IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 October 2023, accepted 14 October 2023, date of publication 18 October 2023, date of current version 23 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325392

RESEARCH ARTICLE

Adaptive Momentum-Based Loss Rebalancing for Monocular Depth Estimation

WON-GYUN YU[1] AND YONG SEOK HEO[1,2]
[1]Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, South Korea
[2]Department of Artificial Intelligence, Ajou University, Suwon 16499, South Korea

Corresponding author: Yong Seok Heo (ysheo@ajou.ac.kr)

This work was supported in part by the Brain Korea 21 (BK21) FOUR Program of the National Research Foundation of Korea through the Ministry of Education under Grant NRF5199991014091; in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2022R1F1A1065702; and in part by the Ministry of Science and Information and Communications Technology (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program, supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2023-2018-0-01424.

ABSTRACT Monocular depth estimation in outdoor scenes presents significant challenges due to ambiguity from occlusions and structural variations. One important challenge lies in effectively incorporating loss functions while considering the distribution of ground truth pixels and structural variations of the scene. The utilization of conventional loss functions, such as scale-invariant loss and gradient loss without considering contribution of each loss in relation to the structural variation of the scene may lead to suboptimal outcomes. To solve this problem, we propose an Adaptive Momentum-based Loss Rebalancing (AMLR) to balance loss functions for monocular depth estimation in outdoor scenes. Our method utilizes the scale-invariant loss and gradient loss, with the proposed balancing term inspired by traditional weight optimizer, Adam. By dynamically updating the loss weights using momentum and considering the increase and decrease of individual losses, we facilitate convergence of the total loss and consequently obtain more accurate results. We observed the gradient loss with an appropriate weight serves the role of assistant to the overall loss convergence. Experimental results on the KITTI benchmark demonstrate that our approach achieves performance comparable to state-of-the-art, achieving an absolute relative difference of 0.049. This work contributes to advancing the field of monocular depth estimation in challenging outdoor scenes.

INDEX TERMS Monocular depth estimation, Adam, loss rebalancing, scale-invariant loss, gradient loss.

I. INTRODUCTION

Estimating accurate depth information has become an essential requirement in numerous computer vision tasks, owing to the growing challenges associated with understanding and analyzing 3D space. Conventionally, spatial information including depth has been acquired through point cloud data generated by Light Detection and Ranging (LIDAR) sensors. Extensive research has been conducted to handle this information, primarily leveraging point cloud data [1], [2]. However, due to high cost of LIDAR sensors and substantial memory usage of point cloud data, generating per-pixel depth prediction map only using monocular image becomes a very attractive task. Although monocular depth estimation is an attractive task, there exist several challenges that need to be addressed. The problem is fundamentally ill-posed, and particularly in outdoor scenes, accurately predicting depth maps is significantly more challenging due to the presence of various structural variations and occlusions compared to indoor scenes.

Due to these challenges, the utilization of proper loss functions, reflecting the characteristics of outdoor scenes has emerged as one of the important problems to be solved. Previously, many monocular depth estimation methods, particularly for outdoor scenes, have been proposed relying solely on the utilization of scale-invariant loss [3]. Conversely, the utilization of the gradient loss [4] has been predominantly limited due to the lack of ground truth

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

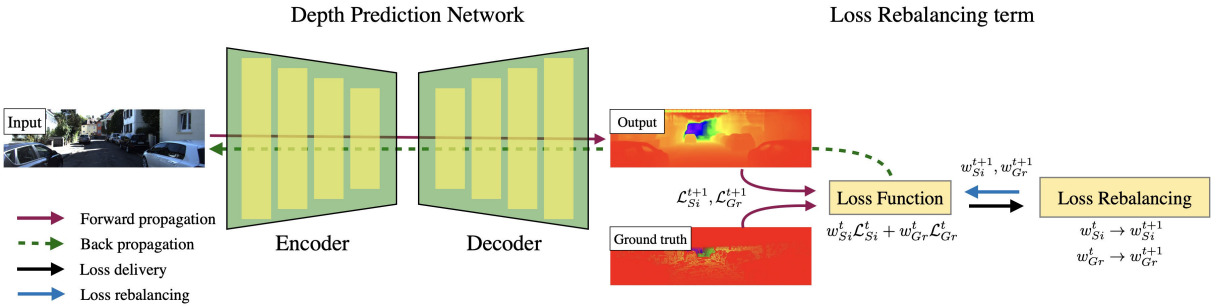

FIGURE 1. Overview of the proposed Adaptive Momentum-based Loss Rebalancing (AMLR). Loss rebalancing is performed using loss function which is computed using output of depth prediction network and ground truth value. The total loss function is dynamically reweighted at each rebalancing step, while ensuring no interference with the backpropagation process.

values and its misalignment with the objective of increasing accuracy. The gradient loss is more sensitive to the absence of a ground truth value compared to the scale-invariant loss because the absence of a single ground truth value hinders the computation of gradient loss for a pair of pixels, while hindering the computation of scale-invariant loss for a single pixel. In addition, the gradient loss compares the relative difference of adjacent pixels, which means it disregards the objective of minimizing the absolute value difference between the ground truth value and the prediction. However, by appropriately weighting gradient loss with scale-invariant loss, they can serve as valuable supplements to the overall loss and enhance the performance of the entire network. Balancing the weights assigned to each loss is a critical component of this process, enabling the allocation of proper weights for each loss.

Recently, loss rebalancing method between several loss functions in indoor depth estimation was proposed by Lee and Kim [7]. However, this method is applicable primarily to indoor scenes, because it applies various loss functions without consideration of effect of missing ground truth value. Furthermore, the rebalancing method merely considers the relative proportions of each loss in the total loss, without accounting for the tendencies of individual loss functions to increase or decrease.

To handle these problems, we propose an Adaptive Momentum-based Loss Rebalancing method (AMLR), with applying the gradient loss [4] and scale-invariant loss [3]. By integrating the advantages of the Adam optimizer [8], our proposed method effectively reflects the past changes of individual loss functions, thereby enabling adaptive rebalancing. With this term, application of gradient loss with an appropriate weight in outdoor scenes can facilitate the convergence of the loss based on per-pixel differences and enhance the generalization capabilities of the overall network. We additionally applied a masking technique to pixels for which gradient computation is not feasible, including missing pixels and those with zero disparity. These masked pixels were then averaged with the number of computable pixels to facilitate more accurate gradient computation. Our proposed rebalancing method can be employed in other tasks involving multiple loss functions without the need for explicit modifications to the loss function or network architecture.

