MONOCULAR DEPTH ESTIMATION USING A DEEP LEARNING MODEL WITH PRE-DEPTH ESTIMATION BASED ON SIZE PERSPECTIVE

Takanori Asano[1] and Yoshiaki Yasumura[2]

[1]Graduate School of Engineering and Science, Shibaura Institute of Technology, Tokyo, Japan
[2]College of Engineering, Shibaura Institute of Technology, Tokyo, Japan

ABSTRACT

In this paper, For the task of the depth map of a scene given a single RGB image. We present an estimation method using a deep learning model that incorporates size perspective (size constancy cues). By utilizing a size perspective, the proposed method aims to address the difficulty of depth estimation tasks which stems from the limited correlation between the information inherent to objects in RGB images (such as shape and color) and their corresponding depths. The proposed method consists of two deep learning models, a size perspective model and a depth estimation model, The size-perspective model plays a role like that of the size perspective and estimates approximate depths for each object in the image based on the size of the object's bounding box and its actual size. Based on these rough depth estimation (pre-depth estimation) results, A depth image representing through depths of each object (pre-depth image) is generated and this image is input with the RGB image into the depth estimation model. The pre-depth image is used as a hint for depth estimation and improves the performance of the depth estimation model. With the proposed method, it becomes possible to obtain depth inputs for the depth estimation model without using any devices other than a monocular camera be forehand. The proposed method contributes to the improvement in accuracy when there are objects present in the image that can be detected by the object detection model. In the experiments using an original indoor scene dataset, the proposed method demonstrated improvement in accuracy compared to the method without pre-depth images.

KEYWORDS

Depth Estimation, Deep Learning, Image Processing, Size Perspective, YOLOv8

1. INTRODUCTION

Depth estimation from RGB images using deep learning is an important task of computer vision and deep learning, as it generates valuable depth information from regular cameras that can be applied across various domains such as autonomous driving and computer graphics. Moreover, it offers a cost-effective alternative to depth measurement compared to often expensive sensors like LiDAR. Previous research has achieved significant accuracy improvements by leveraging state-of-the-art models like ResNet [1]or EfficientNet [2] and others [3] [4]. for estimation. However, the accuracy of estimation is still not sufficient to replace dedicated sensors. One reason why it's challenging to estimate depth from RGB images is the intrinsic ambiguity of the task. The information present in images (such as color and shape and others) has a limited direct relationship with depth, Additionally, in indoor scenes where objects occupy larger areas, we

David C. Wyld et al. (Eds): CSTY, AIMLNET, AI, BINLP -2023
pp. 17-27, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131803

speculate that estimating objects is more challenging than backgrounds because objects tend to have more diverse shapes, with scattered objects in the background potentially interfering with the estimation of the background portion, or vice versa.

Thus, Depth estimation, especially for object parts, is a challenging task, but when it comes to estimating the object parts, much like humans, it is possible to roughly estimate the depth from a size perspective if the size of the object is known. The Size perspective describes the way humans perceive depth, where larger objects appear closer and smaller objects appear farther away. To address the difficulties of depth estimation, in this paper, we propose a method 'Pre-Depth Estimation (based on size perspective)' which pre-estimates depth using size perspective exclusively for object regions. Using Pre-Depth Estimation makes it possible to estimate the depth of the object’s region based on the universal rule of size perspective and reduce ambiguity in the depth estimation tasks. Moreover, it allows to extraction of the separated features between the background and objects. In this paper, we implemented two deep learning networks. The first one is the Size Perspective Model, which estimates the approximate depth of each object within the image using the Size Perspective. The second one is the Depth Estimation Model, which takes the output from the Size Perspective Model (Pre-Depth) and the RGB image as input to produce the final estimated depth. In this paper, we implement the proposed method and experiment to verify the effectiveness of this approach.

