning efficiencies. Moreover, by sharing the backbone layers, the overall network size and computational complexity can be exceptionally reduced which is also beneficial to fast inference requirements.

This paper proposes a lightweight multi-task semantic attention network (MTSAN) for multiple objects detection and semantic segmentation for ADAS applications as in Fig.2. The main contributions of this paper are listed as follows: (i) First, we explored the model backbone that acts as the feature extractor for the model. The backbone should be lightweight and efficient so that it can be implemented on resource-limited embedded devices. Further, we investigated and improved the detection subnet detector and segmentation subnet decoder for more robust prediction. (ii) The paper explores the relationship between object detection and semantic segmentation tasks and proposes ''semantic attention module (SAM)'' that utilizes the semantic clues from segmentation subnet to guide the detection subnet without any additional parameters. (iii) The authors explored and improvised each of the components of the network for better speed and prediction accuracy. (iv) The authors delved the feasibility of deploying the proposed network on low-power embedded devices processing at real-time to demonstrate the low complexity of the proposed network.

The rest of this paper is organized as below. Section II briefs some of the recent state-of-the-art algorithms developed for object detection, semantic segmentation as well as multi-task learning systems. Section III discusses the proposed methods in detail followed by the implementation and the post-processing methods illustrated in Section IV. Finally, Section V concludes the proposed work.

II. RELATED WORK

This below section briefly describes some of the previous state-of-the-art works on object detection, semantic segmentation and multi-task learning systems.

A. OBJECT DETECTION

Recent object detection networks are broadly divided into two types namely, two-stage architecture and single-stage architecture. The two-stage architectures require an extra proposal stage to capture the object proposals [1] and thus they have a strong advantage in terms of accuracy. On the other hand, the single-stage architectures directly classify and localize multiple objects without region proposal mechanism [2], [3] and hence is advantageous in terms of speed.

1) TWO STAGE OBJECT DETECTION

Region with Convolutional Neural Network (RCNN) [4] by Ross Girshick *et al.* combines the region proposals with CNN. First, RCNN extracts about 2k region proposals using a selective search method and then uses CNN to obtain high-level representation features of each proposal. In the end, RCNN classifies each region with class-specific linear support vector machines (SVMs). The main drawback of this method is that it is slow as each proposal is computed through the whole CNN.

Fast Region-based Convolutional Network (Fast R-CNN) [5] by Ross Girshick *et al.* exceedingly improves the speed of RCNN. Fast RCNN crops the region proposals from the convolution feature map and uses region of interest (ROI) pooling layer to pool each cropped features into the same size. Then it predicts class and bounding boxes offset of each ROI by softmax probability function and bounding box regression.

Faster R-CNN [1] proposed by Shaoqing Ren *et al.* further improves the time-consuming compared to Fast-RCNN. Faster R-CNN replaces the original selective search method with Region Proposal Network (RPN). The RPN predicts the region bound and objectness scores of generated anchor boxes. The high scores anchor boxes are viewed as proposals and fed into the ROI pooling layer to get same size features similar to that of Fast RCNN. With this modification, the Faster R-CNN turns out to be a single convolution only network with higher speed.

2) ONE STAGE OBJECT DETECTION

You Only Look Once (YOLO) [3] proposed by Joseph Redmon *et al.* models detection as a regression problem. It divides the image into S x S grid cells. Each grid cell predicts *B* bounding boxes, confidence scores for boxes

and C conditional class probability. These predictions are encoded as an $S\ x\ S\ x(B^*5+C)$ tensor. YOLO architecture is composed of total 24 convolutional layers followed by 2 fully connected layers.

Single Shot MultiBox Detector (SSD) [2] proposed by Wei Liu *et al.* makes boxes prediction easier by locating default boxes over different aspect ratios and scales per feature map location. A single deep convolution network predicts the adjustments to the boxes and confidence for the presence of each object category. The SSD predicts at multiple feature maps at different scales so that it can handle objects of various sizes. The architecture contains several convolution layers and decreases in size progressively to generate multiple scale feature maps.

Focal Loss for Dense Object Detection (RetinaNet) [6] proposed by Tsung-Yi Lin *et al.* points out the foreground-background class imbalance problem for most of the one-stage object detectors during training. Focal loss handles this problem in a better way by modifying cross entropy function and making training process more efficient. It shows the big improvement to the prediction accuracy.

MobileNets [7] proposed by Andrew G. Howard *et al.* are based on a streamlined architecture that uses depth-wise separable convolutions to build lightweight deep neural networks, and the main purpose is to execute on mobile, and embedded platforms. MobileNets demonstrate the effectiveness over a wide range of applications such as object detection and image classification.

B. SEMANTIC SEGMENTATION

Fully Convolution Networks for Semantic Segmentation [8] proposed by Jonathan Long *et al.* extends classification task to dense prediction task by transforming fully connected layers into convolution layers. In addition, deconvolution, also called as transpose convolution, is proposed to connect coarse outputs to dense pixels prediction. Lastly, the element wise summation is introduced to fuse the features from high-resolution feature to lower layer. The FCN architecture is an end-to-end trainable network.

SegNet [9] proposed by Vijay Badrinarayanan *et al.* found that the increasingly loss of image boundary detail has detrimental effect on semantic segmentation task. The decoder feature maps need some clues from encoder to recover boundary information. In order to achieve this, the unpooling operation is proposed. The locations of the maximum value in each max pooling is memorized and applied on unpooling. The overall architecture of SegNet consists of symmetry encoder and decoder.

DeepLab [10] proposed by Liang-Chieh Chen *et al.* points out that transpose convolution can be used to recover the spatial resolution of feature maps. However, it requires additional memory and has more parameters for network to learn. In order to handle this problem, it proposes atrous convolution for dense feature extraction and field-of-view enlargement. It can be integrated with training, and computes responses of any layer at any desirable resolution.

U-Net [11] proposed by Olaf Ronneberger et.al presents a network and effective data augmentation (DA) method for segmentation task. The U-shape network consists of a contracting path for encoding and expansive path for decoding. To deal with the loss of border pixels, the concatenation process is adopted to introduce encoder feature to decoder.

ENet [12] proposed by Aabhish *et al.* aims to perform semantic segmentation task in real-time. It designs a ResNet-like bottleneck module, and follows some rules to progressively down sample the feature maps. The other implementation details are also took into design consideration. The results shows that ENet has high inference speed not only on NVIDIA Titan X GPU but also on embedded NVIDIA TX1.

Dual Attention Network for Scene Segmentation [13] proposed by Jyn Fu *et al.* solves the pixel segmentation task by capturing rich contextual information based on the attention mechanism. The proposed two attention modules, position attention module and chancel attention module, integrate local features with global dependencies through different dimensions. This paper achieves state-of-the-art segmentation results that demonstrate the effectiveness of attention mechanism.

C. MULTI-TASK LEARNING

VPGNet [14] proposed by Seokju Lee *et al.* presents a network to joint detect lanes, road markings and vanishing points. The network is composed of AlexNet-based shared backbone and multiple sub-networks to predict object mask, multi-label, grid box, and vanishing point maps separately.

Fast Scene Understanding for Autonomous Driving [15] proposed by Davy Neven *et al.* tackles semantic segmentation, instance segmentation, monocular depth estimation task with a single integrated network. It uses ENet as backbone. Therefore, the run-time speed is faster.