Our contributions can be summarized as follows:

- We applied a classical optimizer to the loss rebalancing term, thereby enhancing the flexibility and suitability of the loss rebalancing process according to predefined conditions.
- We masked pixels for which we cannot compute the gradient, thus enabling more accurate computation of gradient loss.
- Our method achieves comparable results to state-of-the-art performance for monocular depth estimation on KITTI benchmark both quantitatively and qualitatively. Specifically, we have achieved an absolute relative error of 0.049, surpassing the performance of the previous state-of-the-art (SOTA) methods [6], [9], [10].

II. RELATED WORK

Depth estimation has emerged as a crucial task in various domains, including self-driving, virtual reality (VR), and augmented reality (AR). In prior approaches, depth estimation was predominantly achieved through the utilization of stereo cameras, capitalizing on principles rooted in epipolar geometry. In light of the recent advancements in deep learning and computer vision techniques, numerous methodologies have been introduced, harnessing the potential of stereo cameras for depth estimation [11]. Concurrently, over the past decade, there has been a substantial body of research focused on monocular depth estimation, a technique accomplished using a single camera. These researches have primarily focused on investigating network architectures, loss functions, training methodologies, and other related areas. In this section, we focus on recent deep learning based monocular depth estimation methods, which can be divided into two categories: supervised monocular depth estimation and self-supervised monocular depth estimation. We also discuss about loss functions and their rebalancing methods at the end of the section.

A. SUPERVISED MONOCULAR DEPTH ESTIMATION

Monocular depth estimation using a supervised approach can be regarded as both classification and regression problem.

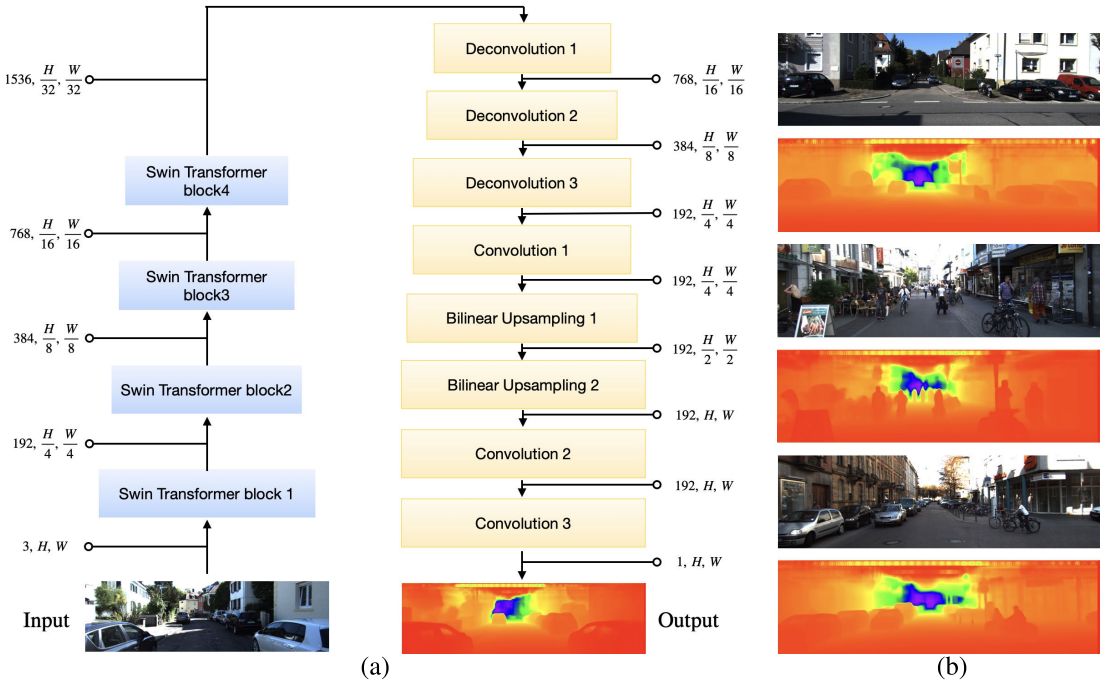

FIGURE 2. Illustration of the network architecture in (a) and multiple depth prediction results obtained using our network in (b). SiLU activation [5] is applied after each Deconvolution and Convolution layer in the Decoder. The Encoder utilizes a pretrained Swin Transformer v2 [6].

Eigen et al. [3] approached the depth estimation problem as a regression task. They trained two separate networks, one for coarse estimation and the other for fine estimation. The outputs of these networks are combined in the final stage to incorporate both coarse and fine perspectives. On the other hand, Fu et al. [12] and Bhat et al. [13] tackled the depth estimation as a classification task. Fu et al. [12] discretized the depth values into bins and employed ordinal regression techniques to predict the depth category. Similarly, Bhat et al. [13] discretized the depth into bins, but with an adaptive binning strategy tailored to the specific domain. Kim et al. [14] proposed selective feature fusion module to reflect both global and local features of the scene. They also introduced vertical cutdepth, data augmentation method for depth estimation, which is based on the understanding that incorporating vertical context plays a vital role in accurate depth prediction [15]. Recently, the training methodology utilizing masked image modeling (MIM) on Swin Transformer v2 [16], as demonstrated by Xie et al. [6], has exhibited remarkable performance in dense prediction tasks such as semantic segmentation and depth estimation.

B. SELF-SUPERVISED MONOCULAR DEPTH ESTIMATION

On the other hand, a self-supervised approach was also utilized for monocular depth estimation. Typically, this approach involves capturing the geometric relationship between consecutive frames and reconstructing one frame using information from the other. Subsequently, a reprojection loss is computed by comparing the reconstructed scene with the original scene. Godard et al. [17] proposed the first self-supervised monocular depth estimation method using the aforementioned approach. They employed separate networks for depth estimation and pose prediction, where the depth network predicts the depth map and the pose network predicts the transformation matrix between two scenes. With utilizing the baseline method of Godard et al. [17], Guizilini et al. [18] suggest encoder-decoder structure using 3D convolution to preserve spatial information. They also proposed a velocity loss based on the concept that the norm of the translation matrix is proportional to the product of velocity and time. Lyu et al. [19] proposed Unet ++ [20] based architecture as an enhancement to the standard encoder-decoder architecture to decrease semantic gap between corresponding stage of encoder and decoder information. Liu et al. [21] specifically focused on depth estimation of nighttime. They only used invariant features to estimate depth, which were extracted using the orthogonality between the a nighttime image synthesized through Cycle-GAN [22] and original image. Recently, Wang et al. [23] attempted to find the transformation matrix by not solely relying on the network but also incorporating a correspondence matching method using RANSAC [24] and finding the fundamental matrix.

