Real-Time Memory Efficient Multitask Learning Model for Autonomous Driving

Shokhrukh Miraliev, *Student Member, IEEE*, Shakhboz Abdigapporov, *Student Member, IEEE*, Vijay Kakani, *Member, IEEE*, and Hakil Kim, *Member, IEEE*

***Abstract*—Developing a self-driving system is a challenging task that requires a high level of scene comprehension with real-time inference, and it is safety-critical. This study proposes a real-time memory efficient multitask learning-based model for joint object detection, drivable area segmentation, and lane detection tasks. To accomplish this research objective, the encoder-decoder architecture efficiently utilized to handle input frames through shared representation. Comprehensive experiments conducted on a challenging public Berkeley Deep Drive (BDD100 K) dataset. For further performance comparisons, a private dataset consisting of 30 K frames was collected and annotated for the three aforementioned tasks. Experimental results demonstrated the superiority of the proposed method's over existing baseline approaches in terms of computational efficiency, model power consumption and accuracy performance. The performance results for object detection, drivable area segmentation and lane detection tasks showed the highest 77.5 mAP50, 91.9 mIoU and 33.8 mIoU results on BDD100K dataset respectively. In addition, the model achieved 112.29 fps processing speed improving both performance and inference speed results of existing multi-tasking models.**

***Index Terms*—Multitask learning, edge device, autonomous driving, object detection, drivable area segmentation, lane detection, convolutional neural networks.**

I. Introduction

As MANY areas in computer vision and deep learning are continuously emerging, vision-based tasks (e.g., drivable area segmentation, lane detection, object detection, etc.) in autonomous driving create many challenges in terms of full understanding of the system. A robust panoptic driving perception system that helps the autonomous driving vehicle to achieve a comprehensive understanding of its surrounding environment via camera- or lidar-based sensors has been developed through continuous efforts. For many tasks in autonomous driving, both lidar-based and camera-based sensors are efficiently utilized. However, camera based sensors are preferred for practical applications because of the low-cost of the hardware and software system implementation. High precision and fast inference are considered extremely important for real-time camera-based scene-understanding systems for autonomous driving. In particular, accurate estimations of locations of obstacles and timely decision-makings on the road are key to providing safety. For instance, the accurate prediction of object detection task provides position and size information of obstacles, which handles the process of timely decision-makings while driving. Correctly estimating lane lines as well as drivable areas is also helpful for route planning. For a safe and diversified real-time panoptic driving system, the implementation of multiple single-tasking networks that benefit each other on the road is essential. For instance, the position information of lane lines compliment the task of drivable area segmentation to move the vehicle correctly on the road while avoiding accidents.

Several studies (e.g., YOLOv3 ([1], [2], [3], [4], [5]), SSD [6] and CenterNet [7]) have been proposed to enhance the performance of object detection task utilizing distinctive algorithms and ImageNet pre-trained CNN models. Similarly, common networks (e.g. U-NET [8] and HRNet [32]) that are applied to drivable area segmentation task and lane detection models (e.g. LaneATT [10] and, other networks [11]) have mainly focused on the performance related problems. However, reducing computational cost is vital for autonomous driving applications with limited resources deployed in real-life. ImageNet pre-trained backbone networks for extracting features from a given input image comprise an essential part of these single-tasking CNN architectures. However, feature extraction of multiple backbones is computationally expensive and requires a large amount of resources. To address this issue, researchers have introduced a single end-to-end encoder-decoder network to handle multiple tasks simultaneously.

This approach is proven to be more computationally efficient as the network share computational resources such as one feature extraction backbone and combined loss functions to simultaneously learn multiple objectives. YOLOP [12], HybridNets [13] and the recently released YOLOPV2 [14] output the aforementioned three tasks using a single encoder and three decoder heads. However, the performance of both networks drops when deployed on an edge device as a real-life application while providing approximately 40 fps video inference speed results. This study proposes a fast, memory efficient, and accurate end-to-end neural network architecture to enhance

Manuscript received 6 April 2023; accepted 22 April 2023. Date of publication 27 April 2023; date of current version 23 February 2024. This work was supported in part by the Ministry of Education (MOE, South Korea) through BK21 Four Program and in part by the National Research Foundation of Korea (NRF). *(Corresponding author: Hakil Kim.)*

Shokhrukh Miraliev, Shakhboz Abdigapporov, and Hakil Kim are with the Department of Electical and Computer Engineering, Inha University, Michuhol-gu 22212, South Korea (e-mail: shoxamiraliyev@inha.edu; shakhboz@inha.edu; hikim@inha.ac.kr).

Vijay Kakani is with the Department of Integrated System Engineering, Inha University, Michuhol-gu 22212, South Korea (e-mail: vjkakani@inha.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3270878.

Digital Object Identifier 10.1109/TIV.2023.3270878

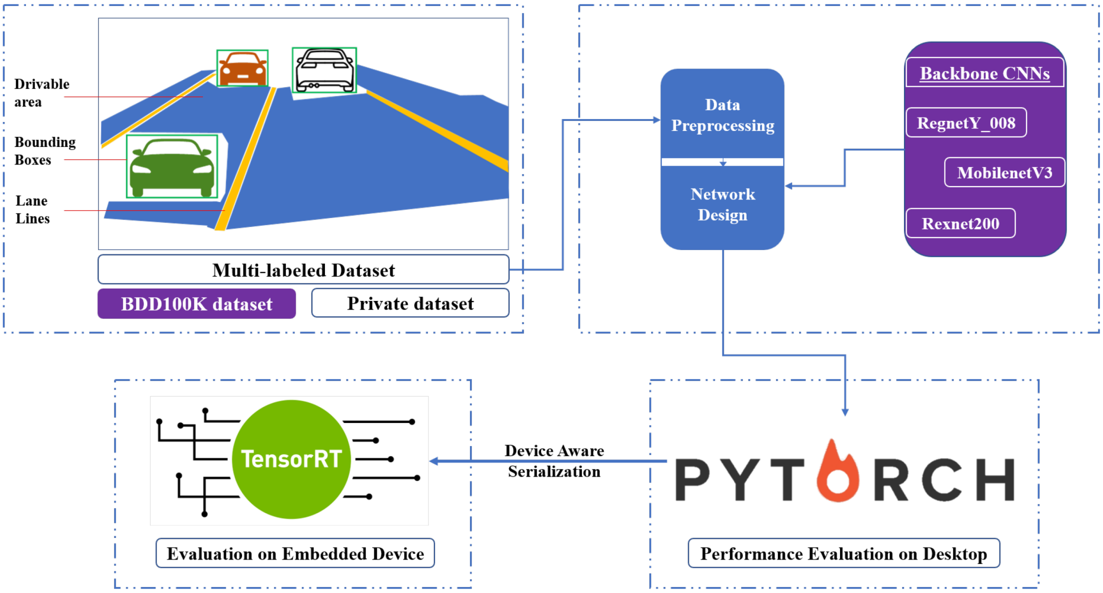

Fig. 1. Overall workflow of the study including multi labeled data preparation, network design, optimization, and performance evaluation.

the simultaneous prediction of objects on the road as well as the estimation of drivable areas and segmentation of lane lines based on a multitask learning approach. The overall real-time system evaluation process on edge devices is shown in Fig. 1. The main objective of this research is to improve the inference of a multi-tasking network on edge devices while maintaining competitive performance results for all three tasks.

To apply all three tasks, shared feature maps were generated from a given input image by the encoder network through a lightweight three-stage feature fusion mechanism and fed into the output-generating decoders. The implemented approach is end-to-end trained on a publicly available Berkeley Deep Drive (BDD100 K) [15] autonomous driving dataset as well as a large scale private dataset and tested on four types of edge devices: Jetson Orin AGX, Jetson Xavier AGX, Jetson Xavier NX and Jetson Nano by efficiently utilizing the TensorRT framework.