MultiNet [16] proposed by Marvin Teichmann *et al.* presents a network to segment drivable area, detect vehicles, and classify street scenes via joint classification, detection and semantic segmentation for autonomous driving. The sharing encoder backbone reduces computation complexity and the whole network is easy to train. However, it does not discuss the task relationship between each task that may be an important clue to further enhance the performance.

End-to-End Multi-Task Learning with Attention Network (MTAN) [17] proposed by Shikun Liu *et al.* presents a novel multi-task learning architecture. Unlike common multi-task learning network that only share the last layer feature of encoder, the MTAN encoder works as a global feature pool, and each subnet learns task specific feature by using soft-attention module. The result shows that MTAN can increase the learning efficiency and is further robust towards different loss weighting schemes.

III. THE PROPOSED MULTI-TASK SEMANTIC ATTENTION NETWORK

The following section introduces the proposed lightweight Multi-task Semantic Attention Network (MTSAN) that can

concurrently deal with object detection and semantic segmentation. The network consists of a shared backbone encoder, a detection subnet, a segmentation subnet and a semantic attention module as shown in Fig.3. The following sections discuss each individual components in detail.

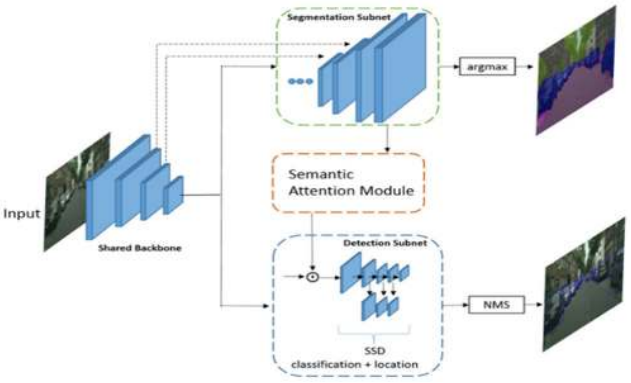

FIGURE 3. Overview of the MTSAN architecture.

A. BACKBONE ENCODER

The function of the backbone encoder is to process an input image and extract rich abstract features from the input image that represent the crucial information in the image. Instead of adopting very deep and wide architectures such as AlexNet [18], GoogleNet [19], DenseNet [20], ResNet101 [21] and VGG16 [22] that comprise numerous parameters and incurs higher computation cost, an open source lightweight architecture named JacintoNet [23] which is designed for embedded devices is adopted in the proposed method. The JacintoNet is a modified ResNet-10 by removing the shortcut connection. In order to reduce the computation complexity, it uses max-pooing instead of convolution with strides to do feature maps down-sampling process. In addition, it adopts group convolution at alternate layers to help in the reduction of the data bandwidth. On the other hand, the single-branch architecture's features of JacintoNet are found to be more efficient and fast on some hardware-embedded devices.

B. SEGMENTATION SUBNET

The architecture of segmentation subnet is designed similarly to U-Net [11] with several learnable up-sampling layers as shown in Fig. 4. The subnets are composed of several conv-blocks, which includes 3×3 convolution layer, batch normalization layer, and ReLU activation layer in order. The width and height of the output tensor of conv-block remain the same with input tensor whereas only the up-sampling process changes size.

In order to extract meaningful semantic features for segmentation, first, a conv-block is applied at the bottom of subnet. Then, instead of using pooling process to explore features at lower resolution, three subsequent conv-blocks with which the convolution layer inside is replaced with dilated one are adopted [6]. Due to the information loss caused from

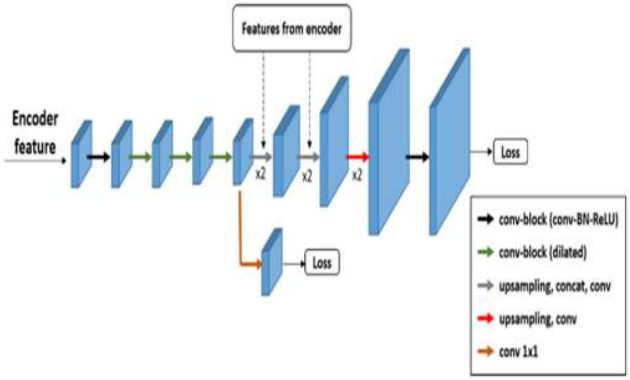

FIGURE 4. The architecture of segmentation subnet.

pooling process, the dilated convolution process is better for extracting dense feature response compared to three subsequent pooling operations with normal convolution. After the dense feature extraction, the implementation of up-sampling process is carried out to recover the spatial resolution. In order to recover the objects boundary efficiently during the process, the encoder features are introduced to decoder for more object-shape clues. The feature concatenation operation is employed instead of the element-by-element summation for better accuracy.

In addition to the loss at the top of subnet, an extra loss is applied at the feature before up-sampling layers to benefit from intermediate supervision. In order to map each feature vector to the desired number of classes for the extra loss, the convolution 1×1 is applied before the loss layer. Due to the intermediate supervision, the front part of subnet is forced to classify each pixel at that scale to fulfill the loss in the middle. Therefore, the remaining part of network can simply focus on the up-sampling process. In real-world applications, the intermediate supervision can also provide extra output choice for users. In this case, the segmentation output can be of higher resolution and more accurate or the one with lower resolution but faster one, depending on the users requirements.

C. DETECTION SUBNET

In order to fulfill the demands of real-time application requirements and achieve faster inference speed, the detection decoder is designed based on an one-stage approach adopting the classical, widely used SSD [2] detector.

The SSD detector generates several anchor boxes over different aspect ratios and scales at each feature map location. More specifically, each vector of feature map tensor along the channel dimension represents the anchor information at each image grid. The grid size depends on the feature map receptive field. The grid size might vary by a great deal due to multi-scale prediction. For instance, from an 8-pixel width to the whole network input size. After the anchor generation, the network output directly predicts the confidence score and location shift of each anchor by a single forward pass. The main advantage of this kind of anchor-based approach is that

it is easy to learn, as the deep network only requirement is to predict the boxes offset instead of learning the whole boxes information from scratch.

1) PROBLEM AND BASELINES

The authors have identified a weakness in the anchor-based approaches like SSD in which it is harder to detect objects in some locations. For instance, an object that is not directly located at an anchor location or an object that is located in the middle of two anchors. This problem usually occurs in objects with high-aspect-ratio such as pedestrians as the overlapping area between two adjacent high-aspect-ratio anchors are small. Fig.5 (a) shows the scenario in which a pedestrian is walking from the left-hand-side to the right-hand-side of the image. It is observed that such a pedestrian cannot be detected successfully in every individual frames. In order to understand the cause of this problem, some failure cases are visualized by drawing the default anchor boxes that are close to the pedestrian's location at the corresponding scale as shown in Fig.5 (b). It can be seen that when the pedestrian walks through the position at which the anchor centers are not located, the prediction fails. This is a major problem for a lightweight network as its regression ability is limited.