C. LOSS FUNCTION AND REBALANCING

Eigen et al. [3] proposed the scale-invariant loss to handle the ambiguity of global scale in monocular depth estimation. Finding global scale in monocular depth estimation is

challenging since we cannot distinguish whether a given photo represents the actual scene or a miniature representation. The scale-invariant loss computes the relationships between pixel values in the scene, irrespective of the absolute global scale. Subsequently, gradient loss [4] was also proposed to deal with comparison of difference between adjacent pixels.

From the perspective of loss rebalancing, achieving a balance between different loss functions has been a fundamental challenge in the domain of multi-task learning [25], [26], [27]. Recently, a multi-loss rebalancing algorithm was introduced for monocular depth estimation specifically focused on indoor scenes [7]. This approach incorporates a total of 78 loss functions, with weight of each loss varying based on spatial scale and the type of loss.

While numerous loss functions can be employed for the task of monocular depth estimation, we specifically utilized the scale-invariant loss [3] and gradient loss [4] to ensure stable convergence of the total loss. Our emphasis is placed on determining the optimal weight ratio between these two loss functions and aiming to enhance the effectiveness of the gradient loss as a supportive component to the scale-invariant loss employed previously.

III. PROPOSED METHOD

Figure 1 presents our proposed AMLR algorithm. In this section, we will begin by reviewing the network architecture and loss function of our proposed method and subsequently provide a detailed description of the proposed rebalancing term.

A. ARCHITECTURE

Figure 2 provides an overview of our overall architecture and several depth prediction results. We utilize the baseline architecture proposed by Xie et al. [6] which consists of a Swin Transformer [16] encoder and a decoder composed of deconvolution and upsampling layers.

The Swin Transformer [16] stands as a specific category within the domain of ViT(Vision Transformer) [28], specifically devised to tackle the inherent difference between the expressive capacity of an individual image patch and that of a single word. Elaborating further, the Swin Transformer initiates its process with smaller patches, each of which can be succinctly represented using a predetermined number of embedding vectors. Subsequently, through a hierarchical approach, these smaller patches are merged to form larger patches. This tiered structure empowers the Swin Transformer with the capacity to effectively process high-resolution images.

In figure 2, each swin transformer [16] block consists of patch merging layer and two self attention blocks. The patch merging layer merges patches and transmits a merged patch to the subsequent layer. A self-attention block consists of two components: a window multi-head self-attention block and a shifted window multi-head self-attention block. Through the shifting of the window during the attention computation process, the shifted window multi-head attention block enables attention computation not only within the designated window but also between adjacent windows.

We made several modifications to enhance performance of previous architecture based on [6] and [16]. One significant modification involved adjusting the channel configuration of the decoder layers. In the original architecture, the channel dimension of the feature map is reduced from 1536 to 32 in the first deconvolutional layer. Furthermore, a channel size of 32 is preserved across three consecutive deconvolutional layers. However, such a drastic reduction in channels can potentially result in information loss. To mitigate this issue, we gradually decreased the channel size in the decoder layers. Additionally, we incorporated the SiLU activation [5] function to introduce more diverse expression in the feature maps.

B. LOSS FUNCTION

We employed both scale-invariant loss [3] and gradient loss [4] in our AMLR method. From Eigen et al. [3], the scale-invariant loss is stated as the subtraction of the ratio between pixel pairs in the predicted depth map from the ratio of pixel pairs in the ground truth depth map. It is commonly employed to mitigate the influence of absolute scale ambiguity in depth estimation tasks as it measures difference between the relationships of pixel pairs rather than the difference of absolute values.

Specifically, the scale-invariant loss $\mathcal{L}_{\mathrm{Si}}$ is defined as

$$\begin{aligned}\mathcal{L}_{Si} &= \frac{1}{n}\sum_{i}^{n} d_i^2 - \frac{1}{n^2}(\sum_{i}^{n} d_i)^2, \\ &= \frac{1}{n^2}\sum_{i,j}^{n} d_i^2 - \frac{1}{n^2}\sum_{i,j}^{n} d_i d_j, \\ &= \frac{1}{2n^2}\sum_{i,j}^{n}((\log y_i - \log y_j) - (\log y_i^* - \log y_j^*))^2, \\ &= \frac{1}{2n^2}\sum_{i,j}^{n}((\log \frac{y_i}{y_j}) - (\log \frac{y_i^*}{y_j^*}))^2, \end{aligned} \tag{1}$$

where n is the total number of pixels, with pixel index i and j. $\mathcal{L}_{\mathrm{Si}}$ is calculated by the difference in logarithmic depth ($d_i = \log y_i - \log y_i^*$) between the ground truth depth (y_i) and the predicted depth (y_i^*).

Meanwhile, the gradient loss $\mathcal{L}_{gr}$ (2) is defined as

$$\mathcal{L}_{Gr} = \frac{1}{n}\sum_{i,j}^{m,n}[(\nabla_x d_{i,j})^2 + (\nabla_y d_{i,j})^2], \tag{2}$$

where the variables m and n represent the width and height of the image, respectively. The indices i and j are used to denote the positions along the x and y axes. The gradient terms $\nabla_x d_{i,j}$ and $\nabla_y d_{i,j}$ are defined by

$$\begin{aligned}\nabla_x d_{i,j} &= (\log y_{i+1,j} - \log y_{i,j}) - (\log y^*_{i+1,j} - \log y^*_{i,j}), \\ \nabla_y d_{i,j} &= (\log y_{i,j+1} - \log y_{i,j}) - (\log y^*_{i,j+1} - \log y^*_{i,j}). \end{aligned} \tag{3}$$