In summary, the contributions of this study are as follows:

- Object detection, drivable area segmentation and lane detection were efficiently integrated into a memory efficient end-to-end framework.
- A novel combined loss function is implemented that consists of the weighted average sum of object detection loss, drivable area segmentation loss and lane detection loss for improving the performance accuracy of all tasks.
- The proposed network architecture is energy conserving, memory efficient and can perform multitask inference in real-time when deployed on embedded devices.
- The effectiveness of the proposed method with different training combinations is demonstrated with ablation studies on the publicly available BDD100 K dataset by comparing the performance accuracy results of each implemented task and processing time.

The reminder of the article is organized as follows: Related works are introduced in Section II; the multi-tasking method which includes network architecture is presented in Section III; Section IV includes the conducted experiments as well as the implementation details for the research paper; quantitative and qualitative performance evaluation results are shown in Section V; Section VI consists of ablation studies conducted and finally, Section VII presents the future works and concludes the article.

II. Related Work

A. Single-Task Networks for Autonomous Driving

1) Object Detection: A number of efficiently developed single-task networks are presented each year for the aforementioned tasks (i.e., object detection, semantic segmentation and lane detection). Generally, object detectors are divided into two groups: one-stage detectors and two-stage detectors. One stage detectors focus on mapping the feature maps directly to the classification score and bounding boxes.

The YOLO series (e.g., [1], [2], [3], [4], [5]) is built as a one-stage approach that is applied to the input image to predict the corresponding category as well as the position of the objects. The success of YOLO networks (e.g., YOLOV7 [5]) depends on directly regressing the bounding boxes from the DarkNet [1] CNN backbone model in a more complete unified detector manner. Another widely used approach, the SSD network [6] consists of a base network, typically VGG [16] or ResNet [17] architecture, which is pre-trained on a large image classification dataset such as ImageNet [18]. This base network extracts features from the input image, which are then passed through a series of additional layers, called the “extra layers,” to produce a set of feature maps at different scales. These feature maps are then fed into a set of convolutional layers called “multibox layers,” which predict the object bounding boxes and class probabilities at different scales.

By contrast, two-stage object detectors are called region-based detectors. Such object detection algorithms first propose a set of regions of interest by a regional proposal network or select

TABLE I
BASELINE NETWORKS FOR OBJECT DETECTION TASK

Model	Network	Dataset	mAP50	FPS
RetinaNet	ResNet50-FPN	BDD100K	55.45	13.7
Faster R-CNN	ResNet50-FPN		57.49	14.8
Cascade R-CNN	ResNet50-FPN		57.97	11.5
HRNet	HRNet-w18		58.69	8.4
ConvNeXt	Faster R-CNN		60.10	14.2
Swin-T	Faster R-CNN		60.55	10.8

TABLE II
BASELINE NETWORKS FOR LANE DETECTION AND DRIVABLE AREA SEGMENTATION TASKS

Model	Network	Lane (mIoU)	Drivable (mIoU)	FPS
ENet	Pascal VGG16	34.12	-	57.3
SCNN	SqueezeNet	35.79	-	45.7
ENet-SAD	Pascal VGG16	32.6	-	30.1
HRNet	ResNet50-D8	-	83.67	8.4
HRNet	HRNet-48	-	83.87	8.7
DeeplabV3+	ResNet50-D8	-	84.35	7.0

a search from an input to be classified and refined. The proposed regions are sparse because the potential bounding box candidates can be infinite. The classifier then processes only the region candidates. Faster R-CNN [19], introduces a region proposal network that shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. For the BDD100K [15] dataset, recently released methods (e.g. ConvNeXt [21] and Swin-T [22]) showed better mean average precision (mAP) results than the widely used RetinaNet [23] or Faster R-CNN [19] as listed in Table I.

2) Lane Detection and Drivable Area Segmentation: State-of-the-art approaches that excellently performed on 2017 TuSimple Lane Detection Challenge seek to learn these hand-crafted features in a more end-to-end manner using convolutional neural networks. To avoid clustering, considering left-left, left, right, and right-right lanes as channels of segmentation has been explored [24]. Projecting pixels onto a ground plane via a learned homography is a strong approach for regularizing curve fitting for individual lanes [29]. The latest improvements in lane line segmentation networks were developed by the SCNN [24], ENet [26] and ENet-SAD [27] networks which aimed at the robustness of the task. The research work by Neven et al. [29] proposed an instance segmentation approach with an encoder-decoder architecture that is modified into two branched network to detect lane lines. The decoder in this architecture was used as the backbone of each separate branch.

With an encoder-decoder architecture, SegNet [31] applied semantic segmentation pixel-by-pixel to find drivable areas to reduce memory and computation requirements by implementing a modified VGG16 backbone network as an encoder. Study by Li et al. [40] proposed a road geometric transformation-based data augmentation method for road detection task. The introduced end-to-end method advantages the characteristics of road boundary and multi-task learning of deep convolutional network. Zhang et al. [42] proposed a wearable system with a novel dual-head Transformer for Transparency perception model which can segment general- and transparent objects on the road. Deng et al. [43] addressed 360-degree road scene semantic segmentation problem using surround view cameras. This article proposed Restricted Deformable Convolution based semantic segmentation model. Some semantic segmentation networks such as HRNet [32] and DeepLabV3+ [33] showed state-of-the-art performances on drivable area segmentation task. Table II lists the baseline performance and inference speed results on for both segmentation type tasks. Lee et al. [44] proposed a research study with performance evaluation method of real-time semantic segmentation models to compare the existing methods under the same conditions on edge devices. The study focused on modern semantic segmentation models providing experimental results of power consumption and performance results on four different of edge devices.

B. Multi-Task Networks

Owing to the importance of sharing designated information between multiple tasks, the use of both encoder-decoder structure inheritance and CNN-based methods has been beneficial in previous real-time networks. DLT-Net [35] is a encoder-decoder based multi-tasking network architecture which uses VGG16 pre-trained backbone network and feature pyramid structure such as feed-forward nets. DLT-Net [35] constructs context tensors between sub-task decoders to share information among the tasks. However, simple construction of a feature pyramid structure led to DLT-Net [35] model performing poorly in lane detection task and heavy architecture of the backbone (VGG16) as well as the decoders resulted in low inference speed which prevents the model from being used in real-life applications.

Another multi-tasking network MultiNet [36] was developed to handle object detection and drivable area segmentation tasks utilizing a simple encoder-decoder network. Because the model did not contain feature fusion for fusing the multi-scaled features prior to feeding them into the decoder, lower performance results for object detection task do not allow the MultiNet [36] model to be deployed on edge devices. YOLOP [12] network share one encoder and combines three decoders to solve different tasks. The encoder network in these architectures consists of a backbone that extracts features from an input image and a neck that is used to fuse the features generated by the backbone. The network architecture (with CSPDarkNet backbone and FPN for feature fusion) is more biased to the object detection task sacrificing the performance of the other two implemented segmentation based tasks. Additionally, Lu et al. [41] addressed the issue of isolating switch accurate localization and state recognition simultaneously by proposing a new multi-task learning framework for isolating switch segmentation and state recognition. The proposed model in this study used strip pooling module (ISS-Net) for isolating switch pixel-level segmentation precisely and the segmentation map yielded from the ISS-Net was fed into the isolating switch recognition network (ISR-Net) to recognize in three stages.

