sensors

MDPI

Article

Optimal Configuration of Multi-Task Learning for Autonomous Driving

Woomin Jun [1,2], Minjun Son [1,2], Jisang Yoo [1,2] and Sungjin Lee [1,2,*]

[1] Electronic Engineering, Dong Seoul University, Seongnam 13117, Republic of Korea
[2] Autonomous Driving Lab., MODULABS, Seoul 06252, Republic of Korea
* Correspondence: sungjinlee@du.ac.kr

Abstract: For autonomous driving, it is imperative to perform various high-computation image recognition tasks with high accuracy, utilizing diverse sensors to perceive the surrounding environment. Specifically, cameras are used to perform lane detection, object detection, and segmentation, and, in the absence of lidar, tasks extend to inferring 3D information through depth estimation, 3D object detection, 3D reconstruction, and SLAM. However, accurately processing all these image recognition operations in real-time for autonomous driving under constrained hardware conditions is practically unfeasible. In this study, considering the characteristics of image recognition tasks performed by these sensors and the given hardware conditions, we investigated MTL (multi-task learning), which enables parallel execution of various image recognition tasks to maximize their processing speed, accuracy, and memory efficiency. Particularly, this study analyzes the combinations of image recognition tasks for autonomous driving and proposes the MDO (multi-task decision and optimization) algorithm, consisting of three steps, as a means for optimization. In the initial step, a MTS (multi-task set) is selected to minimize overall latency while meeting minimum accuracy requirements. Subsequently, additional training of the shared backbone and individual subnets is conducted to enhance accuracy with the predefined MTS. Finally, both the shared backbone and each subnet undergo compression while maintaining the already secured accuracy and latency performance. The experimental results indicate that integrated accuracy performance is critically important in the configuration and optimization of MTL, and this integrated accuracy is determined by the ITC (inter-task correlation). The MDO algorithm was designed to consider these characteristics and construct multi-task sets with tasks that exhibit high ITC. Furthermore, the implementation of the proposed MDO algorithm, coupled with additional SSL (semi-supervised learning) based training, resulted in a significant performance enhancement. This advancement manifested as approximately a 12% increase in object detection mAP performance, a 15% improvement in lane detection accuracy, and a 27% reduction in latency, surpassing the results of previous three-task learning techniques like YOLOP and HybridNet.

Keywords: autonomous driving; multi-task learning; lane detection; object detection; drivable area segmentation; depth estimation

Citation: Jun, W.; Son, M.; Yoo, J.; Lee, S. Optimal Configuration of Multi-Task Learning for Autonomous Driving. *Sensors* **2023**, *23*, 9729. https://doi.org/10.3390/s23249729

Academic Editor: Felipe Jimenez

Received: 12 November 2023
Revised: 29 November 2023
Accepted: 7 December 2023
Published: 9 December 2023

1. Introduction

Recent technical innovations in deep learning have led to a quantum leap in robot technology and autonomous driving technology [1]. In particular, various sensors, such as cameras, lidar, radar, GPS, ultrasonic waves, and IMUs, are used to acquire and process diverse information related to vehicle situational awareness in order to make driving judgments and control the vehicle [1–3].

However, to apply the information gathered from these various sensors to autonomous driving in real time, the corresponding calculations must be lightweight and accelerated [4–18]. Among these sensors, the tasks that require the highest computation and latency are 2D and 3D context-aware computations, which primarily involve cameras, lidar, and radar.

In studies [6–8,13–15], network weight reduction and acceleration efforts were conducted for camera-based 2D object detection and 2D segmentation calculations. In studies [5,14,15], quantization, pruning, and knowledge distillation methods for light weighting of deep learning were studied. Study [18] performs acceleration research for camera-based lane detection.

However, because all these studies focus on single tasks, the corresponding operations must be combined in a real environment where all of them must be used. For this reason, research on MTL (multi-task learning) was initiated in studies [9–12,19–21], allowing multiple tasks noted above to be performed simultaneously as much as possible. Multi-task learning (MTL) is a learning paradigm in machine learning and its aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks [22]. Therefore, due to the parallel execution characteristics of multi-task learning (MTL), essential image recognition tasks for autonomous driving, such as 2D object detection, lane detection, and drivable area segmentation, were conducted using MTL. However, among these studies on multi-task learning, there has been no research addressing optimal design methodologies for three-task configurations. In fact, most multi-task learning (MTL) studies feature complex structures and intricate training processes, making it challenging to reproduce their performance. Particularly in multi-task learning (MTL), the determination of an optimal combination of components is critical. This includes the shared backbone and its lightweight version, subnets for each task, loss functions dictating subnet training performance, and task-specific optimizers and training details, all of which significantly impact the safety of autonomous driving. From an accuracy perspective, concurrently executing multiple tasks can lead to improper training of the shared backbone weights, potentially degrading each task's performance and adversely affecting the safety of autonomous driving. In terms of latency, if the latency of each task slows down beyond the required threshold, it can prevent the decision-making and control stages of autonomous driving from being executed within an appropriate time frame, leading to potentially severe accidents. Additionally, regarding memory size, if each task consumes an increasing proportion of the limited hardware memory in an autonomous vehicle, it can place additional load on the overall system operations, compromising stability. Therefore, in this study, we experimented with various combinations of these details that determine the performance of each task in MTL and proposed a solution through the MDO (multi-task decision and optimization) algorithm to find the optimal configuration.

2. Related Work

In the field of situational awareness for self-driving technology, it is crucial to execute image recognition tasks with high precision in real time. In particular, information from various sensors should be utilized to enable safe and reliable driving decisions.

Among these, representative image recognition tasks based on cameras include 2D tasks such as object detection, semantic segmentation, and lane detection, as well as 3D tasks such as 3D object detection and 3D segmentation. First, 2D detection studies of [8,23] achieve an accuracy of 52 AP with a performance of over 30 FPS based on a single stage. Recently, anchor free-based technologies, as explored in [24,25], have developed a technique that achieves a performance of 280 FPS or more, while also enhancing accuracy. In the field of 2D semantic segmentation, research studies [26,27] have announced a technology that delivers an accuracy performance of 82.4 mAP. In the field of 3D object detection research, studies based on cameras [28] (18.69% AP), lidar [29] (81.8% AP), and a fusion of camera–lidar sensors [30] (82.4% AP) have been announced. In the field of 3D semantic segmentation, lidar-based research [31] has achieved a performance of 74% mIoU. In the studies by [32,33], acceleration of depth estimation was investigated using only cameras, whereas the research conducted by [34,35] focused on exploring camera-based 3D object detection. The research presented in [36,37] dealt with the acceleration of camera-based 3D reconstruction. Lidar-based 3D object detection and 3D segmentation were the subjects of studies by [38–40]. Finally, the research by [41,42] investigated radar-based 3D object detection.

However, we need to note that the technologies discussed above pertain to studies of individual tasks. In practice, within an autonomous vehicle, when all the corresponding recognition models are loaded and executed simultaneously, there can be complications due to synchronization issues among various technologies and potential system overloads. In other words, even if only some of the various image recognition tasks in autonomous driving meet the accuracy or latency requirements, but others do not, it can negatively impact the safety of autonomous driving.As a result, the exploration of MTL (multi-task learning) was initiated specifically for autonomous driving applications [19–21].

MTL aims to leverage useful information contained within related tasks to enhance the generalization performance of all tasks. MTL can be categorized into five technical approaches based on its characteristics: feature learning approach [43] low-rank approach [44], task clustering approach [45], task relation learning approach [46], and decomposition approach [47]. These approaches are being utilized in various domains of deep learning, including natural language processing [48], reinforcement learning [49], medicine [50], and computer vision [43].