In this section, the 'Introduction', we discussed the background related to depth estimation and the objectives we aim to achieve with our approach. In the following Section 2, we delve into previous research on depth estimation and the proposed method. Section 3 provides an overview of the proposed method along with detailed explanations of each step. In Section 4, we discuss the experiments to evaluate the proposed method. Moving on to Section 5, we present insights derived from the experimental results, along with discussions on the limitations and current challenges of the proposed approach. Section 6 summarizes the contributions of the proposed method and outlines prospects.

2. RELATED WORK

Recently, with the advancements in deep learning models for image processing tasks, researchers have been applying these models to depth estimation. In the study by I.Laina et al [1], FCN based on ResNet is developed for depth estimation and they also proposed a loss function suitable for depth estimation. Yasasvy Tadepalli et al [2] developed an estimation model based on Efficient Net. Additionally, generative models such as GAN and Diffusion model are also applied to depth estimation [3][4]. These studies have led to the invention of many effective model architectures and learning methods and significantly contributed to the improvement of accuracy. In addition to these approaches, René Ranftl et al, [5] have also focused on addressing the depth scale gaps between different datasets. They developed various techniques including loss functions and training methods to adjust the scale gaps. They trained models using diverse and large datasets. Like other vision tasks, larger models and datasets can lead to improvements in the accuracy of depth estimation. In this study, we have drawn inspiration from these studies, and have adopted ResNet as a base model and the loss function utilized in the research by I. Laina et al. [1].

In addition to the models mentioned earlier, in a study by David Eigen et al [6], the estimation model has convolutional processing at multiple scales, rather than uniformly extracting features like a typical CNN. This allows for separate extraction of both global and local features of the image. As the complexity of the depth map differs between global features, such as the background, and local features, such as objects, this approach is considered effective. While focused on depth completion tasks, P. Liu, et al’s research [7] has improved accuracy by incorporating the DCN[8] into the model. DCN can adjust the receptive field that each pixel of the input image references. This suggests that the model can estimate depth while considering only the necessary receptive field (values of other pixels) when estimating each pixel's depth value, thereby excluding obstructive regions. In our approach, by pre-estimating the depth of objects, we aim to separate and extract features of both background and objects to a certain extent.

Besides research aiming to estimate a depth with only RGB images as an input, some studies aim to simplify tasks by incorporating additional information beyond RGB images into the model. Dong-hoon Kwak et al. [9] attempted to improve the accuracy by incorporating segmentation images, into the training of the CycleGAN model. H. Lim et al. [10] and Y. Liao et al. [11] attempted to incorporate depth information. They obtained depth information from partial laser scanning. The data is used as an additional input. Additionally, F. Ma et al. [12] included a portion of depth information obtained from LiDAR and used it as an additional input. Because the model tries to estimate depth, especially, it is considered effective to utilize depth information as input. However, these methods require devices other than cameras to obtain the depth information. In this study, similarly, depth information is utilized as an input to the model. However, our method uses size perspective cues to estimate rough depth information. It does not need external sensors to obtain depth information.

3. DEPTH ESTIMATION USING PRE-DEPTH ESTIMATION BASED ON SIZE PERSPECTIVE

3.1. Overview

Figure 1 represents an overview diagram illustrating each step of the proposed method. This method consists of the following five steps.