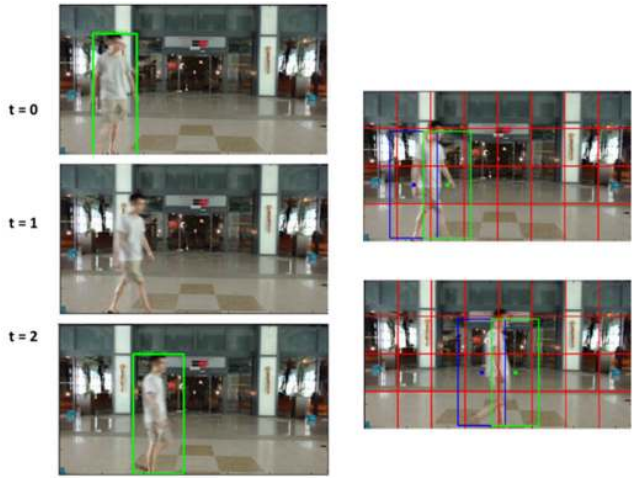

FIGURE 5. (a) Default SSD detector predictions at different time. (b) Failure cases visualization of closest anchors to the pedestrian.

2) PROPOSED MULTI-HEAD DENSE ANCHORS APPROACH

The multi-head dense anchors method as in Fig. 6 (a) is adopted in order to fill the gaps between the adjacent anchors by inserting more anchors at the corners of the grid cells depicted by the blue points in Fig.6 (b).

In order to classify and regress extra anchors, it is necessary to increase the number of SSD detector heads. Fig.7 (a) shows the original SSD detector and detection feature vectors encoding the grid cells information. The proposed multi-head dense anchors' architecture applies 3×3 convolution at the original detector features to generate mixed features as shown in Fig. 7 (b). Due to the constraints of the implementation

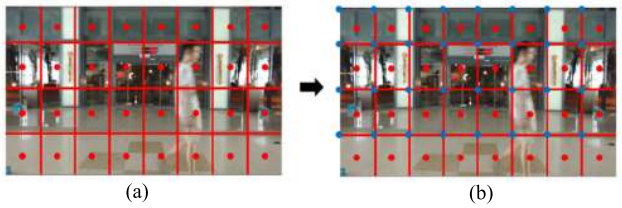

FIGURE 6. (a) Illustration of anchors center. (a) Red points represent the original anchors center. (b) Blue points represent the appended new anchors.

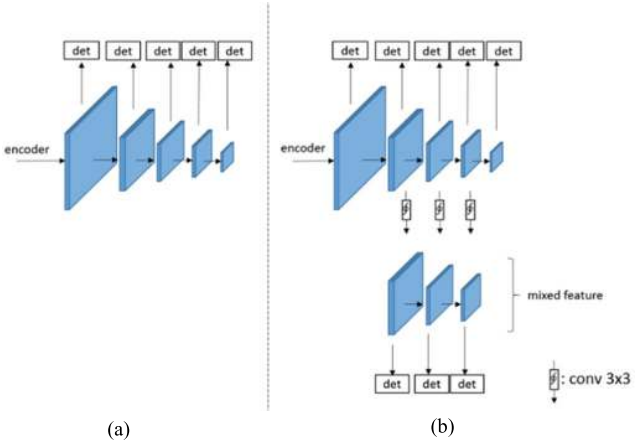

FIGURE 7. Detection subnet architectures. (a) Original SSD detector architecture. (b) Multi-head SSD detector architecture.

platform and increase in the model complexity, we only apply the dense anchors method at three scales as it is experimentally determined that the 3×3 convolution here works as the feature mixer and combine adjacent grid cells information to get the mixed features at corner positions. With the corresponding pre-trained models, the multi-head detection subnet is easy to train and converge fast with the intuitive mixed features concept. Although the multi-head architecture marginally increases the network size and computation cost, we found that it is acceptable and the improvement in quality is significant.

D. SEMANTIC ATTENTION MODULE

For a multi-task learning network to collectively administer detection and semantic segmentation, the two sub-networks share the backbone encoder and distinctly extract the task-specific features. Although the network is easier to generalize a target representation due to multi-task learning and sharing backbone features, there still exists some deficiency. In this work, we have implemented a multi-task learning network without any extra features, and the prediction results of the Cityscape dataset [24] are shown in Fig. 8.

From detection prediction results as in Fig. 8 (a), it can be noted that a pedestrian at the image boundary is not detected. On the contrary, from semantic segmentation prediction shown in Fig. 8 (b), the pedestrian who was missed in detection has been classified well, pixel by pixel. More results

(a) (b)

FIGURE 8. Illustration of common multi-task network output. (a) Detection prediction, (b) Segmentation prediction.

were similar from numerous experiments. Thus, it can concluded that semantic segmentation provides location clues of objects that can be utilized in detection subnet. Additionally, the benefit from the complete alignment between network input and segmentation output maps, the location information can be applied to different scales feature maps through downsampling process.

In order to utilize the semantic information in segmentation subnet, a new approach termed Semantic Attention Module (SAM) is proposed in this paper in order to introduce the features from semantic segmentation subnet to object detection subnet. The SAM builds up a connection between the two tasks as shown in Fig.9.

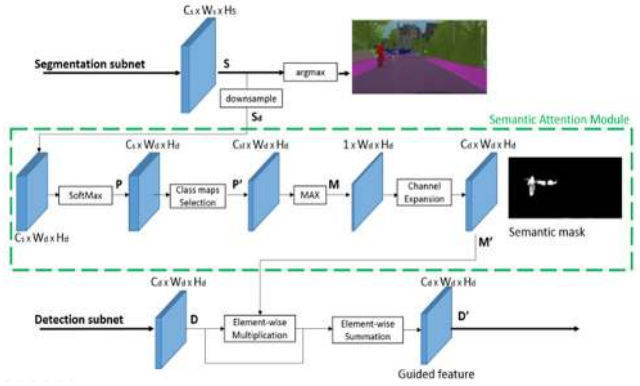

FIGURE 9. The architecture of semantic attention module (SAM).

In order to match the tensor size of detection subnet, the input of the SAM, $S_d \in \mathcal{R}^{C_s \times W_d \times H_d}$ is obtained by rescaling the segmentation output activation maps, $S \in \mathcal{R}^{C_s \times W_s \times H_s}$ given by Eq. (1).

$$\mathrm{S_d} = \mathrm{Downsample\ (S)} \tag{1}$$

where the parameter *Downsample()* represents the downsampling process which can be bilinear interpolation or max-pooling process.

To extract the useful information from the segmentation maps, the first softmax function is applied on each position to get the probability maps. Then, the channel of probability maps related to object detection category such as pedestrian, vehicles etc. is selected. In other words, the unrelated categories, such as road, are discarded. The softmax output probability maps, $P \in \mathcal{R}^{C_s \times W_d \times H_d}$ and selected probability maps $P' \in \mathcal{R}^{C_{sl} \times W_d \times H_d}$ can be presented as in Eq. (2) and Eq. (3), respectively.

$$\mathrm{P} = \mathrm{Softmax}(S_d) \tag{2}$$

$$\mathrm{P'} = \mathrm{Select(P)} \tag{3}$$

where *Softmax()* and *Select()* represent the *softmax-2d*-function and class maps selection function, respectively.