Additionally, in the field of autonomous driving, extensive research is being conducted to improve the performance of related tasks using MTL. In HybridNet, MTL was investigated with respect to three tasks: drivable area segmentation, lane detection, and object detection [19]. Additionally, YOLOP demonstrated potential by enhancing the performance of HybridNet for MTL, focusing on the aforementioned three tasks [20,21].

From the foregoing, it is evident that the accuracy performance of each MTL task is influenced by the efficiency of the underlying backbone network. However, as indicated by the studies referenced in [51,52], the use of a complexly structured backbone network, such as ViT (Vision Transformer) [53], does not necessarily ensure high accuracy across all tasks. This makes achieving the ultimate objective of driving quite challenging. These findings underscore the importance of designing image recognition technology that takes into account the mutually complementary relationship among the relevant technologies.

The principal contributions of this study are as follows:

1. This study proposes an optimal neural network architecture incorporating backbone and loss functions for triple-task learning of drivable area segmentation, object detection, and lane detection. It achieves improvements in all aspects, including accuracy, latency, and size, compared to traditional individual tasks and previous three-task learning methods.
2. The integration of depth estimation within the MTL framework for 2D image recognition was explored, and it was proven to be unsatisfactory due to the low ITC (inter-task correlation).
3. For the performance optimization of MTL, a 3 step MDO algorithm was applied, along with additional training techniques based on SSL (semi-supervised learning). This approach enables enhancements in all aspects, including accuracy, latency, and memory size.

3. System Model

3.1. MTL Architecture

Figure 1 presents a system overview of multi-task learning for autonomous driving. As depicted in Figure 1, tasks such as OD (object detection), DAS (drivable area segmentation), LD (lane detection), DE (depth estimation), and others share a common backbone network model SBM (shared backbone model) denoted as **B**. Subsequently, each individual task utilizes its own dedicated subnet model TSM (task-specific subnet model) denoted as **S**, along with respective loss functions, to derive the final output. Among the networks based on the encoder–decoder structure, UNet [54], FPN [55], Bi-FPN [56], PFPN [26], and Transformer [27] were selected as candidates for SBM. Because all of these networks are based on segmentation tasks, most of them can be shared for tasks such as object detection, segmentation, and depth estimation. As evident from the above, the essence of MTL lies in sharing a common backbone, which serves as a fundamental module or a subset thereof, across diverse image recognition tasks. By sharing the backbone in this manner, the latency required for each multi-task can be reduced, and the accuracy can also

be improved. This is because the shared backbone undergoes learning on diverse data from each multi-task, leading to enhanced performance. However, attempting to share the same backbone for tasks that are too unrelated among these multi-tasks, i.e., low ITC (inter-task correlation), may lead to decreased accuracy for each individual task. This is because the shared backbone cannot be optimized specifically for each task, compromising its performance. In particular, in environments such as self-driving cars, even a minor error in image recognition performance can have a potentially fatal impact on driver safety. Hence, the application of MTL in such scenarios necessitates a cautious approach due to the critical nature of the problem.

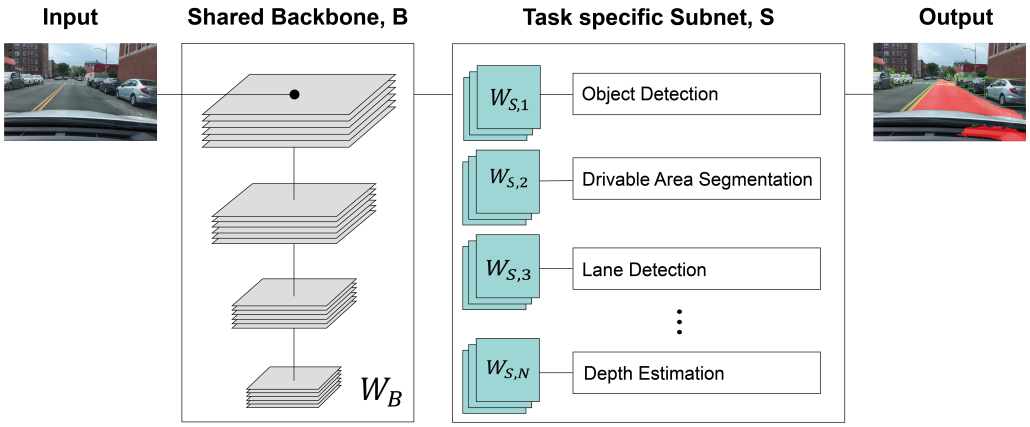

Figure 1. The architecture of multi-task learning for autonomous driving.

3.2. MDO Algorithm

As shown in Figure 2, this paper introduces the MDO (multi-task decision and optimization) algorithm consisting of three steps, designed to optimize accuracy, latency, and size across multiple tasks. The algorithm focuses on minimizing the latency and the size while satisfying the target accuracy considering the target performance of each task. The parameters for the description of the MDO algorithm are defined in Table 1.

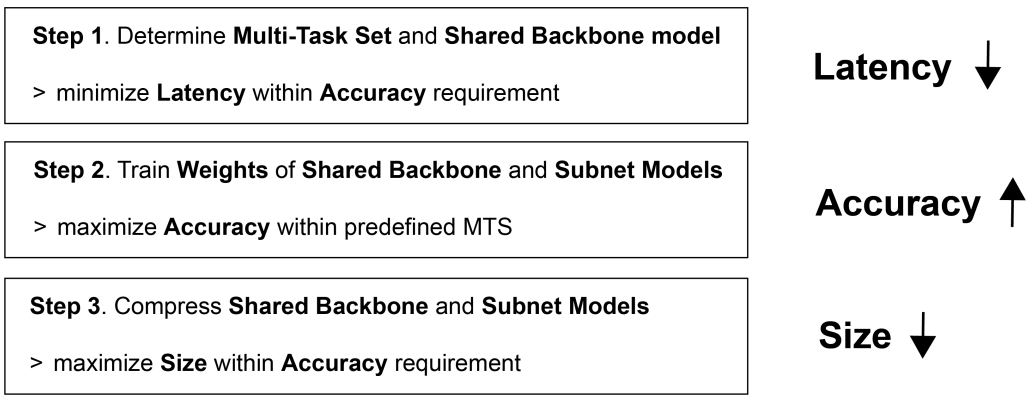

Figure 2. MDO algorithm.

Table 1. Parameters for describing MDO.