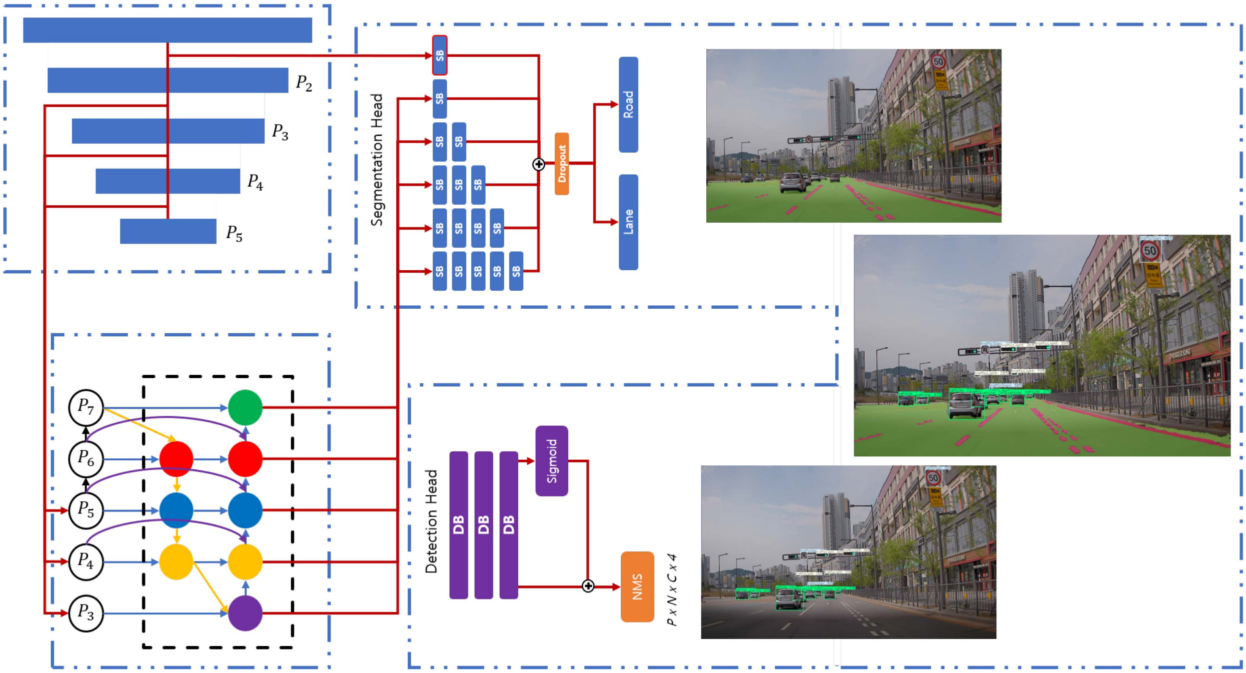

Fig. 2. Proposed model architecture consists of feature extractor (encoder), lightweight feature fusion network and two separate decoders. Encoder is a modified CNN backbone network, multi-scale features are fed into the regression and segmentation type decoders by the feature fusion network.

III. Proposed Method

A. Overall Network Architecture

The overall network architecture of the real-time memory efficient multitask learning model is based on a simple encoder-decoder model with a lightweight feature fusion mechanism for fusing multi-scale feature maps as shown in Fig. 2. Feature extraction is an essential part of the model that helps networks achieve a high performance. The encoder is responsible for extracting features from the image inputs, which are then shared with a three-stage feature fusion network. The resolution of each feature map level is proportional to $1/2^i$ for both width (W) and height (H) of the input image, denoted as P_i. For instance, if the selected input image has an input resolution (W, H) of $(512, 256)$, then P_2 would represent feature level 2 at resolution $(128, 64)$ whereas P_7 would represent the feature level at resolution $(4, 2)$. Recently developed CNN networks (i.e. RegNetY, RexNet and MobileNetV3 Large) that have been pre-trained on ImageNet dataset and have state-of-the-art performance accuracy on ImageNet classification task are utilized as encoders. These design choices create a compact model with fewer parameters, reduced computational complexity, and lower memory requirements, making it suitable for deployment on edge devices with limited resources.

As this study is targeted for testing on edge devices in real-time, CNN networks are selected by considering their model parameters, number of floating-point operations (FLOPs or GFLOPs) as well as flexibility for the output of three tasks with high performance in previous object detection and segmentation studies. Because the multi-tasking network outputs the aforementioned three tasks, CNN networks are selected in accordance with their exceptional performance on each of the tasks, and computational complexity is also considered.

The two specific decoders were used for object detection, lane line prediction, and drivable area segmentation tasks. The regression and classification parts of the proposed architecture are considered for the object detection task. Fused feature maps help locate objects on the input and classify them using classification output. The detector head uses an anchor-based multi-scale detection scheme. Anchor boxes enable deep-learning neural networks to detect signals faster and more efficiently. To determine the anchor boxes, k-means clustering was utilized similar to YOLOv4 [2]. In addition to the position offset as well as height and width scaling, the detection head provides a probability for every category with the predicted confidence for that category.

For lane line prediction and drivable area segmentation, a decoder with different output layers outputs tasks. Segmentation decoder three times up-samples fused feature maps from the feature fusion network and to the size of (W, H, 2) which represents the probability of each pixel for the drivable area and background or lane and background depending on the task.

Overall advantages of proposed model are as followings:

- The proposed multitask learning model is memory efficient and has a small number of parameters, making it suitable for deployment on edge devices with limited computational resources.
- The model achieves state-of-the-art performance on the object detection, lane line prediction, and drivable area segmentation tasks, demonstrating its effectiveness in handling multiple vision-based tasks simultaneously.
- The feature fusion mechanism used in the model allows for efficient feature sharing between the different tasks,

enabling the model to learn from multiple tasks while avoiding overfitting on any one task.

B. Loss Function

Because the network outputs regression and segmentation type tasks (e.g. object detection, lane detection and drivable area segmentation), the loss function for the detection task and segmentation type tasks are implemented. The object detection task utilizes classification loss $L_{classification}$, to improve the prediction confidence, because accurate predictions for object detection on the road highly depend on prediction confidence. Additionally, regression loss $L_{regression}$ was implemented to accurately estimate the distance of the overlap rate, aspect ratio and scale similarity between the ground truth and predicted outputs ([30], [34], [39]). The weighted sum of both losses with α and β tuning parameters was considered for the overall detection loss L_D (1) of the object detection task.

$$L_D = \alpha L_{classification} + \beta L_{regression} \tag{1}$$

The drivable area segmentation task contain a combination of Dice loss (a variant of Dice loss) and a modified cross entropy loss [20]. When the two losses are utilized together, soft Dice loss attempts to leverage the flexibility of class imbalance, and simultaneously cross entropy loss positively affects on curve smoothing.

$$L_{CE} = -\frac{1}{N}\sum_{i}\beta(y - \log(\hat{y})) + (1-\beta)(1-y)\log(1-\hat{y}) \tag{2}$$

where, L_{CE} is a binary cross entropy loss

$$L_{Dice} = 1 - \frac{TP_p(i)}{TP_p(i) + \alpha FN_p(i) + \beta FP_p(i)} \tag{3}$$

where, α and β are parameters that can be tuned for balance. The total combined loss of drivable area segmentation can be defined as follows:

$$L_{DA} = \gamma L_{CE} - (1-\alpha)L_{Dice} \tag{4}$$

For the second lane line segmentation task intersection over union (IoU) loss is proven to be efficient when combined with cross entropy loss (5).

$$L_{IoU} = 1 - \frac{TP}{TP + FP + FN} \tag{5}$$

Combined lane line segmentation loss (6) is defined as follows:

$$L_{Lane} = L_{CE} + L_{IoU} \tag{6}$$

where, LCE is modified cross entropy loss and L_{IoU} is intersection over union loss. Finally, overall network architecture utilizes the weighted sum of all losses together as in (7)

$$L_{Total} = \alpha_1 L_{OD} + \alpha_2 L_{DA} + \alpha_3 L_{lane} \tag{7}$$

where, α_1, α_2, and α_3 are tuning parameters that balance all parts of that total loss.

The proposed loss functions have advantages specific to their respective tasks. For object detection, the combination of classification and regression loss improves prediction confidence and object location prediction. Drivable area segmentation benefits from the combination of modified cross entropy and soft Dice loss, handling class imbalance and curve smoothing. Lane line segmentation improves with the combination of modified cross entropy and IoU loss. The total loss function balances these advantages to create a more robust and accurate network in complex driving scenarios.

IV. Experiments

A. Training Paradigm

Multitask networks ([12], [13], [25], [28], [36]) can be trained using various paradigms. One common approach is to train the network on each task independently and, then fine-tune the shared layers on all tasks jointly. Alternatively, all tasks can be jointly trained from the beginning, leading to faster convergence and better performance. Even if some tasks are unrelated, the model can still learn each task effectively using this paradigm. Algorithm 1, introduces a step-by-step training process that is used to train proposed network.