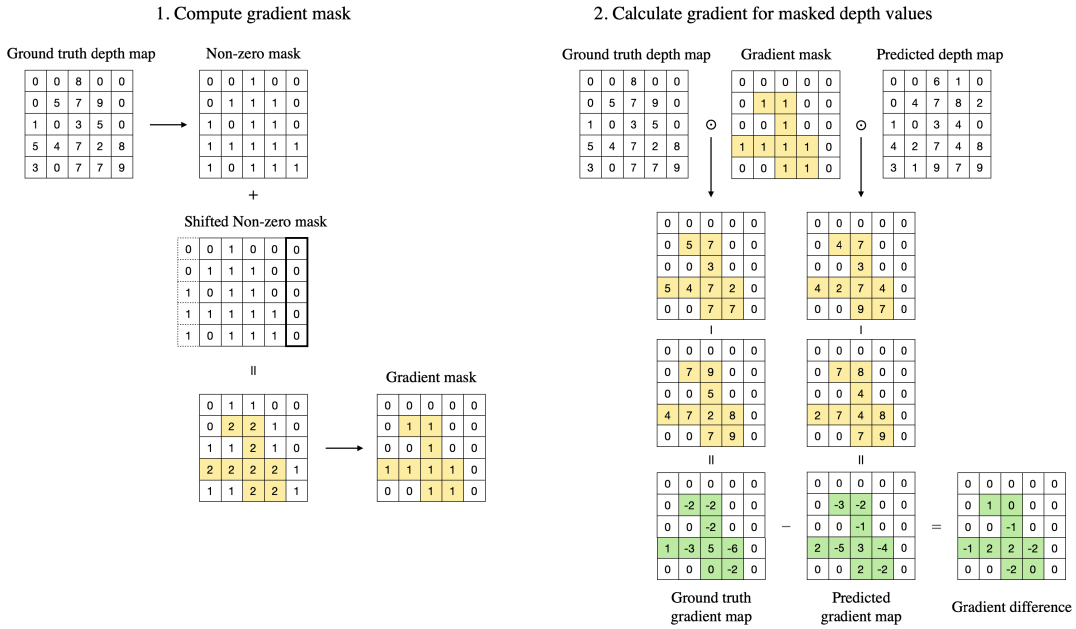

FIGURE 3. Example of computing gradient difference for *x*-axis. Initially, a gradient mask is generated for pixels where the gradient can be computed. In this process, we set the last column of shifted non-zero mask to zero. Subsequently, using this mask, gradient maps are calculated for both the ground truth and predicted depth value. Finally, the computed gradient differences are averaged with respect to the sum of the gradient mask. The gradient difference for the *y*-axis can be computed in the same manner along the vertical axis.

$\mathcal{L}_{gr}$ is calculated by the average of both the horizontal gradient difference and vertical gradient difference. We employed the forward difference method to compute the gradients of each pixel, which is described in equation (3). Additionally, we masked the pixels for which gradient computation was not possible due to missing values or zero disparity. By applying a masking technique, we were able to achieve improved accuracy in the computation of the gradient loss. Computation of gradient with masking is illustrated in Figure 3.

Finally, with the equations (1) and (2), the overall loss $\mathcal{L}_{tot}$ is defined as:

$$\mathcal{L}_{tot} = \sum_{i}^{2} w_i \mathcal{L}_i = w_{Si}\mathcal{L}_{Si} + w_{Gr}\mathcal{L}_{Gr}, \tag{4}$$

where w_{Si} and w_{Gr} represent the respective weights assigned to the scale-invariant loss and gradient loss.

C. LOSS REBALANCING ALGORITHM

In this section, we provide a review of the Adam optimizer and subsequently discuss the two crucial components of our Adaptive Momentum-based Loss Rebalancing method (AMLR), including loss initialization and loss rebalancing. Our loss rebalancing algorithm is rooted in the concept of momentum [29] and penalizing significant changes in loss values while assigning greater weight to loss functions that exhibit relatively smaller variations. This approach aims to stabilize the training process by mitigating the impact of abrupt or drastic fluctuations in the loss function. While rebalancing helps ensure proper weighting, incorrect initialization can lead to catastrophic results due to the utilization of momentum and exponential moving average (EMA) for loss calculation at each timestep. Therefore, we carefully initialized each weight based on fundamental hypotheses regarding gradient loss, number of ground truth pixels and empirical observations. As a result, these two key components contributed to the achievement of comparable results to state-of-the-art approaches. Detailed explanations are given in the following subsections.

1) ADAM OPTIMIZER

In our proposed approach, we incorporated the concept of a weight optimizer, specifically Adam [8]. Adam combines the advantages of RMSprop [30], which applies individual penalties to different parameters, and momentum, resulting in accelerating the training process.

$$m^t = \beta_1 m^{t-1} + (1-\beta_1)\frac{\partial \mathcal{L}^t}{\partial \theta^{t-1}}, \tag{5}$$

$$v^t = \beta_2 v^{t-1} + (1-\beta_2)(\frac{\partial \mathcal{L}^t}{\partial \theta^{t-1}})^2, \tag{6}$$

where m^t represents the momentum term at timestep t, which is calculated using the exponential moving average (EMA) method. Similarly, v^t in equation (6) corresponds to the RMSprop term at timestep t, also computed using the EMA method. Moreover, β_1 and β_2 are decay rate in the EMA for momentum and RMSprop term respectively. Higher decay gives more weight to recent value, resulting faster update of the moving average. $\mathcal{L}^t$ is loss of current timestep with

θ^{t-1} indicating parameter before update. With incorporating these terms and learning rate η and ϵ as a small value(e.g., $\epsilon = 1 \times 10^{-8}$), Adam optimization method [8] is defined follows:

$$\theta^t = \theta^{t-1} - \eta \frac{m^t}{\sqrt{v^t} + \epsilon}. \tag{7}$$

By incorporating the Adam optimizer in the weight rebalancing process, we can take advantage of penalizing loss functions that are relatively easy to optimize. Moreover, this allows us to preserve the weight tendencies for each loss component.

2) LOSS INITIALIZATION

As mentioned earlier, we initialized the weights of each loss function based on the underlying hypotheses and empirical observations. It was hypothesized that in terms of enhancing accuracy, the scale-invariant loss [3] imposes a stronger constraint compared to the gradient loss [4]. This is due to the scarcity of ground truth pixel values in outdoor scenes. In the calculation of gradient loss, the absence of ground truth values for individual pixels hinders the computation of pixel pairs, whereas scale-invariant loss restricts the computation solely to the corresponding pixel. Consequently, in outdoor scenes where ground truth pixel values are often lacking, the limitation of computing a relatively small number of gradient losses imposes a weaker constraint compared to the scale-invariant loss. Furthermore, the gradient loss, by solely measuring the differences between adjacent pixels, may fail to capture the overall image representation adequately. This implies that the scale-invariant loss should assume a more significant role than the gradient loss and thus carry a greater weight. Therefore, we initialized the scale-invariant loss as a primary loss and the gradient loss as a secondary or assistant loss.