Notation	Meaning
$\mathbf{T}_t$	Total task set of image recognition for autonomous driving
$\mathbf{T}_m$	Task set to which MTL is applied, $\mathbf{T}_m \subset \mathbf{T}_t$
t_i	Individual task to which MTL is applied, $t_i \in \mathbf{T}_m$
$\mathbf{T}_{m^c}$	Task sets to which MTL is not applied, $\mathbf{T}_{m^c} = \mathbf{T}_t \setminus \mathbf{T}_m$
t_j	Individual task to which MTL is not applied, $t_j \in \mathbf{T}_{m^c}$
$LM(t_i, \mathbf{B})$	Latency of task t_i with SBM $\mathbf{B}$
$LI(t_j)$	Latency of task t_j
$Acc(t_i)$	Accuracy of task t_i
$Size(t_i, W_B, W_S)$	Size of SBM W_B and TSM W_S for task t_i
γ_i	Accuracy threshold of task t_i
θ_i	Latency threshold of task t_i
$L(t_i)$	Loss function of task t_i for training

- Step 1. Determination of MTS $\mathbf{T}^*_\mathbf{m}$ and SBM $\mathbf{B}^*$ that can minimize total latency with satisfying each accuracy requirement: Based on the weight values pretrained on ImageNet, optimal MTS (multi-task set) $\mathbf{T}^*_\mathbf{m}$ and optimal SBM (shared backbone model) $\mathbf{B}^*$ are determined according to Equation (1). In addition, as optimal MTS $\mathbf{T}^*_\mathbf{m}$ is determined, TSM $\mathbf{S}$ suitable for $\mathbf{T}^*_\mathbf{m}$ is also determined.

$$\begin{aligned} \mathbf{T}^*_m, \mathbf{B}^* \quad &= \quad \arg\min_{\mathbf{T}_m, \mathbf{B}} \{ \sum_{t_i \in \mathbf{T}_m} LM(t_i, \mathbf{B}) + \sum_{t_j \in \mathbf{T}_{m^c}} LI(t_j) \} \\ \text{Subject To} \quad & \quad Acc(t_i) \geq \gamma_i, \;\; t_i \in \mathbf{T}_t, \\ & \quad \mathbf{B} \in \{\textbf{UNet}, \textbf{FPN}, \textbf{BiFPN}, \textbf{PFPN}, \textbf{Transformer}\}. \end{aligned} \tag{1}$$

  where the parameters are defined in Table 1.
  Equation (1) is formulated to select the optimal shared backbone model $\mathbf{B}^*$ and the optimal multi-task set $\mathbf{T}^*_m$, aimed at minimizing the latency of the multi-task set within a multi-task learning framework. For this, each task t_i within the multi-task set must satisfy the accuracy requirement, and each SBM is chosen from UNet, FPN, BiFPN, PFPN, and Transformer. The weights W_B and W_S, being pretrained on ImageNet, do not require additional training. This allows for a swift check of the accuracy conditions for each task t_i in the MTS, thereby selecting the multi-task set that minimizes the overall latency.
  In addition, the accuracy threshold γ_i in Equation (1) is closely linked to the safety and latency in autonomous driving. It plays a crucial role in shaping the overall operational methodology of multi-task learning. First of all, if the goal is to enhance accuracy through MTL, it can be achieved by introducing a new accuracy threshold, $\gamma_i + \delta_i$, where δ_i is added to the existing accuracy threshold, γ_i. Certainly, it should be noted that increasing the accuracy threshold in this manner may result in the non-existence of a feasible solution for the MTS $\mathbf{T}_m$. On the contrary, if the accuracy threshold is decreased to $\gamma_i - \delta_i$, it becomes possible to obtain/larger MTS $\mathbf{T}_m$, thereby reducing the overall system latency. However, this reduction in accuracy threshold may negatively impact the safety of autonomous driving. Based on this observation, it becomes evident that determining $\mathbf{T}_m$ based on Equation (1) introduces a trade-off relationship between safety and speed. Furthermore, because the accuracy threshold γ_i can be customized individually for each task, it provides the flexibility to prioritize specific tasks over others. In other words, by setting a higher threshold for an image recognition task that directly impacts the safety of autonomous driving and a slightly lower threshold for tasks of lesser importance, a viable solution to the problem can be established. These distinct accuracy thresholds also influence the decision-making process of the SBM $\mathbf{B}^*$.
- Step 2. Determination of W^*_B and W^*_S that can further maximize the accuracy of each task within the determined MTS $\mathbf{T}^*_m$:

Using the previously determined MTS $\mathbf{T}_m^*$ and SBM $\mathbf{B}^*$, the optimal weights $\mathbf{W}_B^*$ for the SBM and $\mathbf{W}_S^*$ for the TSM are determined in order to maximize the accuracy across all tasks. Due to the potential variation in accuracy scales and the varying importance of each task, the weights $\mathbf{W}_B$ and $\mathbf{W}_S$ are retrained to maximize the total weighted accuracy sum, incorporating task-specific weights λ_i as depicted in Equation (2),

$$\mathbf{W}_B^*, \mathbf{W}_S^* = \arg\max_{\mathbf{W}_B, \mathbf{W}_S} \{ \sum_{t_i \in \mathbf{T}_m^*} \lambda_i \cdot Acc(t_i, W_B, W_S) \}, \quad (2)$$

$$\approx \arg\min_{\mathbf{W}_B, \mathbf{W}_S} \{ \sum_{t_i \in T_m^*} \lambda_i \cdot L(t_i, W_B, W_S) \}. \quad (3)$$

Here, task-specific weights λ_i are normalized to a total sum of one, $\sum_{t_i \in T_m^*} \lambda_i$, and are proportionally determined for each task based on the respective loss functions, as established through experimentation. Because the accuracy value of Equation (2) can be replaced with a loss function for training, it can be re-derived as in Equation (3) to ensure an optimized allocation of resources across tasks. Additionally, the task-specific loss functions in Equation (3) are detailed in Section 4.

- Step 3. Network compression that can minimize the size of W_B^* and W_S^* while satisfying all accuracy and latency requirements: In Step 3, the memory size of the predetermined optimal weights $\mathbf{W}B^*$ and $\mathbf{W}S^*$ for the shared backbone model and task specific model is minimized. However, this network compression is conducted in a manner that does not compromise the accuracy and latency values obtained in the previous stage. Based on the study by [5], network compression is performed through quantization and pruning to determine the lightweighted W_B and W_S. This task is carefully managed to minimize network size while maintaining the target accuracy, as it can potentially impact accuracy. Based on the research [5], we conduct network compression through quantization and pruning to determine the lightweighted $\mathbf{W}_B^-$ and $\mathbf{W}_S^-$ from $\mathbf{W}_B^*$, $\mathbf{W}_S^*$,

$$\mathbf{W}_B^-, \mathbf{W}_S^- = \arg\min_{\mathbf{W}_B^*, \mathbf{W}_S^*} \{ \sum_{t_i \in \mathbf{T}_m^*} Size(t_i, W_B^*, W_S^*) \}, \quad (4)$$

$$\text{Subject To} \quad Acc(t_i) \geq \gamma_i, \ t_i \in \mathbf{T}_m^*,$$

$$LM(t_i, \mathbf{B}) \geq \theta_i, \ t_i \in \mathbf{T}_m^*.$$

According to [5], most quantization techniques rely on a quantization table, which can lead to latency loss due to value reference time. However, FP16, which performs quantization by merely truncating decimal values from the original FP32 values, is the only method that can achieve quantization without latency loss. Pruning can also achieve network compression without impacting latency. However, unlike FP16, pruning requires additional training. Moreover, the accuracy can be compromised depending on the training technique used, thus necessitating careful application of this method. Considering these factors, this study prioritizes the application of FP16, aiming to achieve network compression without sacrificing the accuracy and latency gains achieved in the previous phase.

4. Subnet for Multi-Task Learning

In this section, we focus on identifying the TSM (task-specific subnet model) **S** for MTL. In particular, the neural network structures and loss functions for the mentioned tasks, i.e., object detection, drivable area segmentation, lane detection, and depth estimation are defined.

4.1. OD (Object Detection)

For object detection, we employ the multi-head subnet structure of FPN [55] based RetinaNet [8] as a subnet. Additionally, the loss functions used for each multi-head branch are the focal loss and regression loss as

$$L_{OD} = L_{Foc} + L_{Reg}.$$

First, focal loss is a cross-entropy-based loss function designed to address the class imbalance problem and is defined as follows:

$$L_{Foc} = -\alpha_t \cdot (1 - p_t)^{\gamma} \cdot \log(p_t),$$

where α_t is a weighting factor for each class, which helps to balance the importance of positive/negative examples, p_t is the model's estimated probability for the class with label 1 (ground truth), and γ is a focusing parameter. A higher value of γ dampens the loss contribution from easy examples and increases the influence of hard examples. In this paper, α_t is equally set to $1/N$ for all classes, and γ is set to 2.