After obtaining the objects probability maps, the maximum operation is applied at each position to get the semantic attention mask to encode the objects response. In practice, the semantic attention mask is multiplied with a parameter λ to control the strength of attention. Furthermore, the semantic attention mask tensor is obtained by the channel expansion function in order to match the tensor size of the detection subnet. Lastly, in order to generate the guided feature, the semantic attention mask tensor is applied on the feature of detection subnet through the element-by-element multiplication and summation that work as the attention operations. The generated guided features $D' \in \mathcal{R}^{C_d \times W_d \times H_d}$ are respectively obtained using Eq. (4), Eq. (5) and Eq. (6).

$$\mathrm{M} = \mathrm{Max(P')} \tag{4}$$

$$\mathrm{M'} = \mathrm{Expand(M}, C_d) \tag{5}$$

$$\mathrm{D'} = ((\lambda \times \mathrm{M'} \otimes)\mathrm{D}) \oplus \mathrm{D} \tag{6}$$

where *Max()* function represents maximum operation through channel axis, *Expand(T, N)* function transforms single channel map $T \in \mathcal{R}^{1 \times W_d \times H_d}$ to N channel tensor, $\otimes$ and $\oplus$ represent element-wise multiplication and summation, respectively. $\mathrm{M} \in \mathcal{R}^{1 \times W_d \times H_d}$ implies semantic attention mask, $\mathrm{M'} \in \mathcal{R}^{C_d \times W_d \times H_d}$ is the semantic attention mask tensor, $\mathrm{D} \in \mathcal{R}^{C_d \times W_d \times H_d}$ is the detection feature, and $\mathrm{D'} \in \mathcal{R}^{C_d \times W_d \times H_d}$ means the guided feature.

After the SAM process, the object response in the original detection feature via the attention mechanism is featured, and the detector utilizes the generated guided feature to capture and localize objects easier. The experimental results and further ablation study are discussed in Section IV.

E. IMPLEMENTATION DETAILS

1) TRAINING STRATEGY

Due to loss imbalance and model capacity, it is found that the network was hard to converge and reach the global minimum using end-to-end training strategy. Hence, we adopt a two-stage training strategy in all our experiments. First, the network is trained with only semantic segmentation subnet by freezing all the parameters of detection subnet. During the first stage of training, it was found that the weight filters learn the global contextual information in images. The first-stage training stops until the loss converges. Then, the backbone encoder and segmentation subnet parameters are frozen and the object detection subnet with semantic attention module is trained. For training the SSD detector, the original multi-box-loss is replaced with Focal Loss [6] to deal with the imbalance problem of foreground and background labels.

a: SEMANTIC SEGMENTATION SUBNET

In data pre-processing, the input images are randomly scaled between 0.5∼1.5 followed by random cropping of a patch from these scaled images. Finally, the images are resized to 512 × 512 during training. The pre-training model trained on ImageNet is used for encoder weights initialization. The softmax cross entropy is used for the pixel level classification task. The Adam optimizer is adopted in this paper with initial learning rate 1e-4 and reduce it on plateau by a factor of 0.1 to optimize the network. The training is terminated when the loss converges.

b: OBJECT DETECTION SUBNET WITH SAM

The training procedures as in the design [16] are employed in this paper with certain modifications. For data pre-processing, the random sample-crop process is swapped by directly resizing input image to network input size during training. Then, we adopt Focal Loss [17] for the classification objective function to deal with imbalance problem of foreground and background labels, and smooth L1 loss is used for bounding box regression. We choose Adam optimizer with initial learning rate 1e-5 and reduce it on plateau by a factor of 0.1 to optimize the subnet. The training is terminated when the loss converges.

2) INFERENCE

During inference, the overall architecture works as a single stage end-to-end model. The softmax function is applied on the top of the output of the segmentation activation maps to get the output probability maps. Then, the probability maps are fed into the SAM to get the guided features. The detection subnet utilizes the guided feature to generate the detection output.

IV. EXPERIMENTAL RESULTS

A. DATASETS AND METRICS

1) CITYSCAPE DATASET

Cityscape Dataset [24] is a large-scale urban street scenes dataset that contains 5000 images collected from 50 different cities across Europe. The annotations contain around seven categories that are further divided into 19 classes and 2 types. The first type called *things* contains the categories that are countable such as, car, people, and so on whereas the other type called *stuff* contains the categories that are uncountable and have amorphous regions such as roads, grass, footpath, and so on. For semantic segmentation task, we classified all the classes. For object detection task, we tested on only the countable *things* type.

2) BERKELEY DEEPDRIVE

Berkeley DeepDrive (BDD) [25] is also a large-scale dataset that contains almost 100K images collected from several cities in America for autonomous driving applications. It is collected in different environments and weather conditions, making it more suitable for real-world applications. For object detection task, the bounding boxes of seven classes related to moving objects are used. For semantic segmentation task, drivable area and lane marking annotations are adopted.

3) METRICS

To evaluate the quantitative performance of the proposed network, the two widely used metrics namely mean intersection over union (mIOU) [8] and mean average precision (mAP) [26] are adopted to measure semantic segmentation task and object detection task, respectively. The mIOU is calculated as per the Eq. 7 where n_{cl} represents the total number of classes, n_{ji} represents the number of pixels of class i predicted to belong to class j, and t_i represents the total number of pixels of class i.

$$\text{mIOU} = \frac{1}{n_{cl}} \frac{\sum_i n_{ii}}{(t_i + \sum_j n_{ji} - n_{ii})} \tag{7}$$

On the other hand, mAP has various versions for its calculation. In this paper, the PASCAL VOC 2007 metric [26] is adopted. The mAP is calculated under the intersection over unit (IOU) threshold of 0.5. With the IOU threshold, the predicted bounding boxes can be classified into either true or false. With the aim to get the mAP, the average precision (AP) of each class should be computed first, and then the mAP will be the average value over AP of all classes. For computing AP, the boxes related to one specific class are sorted by the experimentally set confidence threshold. Following the order, the precision and recall are computed and the precision over recall curve is plotted. The average precision is then computed as the average over the precision value at 11 different recall rates using the Eq. 8 and Eq. 9 where $P_r(a)$ represent the precision value at recall rate a, and n_{cl} represent the total number of classes.

$$\text{AP} = \frac{1}{11} \times (P_r(0) + P_r(0.1) + \cdots + P_r(1.0)) \tag{8}$$

$$\text{mAP} = \sum_i^{n_{cl}} AP_i \tag{9}$$

B. SEGMENTATION SUBNET

1) ABLATION STUDY

The ablation experiments were carried out by decomposing two key parts of the segmentation subnet. First, we train the network but removing the intermediate loss followed by removing the shortcut connection introducing the feature from encoder to decoder. The validation mIOU on Cityscape, number of parameters, and frame rate on Titan X GPU of trained models are shown in Table 1.

The number of parameters here contain the backbone encoder and the segmentation subnet. It can be noted that the shortcut connection results in improved mIOU by almost 1%, demonstrating the importance of boundary information provided by the high-resolution features from the encoder. Moreover, appending the intermediate loss during training further boosts up the performance to 70.17% mIOU, which proves the beneficial effect from the

TABLE 1. Results of the proposed segmentation subnet on the cityscape validation set at 2048 × 1024 resolution.

Symbol	mIOU (%)	Number of Parameters	Frame Rate
Decoder	68.40	2.819 M	11.38
Decoder + shortcut	69.35	2.871 M	10.93
Decoder + shortcut + interLoss	70.17	2.872 M	10.67

intermediate supervision. The mIOU scores of each class are shown in Table 2.

TABLE 2. Detailed mIOU results of the proposed segmentation subnet on the cityscape validation set at 2048 × 1024 resolution.