Specifically, the initialization process of the loss function involves two timesteps. In the first timestep ($t = 0$), we set the loss weights for both the scale-invariant loss and the gradient loss to be the same. In the following timestep ($t = 1$), we establish different weighting factors to the loss weights, with the scale-invariant loss being designated as the primary loss and the gradient loss as the secondary loss. Concretely, the loss weights at timestep $t = 0$ and $t = 1$ are defined as:

$$w_i^0 = \frac{1}{n_w}, \tag{8}$$

$$w_i^1 = \frac{\mathcal{L}_{\text{tot}}^1}{\mathcal{L}_i^1} \times r_{\mathcal{L}_i^1}, \tag{9}$$

where n_w represents the number of loss functions, which is set as 2. In equation (9), $r_{\mathcal{L}_i^1}$ denotes the weighting factor of i^{th} loss component $\mathcal{L}_i^1$ in the timestep $t = 1$. The superscripts of both w and $\mathcal{L}$ indicate the timestep t in our rebalancing algorithm, while subscripts of them indicate the index of the loss function. By appropriately initializing the loss, it became possible to guide the weighting of the loss during the rebalancing step.

3) ADAPTIVE MOMENTUM-BASED LOSS REBALANCING

Loss rebalancing is an important factor when dealing with more than two loss functions. Every individual loss function inherently contributes to the overall convergence of the composite loss, albeit with distinct magnitudes of influence.

In this paper, we propose AMLR employing scale-invariant loss [3] and gradient loss [4]. While conducting training using a supervised approach, it is reasonable to posit that the scale-invariant loss, which is rooted in mean square error, assumes a more substantial role in the training process compared to the gradient loss. From an alternative standpoint, neighboring pixels show a strong interdependence, and gaining insights into these relational complexity might be crucial for developing a understanding of the depth characteristics of objects. For example, the 7th row of Figure 4 demonstrates that by accounting for the interplay among adjacent pixels, the uniformity of the depth pertaining to the traffic sign is significantly improved. Building upon the aforementioned insights, we postulated that the incorporation of gradient loss could offer utility in the context of representing the depth characteristics of individual objects.

Our loss rebalancing term is formulated based on the proportion of a specific loss with respect to the total loss at the current timestep t. The proportion of the specific loss at timestep t can be expressed as $P_i = \mathcal{L}_i^t / \sum_i w_i^t \mathcal{L}_i^t$, where w_i represents the corresponding weight assigned to each loss function and $\mathcal{L}_i$ represents individual loss function. The percentage difference of $\mathcal{L}_i$ between two consecutive timesteps can be computed as $\Delta P_i^t = P_i^t - P_i^{t-1}$.

Based on the aforementioned concepts in equation (5), (6) and (7) and notations, our rebalancing weights at timestep $t \geq 2$ are defined as:

$$m_i^t = \beta_1 m_i^{t-1} + (1 - \beta_1) \frac{\Delta P_i^t}{P_i^t} w_i^{t-1}, \tag{10}$$

$$v_i^t = \beta_2 v_i^{t-1} + (1 - \beta_2) \left(\frac{\Delta P_i^t}{P_i^t} w_i^{t-1} \right)^2, \tag{11}$$

$$w_i^t = w_i^{t-1} - \frac{m_i^{t-1}}{\sqrt{v_i^{t-1}} + \epsilon} \lambda. \tag{12}$$

Here, w_i^t represents the weight of the loss function $\mathcal{L}_i$ at timestep t, while m_i^t and v_i^t denote the corresponding momentum and RMSprop terms respectively. β_1 and β_2 are decay rate in the EMA for momentum and RMSprop term respectively just as mentioned in equation (5) and (6).

These equations provide a mechanism for updating the loss weights based on the percentage differences ($\Delta P_i^t / P_i^t$) and the previous weights (w_i^{t-1}). The momentum term (m_i^t) and the RMSprop term (v_i^t) help adjust the weight updates, taking into account the historical information of the loss variations. Finally, the weight update equation incorporates these terms and the learning rate λ to determine the new weights w_i^t. The employment of momentum and RMSprop terms enhances the effectiveness of the rebalancing process by accounting for historical weights and trends, contrasting with relying