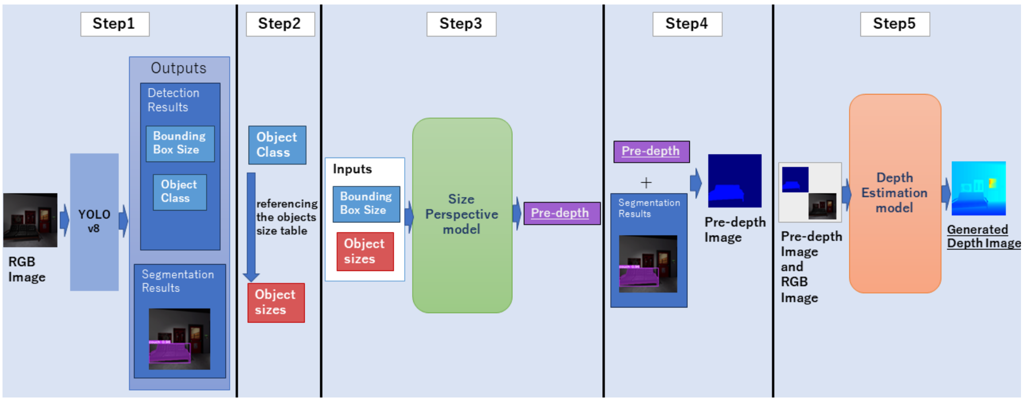

Figure 1 overview diagram of each step of the proposed method

Step 1: Obtain class labels, bounding boxes, and segmentation.

An RGB image is input into YOLOv8 [13]. YOLOv8 outputs information about the classes of the objects, the bounding box of each object in the image, and the segmentation region.

Step 2: Determine the size of the detected objects.

To estimate approximate depth using the size perspective, it is necessary to have information about the size of objects in the image and their actual sizes. In this method, the size of objects in the image is measured from the bounding boxes detected in Step 1. The actual size is determined based on the object's class by referencing a table ("objects size table") that records the corresponding size for each class.

Step 3: Estimate a Pre-depth of objects.

The size perspective model estimates an approximate depth (Pre-depth) based on the object size determined in Step 2. The size perspective model is trained with the object's size and the size of the bounding box as inputs, and the approximate depth of the bounding box region as the target. After the training, the model outputs a Pre-depth of objects.

Step 4: Generate a Pre-Depth image.

A pre-depth image is generated from the pre-depth and segmentation of the detected objects. This pre-depth image is later used as input to the estimation model as a hint. The generation is simply pasting the values of Pre-Depth onto the segmentation region.

Step 5: Train the estimation model and generate a depth image.

The depth estimation model is trained to generate depth images from pre-depth images and RGB images. After training, finally, the depth estimation model generates depth images.

3.2. Datasets

In this study, two datasets are created for the size perspective model and the depth estimation model to ensure the generalization of the pre-depth estimation.

The datasets are created using Unreal Engine 4 [14], a game development platform, and its plugin tools, NVIDIA Deep Learning Dataset Synthesizer (NDDS) [15]. NDDS is a plugin tool designed for computer vision researchers, which allows extracting depth, segmentation, bounding box, and other information from 3DCG within Unreal Engine4.The images in the dataset have a size of 512×512 and a maximum depth of 8192 cm, represented in 8-bit resolution. In the dataset for the size perspective model, each image contains only one object. In the dataset for the depth estimation model, the captured scenes consist of only indoor scenes where the target objects occupy a certain extent of the image.

3.3.Object Detection

In the first stage of the proposed method, it is necessary to detect objects and obtain their classes, segmentation, and bounding boxes. In this study, YOLOv8x-seg[13], trained on the COCO val2017 dataset, is used as the object detection and segmentation model. Furthermore, to avoid false detections, the detection targets were limited to only 17 types of objects. For instance, couch, table, and other furniture are commonly found in indoor scenes.

3.4. Confirmation of Size

The size information is determined based on the size of each object's 3DCG in UnrealEngine 4.The sizes corresponding to each class are recorded in a table (objects size table), and based on this, the sizes of the detected objects are determined. Figure 2 is a part of that table.

class_label	name	Width	Length	Height
2	Chair	80	80	100
8	Bottle	9	9	24
9	Desk	180	90	80

Figure 2 A part of the objects size table

For example, if the class label of the detected object is "8"(Bottle), its size would be [9cm, 9cm,24cm]. These values of size are “Object Width”, “Object Length”, and “Object Height” in

3.5. Pre-Depth Estimation

For each region containing objects, the size perspective model estimates a pre-depth. As shown in Figure 3, the size perspective model has simple network architecture. It is trained with the following inputs and targets.