B. Implementation Details

1) Datasets and Data Processing: All training and testing were conducted using two types of large scale self-driving datasets. BDD100K [37] is the largest publicly available autonomous driving dataset that supports multitask learning related research. The BDD100 K dataset is a video dataset composed of 100 K driving videos. The 10th second frame of each video was annotated for ten different tasks for autonomous driving. The dataset was divided into 70 K train, 10 K validation, and 20 K test sets. As the test set images were not publicly available, evaluations were conducted on the validation set of images.

To evaluate the performance of the model, a private dataset was used which was collected on South Korean roads under different weather conditions in both day and night. The private dataset comprises 30 K frames and is annotated for the output of three tasks (i.e.,object detection, drivable area segmentation and lane line segmentation). To enable the model to be robust in real-time applications, challenging weather conditions such as, foggy, cloudy and rainy were also considered while collecting the dataset. The dataset was divided into 80% train, 10% validation and 10% test set images.

Data augmentation techniques, such as image scaling, translation, random rotation and resizing were utilized to process the images for geometric distortions and increase the variability of the images. The original size of the annotated 100 K images is $1280 \times 720 \times 3$ in the BDD100 K dataset and $1920 \times 1080 \times 3$ for the private dataset. For performance comparisons, the original size of the images on both datasets was resized to $640 \times 384 \times 3$ and $512 \times 256 \times 3$ for the training and testing processes, respectively.

2) Experimental Setting: Performance results were compared with both single-tasking networks that handle drivable

Algorithm 1: This algorithm trains a neural network with multiple heads for object detection, drivable area segmentation and lane detection tasks. The training is performed in three phases, wherein the backbone, fusion, and detection heads are trained in the first phase, followed by the drivable and lane heads in the second phase. In the third and final phase, all heads are jointly trained. The algorithm uses the Adam optimizer in the first two phases and switches to the stochastic gradient descent (SGD) for the final phase. The training process continues until the loss function becomes less than a certain threshold, indicating that the network has converged.

Require: Target neural network with parameters $\mathcal{N}$ group: $\theta = \{\theta_{backbone}, \theta_{fuse}, \theta_{det}, \theta_{drive}, \theta_{lane}\}$; training set: $\mathcal{T}$; threshold for convergence: *thr*; loss: L_{all}; optimizer: η.
Ensure: Well-trained network: $\mathcal{N}(x; \theta)$.

```
procedure TRAIN($\mathcal{N}, \mathcal{T}$)
    while $\ell < thr$ do
        $\ell \leftarrow L_{all}(\mathcal{N}(x_b, \theta); y_b)$
        $\theta \leftarrow arg\ min_\theta\ \ell$
    end while
end procedure
$\theta \leftarrow \{\theta_{backbone}, \theta_{fuse}, \theta_{det}\}$
Freeze $\{\theta_{drive}, \theta_{lane}\}$
$\eta \leftarrow$ Adam optimizer
TRAIN($\mathcal{N}, \mathcal{T}$)
$\theta \leftarrow \{\theta_{drive}, \theta_{lane}\}$
Freeze $\{\theta_{backbone}, \theta_{fuse}, \theta_{det}\}$
TRAIN($\mathcal{N}, \mathcal{T}$)
$\theta \leftarrow \{\theta_{backbone}, \theta_{fuse}, \theta_{det}, \theta_{drive}, \theta_{lane}\}$
$\eta \leftarrow$ SGD optimizer
TRAIN($\mathcal{N}, \mathcal{T}$)
return Trained model $\mathcal{N}(x; \theta)$
```

area segmentation, lane line segmentation or object detection as well as state-of-the-art multi-tasking networks which handled all three tasks simultaneously. Recent multi-tasking models, Multi-Net [36] and DLT-Net have achieved excellent performance results for multiple tasks on the BDD100 K dataset. Because their inference speed is comparatively low on embedded systems, the implemented real-time multi-tasking model's memory efficiency and inference speed were not compared with those of these models on edge devices. By contrast, the more recent multi-tasking networks, HybridNets [13] and YOLOP [12] showed better performance accuracy for object detection and mean intersection over union (mIoU) performance on both segmentation type tasks and were comparatively faster in edge devices.

To compare developed multi-tasking network with single-task networks, one network for each task was selected. For object detection task evaluations Faster R-CNN [19] with FPN mechanism, for drivable area segmentation task performance evaluations HRNet [32] and finally for lane line segmentation task evaluations ENet-SAD [26] models were compared with the proposed multi-tasking model's performance. All experiments on the BDD100 K and private dataset are conducted on an NVIDIA RTX 3060 Ti GPU. The network was trained for 300 epochs with a batch size of eight. The first 200 epochs were trained using the Adam optimizer with a learning rate of 1e-4 and the remaining 100 epochs utilized the Stocastic Gradient Descent (SGD).

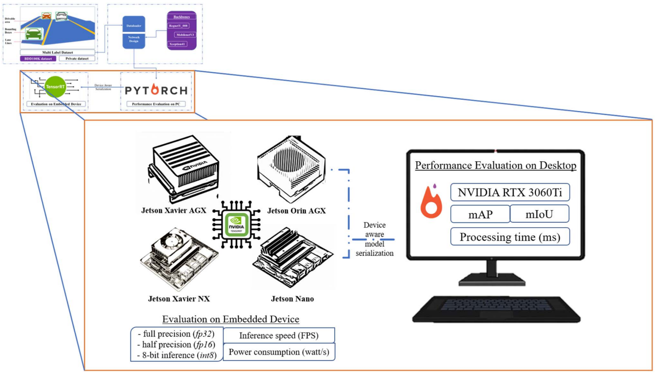

Fig. 3. Flow chart of the proposed network optimization and deployment on embedded devices. Detailed performance evaluation process of network acceleration before and after device aware model serialization on desktop and embedded systems. For evaluation, input sizes of 256 × 512 and 384 × 640. Pytorch and TensorRT frameworks are utilized for experiments on desktop and edge devices, respectively.

3) Edge Device Implementation: The multi-tasking network was deployed on NVIDIA's four types of embedded devices (i.e., Jetson Orin AGX, Jetson Xavier AGX, Jetson XAvier NX and Jetson Nano). Detailed information on the target edge devices are listed in Table III. Three models with different backbone networks were first trained and evaluated on a desktop. After the training, all models were converted through the TensorRT framework as for most real-time applications low-latency is crucial for achieving high performance in the resource constrained environments (e.g. embedded devices and etc.,). TensorRT framework results in a significance speedup of the inference process, reducing the latency and improving the throughput of the system [45].

The evaluation metrics are mAP for object detection task and, mean Intersection over Union (mIoU) for drivable area segmentation as well as lane prediction and processing time is also considered before converting the model to implement on edge devices. After deploying the models, the full precision, half precision, and 8-bit inference performances were compared with the previously developed YOLOP [12] and HybridNets [13] multi-tasking networks.

Additionally, the power consumption of all networks was measured using the Jetson power GUI on all devices. The overall network optimization and embedded device deployment process charts are shown in Fig. 3.