Regression loss L_{Reg} is a loss function introduced for bounding box regression. It uses Smooth $L1$ loss (Huber loss) and is defined as follows:

$$L_{Reg} = \begin{cases} 0.5 \times (\delta y)^2 & \text{if } |\delta y| < 1, \\ |\delta y| - 0.5 & \text{otherwise,} \end{cases}$$

where δy represents the difference between the predicted value and the ground truth for each aspect of the bounding box (e.g., center coordinates, width, and height).

The accuracy index utilizes AP (average precision) for each major class and mAP (mean average precision) for all classes:

$$AP = \int_0^1 p(r)dr, \quad mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i,$$

where $p(r)$ is the precision at recall r, N is the number of classes. and AP_i is the AP for the ith class.

4.2. DAS (Drivable Area Segmentation)

For DAS (drivable area segmentation) tasks, the subnet depending on the each semantic segmentation based backbone [26,27,51,54–56] is primarily utilized. In terms of the loss function, Dice loss, Tyversky loss, and BCE loss are selectively employed based on their performance.

BCE (binary cross-entropy) loss L_{BCE} is based on pixel-wise classification, that is, each pixel in an image is classified as either belonging to the foreground class or the background class,

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

where N is the total number of pixels in the image, y_i is the ground truth label for the ith pixel, which is 1 if the pixel belongs to the foreground and 0 if it belongs to the background, p_i is the predicted probability that the ith pixel belongs to the foreground class.

The Dice loss L_{Dice} utilizes the DC (Dice coefficient), a measure used to quantify the degree of overlap between two sets. This coefficient computes the extent of overlap between the predicted area and the actual ground truth, normalizing it to a decimal value less than 1. The Dice loss L_{Dice} is then derived by calculating the difference from 1,

$$DC = \frac{2 \times |P \bigcap G|}{|P| + |G|} = \frac{2 \times \sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2}, \quad L_{Dice} = 1 - DC,$$

where p_i refers to the predicted probability of pixel i belonging to the foreground class, g_i is the ground truth label for pixel i, which is 1 for foreground and 0 for background, and N is the number of total pixels in the predicted and ground truth images.

Tyversky loss L_{Tyv} is a generalization of the Dice loss, providing more flexibility in handling false positives and false negatives:

$$L_{Tyv} = 1 - \frac{\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i g_i + \alpha \sum_{i=1}^{N} p_i(1 - g_i) + \beta \sum_{i=1}^{N} (1 - p_i) g_i + \epsilon},$$

where p_i is the predicted probability of pixel i belonging to the foreground class, g_i is the ground truth label for pixel i, which is 1 for foreground and 0 for background, α and β are weights to control the relative importance of false negatives and false positives, respectively, and ϵ is a small constant (like 1×10^{-5}) added for numerical stability; N is the number of pixels in the predicted and ground truth images.

The accuracy metric is based on the accuracy calculated from segmentation mask as the measure of performance.

4.3. LD (Lane Detection)

For lane detection, the lane area was derived by treating it as a branch of segmentation, in a similar way as DAS. In recent lane detection studies UFLD [18] and CLRNet [57], row anchor-based approaches have demonstrated superior accuracy performance. However, as these approaches necessitate additional backbones, subnets, and post-processing steps, they result in increased latency. To minimize latency and leverage the potential of the existing backbone and subnet, a segmentation-based technology was employed in this study.

Similar to the case of DAS (drivable area segmentation), the accuracy metric for this task is based on the accuracy calculated from the segmentation mask, serving as the performance measure.

4.4. DE (Depth Estimation)

A subnet was designed based on MonoDepth [58], which is an FPN based depth estimation technology.

SigLoss (scale-invariant gradient loss) L_{Sig} and BerhuLoss L_{Berhu} [59] were employed as loss functions, with the more effective loss function value being selected and utilized based on experimental outcomes.

SigLoss L_{Sig} is used to ensure that the estimated depth maps maintain correct local structures and gradients relative to the ground truth. This loss is particularly useful for preserving edge information and relative depth differences, regardless of the absolute scale:

$$L_{Sig} = \frac{1}{N} \sum_{i=1}^{N} \left(\Delta d_i^{pred} - \Delta d_i^{true} \right)^2 \tag{5}$$

where N is the total number of pixels, and Δd_i^{pred} and Δd_i^{true} are the gradients (spatial derivatives) of the predicted and true depth values at pixel i, respectively. The sum of squared differences in gradients across all pixels is calculated and normalized by the number of pixels.

BerhuLoss L_{Berhu} is a loss function that combines the properties of $L1$ for small errors and $L2$ losses for larger errors:

$$L_{Berhu} = \begin{cases} |y - \hat{y}| & \text{for } |y - \hat{y}| \leq c \\ \frac{(y-\hat{y})^2 + c^2}{2c} & \text{for } |y - \hat{y}| > c \end{cases} \tag{6}$$

where y is the true value (in this case, the true depth), $\hat{y}$ is the predicted value (the estimated depth), and c is a threshold that determines the switch between the $L1$-like and $L2$-like behavior.

For accuracy metrics, the *REL* (absolute relative error) values for depth information of each pixel were utilized.

$$REL = \frac{1}{N}\sum_{i=1}^{N}\frac{|d_i^{pred} - d_i^{true}|}{d_i^{true}}, \tag{7}$$

where N is the total number of pixels (or points) for which the depth is being estimated, d_i^{pred} is the predicted depth for the ith pixel (or point), and d_i^{true} the ground truth depth for the ith pixel (or point).

5. Simulation Results

The evaluation of various tasks are conducted under the umbrella of MTL, analyzing them both individually and in integrated configurations, ranging from single-task scenarios to combinations of up to four tasks. Furthermore, the results were benchmarked against previous MTL techniques, notably YOLOP [20] and HybridNet [19]. This comparison was extended to traditional OD (object detection) strategies, such as RetinaNet [8], LD (lane detection) methods such as UFLD [18] and CLRNet [57], DE (depth estimation) such as DepthFormer, and DAS (drivable area segmentation) approaches utilizing architectures like FPN, PFPN, BiFPN, and Transformer (SegFormer) mentioned in Section 3.

To assess the MDO algorithm, the optimal multi-task set T_m^*, SBM B, and TSM S are determined within parameters exceeding the targeted accuracy of 95 % for LD and DAS, surpassing the targeted mAP of 0.80 for OD, and falling below the targeted absolute REL (relative error) of 0.06 for DE. Subsequently, the accuracy and latency performances of these optimized sets are evaluated.

The BDD 100K and the KITTI datasets in [60,61] were employed for training and evaluation. Specifically, whereas the BDD 100K dataset contains labels for DAS, such labels are absent in the KITTI dataset. To address this, SSL (semi-supervised learning) was applied to the KITTI dataset for additional training. More precisely, a pseudo label was created using an InterImage model [51] pretrained on Cityscapes [62], which was then utilized to apply semi-supervised learning for the DAS task. Experiments were executed using implementations in TensorFlow, facilitated by an NVIDIA GPU equipped with a 2-way 4090 architecture. A piecewise constant decay strategy was adopted for the learning rate schedule. Model performances were assessed across a span of 50 epochs, with the most optimal outcome within this range being chosen for further analysis. The AdamW was employed as the optimizer algorithm. Each task-specific loss value in MTL was trained through the summation of loss values derived in Equation (3), using the functions mentioned in Section 4. The weight λ_i for each loss function was set to 2 for object detection and maintained at 1 for the remaining tasks.