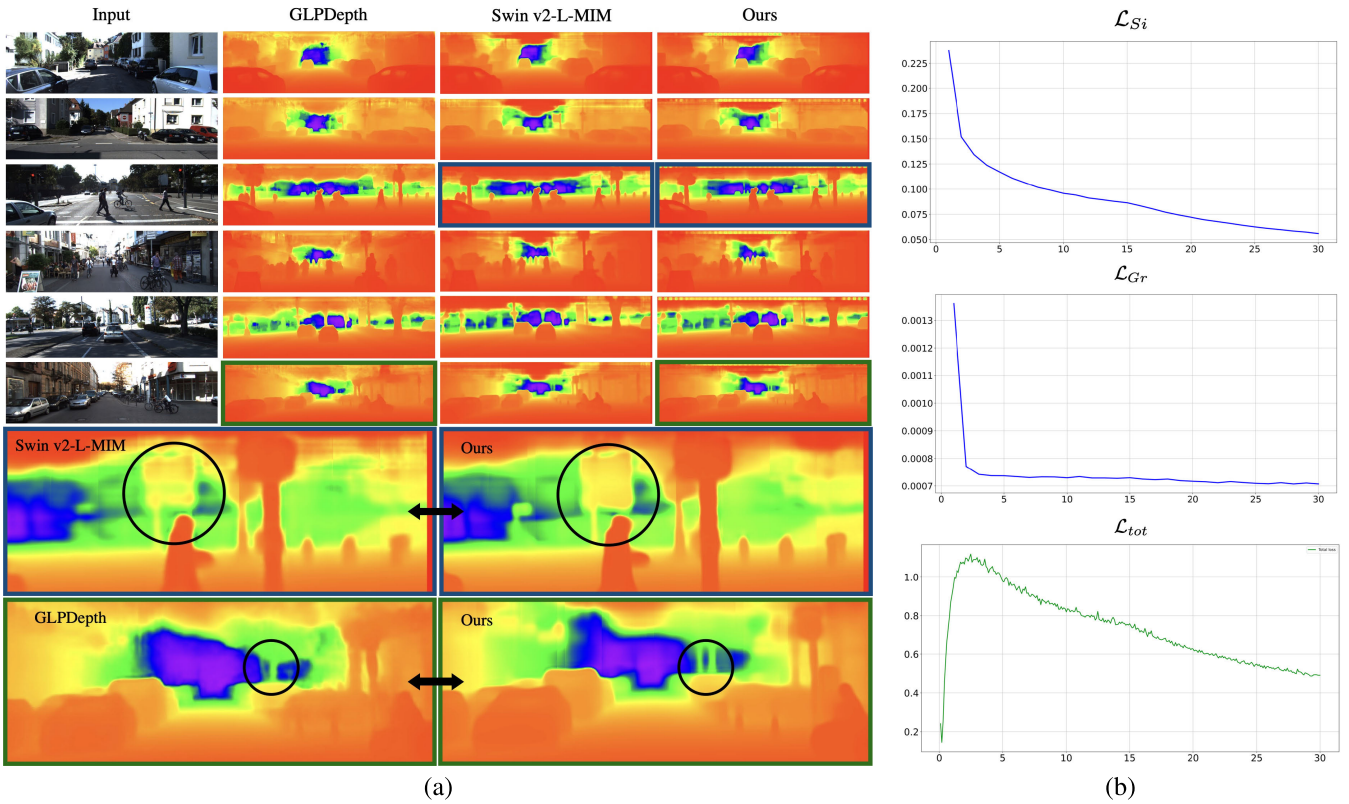

FIGURE 4. Qualitative comparison results of GLPDepth [14], Swin v2-L-MIM [6] and Ours with KITTI [31] dataset in (a), and scale-invariant loss, gradient loss and total loss on training process of KITTI dataset in (b). Several qualitative improvements are observed in the output results between our network and previous networks. In row 7 of (a), more explicit prediction of a billboard and upper body of pedestrian is discernible. In row 8, we can observe that our network successfully distinguishes two adjacent trees that were previously indistinguishable by the preceding network.

exclusively on the proportional adjustment of individual loss components [7].

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

For KITTI [31] dataset, we employ a single NVIDIA A100 GPU for training our model with 30 epochs and batch size of 6. In the timestep $t = 1$, the initial weighting factor $r_{\mathcal{L}_1^1}$ for the scale-invariant loss and $r_{\mathcal{L}_2^1}$ for the gradient loss in equation (9) were empirically determined as 0.99 and 0.01, respectively. This determination was based on the observations and hypothesis that the scale-invariant loss serves as the primary loss while gradient loss serves as secondary loss. We adopted the learning rate schedule proposed by Xie et al. [6] for our training process. The learning rate gradually increased until the midpoint of the entire epoch and then reverted to its initial value, with an initial learning rate of 3e-5 and a midpoint learning rate of 5e-4. We updated the learning rate after every single batch and the depth range was defined as 0 to 80 meters.

Furthermore, we conducted experiments on the NYUv2 [32] dataset to substantiate the efficacy of our approach. The experiments involving the NYUv2 dataset was conducted under nearly identical conditions to those of the KITTI [31] dataset, with the exception of variations in depth range and training epochs. For the NYUv2 dataset, we executed training for a total of 20 epochs and with depth range as 0 to 10 meters.

During the evaluation step, we incorporated the concepts of the flip test and shifted window test. Firstly, the validation image is divided into several overlapping sub-images, each taking the form of a square with sides equal to the height of the original image. Subsequently, through augmenting these sub-images and obtaining their respective depth maps, we were able to reconstruct the depth map of the original image by averaging the depth maps of the sub-images. This method resulted in notable improvements in our outcomes. Table 1 demonstrates the improved performance obtained by incorporating both the shifted window test and flip test.

B. DATASET AND EVALUATION METRICS

We used KITTI dataset [31] with eigen split [3] and garg crop [38]. The dataset comprises a total of 23,158 training images and 652 test images. Each image in the dataset is accompanied by four corresponding sections. Notably, the KITTI dataset is constructed based on a stereo camera setup, resulting in the provision of both color and grayscale images captured by the left and right cameras respectively. Since our task only utilizes a single image per scene, our approach only considers color images captured by the left stereo camera. Prior to training, we crop image to size of

TABLE 1. The comparison of the evaluation results obtained from the shift window test and flip test, juxtaposed with the results obtained without these tests. "None" refers to the absence of the shifted-window and flip test, while "S/W, Flip" denotes testing with the inclusion of both the shifted-window and flip test. The best performing results are highlighted in bold. The upward (↑) and downward (↓) arrows indicate the direction of improvement in each metric.

	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE Log↓	Log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
None	0.053	0.153	2.065	0.081	0.023	0.973	0.997	**0.999**
S/W, Flip	**0.049**	**0.137**	**1.971**	**0.074**	**0.021**	**0.978**	**0.998**	**0.999**

TABLE 2. Quantitative comparison results on the KITTI Eigen split dataset. We reported result with 7 widely used metrics. State-of-the-art results are highlighted in bold. We have achieved state of the art or comparable results in most metrics. We also reported result of our method with GLPDepth [14] baseline. The upward (↑) and downward (↓) arrows indicate the direction of improvement in each metric.

Methods	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE Log↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Eigen *et al.* [3]	0.203	1.548	6.307	0.282	0.702	0.898	0.967
DORN [12]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
BTS [33]	0.061	0.261	2.834	0.099	0.954	0.992	0.998
AdaBins [13]	0.058	0.190	2.360	0.088	0.964	0.995	0.999
SfM-Revisited [23]	0.055	0.224	2.273	0.091	0.956	0.984	0.993
MonoDELSNet [34]	0.053	0.161	2.101	0.082	0.969	0.996	0.999
GLPDepth [14]	0.057	0.187	2.297	0.086	0.967	0.996	0.999
NewCRFs [35]	0.052	0.155	2.129	0.079	0.974	0.997	0.999
DepthFormer [36]	0.052	0.158	2.143	0.079	0.975	0.997	0.999
BinsFormer [37]	0.052	0.151	2.098	0.079	0.974	0.997	0.999
Swin v2-L-MIM [6]	0.050	0.139	**1.966**	0.075	0.977	**0.998**	**1.000**
URCDC-Depth [9]	0.050	0.142	2.032	0.076	0.977	0.997	0.999
iDisc [10]	0.050	0.145	2.067	0.077	0.977	0.997	0.999
GLPDepth [14] +AMLR	0.054	0.148	2.069	0.080	0.975	0.997	0.999
Ours	**0.049**	**0.137**	1.971	**0.074**	**0.978**	**0.998**	0.999

1216 × 352 and applied common data augmentations such as horizontal flipping and random cropping.

We evaluate the results of our method in indoor environments using the NYUv2 [32] dataset. NYUv2 dataset is comprised of video sequences from a variety of indoor scenes with its depth captured by Microsoft Kinect camera. It consists of 24k training split from 464 indoor scenes and result was evaluated by 654 images from 215 indoor scenes. Crop was done with the size of 448 × 576 and common data augmentations such as horizontal flipping and random cropping were also applied.