Input: the height and width of the object bounding box, the overall length, width, and height obtained from the objects size table (Figure 2).

Target: The average of the 10% lowest pixel values inside the bounding box.

The relationship between the input and output implies that this model plays the role of size perspective. Thus, the model is named the size perspective model. As mentioned in Section 3.2 we used images that only contain a single object as training data.

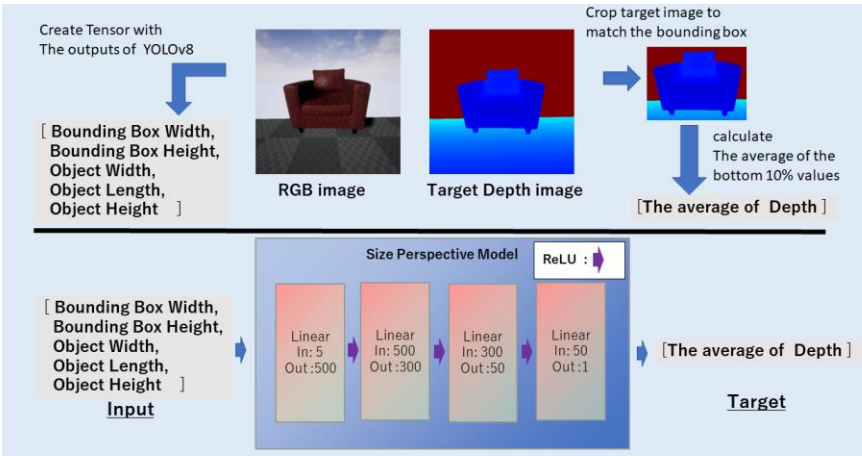

Figure 3 Size Perspective Model

The target values are the average of the 10% lowest pixel values. This is not simply an average because the bounding box includes the background, which typically has significantly higher depth values compared to the object parts. Using a regular average would lead to a misalignment with the actual values of the object parts.

3.6.Generate Pre Depth Image

As shown in Figure 4, a pre-depth image is generated by pasting the pre-depth value onto each segmentation region. The image size is kept the same as the RGB image, and the pixel values in the background region are set to 0. This image is used as an additional input for the depth estimation model.

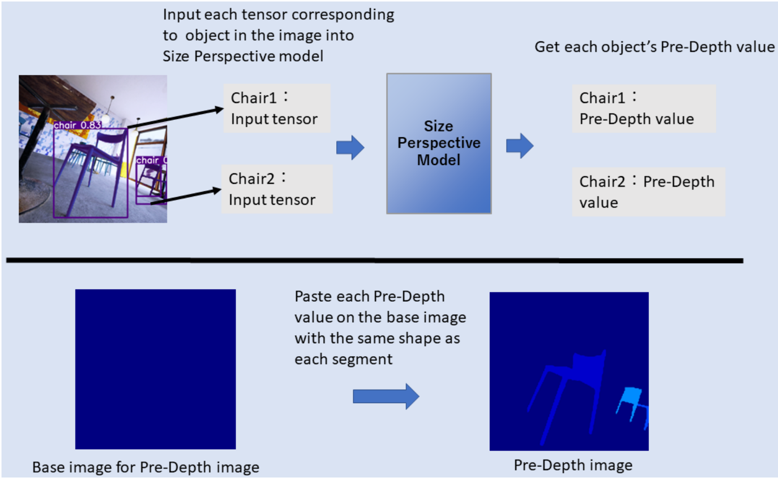

Figure 4 The process of generating pre-depth image

4. Experiments

To verify the improvement in accuracy through the proposed method, we compared the accuracy between the cases using pre-depth images for training the depth estimation model (our method) and not using pre-depth images.