For the performance analysis of each task (OD, LD, DAS, and DE) in MTL, experimental groups were set up with a dedicated 1-task model, a 2-task model (DAS+LD, DAS+OD, OD+LD, DE+DAS) and a 3-task model (OD+LD+DAS), and their respective performances were compared. Table 2 shows the performance of DAS, Table 3 presents the performance of OD, Table 4 illustrates the performance of LD, and Table 5 presents the performance of DE. Additionally, Figures 3–5 provide visual examples of the application of these 2-task and 3-task scenarios.

Based on the results of all experimental groups presented in Tables 3–5, it is evident that the applications of depth estimation are insufficient for ensuring safe autonomous driving. As evidenced by Tables 2 and 5, the results for the 2-task (DAS + DE) setup indicate that DAS does not meet its target accuracy of 95%, and, similarly, DE falls short of the target REL of 0.06. In contrast, other experimental sets excluding DE, such as 1-task, as well as 2-task and 3-task configurations, generally satisfy their target performance.

This can be attributed to the fact that tasks such as DAS and LD have high ITC, leading to their backbone weights being trained to exhibit similar distributions, which in turn enhances their collective performance. Conversely, tasks like OD and DE have less ITC, resulting in them being trained with different backbone weight distributions, ultimately leading to a mutual degradation in performance.

Therefore, it can be inferred that tasks for multi-task learning can be readily trained to assist each other in improving accuracy, whereas some MTS configurations may not offer such benefits. Consequently, constructing an MTS with such complementary tasks is instrumental in enhancing the safety of autonomous driving. Additionally, the multi-task learning examples presented in this study reveal that operating with only three tasks—OD, LD, and DAS—excluding DE, provides a more secure and efficient approach to securing an autonomous driving image recognition model.

Table 2. Performance results of MTL for drivable area segmentation task.

1 Task (DAS)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.93	0.93	0.95	0.95	0.95		
Perf. Req.	X	X	O	O	O		
Lat (ms)	24	24	25	25	60		
2 Tasks (DAS + LD)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.92	0.94	0.95	0.95	0.95		
Perf. Req.	X	X	O	O	O		
Lat (ms)	26	26	27	27	62		
2 Tasks (DAS + DE)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.59	0.62	0.66	0.68	0.71		
Perf. Req.	X	X	X	X	X		
Lat (ms)	31	31	32	32	61		
2 Tasks (DAS + OD)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.92	0.94	0.95	0.95	0.92		
Perf. Req.	X	X	O	O	X		
Lat (ms)	26	26	27	28	63		
	3 Tasks (DAS + LD + OD)					**Prev 3 Tasks**	
Model	UNet	FPN	BiFPN	PFPN	TRN	YOLOP	HybridN
Best Loss	Dice	Dice	Dice	Dice	Dice	Tyv	Tyv
ACC	0.90	0.91	0.95	0.95	0.90	0.97	0.91
Perf. Req.	X	X	O	O	X	O	X
Lat (ms)	30	30	31	32	72	44	61

Table 3. Performance results of MTL for object detection task.

1 Task							
Metric	UNet	FPN	BiFPN	PFPN	TRN		
mAP	0.83	0.88	0.88	0.88	0.88		
Lat (ms)	26	25	25	25	61		
2 Tasks (OD + DAS)							
Metric	UNet	FPN	BiFPN	PFPN	TRN		
mAP	0.82	0.85	0.87	0.86	0.88		
Lat (ms)	26	26	27	28	63		
2 Tasks (OD + LD)							
Metric	UNet	FPN	BiFPN	PFPN	TRN		
mAP	0.82	0.86	0.87	0.86	0.87		
Lat (ms)	26	26	26	28	62		
	3 Tasks (OD + LD + DAS)					**Prev 3 Tasks**	
Metric	UNet	FPN	BiFPN	PFPN	TRN	YOLOP	HybridN
mAP	0.81	0.86	0.87	0.87	0.85	0.76	0.77
Perf. Req.	O	O	O	O	O	X	X
Lat (ms)	30	30	31	32	72	44	61

Table 4. Performance results of MTL for lane detection task.

1 Model			**1 Task (LD)**				
Metric	UFLD	CLRNet	UNet	FPN	BiFPN	PFPN	TRN
Best Loss	Dice	Dice	Dice	Dice	Dice	Dice	Dice
mAP	0.98	0.99	0.95	0.97	0.98	0.97	0.98
Lat (ms)	10	11	24	24	25	25	60
2 Tasks (LD + DAS)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.96	0.96	0.98	0.97	0.95		
Lat (ms)	26	26	27	27	62		
2 Tasks (LD + OD)							
Model	UNet	FPN	BiFPN	PFPN	TRN		
Best Loss	Dice	Dice	Dice	Dice	Dice		
ACC	0.93	0.96	0.98	0.95	0.90		
Lat (ms)	26	26	26	28	62		
	3 Tasks (LD + DAS + OD)					**Prev 3 Tasks**	
Model	UNet	FPN	BiFPN	PFPN	TRN	YOLOP	HybridN
Best Loss	Dice	Dice	Dice	Dice	Dice	Tyv	Tyv
ACC	0.89	0.94	0.98	0.98	0.86	0.70	0.85
Perf. Req.	X	X	O	O	X	X	X
Lat (ms)	30	30	31	32	72	44	61

Table 5. Performance results of MTL for depth estimation task.

1 Model	**1 Task (DE)**					
Metric	DepthF	UNet	FPN	BiFPN	PFPN	TRN
Best Loss	Berhu	Berhu	Berhu	Berhu	Berhu	Berhu
REL	0.0528	0.122	0.098	0.096	0.095	0.074
Perf. Req.	O	X	X	X	X	X
Lat (ms)	37	24	24	25	55	60
2 Tasks (DE + DAS)						
Model	UNet	FPN	BiFPN	PFPN	TRN	
Best Loss	Berhu	Berhu	Berhu	Berhu	Berhu	
REL	0.182	0.167	0.155	0.154	0.116	
Perf. Req.	X	X	X	X	X	
Lat (ms)	31	31	32	32	61	

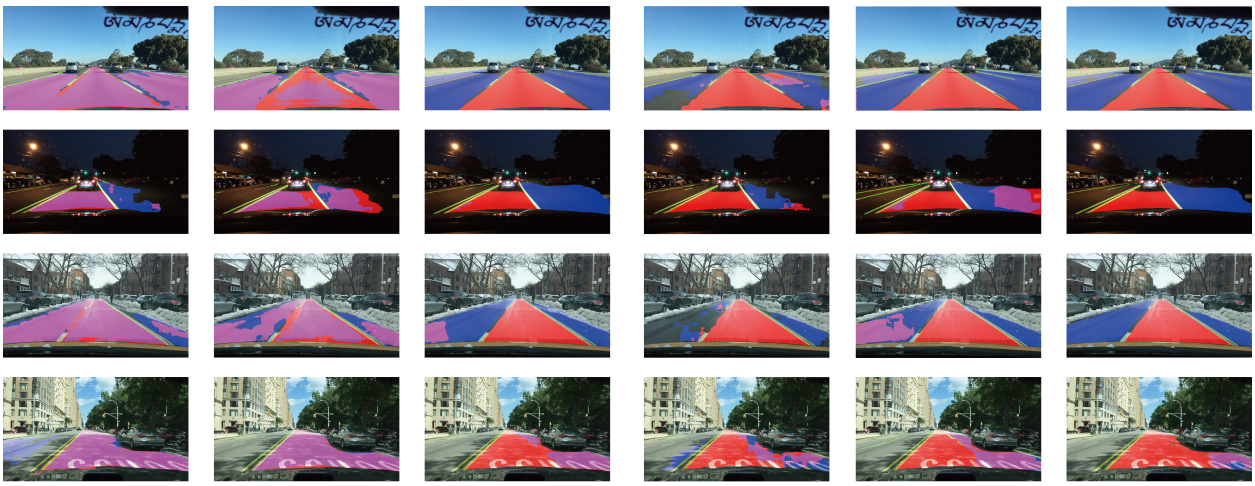

Figure 3. The example results of two tasks.