Class	mIOU (%)	Class	mIOU (%)
road	97.49	sky	93.33
sidewalk	81.60	person	76.58
building	90.35	rider	53.36
wall	44.35	car	93.16
fence	49.78	truck	54.19
pole	57.62	bus	77.43
traffic light	60.48	train	55.66
traffic sign	72.36	motorcycle	51.55
vegetation	91.32	bicycle	72.51
terrain	60.13		
average		**70.17%**	

2) COMPARISON

With the purpose of comparing the proposed design with the state-of-the-art methods, the test set predictions of the other methods and the proposed method on the Cityscape evaluation server is determined. The results of the same are as tabulated in Table 3. The proposed method yields the mIOU 70.17% with adequate frame rate as required for the real-time ADAS applications. Considering the time consumption, we only include the methods that have reported the corresponding run times. Our method strikes a good trade-off between the accuracy, inference speed and model size, and the simple straight architecture is more suitable for hardware-embedded devices.

TABLE 3. Comparison with the state-of-the art works on cityscape test set.

Components	mIOU (%)	Number of Parameters	Frame Rate
ENet [18]	58.30	0.359 M	76.9
SQ [19]	59.80	Unknown	16.7
FCN-8S [4]	65.30	134 M	2.0
Dilation10 [20]	67.10	134 M	0.25
PSPNet [21]	81.20	56.27 M	0.78
Proposed Method	**70.17**	**2.87 M**	**10.67**

C. DETECTION SUBNET

In this section, the evaluation on the detection subnet individually by training the backbone encoder and detection decoder together and ignore the segmentation decoder and SAM is performed.

The multi-head anchor SSD architecture has enabled us to overcome the discontinuity detection problem caused by sparse anchors distribution as shown in Fig. 5. Fig. 10 shows the prediction results of two models. It can be noted that the predictions of multi-head detector is successful in all the frames. Further, the variation of the confidence values can be observed from the plot of confidence values verses frame index as shown in Fig. 11 where the green curve represents original SSD detector predictions whereas the blue curve represents the multi-head SSD detector predictions. The mean and standard deviation of the two curves are listed in Table 4. Although the confidence values predicted by the proposed method varies frequently due to the engagement of more anchors, the overall values are higher than the results predicted by the original SSD detector implementation.

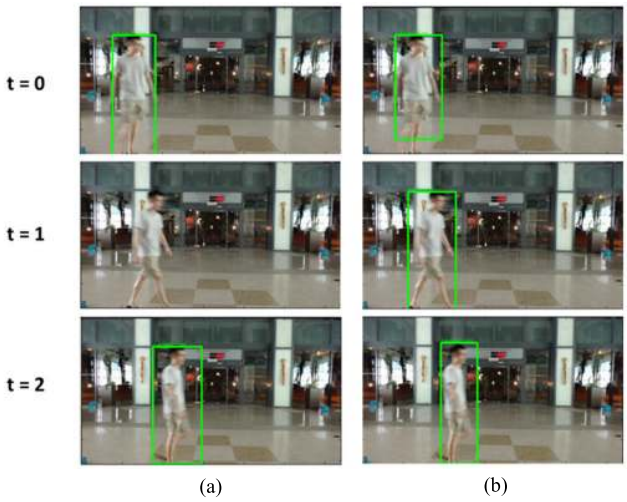

FIGURE 10. Pedestrian prediction of two methods at different time. (a) Original SSD. (b) Multi-head SSD.

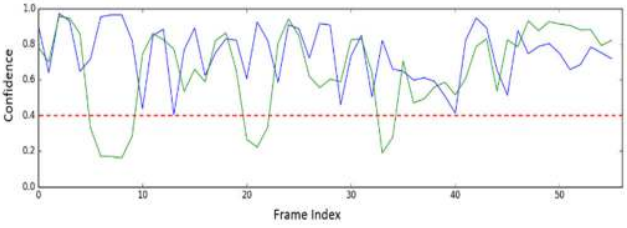

FIGURE 11. Confidence curves of pedestrian in different SSD detectors.

In order to compare the proposed design with other works, we have re-implemented two of the popular networks using

TABLE 4. The mean and standard deviation of confidence values of the two methods.

Method	Mean	Frame Rate
Original SSD (Green curve)	65.77	23.43
Multi-head SSD (Blue curve)	74.68	15.36

highly optimized tensorflow-object-detection-API [27]. The first one is ResNet101-Faster-RCNN pre-trained on the COCO dataset [28] and we view it as the upper bound of the Cityscape detection. The other one is the MobileNet-SSD with default settings and pre-trained on COCO dataset and it is viewed as the contemporary of the proposed method. The input sizes of all these networks are 1024×512 and networks are implemented on a NVIDIA Maxwell Titan X GPU. The results are shown in Table 5. The performance of the proposed detector with default setting is comparatively better than the MobileNet-SSD, which demonstrates the best features from JacintoNet. For the proposed multi-head version, we sacrifice the inference speed and model size to get the performance boosting, and it is found that it is an acceptable trade-off.

TABLE 5. Comparison with other works on cityscape validation set.

Method	mAP (%)	# of Parameters	Frame Rate (fps)
ResNet101-Faster RCNN	47.10	62.43 M	8.93
Darknet53-YOLOv3	46.24	61.61M	10.31
MobileNet-SSD	35.26	4.76 M	66.30
Our (default setting SSD)	35.47	3.75 M	69.93
Our (Multi-head SSD)	36.91	4.62 M	57.62

D. MULTI-TASK SEMANTIC ATTENTION NETWORK

1) ABLATION STUDY AND DIFFERENT λ PARAMETERS

For ablation study, we directly trained a multi-task network without SAM and compared with the proposed MTSAN. Then, we explore MTSAN with different λ parameters that work as the multiplication factors during the attention operation. The results are shown in Table 6.

Comparing the network that has been only trained for object detection with Multi-task and without SAM method, we can see a drop in accuracy by 3.4%, which might be due to fewer learnable parameters caused by fixing backbone parameters during two-stage training. However, with the SAM, the MTSAN boosts up the performance from 33.50% to 35.92% with an increase of mAP by 2.42% when $\lambda = 1.0$. This demonstrates the effectiveness of spatial information provided by attention module.

Further experiments were conducted with the increased value of λ parameters that represent the increase of attention response applied on the detection feature. With $\lambda = 1.3$, the mAP of the detection results increase to 39.78%, proving the significant improvement compared to $\lambda = 1.0$. As there were no further accuracy improvements with respect to λ, we did not observe further accuracy improvement

TABLE 6. Detection results of MTSAN with different λ values and Other works.

Method	mAP (%)
ResNet101-Faster RCNN	47.10
Darknet53-YOLOv3	46.24
MobileNet-SSD	35.26
Ours - detection only	36.91
Multi-task without SAM	33.50
MTSAN, λ=1.0	35.92
MTSAN, λ=1.1	36.61
MTSAN, λ=1.2	38.45
MTSAN, λ=1.3	**39.78**
MTSAN, λ=1.4	37.46
MTSAN, λ=1.5	37.66

for $\lambda > 1.5$. The experiments prove that the appropriate increase of attention clue is helpful for detection prediction. To show the effectiveness of the SAM, we provide qualitative results comparison obtained by network with and without SAM as shown in Fig.12. The visualization results show that the MTSAN is better for reducing missed predictions and for localizing objects more accurately, and with higher probability to capture small objects at a farther distance.