4.1. Implementation of the Network

The depth estimation model, which takes only RGB images as input, is illustrated in Figure 5(a).As an image feature extractor, ResNet18 [16] is utilized (The last average pooling layer and linear transformation layer of the original ResNet have been removed), and thereafter, 3x3 convolutions, batch normalization, ReLU, and upsampling layer are repeated until the output size matches the target size. In the final stage, the upsampling mode is changed to bilinear interpolation. When using the pre-depth images(Figure 5(b)), an additional network, "Pre-depth Image Encoder", that extracts features from these pre-depth images is added. The results of this extraction are then merged with the upsampling process of the RGB image. Pytorch [17] was used to implement these networks.

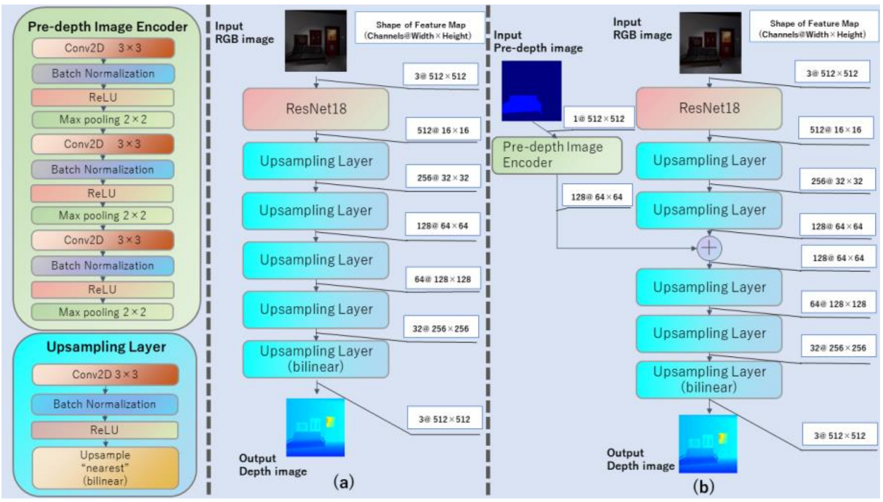

Figure 5 Network architecture use only RGB image (a), RGB and pre-depth image (b)

4.2.Training

In the training process of the size perceptive model, the following settings are configured.

Epoch size : 50
Batch size : 32

Optimizer : Adam (The initial learning rate was set to lr = 0.001)

Loss Function：MSELoss

A total of 5280 training data and 1320 evaluation data were prepared for the size perspective model.

In the training process of the depth estimation model, the dataset (mentioned in III.I) was used, and the following settings is configured.

Epoch size : 25
Batch size : 8

Optimizer : Adam (The initial learning rate was set to lr=0.001, and a scheduler was used to perform a learning rate decay of 0.8 at the epochs [5, 10, 15, 20, 22].),

Loss Function：Inverse Huber loss (referring to the study by I. Laina et al. [1]).

A total of 1000 training data and 200 evaluation data were prepared for the depth estimation model.

4.3.Evaluation

As evaluation metrics, MSE loss, Inverse Huber loss, and following evaluation metrics (δ_1,δ_2,δ_3) as an image accuracy are adopted. where y_i and $\hat{y}_i$ are respectively the ground truth and the prediction, and card is the cardinality of a set. This evaluation metric (δ_1, δ_2, δ_3) is a commonly used measure in prior studies [1][4][5][6][9]. It evaluates how close each predicted value is to the target value for the entire image by setting a threshold and indicates the proportion of values that meet this criterion. A larger value indicates better accuracy.

$$\delta_i = \frac{\mathbf{card}\left(\left\{\hat{y}_i : \max\left\{\frac{\hat{y}_i}{y_i}, \frac{y_i}{\hat{y}_i}\right\} < 1.25^i\right\}\right)}{\mathbf{card}\left(\{y_i\}\right)}$$

y_i: Each pixel value of the output image
$\hat{y}_i$: Each pixel value of the ground truth image

4.4. Result

The result of the size perspective model for the evaluation data is MSE Loss = 1017.48. The loss is not extremely small, but overall, the predicted values were within an error range of around 20 to 30 compared to the target pixel values.