V. Results

A. Quantitative Performance Analysis on Desktop

1) Results Comparison on BDD100 K Dataset: The comparative analysis results of the real-time memory efficient multitask learning model with both, existing baseline single-task and multi-tasking models on the BDD100 K dataset are listed in

TABLE III
DETAILED INFORMATION OF THE TARGET DEVICES

Device	**CPU**	**GPU**	**RAM**	**AI Perf.**	**JetPack**	**Power mode**
Jetson Orin AGX	8-core (Cortex)	1792-core (Ampere)	32GB	200 TOPs	5.0.2	15W/30W/50W
Jetson Xavier AGX	8-core (Cortex)	512-core (Volta)	16GB	32 TOPs	5.0.2	10W/15W/30W
Jetson Xavier NX	6-core (Cortex)	384-core (Volta)	8GB	21 TOPs	5.0.1	10W/15W
Jetson Nano	4-core (Cortex)	128-core (Maxwell)	4GB	472 GFLOPs	4.6	5W/10W

TABLE IV
DESKTOP PERFORMANCE EVALUATION RESULTS ON BDD100K DATASET

Model	**Object detection**		**Drivable area**		**Lane detection**		**Number of parameters**	**MACs**	**Speed**
	mAP	Recall	mIoU	F1-score	mIoU	Accuracy			
Input 384 × 640									
Faster R-CNN [19] + FPN	57.5	60.5	-	-	-	-	65.8 M	9.01 G	14.8 *fps*
HRNet [32]	-	-	83.2	90.1	-	-	28.5 M	8.45 G	8.7 *fps*
Enet-SAD [26]	-	-	-	-	32.4	78.7	0.46 M	0.61 G	30.1 *fps*
MultiNet [36]	60.2	81.3	71.6	82.5	-	-	35.6 M	20.2 G	12.6 *fps*
YOLOP [12]	76.5	89.2	91.5	89.1	26.2	70.5	8.25 M	4.54 G	42.0 *fps*
HybridNets [13]	77.3	**92.5**	90.5	88.8	31.6	**85.4**	4.89 M	**0.91 G**	36.0 *fps*
Ours (MobileNetV3)	75.2	82.7	89.8	90.8	28.8	75.9	**3.98 M**	2.71 G	**49.4** ***fps***
Ours (RegNetY)	**77.5**	86.1	**91.9**	**91.5**	**33.8**	76.9	6.52 M	5.91 G	46.9 *fps*
Input 256 × 512									
Ours (MobileNetV3)	56.4	65.2	85.0	85.1	25.6	69.1	**3.98 M**	1.45 G	63.8 *fps*
Ours (RegNetY)	58.5	69.5	88.3	87.8	25.1	68.0	6.52 M	3.19 G	59.6 *fps*

The bold entities indicate the best/highest performance in each performance metric.

Table IV. The desktop performance of the model's object detection task results was compared with Faster R-CNN [19] network with an additional FPN mechanism. The evaluation metrics of detection accuracy mAP50 and recall were utilized. For drivable area segmentation task evaluation, the mIoU and F1-scores were compared with the HRNet [32] model. For lane detection task comparisons, the Enet-SAD [26] model was selected and the performance evaluation metrics mIoU and accuracy were used. Multinet [36], YOLOP [12] and HybridNets [13] multi-tasking models are also included in performance evaluation comparison table. Multiply-accumulate operations (MACs), number of parameters of models and inference speed metrics are included in the comparative results to evaluate the efficiency.

For the analysis of the results, tiny models which were tested with an image size of 256 × 512 and small models 384 × 640 were evaluated and compared with other methods as listed in Table IV. The object detection results indicated that our model with the RegNetY backbone CNN network has the highest 77.5 mAP result, outperforming Faster R-CNN [19] + FPN, MultiNet [36], YOLOP [12] and HybridNets [13] in terms of detection accuracy. The proposed memory efficient model adopts a scale-aware module that adjusts the object scales in the feature maps to match the scales of the objects in the image, leading to improved localization accuracy. The results suggest the significance of the multi-head mechanism, scale-aware module, and dynamic sampling strategy in improving the performance of object detection.

The drivable area results showed that the introduced memory efficient multitask learning model for autonomous driving with the RegNetY [38] backbone provides the highest mIoU result of 91.9 mIoU and F1-score of 91.5 in comparison with HRNet [32] model for semantic segmentation.

Finally, the lane detection task also achieved the highest 33.8 mIoU performance owing to the combined loss function with the additions of IoU loss. Enet-SAD [26] network is also a single-tasking lane detection model implemented for embedded devices. However, our multi-tasking network showed superiority in mIoU performance and achieving 14.9 fps higher inference speed when trained and tested for multiple tasks. All models tested with both tiny and small input size showed higher inference speed than existing three single-tasking models and all multi-tasking models, with as high as 63.8 fps and as low as 46.9 fps, which is 4.9 fps higher than the fastest YOLOP [12] multi-tasking model on the presented table.

The comparative performance results in Table IV show that both our proposed model and Hybridnets model have strengths and weaknesses. Even though the proposed model showed better accuracy performance in two of the three tasks mentioned (i.e., object detection and drivable area segmentation) with faster inference speed, the performance accuracy for lane detection task was slightly lower. To further evaluate the model's overall performance, new Efficiency Score metric (8) is utilized which takes into account for both the accuracy and processing time.

$$EfficiencyScore = \frac{Performance}{InferenceTime} \tag{8}$$

Table V represents the comparative results of Efficiency Score for related architectures and our proposed memory efficient model. The results indicate that our proposed model achieved higher Efficiency Score than MultiNet, YOLOP and HybridNets models proving that our model is more suitable for practical applications.

2) Results Comparison on Private Dataset: Table VI lists the comparisons of the memory efficient multitask learning model

TABLE V
EFFICIENCY SCORE EVALUATION OF MODELS ON BDD100 K DATASET

Model	OD		Drivable		Lane		Mean
	mAP	Recall	mIoU	F1	mIoU	Acc.	
MultiNet [36]	0.76	1.03	0.91	1.04	-	-	0.94
YOLOP [12]	3.19	3.72	3.81	3.71	1.09	2.94	3.08
HybridNets [13]	2.76	3.30	3.23	3.17	1.13	3.05	2.77
Ours (RegNetY)	**3.69**	**4.10**	**4.38**	**4.36**	**1.61**	**3.65**	**3.63**

The bold entities indicate the best/highest performance in each performance metric.

TABLE VI
DESKTOP PERFORMANCE EVALUATION RESULTS ON PRIVATE DATASET

Model name	OD mAP	Drivable mIoU	Lane mIoU	Number of params	MACs
Input 384 × 640					
YOLOP [12]	46.9	89.0	58.2	8.25 M	4.54 G
HybridNets [13]	47.5	88.3	60.5	4.89 M	**1.78 G**
Ours (RexNet)	45.0	**90.6**	**63.6**	13.8 M	9.41 G
Ours (MobileNetV3)	49.7	87.8	60.1	**3.98** M	2.71 G
Ours (RegNetY)	**51.2**	88.9	62.1	6.52 M	5.91 G
Input 256 × 512					
YOLOP [12]	31.9	86.3	40.1	8.25 M	3.25 G
HybridNets [13]	32.1	85.6	44.5	4.89 M	**0.91 G**
Ours (RexNet)	35.1	**86.8**	**48.6**	13.8 M	5.01 G
Ours (MobileNetV3)	36.1	85.4	46.4	**3.98** M	1.45 G
Ours (RegNetY)	**36.9**	84.1	46.1	6.52 M	3.19 G

The bold entities indicate the best/highest performance in each performance metric.

results with those of the other two multi-tasking models on a private dataset. Because the performance and inference speed results of YOLOP [12] and HybridNets [13] were comparatively higher than those of the tested single-tasking models (i.e., Faster R-CNN [19] + FPN, HRNet [32] and Enet-SAD [26]) and multi-tasking MultiNet [36] model on the BDD100 K dataset, only these two multi-tasking models were selected for the evaluations on the private dataset.

With an input image size of 256 × 512, the object detection and lane detection performance results were higher for all of our models than for multi-tasking models.

For drivable area segmentation, our model with the RexNet encoder network showed the best 86.8 mIoU result. When all models were tested with an input image size of 384 × 640, our model with RegNetY encoder networks showed the highest 51.2 mAP result on the object detection task, where our model with the RexNet backbone showed best 90.6 mIoU Drivable area segmentation task performance and 63.6 mIoU lane detection task performance. The modified combination of soft Dice loss and cross-entropy loss used in the drivable area segmentation and lane detection modules further enhances the accuracy and robustness of the model, enabling it to be suitable for various scenarios, including challenging lighting and weather conditions.

B. Qualitative Performance Analysis

1) On BDD100 K Dataset: The qualitative results of the memory efficient multi-tasking model were compared with existing multi-tasking models as shown in Fig. 4. On the BDD100K [15] dataset, image samples with different photometric scenes (i.e. highway, dirty camera, rain and snow) with two types of light conditions (i.e. day and night) on the road are provided for model evaluations. The first row shows the raw image inputs. As can be seen from Fig. 4, our model with the RegNetY encoder network predicted lane lines more accurately than any other developed models. MultiNet [36] and YOLOP [12] models both failed to detect cars on the side of the road on the first and third input frame respectively. The HybridNets [13] model failed to detect traffic lights and signs on the side of the road as indicated by the red circles on the last row of result images of our model. In addition, our model achieved better lane detection performance results than HybridNets [13] and YOLOP [12] models as shown in the qualitative results. Overall, the qualitative results suggest that the proposed memory efficient multi-tasking model can achieve better performance than other recently implemented multi-tasking models, rendering it a promising solution for real-world applications in autonomous driving and intelligent transportation systems.