2) COMPARISON WITH OTHER FUSION METHODS

The MTSAN introduces semantic features through semantic attention module, while we also explore two other methods to introduce features from segmentation subnet. The first one has adopted element-wise summation to add the segmentation features with detection features and the extra 1×1 convolution is applied before operation due to the different channel dimension. The other one is concatenating the features at the top of segmentation subnet with the detection features. As shown in Table 7, both the methods have effects on the detection results, but the result predicted by SAM is much better than these two methods, which demonstrates the effectiveness of SAM.

TABLE 7. Comparison with other works on cityscape validation set.

Method	mAP (%)
Multi-task without any feature fusion	33.50
Multi-task with feature element-wise summation	34.53
Multi-task with feature concatenation	35.92
MTSAN, λ=1.3	**39.78**

3) VISUALIZATION MASKS AND FEATURES

To further understand the attention modules, we visualize the attention masks and guided features before and after attention operation as shown in Fig.13. The visualization results are obtained by the normalization process and the 2D feature maps are chosen from the feature tensor randomly. From the

FIGURE 12. Visualization of MTSAN on Cityscape validation set. From left to right with (a) Input image, (b) Segmentation prediction, (c) Detection prediction without SAM, (d) Detection prediction of MTSAN ($\lambda = 1.3$).

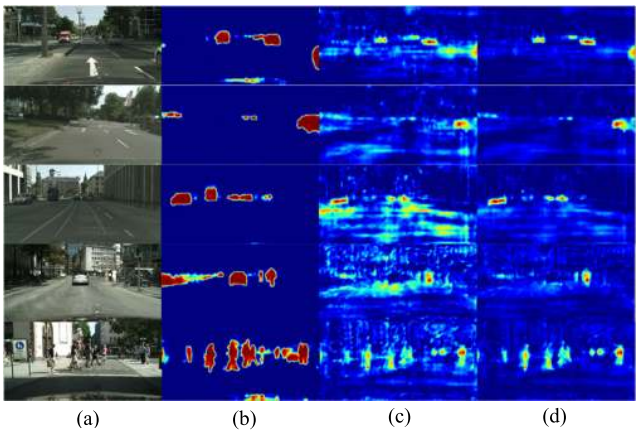

FIGURE 13. The semantic attention mask and its guided features. (a) Input image, (b) Semantic attention mask, (c) Features before attention process, (d) Guided features after attention process.

results, it can be seen that applying semantic attention mask can enhance the objects response and degrade unimportant noise from the feature maps making it easier for the network to focus on appropriate objects.

4) INFERENCE SPEED AND MODEL SIZE ANALYSIS

The proposed MTSAN consists of a backbone encoder, a segmentation subnet, a detection subnet and a semantic attention module. The parameter sizes and the inference time of each component are given in Table 8. For inference time analysis, the input size of the network is 1024×512 and the GPU device is NVIDIA Maxwell Titan X. The overall light-weight network contains only 4.94 million parameters. Most of the parameters are in the backbone encoder and detection subnet due to the deeper architectures.

TABLE 8. Model size and inference time analysis.

Network Component	Number of Parameters	Inference Time (ms)
Backbone Encoder	2.55 M	5.71
Segmentation Subnet	0.32 M	11.39
Detection Subnet	2.07 M	12.13
Semantic Attention Module	0 M	0.14
Total	4.94 M	29.37

The segmentation subnet required only a few parameters but the inference time is long due to the bigger feature maps obtained via up-sampling process. The detection subnet is slowest due to several bounding boxes generation, regression process and non-maximum suppression. For the proposed SAM, it does not require any extra parameters, and takes only a little inference time, which results in the low cost method.

5) IMPLEMENTATION ON BDD DATASET

Compared to the Cityscape dataset, the segmentation prediction in BDD dataset does not contain any classes that detection process tries to predict. Therefore, the formulation to adapt it to the BDD dataset is modified as follows: (i) First, in the *Select()* function, all the classes in the segmentation maps activation are selected and sent into the SAM. After the *Max()* operation, the semantic mask here represents the drivable area region. The example masks are given in Fig.14.

TABLE 9. Comparison with the state-of-the art works on cityscape test set.

Main Lane	Alternative Lane	Double Line	Dash Line	Single line	Background	mIOU
71.47	56.79	53.80	49.59	48.23	94.50	62.40

TABLE 10. Detection prediction results comparison on BDD validation set (%).

Method	Car	Bus	Truck	Person	Rider	Bike	Motor	mAP
Multi-Task without SAM	55.6	25.8	28.4	29.7	12.5	14.5	9.1	25.1
MTSAN, λ=1.0	60.8	27.7	30.6	37.6	14.9	18.1	12.1	28.8

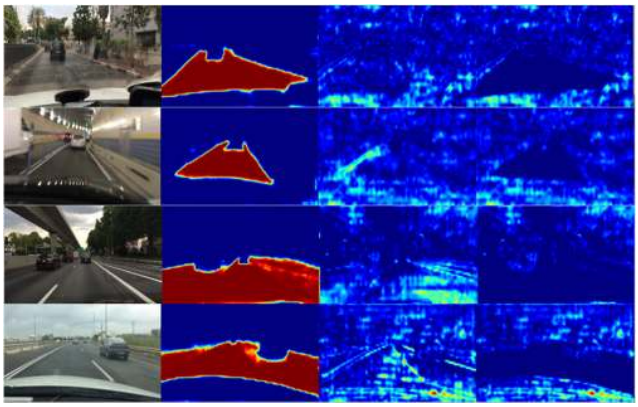

FIGURE 14. The semantic attention mask and its guided features on BDD dataset. (a) Input image, (b) Semantic attention mask, (c) Features before attention process, (d) Guided features after attention process.

The attention operation is modified as in Eq. (10).

$$\mathrm{D}' = \mathrm{D} \ominus ((\lambda \times \mathrm{M}') \otimes \mathrm{D}) \tag{10}$$

where $\otimes$ and $\ominus$ represent element-wise multiplication and subtraction, respectively. $\mathrm{M}' \in \mathcal{R}^{C_d \times W_d \times H_d}$ means semantic attention mask tensor, $\mathrm{D} \in \mathcal{R}^{C_d \times W_d \times H_d}$ means semantic attention mask tensor, $\mathrm{D} \in \mathcal{R}^{C_d \times W_d \times H_d}$ means detection feature, $\mathrm{D}' \in \mathcal{R}^{C_d \times W_d \times H_d}$ means guided feature.

The training results of MTSAN are shown in Table 9 and Table 10. Even though the segmentation task does not predict objects categories, the SAM still can boost up the detection performance significantly by degrading the background response. In addition, the detection results suffer from data imbalance in the BDD dataset and requires dataset fine-tuning. Qualitative results of the MTSAN prediction are provided in Fig. 15.

6) FAILURE CASES

Although the SAM has proved beneficial from the results in the previous sections, there exists certain failure cases when tested on Cityscape dataset as shown in Fig. 16. Since we have applied segmentation attention masks on detection features and rely on semantic spatial hints, the false alarms, though minimal, may introduce wrong information to the detection features and cause incorrect detection predictions. Considering this as a pivotal issue, we list it as a future work of the proposed method.

E. POST-PROCESSING

The post-processing methods are employed to deal with the semantic segmentation predictions for further applications. For BDD dataset, we mainly classify the output maps into two categories such as lanes and lane markings. The following sections introduces the proposed post-processing methods on these two categories, respectively.

