IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 April 2023, accepted 24 April 2023, date of publication 1 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272292

RESEARCH ARTICLE

TP-GAN: Simple Adversarial Network With Additional Player for Dense Depth Image Estimation

ANDI HENDRA AND YASUSHI KANAZAWA, (Member, IEEE)
Department of Computer Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

Corresponding author: Andi Hendra (andi.hendra.dt@tut.jp)

This work was supported in part by C-BEST Program of the Japan International Cooperation Agency (JICA), and in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP20K06319.

ABSTRACT We present a simple yet robust monocular depth estimation technique by synthesizing a depth map image from a single RGB input image using the advantage of generative adversarial networks (GAN). We employ an additional sub-model termed refiner to extract local depth features, then combine it with the global scene information from the generator to improve the GAN's performance compared to the standard GAN architectural scheme. Notably, the generator is the first player to learn to synthesize depth images. The second player, the discriminator, classifies the generated depth. In the meantime, the third player, the refiner, enhances the final reconstructed depth. Complementing the GAN model, we apply a conditional generative network (cGAN) to lead the generator in mapping the input image to the respective depth representation. We further incorporate a structured similarity (SSIM) as our loss function for the generator and refiner in GAN training. Through extensive experiment validation, we confirmed the performance of our strategy on the publicly indoor NYU Depth v2 and KITTI outdoor data. Experiment results on the NYU depth v2 dataset show that our proposed approach achieves the best performance by 96.0% on threshold accuracy ($\delta < 1.25^2$) and the second-best accuracy on all thresholds on the KITTI dataset. We discovered that our proposed method compares favorably to numerous existing monocular depth estimation strategies and demonstrates a considerable improvement in the accuracy of image depth estimation despite its simple network architecture.

INDEX TERMS Depth estimation, single image, conditional GAN, generative adversarial network (GAN), third player GAN.

I. INTRODUCTION

Estimating depth from a single image is a fundamentally challenging task and a wide area of research in computer vision. Knowledge of the scene depth information has been applied in many vision applications such as 3-D modeling [1], robotics [2], and autonomous driving [3], as well as potentially leading to improve related studies in pedestrian detection tasks [4], [5], [6].

Significant progress has been achieved in obtaining depth information from a single image using machine learning. Various approaches have accomplished remarkable improvements in extracting depth information using Convolutional Neural Networks (CNN). Several techniques have made significant advancements in extracting deep information, whether supervised [7], [8], [9], [10], semi-supervised [11], [12], [13] or unsupervised [14], [15], [16]. The first impressive single image depth estimation based on CNN, Eigen et al. [7] estimated depth information using two independent deep neural networks. One makes a broad global prediction, while the other offers a more precise local prediction.

In the meantime, the generative adversarial network (GAN) has significantly improved the learning of mapping high-dimensional data distributions. It has been demonstrated that a generative adversarial network is highly effective at capturing the global structure of a scene and producing realistic images. In the adversarial network, the generator model (G) is responsible for reconstructing newly

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

synthesized images. At the same time, the discriminator (D) evaluates the probability that a given input image is either derived from training data or is synthetically generated. On top of its excellent performance in constructing synthetic photo realistic [17], the adversarial model has also been utilized for image-to-image translation tasks such as image segmentation [18] and, more recently, for single image depth prediction [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. However, there is still room for improvement in accurately reconstructing depth information.

There are likely several challenges in this area, including the fact that it is computationally expensive, requires large data to train, generates poor depth image reconstruction, and causes a problem with uncertainty. An example of a typical symptom is a blurred reconstruction of object boundaries or distortion of a certain object in the scene. The motivation of our work is to utilize contextual global image structure and local feature information for better depth reconstruction results. The global feature describes the scene as a whole in order to understand the global relations between pixels in the image. For illustration, a patch of white pixels in the indoor dataset could represent a wall in the distance or a nearby white chair. On the other hand, the local feature information is necessary to align with the local details of the object in the scene.