Table1,2 shows the experimental results of depth estimation models. Each value in the table represents the average of 30 experimental runs. It has significant differences by t-test.

Table 1 Exprememtal results MSE Loss and Inverse Huber Loss

	MSELoss	Inverse Huber Loss
RGB	0.134	1.114
Ours	0.123	1.046

Table 2 Exprememtal results accuracy

	δ_1	δ_2	δ_3
RGB	73.99 [%]	89.69 [%]	94.66 [%]
Ours	77.01 [%]	91.33 [%]	95.61 [%]

Table 1,2 shows that there are improvements for MSELOSS, Inverse Huber loss, and accuracy.

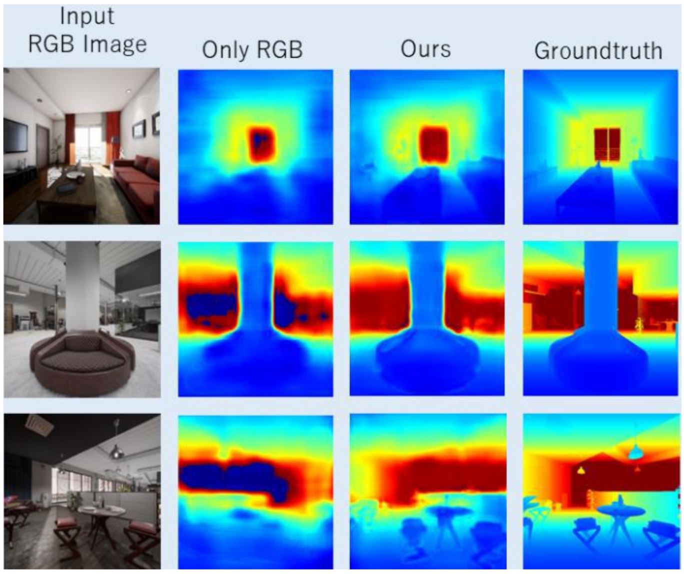

Figure 6 Example of generated depth image

Figure 6 shows an example of generated depth images. The pre-depth image is also one kind of segmentation image and the example images represent clearer contours of the objects in the regions where the objects appear.

5. DISCUSSION

The experimental results indicate that the proposed method can achieve improved accuracy of the model. It implies that the successful use of the pre-depth images during the image reconstruction stage within the model contributed to the simplification of the task. On condition that the number of input channels is changed from 3 to 4 in order that the pre-depth image can be directly entered into the model from the beginning, the accuracy decreased compared to the experimental results in Table 1,2. It suggests the model processed the pre-depth images with excessive parameters or batch normalization. It might be essential to preserve some of the pre-depth image features and make the model rely on them to achieve better results. When the batch size was changed from 8 to 16, the accuracy also decreased. This might be due to larger batch sizes leading to the loss of individual features of each pre-depth image through batch normalization, thereby reducing the impact of hints in the training process. Depending on the dataset used, these hyperparameters and model architecture may need to be adjusted.

The following Table 3 presents the results of the accuracy evaluation for the segmentation parts containing objects.

Table 3 Accuracy for segmentation area

	δ_1	δ_2	δ_3
RGB	58.05 [%]	81.72 [%]	91.30 [%]
Ours	60.39 [%]	84.56 [%]	93.36 [%]

Table 3 shows that the proposed method improves the accuracy in all cases. But it also shows a significant decrease in accuracy compared to Table 1 and Table 2. As anticipated in Section 1, It suggests there is more randomness in the depth variation patterns in the regions where objects are captured compared to the background. Additionally, using a dataset with fewer regions containing objects compared to the background (only 17% of the validation dataset contains object regions) led the model to focus more on estimating the background parts, which may cause a decrease in accuracy compared to the results in Table 1 and Table 2. Nevertheless, the proposed method manages to mitigate this decrease, Therefore, the result suggests that the approach of separating and estimating the depth of objects and background, as in the proposed method, might be more effective than usual depth estimating. Alternatively, a step-by-step generation approach, such as the Diffusion Model could also be effective.