We evaluate our method with various metrics. With the depth space $\mathbf{d}$, the metrics consist of the absolute relative error (Abs Rel) $= \frac{1}{M}\sum_{i=1}^{M} |\mathbf{d}_i - \hat{\mathbf{d}}_i|/\mathbf{d}_i$, the square relative error (Sq rel) $= \frac{1}{M}\sum_{i=1}^{M} (\mathbf{d}_i - \hat{\mathbf{d}}_i)^2/\mathbf{d}_i$, the root mean squared error (RMSE) $= (\frac{1}{M}\sum_{i=1}^{M} (\mathbf{d}_i - \hat{\mathbf{d}}_i)^2)^{\frac{1}{2}}$, the log root mean squared error (RMSE Log) $= (\frac{1}{M}\sum_{i=1}^{M} (\log_{10}\mathbf{d}_i - \log_{10}\hat{\mathbf{d}}_i)^2)^{\frac{1}{2}}$, the average $\log_{10}$ error (Log_{10}) $= \frac{1}{M}\sum_{i=1}^{M} |\log_{10}\mathbf{d}_i - \log_{10}\hat{\mathbf{d}}_i|$, and the threshold accuracy $\delta_n =$ percent of pixels, such that $\max(\mathbf{d}_i/\hat{\mathbf{d}}_i, \hat{\mathbf{d}}_i/\mathbf{d}_i) < 1.25^n$ for $n = 1, 2, 3$, where $\mathbf{d}_i$ and $\hat{\mathbf{d}}_i$ denote ground truth and predicted depth value at pixel index i respectively and M is the total number of pixels in the image.

C. COMPARATIVE RESULTS

For KITTI [31] dataset, we compared our proposed method with previous methods using 7 widely used metrics. Our proposed method outperformed previous methods for almost measures and achieved comparable results on others. The results are reported in Table 2. Specifically, we achieved state-of-the art result on Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), and Root Mean Square Error (RMSE) metrics while achieving comparable results in other metrics.

Qualitative results for KITTI dataset are illustrated in Figure 4. The upper six rows present the outputs of different networks for the same image. Specifically, in the seventh row, our method demonstrates improved clarity in capturing the shape of the billboard, background texture, and upper body of the pedestrian compared to Swinv2-L-MIM [6]. Moreover, in the eighth row, our method successfully achieves the distinction between two adjacent trees, which the previous network fails to accomplish. These achievements appear attributable to the adaptive incorporation of the scale-invariant loss and the gradient loss [4], which facilitates object elucidation by learning the interplay between neighboring pixels.

TABLE 3. Quantitative comparison results on the NYUv2 dataset. We reported result with 6 widely used metrics. State-of-the-art results are highlighted in bold. We also have achieved comparable results in most metrics even though our method was done for outdoor scene just as on KITTI dataset. The upward (↑) and downward (↓) arrows indicate the direction of improvement in each metric.

Methods	Abs Rel↓	RMSE↓	Log_{10}↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Eigen *et al.* [3]	0.158	0.641	-	0.769	0.950	0.988
DORN [12]	0.115	0.509	0.051	0.828	0.965	0.992
BTS [33]	0.113	0.407	0.049	0.871	0.977	0.995
AdaBins [13]	0.103	0.364	0.044	0.903	0.984	0.997
GLPDepth [14]	0.098	0.344	0.042	0.915	0.988	0.997
NewCRFs [35]	0.095	0.334	0.041	0.922	0.992	0.998
DepthFormer [36]	0.096	0.339	0.041	0.921	0.989	0.998
BinsFormer [37]	0.094	0.330	0.040	0.925	0.989	0.997
Swin v2-L-MIM [6]	**0.083**	**0.287**	**0.035**	**0.949**	**0.994**	**0.999**
URCDC-Depth [9]	0.088	0.316	0.038	0.933	0.992	0.998
iDisc [10]	0.086	0.313	0.037	0.940	0.993	**0.999**
Ours	0.087	0.300	0.037	0.935	0.991	0.998

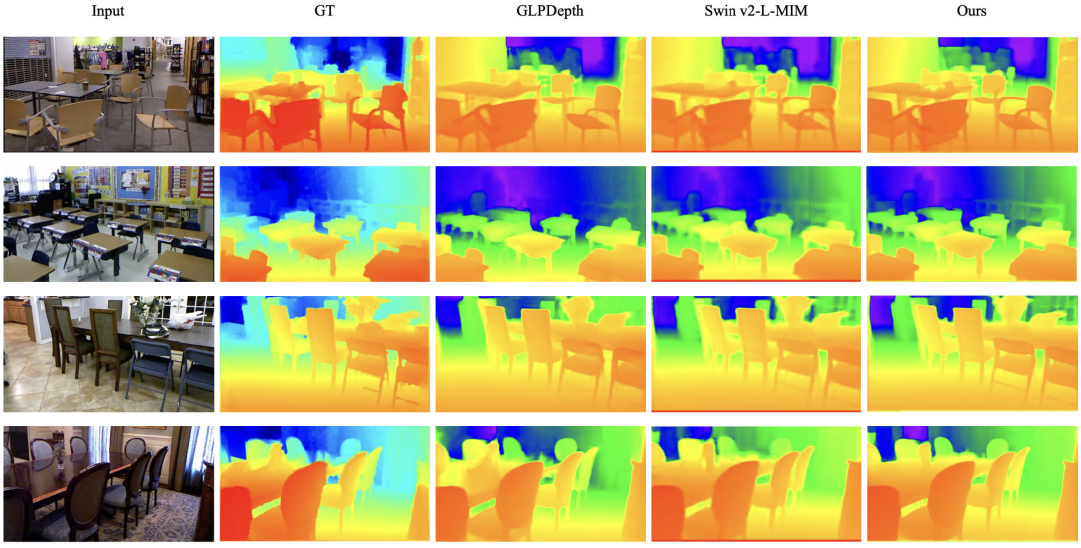

FIGURE 5. Qualitative comparison results of GLPDepth [14], Swin v2-L MIM [6] and Ours on NYUv2 [32] dataset.

In the realm of this backbone architectures and their associated training methodologies, the foundation of our works was laid by the Swin Transformer [16] and the work of Xie et al. [6]. By comparing with previous method by Xie et al. [6], incorporating adaptations such as gradient loss integration and embracing a loss rebalancing approach, we demonstrated competitive performance across a spectrum of metrics, with particularly in the domain of absolute relative error.

We also evaluated our method using NYUv2 [32] dataset. While our method is primarily designed to address outdoor environments, it achieved comparable results to previous methods in indoor scenes. Quantitative comparison results for NYUv2 dataset are reported in Table 3, where we compared our method with previous methods using 6 widely used metrics. We also presented comparisons between qualitative results for several indoor scenes in Figure 5. Figure 5 shows that objects in our results are clearer and sharper than those of previous methods [6], [14]. For instance, our method enhances the perceptual clarity of background chairs in the first row and shelves in the second row within the obtained results. This enhancement is attributed to the incorporation of the gradient loss with the proposed adaptive loss weighting algorithm through the consideration of relationships between adjacent pixels.