2) On Private Dataset: Fig. 5 represents the qualitative performance analysis on the private dataset collected on South Korean roads. Main objective of the qualitative analysis on the privately collected dataset is that most of the publicly available datasets (e.g., BDD100K [15], Cityscapes [46]) can not be generalized for real life applications. To investigate the practical usage of our model in depth, the additional experiments on private dataset were conducted. Proposed memory efficient multi-tasking model showed excellent individual and combined qualitative performance results.

C. Processing Speed Evaluations on Embedded Device

Experiments were conducted to evaluate the processing speed of our model in comparison with other previously developed YOLOP [12] and HybridNets [13] multi-tasking networks in an embedded devices. The models were converted using the TensorRT framework and evaluated using four modes (i.e., FP32, FP16 and INT8) as listed in Table VII. In summary, the experiments showed that the low precision mode helped to increase the processing speed in all devices. Two of our models showed more than 100 fps on the Jetson Orin AGX device and still over 50 fps on the Jetson Xavier NX device using the FP16 and INT8 data types. YOLOP [12] model showed similar results with 98.07 fps using INT8 data type on Jetson Orin AGX device and 55.21 fps on Jetson Xavier NX device.

However, the accuracy performances of the models for the three implemented tasks are lower when 256 × 512 sized input images are utilized as listed in Tables IV and VI. For better performance accuracy and inference speed trade-off, memory efficiency evaluations of our models with 384 × 640 sized input images were obtained. Results indicated that our models can achieve up to 83.72 fps on Jetson Orin AGX embedded device, 52.88 fps on Jetson Xavier AGX device and 46.53 fps on Jetson Xavier NX device using FP16, INT8 and INT8 data types respectively.

The lowest inference speed for all models was recoded on the Jetson Nano. Evaluations with an input image size of 256 × 512

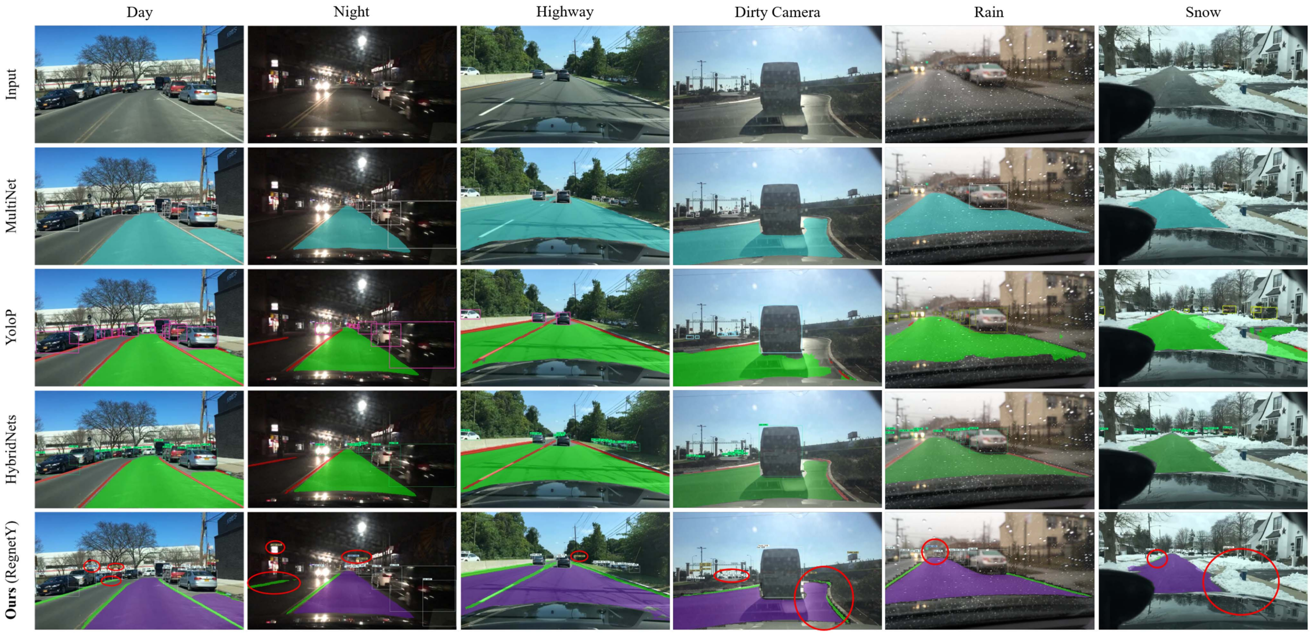

Fig. 4. Qualitative performance comparison of proposed model with related studies on BDD100 K dataset. Input images with different light conditions and scenes are selected to demonstrate the practicality of the proposed model. Highlighted red circles on the results of our model focuses on problematic regions of previous multi-tasking models.

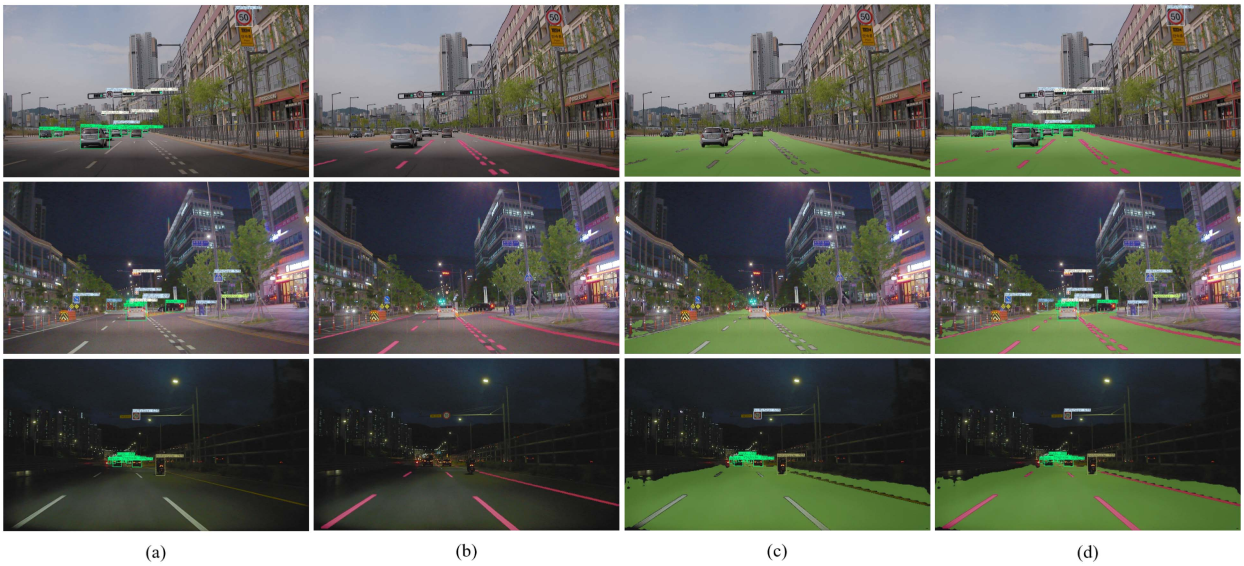

Fig. 5. Qualitative performance analysis on private dataset for the implemented tasks. The results of (a) object detection, (b) lane detection, (c) drivable area segmentation, and (d) combined output is shown.

were achieved over 3 fps for two of our models. Both models recorded faster processing times for both the input image sizes.

D. Energy Consumption

The power consumption of a network model is critical when implementing real-time multitask learning models for embedded devices. Comparative experiments were conducted to measure the power consumption of all tested models of a memory efficient multitask learning network and two related studies (i.e., HybridNets [13] and YOLOP [12]). Figs. 6 and 7 show the average power consumption of each multi-tasking model (the average amount of power used to process an image) with input images size of 384 × 640 and 256 × 512 respectively. The results are used the average values from five repetitive experiments.