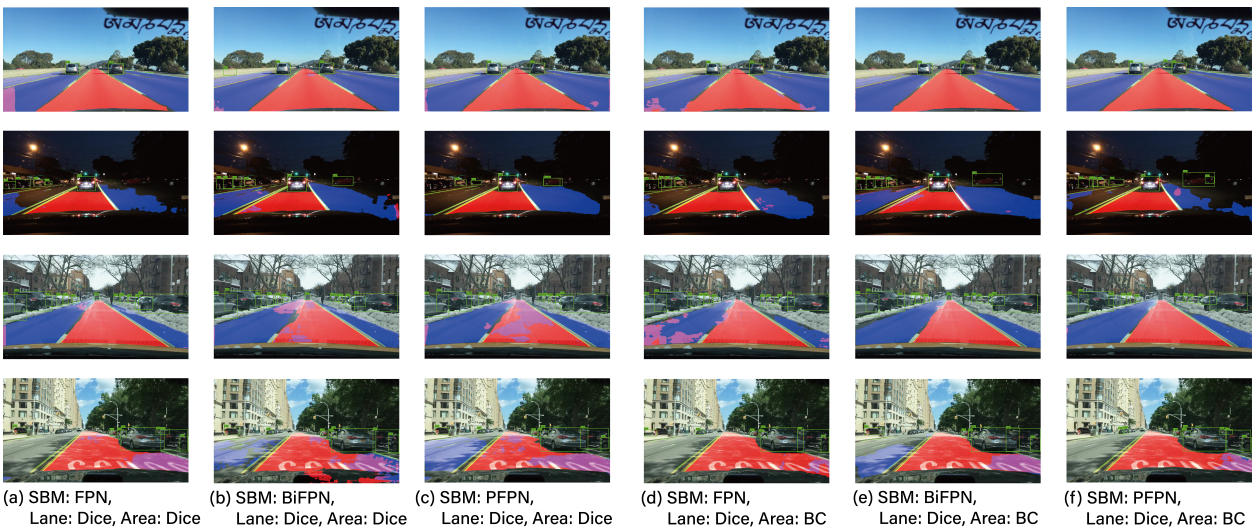

Figure 4. The example results of three tasks.

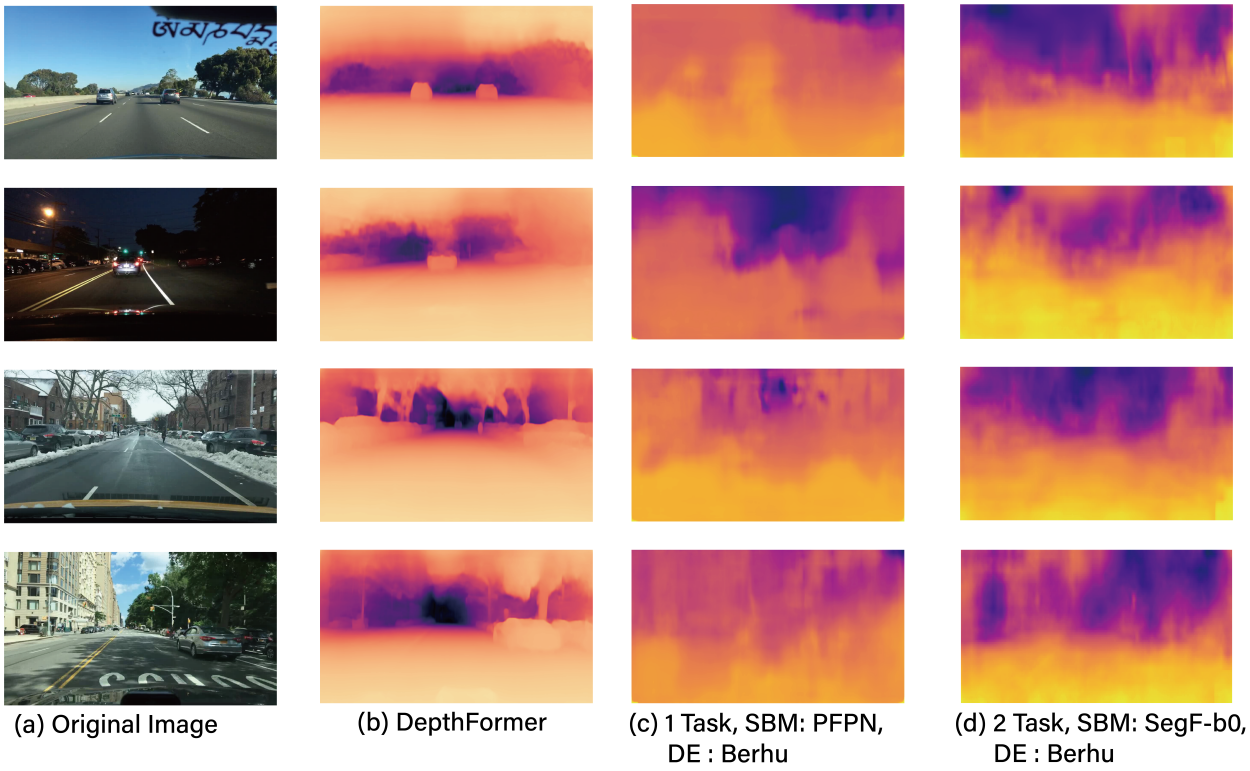

Figure 5. The example results for depth estimation.

Moreover, although the backbone models generally exhibit similar performance, it is noteworthy that in the 3-task configuration, BiFPN demonstrates the best performance. This surpasses even the Transformer-based SegFormer and PFPN, which have the highest number of parameters. This suggests that for the KITTI dataset of DAS, OD, and LD tasks, the BiFPN model, with fewer parameters than the SegFormer and PFPN, is less prone to overfitting and offers better generalization effects.

Furthermore, it can be observed that this parallel processing approach in multi-task learning offers significant advantages in terms of latency. As indicated in Tables 2–4, for the 2-task configuration, there is an approximate 50% reduction in latency, while for the 3-task setup, the latency reduction effect can reach around 60% compared to the individual task learning.

Table 6 selectively compares the system load of 3-task learning in experimental sets that meet the performance requirements of each task. The results presented are after the application of step 3 of the MDO algorithm, which is TensorFlow-Lite based FP16 quantization [63]. The rationale for employing FP16 quantization is that, compared to other quantization techniques, it incurs the least accuracy loss and can halve the memory size, while having no impact on operational latency [5]. This demonstrates that the application of the MDO algorithm can achieve optimal adjustments suitable for autonomous driving in terms of accuracy, latency, and memory size.