We explore the benefit of the adversarial network, which has been demonstrated to be effective in capturing global scene structure with fewer training data than a standard encoder-decoder CNN. We utilize a conditional GAN (cGAN) to provide additional information to boost the model performance, allowing it to converge faster and reduce training time significantly. In particular, we propose a three-player GAN (TP-GAN) that uses an additional sub-model (refiner) to complement the cGAN performance. The broad idea here is that the generator will extract the global scene layout while the refiner learns to improve depth structure by integrating updated weight from the generator with local scene information and expressing feedback from the discriminator throughout each mini-batch training session. Hence, our strategy concurrently integrates global scene structure and local scene information to enhance the performance of the adversarial network for a single image depth estimation. In addition, the SSIM loss will further evaluate the structural feature similarity rather than pixel-by-pixel between two images, which is a more effective strategy for image reconstructing tasks, including image depth estimation. Fig. 1 shows the overview of the schematic of our proposed model architecture, to be described in detail in the next section.

The remainder of this paper is organized as follows. Section II reviews several related works. The theory and procedure of the proposed method are described in Section III. Section IV shows the implementation of our strategy. We describe our setup for the experiment in Section V. The result of our method is discussed in Section VI, including a comparison with the previous works. Finally, the paper is concluded in Section VII.

II. RELATED WORKS

Several techniques for extracting depth information using Convolutional Neural Networks (CNN) have been developed in recent years. The related works are addressed in the following paragraphs.

A. NON-ADVERSARIAL BASED MODELS

Eigen et al. [7] estimated depth information from a monocular image using a multi-scale structure that stage-wisely refines the estimated depth map from low to high spatial resolution via independent networks. Following this seminal work, Liu et al. [29] discover the unary and pairwise potential of continuous Conditional Random Field (CRF) and train it using a CNN. Laina et al. [30] proposed a fully convolutional architecture to learn feature map up-sampling to generate higher resolution output dense maps. Godard et al. [11] considered unsupervised learning for a monocular image depth estimation using a deep CNN network. Their studies constructed disparity images using a left-right consistency image reconstruction loss.

Later on, Chen et al. [8] presented a residual pyramid decoder (RPD) that takes into account the underlying image structure at many scales. Yin et al. [9] introduced a framework that consists of two primary modules; a depth prediction and a point cloud module, to improve the structure of point clouds derived from depth maps in order to recover more accurate 3-D shape from a single image. Gur et al. [14] proposed a deep learning-based method to estimate depth from a single image based on depth focus cues. In their method, the model requires at least one focused image of the same scene from the same viewpoint. Bian et al. [15] proposed an Auto-Rectify network to enhance unsupervised depth estimation by removing relative rotational motions in addition to their innovative loss functions. Eventually, Ye et al. [10] introduced a transformer framework for multi-task dense prediction. They used an inverted pyramid multi-task transformer (InvPT) to learn long-range interaction in both spatial and all-task contexts in a unified architecture. Subsequently, studies on enhancing the quality of depth information using deep learning have been readily conducted.

B. ADVERSARIAL BASED MODELS

Generative adversarial network (GAN) [17], also known as two players deep learning network, have already been explored for depth estimation. To mention a few, Aleotti et al. [24] introduced monocular depth estimation based on the intrinsic ability of GAN to detect inconsistencies in images. In their research, the generator network learns to estimate depth from the reference image to generate a warped target image. Simultaneously, the discriminator learns to differentiate between generated depth and target ground truth.

Several more studies are then presented to improve depth estimation based on the benefits of adversarial networks for image reconstruction tasks. Zheng et al. [27] proposed a two-module domain adaptive network with a generative