The absolute values of the average power consumption differed depending on the embedded device; however, the results showed that our model with RegNetY backbone exhibited the lowest power consumption of all devices. Both of the related multi-tasking models showed higher power consumption than all of our models on NVIDIA's AGX Orin and AGX Xavier devices. On NVIDIA's Xavier NX and Jetson nano devices, two of our models achieved lower power consumption. By comprehensively considering the accuracy performance for all

TABLE VII
COMPARATIVE PROCESSING SPEED EVALUATIONS OF OUR MODELS WITH RELATED STUDIES ON EDGE DEVICES

Model	Jetson Orin AGX			Jetson Xavier AGX			Jetson Xavier NX			Jetson Nano		
	FP32	FP16	INT8	FP32	FP16	INT8	FP32	FP16	INT8	FP32	FP16	INT8
Input 384 × 640												
YOLOP [12]	57.14 *fps*	75.02 *fps*	78.97 *fps*	43.95 *fps*	46.84 *fps*	51.34 *fps*	25.31 *fps*	37.64 *fps*	43.71 *fps*	0.91 *fps*	1.65 *fps*	1.02 *fps*
HybridNets [13]	51.73 *fps*	67.76 *fps*	71.79 *fps*	41.81 *fps*	43.51 *fps*	45.74 *fps*	21.25 *fps*	36.98 *fps*	40.14 *fps*	1.21 *fps*	1.48 *fps*	1.42 *fps*
Ours (RexNet)	44.91 *fps*	60.64 *fps*	63.66 *fps*	26.42 *fps*	29.85 *fps*	36.39 *fps*	17.59 *fps*	28.37 *fps*	40.55 *fps*	-	-	-
Ours (MobileNetV3)	**59.51** *fps*	**79.80** *fps*	**82.22** *fps*	**44.85** *fps*	**49.70** *fps*	**52.88** *fps*	**29.91** *fps*	**43.46** *fps*	**46.53** *fps*	**1.54** *fps*	**1.85** *fps*	**1.33** *fps*
Ours (RegNetY)	53.88 *fps*	83.72 *fps*	79.28 *fps*	41.91 *fps*	46.33 *fps*	47.23 *fps*	24.95 *fps*	38.05 *fps*	44.43 *fps*	1.28 *fps*	1.72 *fps*	1.25 *fps*
Input 256 × 512												
YOLOP [12]	69.26 *fps*	94.31 *fps*	98.07 *fps*	48.28 *fps*	54.07 *fps*	56.41 *fps*	37.11 *fps*	47.35 *fps*	55.21 *fps*	1.83 *fps*	2.35 *fps*	1.98 *fps*
HybridNets [13]	59.88 *fps*	72.34 *fps*	88.71 *fps*	41.73 *fps*	51.39 *fps*	59.34 *fps*	31.81 *fps*	44.91 *fps*	53.17 *fps*	2.11 *fps*	2.79 *fps*	2.34 *fps*
Ours (RexNet)	55.40 *fps*	64.15 *fps*	81.49 *fps*	37.14 *fps*	43.21 *fps*	44.53 *fps*	23.09 *fps*	38.53 *fps*	48.38 *fps*	1.21 *fps*	1.69 *fps*	1.44 *fps*
Ours (MobileNetV3)	**72.26** *fps*	**100.07** *fps*	**112.29** *fps*	46.38 *fps*	53.79 *fps*	64.84 *fps*	**42.59** *fps*	**55.33** *fps*	**60.24** *fps*	**2.87** *fps*	3.04 *fps*	3.41 *fps*
Ours (RegNetY)	68.42 *fps*	97.63 *fps*	103.60 *fps*	**56.74** *fps*	**62.66** *fps*	**65.93** *fps*	39.47 *fps*	50.59 *fps*	57.42 *fps*	2.78 *fps*	**3.10** *fps*	**3.53** *fps*

The bold entities indicate the best/highest performance in each performance metric.

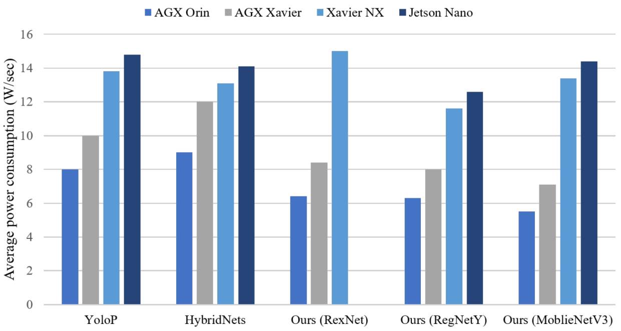

Fig. 6. Comparison of average power consumption of the proposed memory efficient multitask learning models with existing multi-tasking models. Input image size is 384 × 640 for all models provided in the chart.

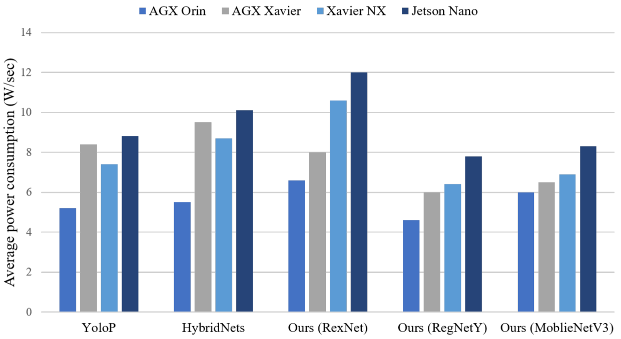

Fig. 7. The average power usage of our memory efficient multitask learning models is compared with existing models, considering a consistent input image size of 256 × 512 for all models in the chart.

three tasks (i.e., object detection, drivable area segmentation and lane detection), inference speed, energy consumption and processing time of our real-time memory efficient multi-tasking model and other recently developed related studies, the experimental results showed that our model with RegNety backbone best suits embedded devices in real life applications.

VI. Ablation Studies

The first ablation experiment was conducted to demonstrate the effectiveness of the different training combinations of our model on the processing time and performance accuracy of each implemented task. Table VIII lists the model evaluation results when the baseline model is trained with six repetitive BiFPN network blocks for feature fusion and cross entropy loss and compared when different optimization techniques are utilized (e.g., changing the loss function, applying data augmentation techniques and replacing the swish activation function with a memory efficient swish (ME)). A notable increase in inference time can be observed when the repetitive feature fusion blocks are reduced where the performance of each implemented method drops. The results showed that the combined loss functions and data augmentation increased the performance accuracy results while maintaining inference speed.

TABLE VIII
DIFFERENT TRAINING COMBINATIONS WITH OPTIMIZATION AND NEW LOSS FUNCTION. N - BASELINE ENCODER-DECODER NETWORK, FF - FEATURE FUSION, CE - CROSS-ENTROPY LOSS, CL - COMBINED LOSS, DA - DATA AUGMENTATION, ME - MEMORY EFFICIENT SWISH

	Detection	**Drivable**	**Lane**	**Inference**
Method	mAP50	mIoU	mIoU	time
N+6xFF+CE	78.0	92.6	34.6	27.6 *ms*
-3xFF	75.3	89.2	32.5	23.1 *ms*
-CE & +CL	75.7	90.8	32.9	23.1 *ms*
+DA	76.5	91.2	33.2	23.1 *ms*
+ME	77.5	91.9	33.8	21.3 *ms*

TABLE IX
MULTITASK VS SINGLE-TASK TRAINING

Model	**Detection**	**Drivable**	**Lane**	**Inference**
name	**mAP**	**mIoU**	**mIoU**	**time**
Detection only	76.3	-	-	14.1 *ms*
Drivable only	-	93.0	-	12.6 *ms*
Lane only	-	-	33.4	12.6 *ms*
Joint training	77.5	91.9	33.8	21.3 *ms*

In addition, the change in the activation function (from swish activation function to memory efficient swish function) lead to a slight increase in the performance accuracy of each task while dropping the processing time results.