Next, let us delve into an analysis of the loss functions utilized for the DAS and LD tasks, as presented in Tables 2 and 4. Traditionally, in image segmentation problems, the binary cross-entropy function is predominantly employed. However, to address class imbalance issues, the Dice and Tversky functions are utilized [52]. As can be discerned from Figures 3 and 4, the proportion of the foreground area is relatively small compared to the entirety of the image. Consequently, based on the results presented in Tables 2 and 4, the proposed DAS and LD techniques demonstrate superior performance with the Dice function rather than the BC. Notably, conventional methods such as YOLOP and HybridNet also employ a similar Tversky function as their loss function. Given this context, it would be prudent to utilize the Dice or Tversky loss functions in autonomous driving applications, taking into account the dimensions of the foreground areas.

Table 6. System load of triple-task learning after implementing step 3 of the MDO algorithm (target accuracy of 95% for LD, DAS, and target mAP of 0.80 for OD).

	3 Tasks (DAS + OD + LD)					**Prev 3 Tasks**	
Model	UNet	FPN	BiFPN	PFPN	TRN	YOLOP	HybridN
Perf. Req.	X	X	O	O	X	X	X
Parameters (Mega)	4.3	3.8	3.5	4.9	3.8	7.9	12.8
Base (MB)	522	456	425	597	466	91	54
Compressed (MB)	97	76	70	99	77	-	-
Lat (ms)	30	30	31	32	72	44	61

From the aforementioned results, it can be observed that the REL of the dedicated model for the DE task, i.e., DepthFormer, demonstrates superior performance compared to other experimental groups. It is imperative to note that even the singular task configuration showcases a suboptimal REL metric, which further deteriorates when subjected to multi-task operational paradigms, as exemplified by the 2-task model. From the foregoing analysis, it becomes apparent that implementing depth estimation via MTL frameworks is suboptimal, given the intrinsically low degree of ITC between the DE task and other associated tasks. Furthermore, as elucidated in [64], the particular task under consideration is not optimally aligned for integration within MTL frameworks. This is attributed to its heightened dependence on supplementary operations external to the backbone structure (e.g., T-Net), rather than on the primary backbone task. This aspect categorically renders it as a technically misaligned group for MTL applications. Additionally, an examination of the performance metrics associated with the 2-task (DAS+DE) configuration, as detailed in Table 2, reveals a concurrent degradation in the performance of DAS, another task intricately connected with DE. This observation underscores the inappropriateness of sharing a common backbone between the DE and DAS tasks. However, as depicted in Table 6, the dedicated model approach, exemplified by DepthFormer, necessitates the utilization of additional parameters compared to the MTL methodology, resulting in augmented costs for securing the requisite resources. Consequently, a comprehensive consideration of both the additional resource costs and accuracy is imperative when determining the application of MTL to DE.

6. Conclusions

This study explores MTL for maximizing the efficiency of various image recognition tasks performed in autonomous driving, considering the task characteristics and the given hardware conditions. Additionally, MDO algorithm, an optimal configuration algorithm for this purpose, is proposed. The MDO algorithm targets drivable area segmentation, object detection, lane detection, and depth estimation as the tasks for recognition, and is comprised of three stages: minimizing latency, maximizing accuracy, and minimizing size. Through the MDO algorithm, an optimal neural network design including the backbone and loss functions is achieved. Additional training based on the SSL led to improvements in all aspects—accuracy, latency, and size—compared to traditional single-task methods and existing three-task learning approaches. The experimental results reveal that integrated accuracy performance is crucial in the configuration and optimization of MTL, and this integrated accuracy is determined by the ITC. Considering these characteristics, it was proven important to design multi-task sets comprising tasks with high ITC. The proposed MDO algorithm facilitated approximately a 12% improvement in object detection mAP performance, a 15% enhancement in lane detection ACC, and a 27% reduction in execution time. Additionally, it has been found that depth estimation has a low ITC with tasks such as drivable area segmentation, object detection, and lane detection. Forming a multi-task set with these tasks could potentially lead to mutual performance degradation. Therefore, to achieve stable performance in depth estimation, it is concluded that it should

be implemented either through a dedicated independent neural network or conducted using additional sensors like lidar. In the future, research is planned to extend MTL based on sensor fusion, not only through single-sensor inputs from cameras but also incorporating inputs from lidar sensors. This expansion aims to enhance the currently limited performance of depth estimation. Additionally, the scope of research will be extended to encompass the entire process of perception, decision-making, and control in autonomous driving, achieving an end-to-end learning approach. This will facilitate both horizontally and vertically integrated optimization in the field.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; software, W.J., M.S. and J.Y.; validation, W.J., M.S. and J.Y.; formal analysis, S.L.; investigation, S.L. and J.Y.; resources, S.L.; data curation, W.J., M.S. and J.Y.; writing original draft preparation, S.L.; writing review and editing, S.L.; visualization, S.L., W.J., M.S. and J.Y.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education and Brian Impact, a non-profit organization dedicated to the advancement of science and technology.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results

References

1. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A Survey of Deep Learning Techniques for Autonomous Driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
2. Károly, A.I.; Galambos, P.; Kuti, J.; Rudas, I.J. Deep Learning in Robotics: Survey on Model Structures and Training Strategies. *IEEE Trans. Syst. Man Cybern.* **2021**, *51*, 266–279. [CrossRef]
3. Kwak, D.; Yoo, J.; Son, M.; Choi, D.; Lee, S. Rethinking Real-Time Lane Detection Technology for Autonomous Driving. *J. Korean Inst. Commun. Inf. Sci.* **2023**, *48*, 589–599. [CrossRef]
4. Bae, E.; Lee, S. Efficient Training Methodology in an Image Classification Network. *J. Korean Inst. Commun. Inf. Sci.* **2021**, *46*, 1087–1096. [CrossRef]
5. Lee, H.; Lee, N.; Lee, S. A Method of Deep Learning Model Optimization for Image Classification on Edge Device. *Sensors* **2022**, *22*, 7344. [CrossRef]
6. Liu , W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016. [CrossRef]
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
8. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. [CrossRef]
9. Lee, D. Fast Drivable Areas Estimation with Multi-Task Learn-ing for Real-Time Autonomous Driving Assistatnt. *Appl. Sci.* **2021**, *11*, 10713. [CrossRef]
10. Ishihara, K.; Kanervisto, A.; Miura, J.; Hautamäki, V. Multi-task Learning with Attention for End-to-end Autono-mous Driving. In Proceedings of the 2021 IEEE/CVF Con-ference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 2896–2905. [CrossRef]
11. Teichmann, M.; Weber, M.; Zöllner, M.; Cipolla, R.; Urtasun, R. MultiNet: Real-time Joint Semantic Reasoning for Au-tonomous Driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Chang-shu, China, 26–30 June 2018; pp. 1013–1020. [CrossRef]
12. Guo, J.; Wang, J.; Wang, H.; Xiao, B.; He, Z.; Li, L. Research on Road Scene Understanding of Autonomous Vehicles Based on Multi-Task Learning. *Sensors* **2023**, *23*, 6238. [CrossRef]
13. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; An-dreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Net-works for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
14. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted Residuals and Linear Bottle-necks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogni-tion (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
15. Howard, A.; Sandler , M.; Chen, B.; Wang, W.; Chen, Li.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for Mo-bileNetV3. In Proceedings of the IEEE/CV FInternational Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]