Table 4 shows the accuracy of the generated image's object-occupied regions when the values of the object-occupied regions in the pre-depth image were adjusted to be closer to the ground truth.

Table 4Accuracy for segmentation area of Ours, Target, GT

	δ_1	δ_2	δ_3
Ours	60.39 [%]	84.56 [%]	93.36 [%]
Target	68.65 [%]	89.15 [%]	95.51 [%]
GT	85.09 [%]	95.49 [%]	98.04 [%]

("Target": using target value of validation data for size perspective model as pre-depth values.
"GT": using the ground truth values as pre-depth values.)

Table 4 shows that using pre-depth images closer to the GT values improves the accuracy. Therefore, the result indicates that the improvement in pre-depth estimation leads to an enhancement in accuracy.

The experiments demonstrated that obtaining the depth of the area occupied by objects beforehand through Pre-Depth Estimation using the Size Perspective Model facilitated easier estimation of the depth map of objects and improved the estimation accuracy.

From this point, we discuss the current limitations and points to be improved regarding Pre-Depth Estimation. Firstly, the current Size Perspective Model estimates depth based on the size of the bounding box and the actual size of the object, without utilizing the image containing the object. Depending on factors like the capture angle, a larger bounding box size doesn't necessarily imply a shallower depth. Moreover, since images are not used as inputs, it's impossible to obtain pixel-level Pre-Depth estimations. For a more refined Pre-Depth Estimation, we believe it's necessary to develop an architecture that includes images of detected objects as input. However, the image sizes of detected objects can vary widely, some being very small, Therefore, we need to develop an architecture that takes this into consideration. Next, our approach assumes that there are a certain number of detectable objects within the image. In cases where this assumption doesn't hold, the Pre-Depth Image might end up being an image consisting solely of zeros, which could potentially have a negative impact on the estimation process. Similarly, if the area occupied by detected objects is too small or if the Pre-Depth values have significant discrepancies, it could also negatively affect the results. In future research, we intend to address these challenges by revising the architecture of the Size Perspective Model and considering conditioning (such as not using the Pre-Depth Image when there are no detected objects.) on input data for the Depth Estimation Model

6. CONCLUSION

In this paper, we proposed a method for depth estimation using the size perspective. In this study, the proposed method improved the accuracy of depth image generation from monocular RGB images using deep learning. Size perspective enables depth estimation based on universal rules, allowing the separation of background and objects for estimation. Size Perspective Model estimates pre-depth and pre-depth images are generated from pre-depth. Then, pre-depth images made the estimation task easier by serving the role of hint. The proposed method successfully enhanced the accuracy of the generated images. However, the accuracy of the pre-depth estimation is insufficient, and its poor performance may adversely affect the model's training. Therefore, improvement in the accuracy of the pre-depth estimation is necessary. Specifically, modifications to the Size Perspective Model architecture are necessary to input detected object images and address cases where no detected objects in the image. Additionally, there is a need for innovations to bridge the accuracy gap between the background and object regions. We will continue research to address these issues and make further improvements.