1) LANE MARKING POST-PROCESSING

The proposed lane marking post-processing method is divided into three steps namely: (i) the local maximum extraction, (ii) clustering, and (iii) the polynomial curve fitting. First, the lane marking probability maps are stacked into a single channel binary response map, and then we scan all the regions in the maps through the *y-axis*. For each value of y, we can get one x vector, and if there is any lane response in the vector, we pick the mid-point of each response as a local maximum point. After scanning through all the y values, all the possible local maximum points of lane marking are stored.

After capturing all the local maximum points, we adopt our proposed clustering methods. In brief, we cluster the local maximum points through the y-direction and follow the two main constraints namely, the minimum distance and angle between the cluster and candidate point both need to be small.

After the clustering step, each cluster will define their class type by majority vote. Last, the polynomial curve fitting is used to get the formulation of each lane marking.

$$\begin{aligned} \textit{Detection rate} &= \frac{TP}{TP + TN} \\ &= \frac{\textit{Correct predictions}}{\textit{Departure ground truth}} \end{aligned} \tag{11}$$

$$\begin{aligned} \textit{False alarm rate} &= \frac{FP}{TP + FP} \\ &= \frac{\textit{False predictions}}{\textit{All predictions}} \end{aligned} \tag{12}$$

The Lane Departure Warning System (LDWS) is implemented using the lane marking post-process results. First, we define two symmetry boundary points on the vehicle say, car's hood in order to judge the occurrence of the lane departure. Then, for each lane marking, we obtain the extension

FIGURE 15. Visualization of MTSAN prediction on BDD validation set. (a) Input image, (b) Ground truth label, (c) MTSAN prediction result.

point by calculating the polynomial curve output x with the same y-coordinate as the boundary points. If the output coordinate x is located between the two boundary points, the lane departure has occurred. In order to evaluate the reliability of the proposed system, we pick up several inclement weathers including highway-driving videos captured in Taiwan and calculate the detection rate and false alarm rate, as defined by Eq. 11 and Eq. 12, respectively. As shown in Table 11, our system achieves 98.31% detection rate and 3.45% false alarm rate averagely and qualitative results are shown in Fig. 17.

TABLE 11. Lane departure warning system experimental results.

#	Weather	Departure (Ground Truth)	Correct predictions	False predictions
1	Day	9	9	0
2	Day	13	13	0
3	Night	8	8	0
4	Night	8	8	0
5	Night	8	8	0
6	Rainy day	7	7	1
7	Rainy day	6	5	1
Total		59	58	2
Detection rate = 98.31% (58/59);			**False alarm rate = 3.45% (2/58)**	

2) LANE POST-PROCESSING

The lane prediction results from segmentation subnet can be classified into two categories like (i) the main lane; (ii) the alternative lane. In our application, it is viewed as one class. First, we define the path of interest that represents the path that a driver will pass through, and we divide all the cases into two circumstances. The first case is that the main lane is surrounded by lane markings, and we define the region surrounded by the lane markings as path of interest. The other circumstance is that there is no lane marking. We have to pre-define the path that the drivers might pass through by ourselves. Since we cannot get the actual steering wheel angle and direction from the simulation data, we can only assume the vehicle to go straight and define a fixed path. After defining the path, in the same way, we get the path of interest. After obtaining the path of interest, it is overlapped with the drivable area which is the region predicted by all the segmentation classes. The overlapping region represents the drivable region along the path that vehicle might take. Importantly, the region in path of interest but not in drivable area represent the non-drivable region along the path that vehicle might take, and the point that contains the smallest y-coordinate in this region is the target considered as the closest point. After we get the closest point, we draw the stop line for visualization. The process is shown in Fig. 18.

After getting the stop line, we use this information to implement the function of Forward Collision Warning

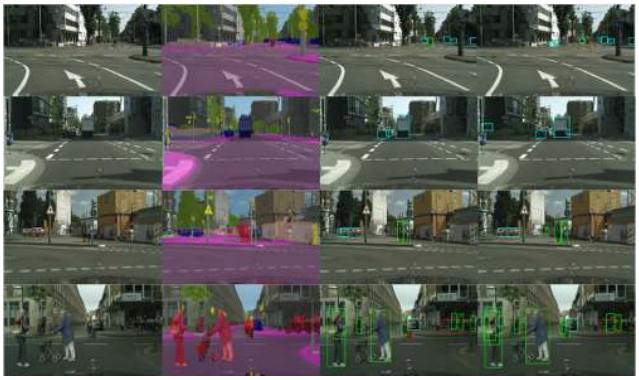

FIGURE 16. Visualization of failure cases: (a) Input image, (b) Segmentation prediction, (c) Detection prediction without SAM, (d) Failure cases predicted by MTSAN ($\lambda = 1.3$).

FIGURE 17. Qualitative results of the lane departure warning system (LDWS).

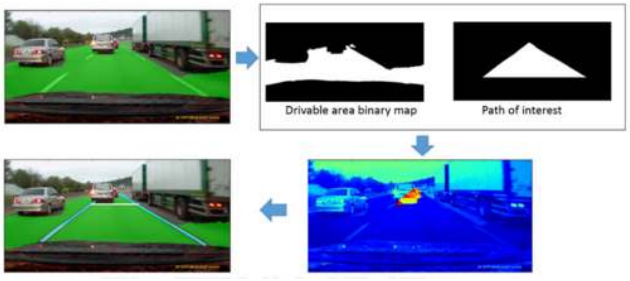

FIGURE 18. The process flow of identifying the stop line.

System (FCWS). Theoretically, a monotonous camera cannot estimate depth. However, with prior knowledge as in [24] that assumes the road is flat, it is possible to estimate the distance of an object using a single monotonous camera. That is, if we assume the road is flat, we can use geometric relation between the road and the camera to estimate the object distance. Fig. 19 shows the qualitative results. If the estimated distance of stop line is smaller than 15 meters, the line is colored red indicating a warning signal.

F. IMPLEMENTATION ON HARDWARD PLATFORMS

In this section, we explore two embedded devices, NVIDIA Jetson Xavier [28], and Texas Instrument TDA2x [29], to prove the porting ability of the proposed methods. The specification of device and performance evaluation are included.

1) NVIDIA JETSON XAVIER

NVIDIA Jetson Xavier [28] as shown in Fig. 20 (a) contains commonly used Linux environment, includes many

FIGURE 19. Qualitative results of forward collision warning system.

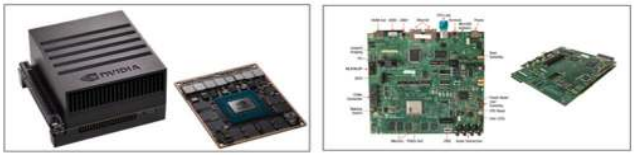

FIGURE 20. (a) NVIDIA Jetson Xavier, (b) TI TDA2x.

common APIs, and is supported by NVIDIA's complete development tool chain. The specification of NVIDIA Jetson Xavier is shown in Table 12.

TABLE 12. Lane departure warning system experimental results.

Parameters	Specifications
CPU	8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3
GPU	512-core Volta GPU with Tensor Cores
Memory	16GB 256-Bit LPDDR4x | 137GB/s
CUDA Version	CUDA-10.0
I/O	HDMI x 1, USB3.0 x 1, …

The inference speed on Jetson Xavier compared to the powerful GPU, such as Titan X, is almost 10 times slower due to the number of CUDA cores and clock rate. In order to port our algorithm on it, we have to downsize the input resolution to 512×256, and retrain the network. It achieves a run-time of 10FPS on Jetson Xavier. Some qualitative results are shown in Fig.21.