16. Lee, Y.; Moon, Y.; Park, J.; Min, O. Recent R&D Trends for Lightweight Deep Learning. *Electron. Telecommun. Trends* **2019**, *34*, 40–50. [CrossRef]
17. Liu, Z.; Sun, M.; Zhou, T.; Huang, G. Trevor Darrell Rethinking the value of network pruning. *arXiv* **2018**, arXiv:1810.05270. [CrossRef].
18. Qin, Z.; Wang, H.; Li, X. Ultra Fast Structure aware Deep Lane Detection. In Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020. [CrossRef]
19. Vu, D.; Ngo, B.; Phan, H. HybridNets: End-to-End Perception Network. *arXiv* **2022**, arXiv:2203.09035v1.
20. Wu, D.; Liao, M.; Zhang, W.; Wang, X.; Bai, X.; Cheng, W.; Liu, W. YOLOP: You Only Look Once for Panoptic Driving Perception. *Mach. Intell. Res.* **2022**, *19*, 550–562. [CrossRef]
21. Han, C.; Zhao, Q.; Zhang, S.; Chen, Y.; Zhang, Z.; Yuan, J. YOLOPv2: Better, Faster, Stronger for Panoptic Driving Perception. *arXiv* **2022**, arXiv:2208.11434. [CrossRef].
22. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5586–5609. [CrossRef]
23. Du, X.; Lin, T.Y.; Jin, P.; Ghiasi, G.; Tan, M.; Cui, Y.; Le, Q.V.; Song, X. SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization. In Proceedings of the 2020 Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]
24. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430v2.
25. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *arXiv* **2023**, arXiv:2304.00501.
26. Kirillov, A.; Girshick, R.; He, K.; Dollár, P. Panoptic Feature Pyramid Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]
27. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the Advances in Neural Information Processing Systems 34, NeurIPS 2021, Virtual, 6–14 December 2021. [CrossRef]
28. Hong, Y.; Dai, H.; Ding, Y. Cross-Modality Knowledge Distillation Network for Monocular 3D Object Detection. In Proceedings of the 17th European Conference, ECCV 2022, Tel Aviv, Israel, 23–27 October 2022. [CrossRef]
29. Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; Li, H. PV-RCNN++: Point-Voxel Feature Set Abstraction with Local Vector Representation for 3D Object Detection. *arXiv* **2021**, arXiv:2102.00463. https://doi.org/10.48550/arXiv.2102.00463.
30. Kim, Y.; Park, K.; Kim, M.; Kum, D.; Choi, J. 3D Dual-Fusion: Dual-Domain Dual-Query Camera-LiDAR Fusion for 3D Object Detection. *arXiv* **2022**, arXiv:2211.13529. [CrossRef].
31. Lai, X.; Chen, Y.; Lu, F.; Liu, J.; Jia, J. Spherical Transformer for LiDAR-based 3D Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, 17–24 June 2023. [CrossRef]
32. Sada, Y.; Soga, N.; Shimoda, M.; Jinguji, A.; Sato, S.; Nakahara, H. Fast Monocular Depth Estimation on an FPGA. In Proceedings of the IPDPSW 2020, New Orleans, LA, USA, 18–22 May 2020; pp. 143–146. [CrossRef]
33. Zhang, J.; Yang, H.; Ren, J.; Zhang, D.; He, B.; Cao, T.; Li, Y.; Zhang, Y.; Liu, Y. MobiDepth: Real-time depth estimation using on-device dual cameras. In Proceedings of the MobiCom'22: Proceedings of the 28th Annual International Conference on Mobile Computing and Networking, Sydney, NSW, Australia, 17–21 October 2022; pp. 528–541. [CrossRef]
34. Wang, Y.; Chao, W.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 2019, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
35. You, Y.; Wang, Y.; Chao, W.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; Weinberger, K. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. In Proceedings of the ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020. [CrossRef]
36. Mildenhall, B.; Srinivasan, P.; Tancik, M.; Barron, J.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In Proceedings of the ECCV 2020, Glasgow, UK, 23–28 August 2020. [CrossRef]
37. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. In Proceedings of the SIGGRAPH 2022, Vancouver, BC, Canada, 7–11 August 2022. [CrossRef]
38. Qi, C.; Su, H.; Mo, K.; Guibas, L. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 2017, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
39. Lang, A.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection from Point Clouds. In Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 2019, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
40. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 2019, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
41. Nobis, F.; Shafiei, E.; Karle, P.; Betz, J.; Lienkamp, M. Radar Voxel Fusion for 3D Object Detection. *Appl. Sci.* **2021**, *11*, 5598. [CrossRef]
42. Nabati, R.; Qi, H. CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection. In Proceedings of the WACV 2021, Waikoloa, HI, USA, 3–8 January 2021. [CrossRef]
43. Lapin, M.; Schiele, B.; Hein, M. Scalable multi-task representation learning for scene classification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1434–1441. [CrossRef]

44. Yuan, X.; Yan, S. Visual classification with multi-task joint sparse representation. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3493–3500. [CrossRef]
45. Cheng, B.; Liu, G.; Wang, J.; Huang, Z.; Yan, S. Multi-tasklow-rank affinity pursuit for image segmentation. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2439–2446. [CrossRef]
46. An, Q.; Wang, C.; Shterev, I.; Wang, E.; Carin, L.; Dunson, D.B. Hierarchical kernel stick-breaking process for multi-task image analysis. In Proceedings of the ICML '08: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008. [CrossRef]
47. Hong, Z.; Mei, X.; Prokhorov, D.V.; Tao, D. Tracking via robust multi-task multi-view joint sparse representation. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 649–656. [CrossRef]
48. Zhang, Z.; Yu, W.; Yu, M.; Guo, Z.; Jiang, M. A Survey of Multi-task Learning in Natural Language Processing: Regarding Task Relatedness and Training Methods. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 3–5 May 2023.
49. Vithayathil Varghese, N.; Mahmoud, Q.H. A Survey of Multi-Task Deep Reinforcement Learning. *Electronics* **2020**, *9*, 1363. [CrossRef]
50. Sergey, S.; Mariia, V.; Michael, W.; Pavel, K.; Maxim, F.; Igor, V.T. A Survey of Multi-task Learning Methods in Chemoinformatics. *Mol. Inform.* **2019**, *38*, e1800108. [CrossRef]
51. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; Wang, X.; Qiao, Y. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, 17–24 June 2023 . [CrossRef]
52. Kwak, D.; Choi, J.; Lee, S. A Method of the Breast Cancer Image Diagnosis Using Artificial Intelligence Medical Images Recognition Technology Network. *J. Korean Inst. Commun. Inf. Sci.* **2023**, *48*, 216–226. [CrossRef]
53. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the ICLR 2021, Vienna, Austria, 4 May 2021. [CrossRef]
54. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the MICCAI 2015, Munich, Germany, 5–9 October 2015. [CrossRef]
55. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
56. Tan, M.; Pang, R.; Le Quoc, V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]
57. Zheng, T.; Huang, Y.; Liu, Y.; Tang, W.; Yang, Z.; Cai, D.; He, X. CLRNet: Cross Layer Refinement Network for Lane Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022. [CrossRef]
58. Godard, C.; Aodha, O.; Firman, M.; Brostow, G. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 17, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
59. Carvalho, M.; Saux, B.L.; Trouvé-Peloux, P.; Almansa, A.; Champagnat, F. On Regression Losses for Deep Depth Estimation. In Proceedings of the 2018 IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018. [CrossRef]
60. Yu, F. BDD100K: A Large-Scale Diverse Driving Video Database. Available online: https://bair.berkeley.edu/blog/2018/05/30/bdd/ (accessed on 30 November 2023).
61. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
62. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
63. Available online: https://www.tensorflow.org/lite/performance/model_optimization (accessed on 9 December 2023).
64. Agarwal, A.; Arora, C. Depthformer : Multiscale Vision Transformer For Monocular Depth Estimation With Local Global Information Fusion. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022. [CrossRef]