REFERENCES

[1] I. Laina & C. Rupprecht et al., (2016) "Deeper depth prediction with fully convolutional residual networks", 2016 Fourth International Conference on 3D Vision (3DV), pp. 239 – 248, doi: 10.1109/3DV.2016.32

[2] Yasasvy Tadepalli & Meenakshi Kollati et al., (2021) "EfficientNet-B0 Based Monocular Dense-Depth Map Estimation" IIETA Vol. 38, No. 5, pp. 1485-1493, https://doi.org/10.18280/ts.380524

[3] A. C. Kumar & S. M. Bhandarkar & M. Prasad, (2018) "Monocular Depth Prediction Using Generative Adversarial Networks", IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, pp.413-4138, doi: 10.1109/CVPRW.2018.00068

[4] Saurabh Saxena & Abhishek Kar & Mohammad Norouzi & David J. Fleet,(2023) "Monocular Depth Estimation using Diffusion Models" arXiv:2302.14816

[5] Ranftl Rene& Lasinger Katrin & Hafner David & Koltun Vladlen, (2020)"Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer.", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1-1,doi:10.1109/TPAMI.2020.3019967.

[6] David Eigen & Christian Puhrsch & Rob Fergus, (2014). "Depth map prediction from a single image using a multi-scale deep network." Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, pp.2366–2374.

[7] P. Liu, Z. Zhang & Z. Meng & N. Gao, (2022) "Deformable Enhancement and Adaptive Fusion for Depth Map Super-Resolution," IEEE Signal Processing Letters, vol. 29, pp.204-208, doi: 10.1109/LSP.2021.3132552

[8] J. Dai et al., (2017) "Deformable Convolutional Networks," IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 764-773, doi: 10.1109/ICCV.2017.89.

[9] Dong-hoon Kwak & Seung-ho Lee., (2020) "A Novel Method for Estimating Monocular Depth Using Cycle GAN and Segmentation" Sensors 20, no. 9: 2567, https://doi.org/10.3390/s20092567

[10] H. Lim & H. Gil & H. Myung, (2020) "MSDPN: Monocular Depth Prediction with Partial Laser Observation using Multi-stage Neural Networks", 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, pp.10750-10757, doi:10.1109/IROS45743.2020.9340767.

[11] Y. Liao & L. Huang& Y. Wang& S. Kodagoda & Y. Yu & Y. Liu, (2017) “Parse geometry from a line: Monocular depth estimation with partial laser observation”, 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 5059–5066.

[12] F. Ma & S. Karaman, (2018) “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,”, 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–8.

[13] Glenn Jocher & Chaurasia Ayush & Jing Qiu, (2023) "YOLO by Ultralytics", https://github.com/ultralytics/ultralytics

[14] Epic Games, (2019) “Unreal Engine”, Available at: https://www.unrealengine.com.

[15] Thang To & Jonathan Tremblay & Duncan McKay & Yukie Yamaguchi & Kirby Leung, Adrian Balanon & Jia Cheng & William Hodge & Stan Birchfield, (2018) “: {NVIDIA} Deep Learning Dataset Synthesizer”, https://github.com/NVIDIA/Dataset_Synthesizer

[16] K. He & X. Zhang & S. Ren & J. Sun, (2016) "Deep Residual Learning for Image Recognition,", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778. doi:10.1109/CVPR.2016.90

[17] A. Paszke & S. Gross& F. Massa & A. Lerer & J. Bradbury & G. Chanan&T. Killeen & Z. Lin& N. Gimelshein & L. Antiga A. Desmaison & A. Köpf & E. Yang & Z. DeVito & M. Raison & A. Tejani & S. Chilamkurthy & B. Steiner &L. Fang& J. Bai & S. Chintala, (2019) “PyTorch: An Imperative Style, High-Performance Deep Learning Library” Neural Information Processing Systems 32, pp. 8024–8035

Authors

Takanori Asano is a graduate student of Graduate School of Shibaura Institute of Technology. He obtained his bachelor's degree in the Department of Electrical Engineering at Shibaura Institute of Technology and currently research on deep learning, image processing, and related fields.

Yoshiaki Yasumura receives the PhD degree from Osaka University in 1998. He is a professor of Electrical Engineering at Shibaura Institute of Technology. His research interests include machine learning, image processing, and natural language processing.