D. ABLATION STUDY

During the training process, we conduct an analysis of the convergence of the loss functions. While the scale-invariant loss remained dominant throughout the entire training process, we observe the convergence of the gradient loss as well. This observation indicates that despite its relatively

TABLE 4. Comparative analysis of incorporating different loss functions. The table presents the results of incorporating scale-invariant loss and previous loss rebalancing method proposed by Lee *et al*. [7]. We also reported our result on the last row. The best performing results are highlighted in bold. The upward (↑) and downward (↓) arrows indicate the direction of improvement in each metric.

	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE Log↓	Log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Only $\mathcal{L}_{Si}$	0.050	0.139	1.981	0.075	0.022	**0.978**	**0.998**	**0.999**
Lee *et al*. [7]	**0.049**	0.139	2.000	0.075	0.022	**0.978**	**0.998**	**0.999**
Ours	**0.049**	**0.137**	**1.971**	**0.074**	**0.021**	**0.978**	**0.998**	**0.999**

small portion, the gradient loss still exerted influence on the training process. Training loss of both loss functions are visualized in Figure 4. We observe an initial rise in the total training loss, consistent with the expected behavior during the initialization and rebalancing stages where loss weights are being stabilized. As the training progresses, this upward trend subsides, indicating the achievement of loss weight stabilization and subsequent convergence of the loss function.

Furthermore, we conduct separate tests under identical conditions using only the scale-invariant loss and the gradient loss, respectively. Incorporating our rebalancing term, we observe a notable enhancement in performance when utilizing both the scale-invariant loss and the gradient loss, surpassing the individual utilization of each loss function. The comparative analysis regarding the incorporation of different loss functions and rebalancing algorithm is presented in Table 4. We also applied our method to backbone of GLPDepth [14] architecture to validate the effectiveness of our proposed AMLR approach. We noted performance improvement of GLPDepth upon the application of the proposed AMLR. Detailed comparisons using various metrics are presented in Table 2.

We further note a degradation in the performance of our network when employing skip connections. This outcome can be attributed to the placement of the skip connection at an early stage of the decoder, impeding its intended function of transmitting comprehensive global information from the encoder to the decoder.

V. CONCLUSION

In this paper, we propose several changes to the architecture of a previous monocular depth estimation network including SiLU activation and modification of decoder channel. We also incorporated a gradient loss into the depth prediction task. To ensure that the gradient loss not to be dominant loss during the learning process but rather serves as an assisting loss to the scale-invariant loss, we introduce a novel loss rebalancing term called as AMLR. Through our methodology, we achieve state-of-the-art performance in monocular depth estimation on the KITTI dataset with eigen split.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

[2] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12697–12705.

[3] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[4] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[5] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, Nov. 2018.

[6] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao, "Revealing the dark secrets of masked image modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2023, pp. 14475–14485.

[7] J.-H. Lee and C.-S. Kim, "Multi-loss rebalancing algorithm for monocular depth estimation," in *Proc. Comput. Vis. ECCV 16th Eur. Conf.* Glasgow, U.K.,: Springer, Aug. 2020, pp. 785–801.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[9] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, and Z. Li, "URCDC-depth: Uncertainty rectified cross-distillation with CutFlip for monocular depth estimation," 2023, *arXiv:2302.08149*.

[10] L. Piccinelli, C. Sakaridis, and F. Yu, "IDisc: Internal discretization for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21477–21487.

[11] K. Zhang, M. Liu, J. Zhang, and Z. Dong, "PA-MVSNet: Sparse-to-dense multi-view stereo with pyramid attention," *IEEE Access*, vol. 9, pp. 27908–27915, 2021.

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[13] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4009–4018.

[14] D. Kim, W. Ka, P. Ahn, D. Joo, S. Chun, and J. Kim, "Global–local path networks for monocular depth estimation with vertical CutDepth," 2022, *arXiv:2201.07436*.

[15] T. Van Dijk and G. De Croon, "How do neural networks see depth in single images?" in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2183–2191.

[16] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, and L. Dong, "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2022, pp. 12009–12019.

[17] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jul. 2019, pp. 3828–3838.

[18] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 2485–2494.

[19] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan, "HR-depth: High resolution self-supervised monocular depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 3, 2021, pp. 2294–2301.

[20] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Granada, Spain: Springer, Sep. 2018, pp. 3–11.

[21] L. Liu, X. Song, M. Wang, Y. Liu, and L. Zhang, "Self-supervised monocular depth estimation for all day images using domain separation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12737–12746.
[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2223–2232.
[23] J. Wang, Y. Zhong, Y. Dai, S. Birchfield, K. Zhang, N. Smolyanskiy, and H. Li, "Deep two-view structure-from-motion revisited," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, May 2021, pp. 8953–8962.
[24] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[25] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 794–803.
[26] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.
[27] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.
[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[29] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, Jan. 1964. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0041555364901375
[30] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[32] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Comput. Vis. ECCV 12th Eur. Conf. Comput. Vis.* Florence, Italy: Springer, Oct. 2012, pp. 746–760.
[33] J. Han Lee, M.-K. Han, D. Wook Ko, and I. Hong Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*.
[34] A. Gurram, A. F. Tuna, F. Shen, O. Urfalioglu, and A. M. López, "Monocular depth estimation through virtual-world supervision and real-world SfM self-supervision," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12738–12751, Aug. 2022.
[35] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "NeW CRFs: Neural window fully-connected CRFs for monocular depth estimation," 2022, *arXiv:2203.01502*.
[36] Z. Li, Z. Chen, X. Liu, and J. Jiang, "DepthFormer: Exploiting long-range correlation and local information for accurate monocular depth estimation," 2022, *arXiv:2203.14211*.
[37] Z. Li, X. Wang, X. Liu, and J. Jiang, "BinsFormer: Revisiting adaptive bins for monocular depth estimation," 2022, *arXiv:2204.00987*.
[38] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Comput. Vis. ECCV 14th Eur. Conf.*, Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 740–756.

WON-GYUN YU received the B.S. degree from the Department of Electrical and Computer Engineering, Ajou University, Suwon, South Korea. His research interests include computer vision, deep learning, and 3D vision.

YONG SEOK HEO received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, South Korea, in 2005, 2007, and 2012, respectively. From 2012 to 2014, he was with the Digital Media and Communications Research and Development Center, Samsung Electronics. He is currently with the Department of Electrical and Computer Engineering and the Department of Artificial Intelligence, Ajou University, as a Professor. His research interests include segmentation, stereo matching, 3D reconstruction, and computational photography.