Table IX lists the comparisons of the performance and inference speed results of the single-task and multiple-tasks training. To verify the effectiveness of the proposed multitask model, all

TABLE X
MEMORY REQUIREMENTS COMPARISON OF PROPOSED MODEL WITH EXISTING METHODS

Model	CPU Inference		CUDA Inference	
	Mem. alloc.	Run time	Mem. alloc.	Run time
YoloP	1387.57 MB	238 ms	110.52 MB	24 *ms*
HybridNets	1073.82 MB	388 ms	29.87 MB	28 *ms*
Ours (RegNetY)	**953.92 MB**	**131 ms**	**16.46 MB**	**21 *ms***

The bold entities indicate the best/highest performance in each performance metric.

implemented tasks were trained simultaneously and compared with the results of separately trained object detection, drivable area segmentation and lane detection tasks. The results showed that our multitask model saved considerable time compared to executing each task individually on embedded device.

Table X represents the comparative results of the proposed real-time memory efficient multi-tasking model with previously developed multi-tasking models. As can be seen from the table, our model with RegNetY backbone network requires less memory with lowest run time compared to previous YoloP and HybridNets models for both CPU and CUDA inference.

VII. CONCLUSION AND FUTURE WORKS

In this study, a real-time memory efficient multitask learning model for three of the most important tasks in self-driving, object detection, drivable area segmentation and lane detection is proposed. The proposed model can simultaneously handle all three tasks with end-to-end training. Our model showed excellent performance accuracy results for all three tasks and comparatively faster inference speed results than other previously developed multi-tasking models on challenging BDD100 K and privately collected datasets.

All comparative experiments for the memory efficiency and power consumption of our model are conducted on commonly used embedded devices for real-life applications. Our ablation studies showed that our model could be optimized by altering the training schemes. However, this multi-tasking model was limited to only three tasks. Other essential autonomous driving tasks can still be added to our memory efficient multi-tasking model to create complete practical application for self-driving.

REFERENCES

[1] J. Redmon and A. Farhadi, "YoloV3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://doi.org/10.48550/arXiv.1804.02767

[2] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: https://doi.org/10.48550/arXiv.2004.10934

[3] G. Jocher et al., "Ultralytics/Yolov5: V7.0YOLOv5 SOTA realtime instance segmentation," Zenodo, Nov. 2022, doi: 10.5281/zenodo.7347926.

[4] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*. [Online]. Available: https://doi.org/10.48550/arXiv.2209.02976

[5] C. Y. Wang, A. Bochkovskiy, and H. Y. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*. [Online]. Available: https://doi.org/10.48550/arXiv.2207.02696

[6] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: https://doi.org/10.48550/arXiv.1701.06659

[7] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*. [Online]. Available: https://doi.org/10.48550/arXiv.1904.07850

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp 234–241 .

[9] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[10] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 294–302.

[11] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3773–3782.

[12] D. Wu et al., "YoloP: You only look once for panoptic driving perception," *Mach. Intell. Res.*, vol. 19, pp. 550–562, Nov. 2022.

[13] D. Vu, B. Ngo, and H. Phan, "Hybridnets: End-to-end perception network," 2022, *arXiv:2203.09035*. [Online]. Available: https://doi.org/10.48550/arXiv.2203.09035

[14] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, "YoloPv2: Better, faster, stronger for panoptic driving perception," 2022, *arXiv:2208.11434*. [Online]. Available: https://doi.org/10.48550/arXiv.2208.11434

[15] F. Yu et al., "Bdd100 k: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2636–2645.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556.Z.* [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[18] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[19] 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.

[20] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol.*, 2020, pp. 1–7.

[21] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.

[22] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *Proc. IEEE/CVF Int. Conf. Comp. Vis.*, 2021, pp. 9992–10002.

[23] T. Lin, P. Goyal, R.B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE/CVF Int. Conf. Comp. Vis.*, 2017, pp. 2980–2988.

[24] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 7276–7283.

[25] S. Miraliev, S. Abdigapporov, J. Alikhanov, V. Kakani, and H. Kim, "Edge device deployment of multi-tasking network self-driving operations," *arXiv:2210.04735*. [Online]. Available: https://doi.org/10.48550/arXiv.2210.04735

[26] 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: https://doi.org/10.48550/arXiv.1606.02147

[27] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNs by self attention distillation," in *Proc. IEEE/CVF Int. Conf. Comp. Vis.*, 2019, pp. 1013–1021.

[28] S. Abdigapporov, S. Miraliev, J. Alikhanov, V. Kakani, and H. Kim, "Performance comparison of backbone networks for multi-tasking in self-driving operations," in *Proc. IEEE 22nd Int. Conf. Control, Automat. Syst.*, 2022, pp. 819–824.

[29] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 286–291.

[30] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comp. Vis.*, 2019, pp. 9656–9665.

[31] 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.

[32] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[33] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[34] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comp. Vis.*, 2019, pp. 9626–9635.

[35] Y. Qian, J. M. Dolan, and M. Yang, "DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4670–4679, Nov. 2020.

[36] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun, "Multi-Net: Real-time joint semantic reasoning for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1013–1020.

[37] F. Yu et al., "BDD100 K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2633–2642.

[38] I. Radosavovic, R. Kosaraju, R.B. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10425–10433.

[39] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14449–14458.

[40] K. Li, H. Xiong, D. Yu, J. Liu, Y. Guo, and J. Wang, "An end-to-end multi-task learning model for drivable road detection via edge refinement and geometric deformation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8641–8651, Jul. 2022.

[41] X. Lu et al., "A segmentation-based multitask learning approach for isolating switch state recognition in high-speed railway traction substation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15922–15939, Sep. 2022.

[42] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19173–19186, Oct. 2022.

[43] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, "Restricted deformable convolution-based road scene semantic segmentation using surround view cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4350–4362, Oct. 2020.

[44] M. Lee, M. Kim, and C. Y. Jeong, "Real-time semantic segmentation on edge devices: A performance comparison of segmentation models," in *Proc. IEEE 13th Int. Conf. Inf. Commun. Technol. Convergence*, 2022, pp. 383–388.

[45] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 2704–2713.

[46] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

Shokhrukh Miraliev (Student Member, IEEE) received the B.B.A. degree from Inha University, Incheon, South Korea, in 2021, where he is currently working toward the M.S. degree. He is also a dedicated and passionate Researcher of electrical and computer engineering. His research interests include application of deep learning to computer vision and autonomous vehicles. Specifically, he is exploring innovative techniques that enable machines to perceive and analyze visual information in a manner similar to that of human beings, with the goal of enhancing the accuracy and efficiency of computer vision systems.

Shakhboz Abdigapporov (Student Member, IEEE) received the B.B.A. degree from Inha University, Incheon, South Korea, in 2021, where he is currently working toward the M.S. degree in electrical and computer engineering. He is a dedicated and ambitious Researcher of electrical and computer engineering. Throughout his academic career, has developed a strong interest in the application of deep learning techniques to computer vision and autonomous vehicles. His research focuses on developing novel approaches that can improve the accuracy and efficiency of computer vision systems, particularly in challenging real-world scenarios.

Vijay Kakani (Member, IEEE) received the B.S. degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Kakinada, India, in 2012, the M.S. degree in computers and communication systems from the University of Limerick, Limerick, Ireland, in 2014, and the Ph.D. degree in information and communication engineering and future vehicle engineering from Inha University, Incheon, South Korea, in 2020. He is currently an Assistant Professor with the Department of Integrated System Engineering, School of Global Convergence Studies, Inha University. His research interests include autonomous vehicles, sensor signal processing, applied computer vision, deep learning, systems engineering, and machine vision applications.

Hakil Kim (Member, IEEE) received the M.Sc. and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1985 and 1990, respectively. In 1990, he joined the College of Engineering, Inha University, Incheon, South Korea, where he is currently a Full Professor with the Department of Information and Communication Engineering. In order to retain the balance between academic research and commercial development, he founded Vision Inc., in 2014, where he is also the CEO. His research interests include biometrics, intelligent video surveillance, and embedded vision for autonomous vehicles.