FIGURE 21. Qualitative results of MTSAN 512 × 256 on Jetson Xavier.

2) TEXAS INSTRUMENT TDA2X

Texas Instrument TDA2x evaluation module (EVM) [29] is as shown in Fig. 20 (b) is designed to speed up the development efforts and reduce time to market of the ADAS applications. It is delivered with scalable, highly integrated SoCs consisting of several DSP based accelerators with low-power footprint. The specifications are shown in Table 13. Due to the device and library limitation, we could not implement

TABLE 13. Specifications of texas instrument TDA2x.

Parameters	Specification
MPU	2x ARM Cortex-A15
Cores	2x dual-Cortex-M4, 2x C66x DSP, 4x Embedded Vision Engine(EVE)
Memory	4GB DDR3L
I/O	HDMI, JTAG, Micro-SD, USB …

our MTSAN on it. Instead, we split our models into two separate models for detection and segmentation, respectively. Then, through the model pruning process to reduce the model size and computation, we successfully port two models onto the platform. Although two separate models cannot get the benefit of sharing encoder, the run-time performance can reach almost 15FPS with 512 × 256 input resolution. Some of the qualitative results are shown in Fig.22.

FIGURE 22. Qualitative results of object detection and segmentation predictions on TDA2x.

V. CONCLUSION

In this paper, we have proposed, developed and implemented a Multi-task Semantic Attention Network (MTSAN) to jointly deal with multiple objects detection and the semantic segmentation tasks. The design concepts of each component are introduced. This paper has also proposed an efficient semantic attention module (SAM) to boost up the detection performance by introducing semantic information. The effectiveness of the proposed method is demonstrated on the benchmark datasets, and it is demonstrated that the predictions of MTSAN can be utilized for real-time applications such as lane departure warning, and forward collision warning. The proposed MTSAN network is a lightweight, low computation cost network and achieves 10FPS @ 512 × 256 on the NVIDIA Jetson Xavier and 15FPS @ 512 × 256 on the Texas Instrument TDA2x.

Alongside, we believe that the proposed MTSAN method can be robust to other object detection applications with suitable training and certain modifications corresponding to target applications.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," in *Computer Vision–(ECCV)* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–(MICCAI)* (Lecture Notes in Computer Science), vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[12] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: http://arxiv.org/abs/1606.02147

[13] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3141–3149, doi: 10.1109/CVPR.2019.00326.

[14] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1965–1973, doi: 10.1109/ICCV.2017.215.

[15] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Fast scene understanding for autonomous driving," in *Proc. IEEE Symp. Intell. Vehicles*, Redondo Beach, CA, USA, Jun. 2017.

[16] M. Teichmann, M. Weber, M. Zollner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time joint semantic reasoning for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 1013–1020, doi: 10.1109/IVS.2018.8500504.

[17] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1871–1880, doi: 10.1109/CVPR.2019.00197.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, pp. 1–9.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[23] M. Mathew, K. Desappan, P. K. Swami, and S. Nagori, "Sparse, quantized, full frame CNN for low power embedded devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 328–336, doi: 10.1109/CVPRW.2017.46.
[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.
[25] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," 2018, *arXiv:1805.04687*. [Online]. Available: http://arxiv.org/abs/1805.04687
[26] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015, doi: 10.1007/s11263-014-0733-5.
[27] (2020). *Tensorflow/Models*. GitHub. Accessed: Oct. 17, 2019. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection
[28] (2021). *NVIDIA Jetson AGX Xavier: The AI Platform for Autonomous Machines*. Accessed: Mar. 10, 2020. [Online]. Available: https://www.nvidia.com/zh-tw/autonomous-machines/jetson-agx-xavier/
[29] (2021). *TDA2x Vision EVM Kit–Spectrum Digital (Includes CPU Board and Vision Application Board): Spectrum Digital Inc.–TDA2EVM5777–Third Party Tool Folder*. Ti.com. Accessed: Jul. 18, 2019. [Online]. Available: http://www.ti.com/tool/TDA2EVM5777
[30] M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, and A. Mayr, "Speeding up semantic segmentation for autonomous driving," in *Proc. 29th Conf. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, 2016, pp. 1–7.
[31] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. Accessed: Mar. 15, 2020. [Online]. Available: http://arxiv.org/abs/1511.07122
[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.
[33] G. P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: Bounds on range and range rate accuracy," in *Proc. IEEE IV Intell. Vehicles Symp.*, Columbus, OH, USA, Jun. 2003, pp. 120–125, doi: 10.1109/IVS.2003.1212895.

CHUN-YU LAI was born in Taoyuan, Taiwan, in 1995. He received the Master of Science degree in electronics engineering from the Department of Electronics Engineering, Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2019.

He is currently working as a Software Engineer with Mediatek Inc., Taiwan. His research interests include computer vision, embedded deep learning applications, and deep learning-based object detection. He has received the Academic Excellence Award and the Outstanding Graduate Student Award in 2015 and 2019, respectively.

BO-XUN WU was born in Taipei, Taiwan, in 1997. He received the Bachelor of Science degree in electronics engineering from the Department of Electronics Engineering, Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, in 2019, where he is currently pursuing the master's degree in electronics engineering.

His research interests include computer vision, embedded systems with deep learning applications, and object detection algorithms.

VINAY MALLIGERE SHIVANNA was born in India. He received the Master of Science degree in electronics engineering from the Department of Electronics Engineering, Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan in 2015, where he is currently pursuing the Ph.D. degree in electronics engineering.

He has worked with OnMobile Global Ltd., Bengaluru, India, from 2011 to 2013. His research interests include images, multimedia, and digital signal processing, computer vision, object detection and segmentation, artificial intelligence and deep learning, data augmentation, and SOC design.

JIUN-IN GUO received the B.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1989 and 1993, respectively.

He is currently a Distinguished Professor of the Institute of Electronics, the Associated Dean of electrical and computer engineering, and the Director of Embedded Artificial Intelligence Research Center, National Yang Ming Chiao Tung University, Hsinchu. He was the Director of Institute of Electronics, National Chiao-Tung University, from 2013 to 2015. Before joining in National Chiao-Tung University, he was an Associate Professor of the Department of Computer Science and Information Engineering, National Chung-Cheng University, from 2001 to 2003. He has been promoted as a Professor, since 2003. He also served as the Director for the SOC Research Center, National Chung-Cheng University, from 2005 to 2008, and the Director for the Department of Computer Science, National Chung-Cheng University, Taiwan, from 2009 to 2011, and the Research Distinguished Professor of National Chung-Cheng University in 2008. From 1994 to 2001, he served as an Associate Professor for the Department of Electronics Engineering, National Lien-Ho Institute of Technology, Miaoli, Taiwan. He is also the author of 243 technical articles on the research areas. His research interests include images, multimedia, and digital signal processing, VLSI algorithm/architecture design, digital SIP design, SOC design, and intelligent vision processing applications including ADAS/Self-driving vehicles. He received the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering in 2010, the Outstanding Engineering Professor Award from the Chinese Institute of Engineers in 2014, the Outstanding Research Award from Minister of Science (MOST) in 2017, and the Outstanding Technology Transferring Award from MOST in 2018 with the topic of ADAS system.