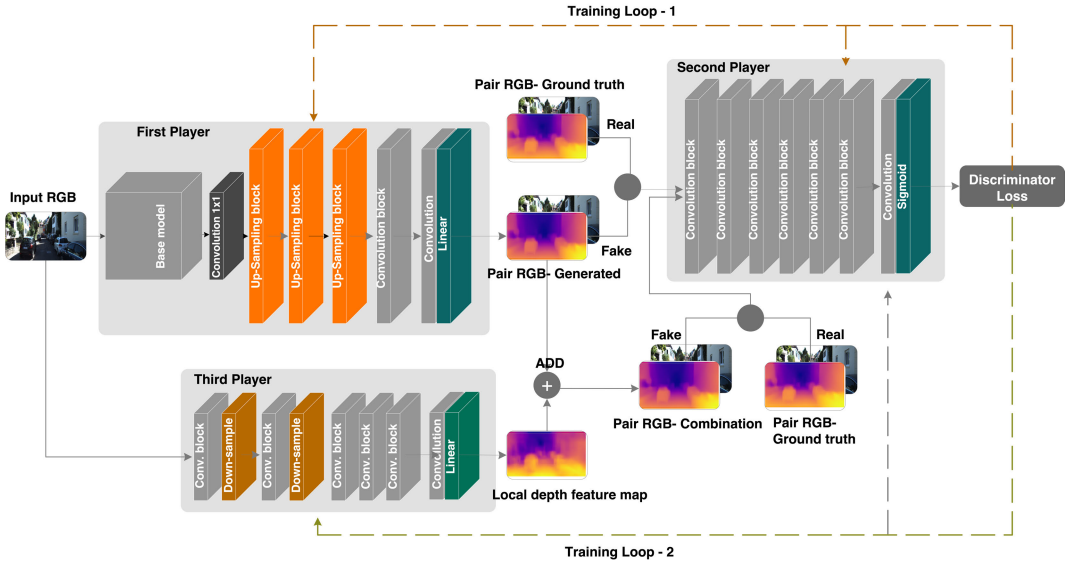


FIGURE 1. Outline of our proposed TP-GAN method.

adversarial loss to map real and synthetic images to the real domain. Kumar et al. [26] presented an adversarial network-based model in which their generator network consists of depth and relative object pose in addition to their adjustable loss functions. Subsequently, Pilzer et al. [25] and Kwak et al. [28] explored unsupervised deep learning depth generation based on a cycled generative adversarial network. Their model estimates a disparity map given input left and right images from a calibrated stereo camera. Recently, Zhao et al. [19] developed a Masked GAN framework for monocular depth estimation and ego-motion utilizing their scale-consistency loss.

However, the methods mentioned earlier, their depth network either requires a complex network, focuses on capturing local information, or simply obtains scene structure globally without taking scene local features into account. In addition, they merely limit the consistency of values across depth maps while ignoring the consistency of image structures, resulting in poor performance. In this work, we use the advantage of the adversarial learning-based model for image generation tasks utilizing SSIM loss. Our proposed approach employs a conditional GAN (cGAN) [31], in which both the generator and the discriminator are conditioned on some extra information.

Motivated by these insights and the impressive performance of the conditional adversarial network (cGAN) model, including the ability to converge faster than the standard CNN, we present a novel single image depth estimation by expanding the cGAN model into a three-player GAN (TP-GAN). In our model, the residual networks (ResNet) proposed by He et al. [32] were implemented as the backbone for our generator (G) sub-model. Our second sub-model, discriminator (D), is designed as a patch GAN model encouraged by [18] that only penalizes structure at the scale of local image patches in the $N \times N$ output vector as opposed to outputting a single value indicating whether an image is fake or real. Then, we stack six convolutional layers to capture local feature information in our refiner (R) sub-model, later referred to as the third player in our GAN model.

III. PROPOSED METHOD

We propose a simple architecture for a single image depth prediction based on an adversarial network, implemented as three sub-models instead of standard two players. We study the advantages of incorporating an additional sub-model into the cGAN architecture to improve the prediction by incorporating global scene structure and local image information along with the structure similarity (SSIM) loss.

We elaborate on our proposed strategy. First, we define the outline of our technique by formulating problems, and then we describe each sub-model in our three-player conditional adversarial deep learning network in detail. Next, we specify the reconstruction loss and explain our SSIM loss function, which analyzes the structural difference between the predicted and ground truth depth.

A. PROBLEM FORMULATION

In this research, we utilize adversarial learning advantages to formulate the problem of learning depth from monocular inputs as an image translation problem. While the discriminator discovers how to distinguish between ground truth and synthetic depth maps, the generator learns how to create more realistic depth maps. In fact, the generator continuously seeks the output that appears plausible to the discriminator.

Our proposed adversarial model is a conditional generative adversarial neural network (cGAN) to assist the generator and refiner in mapping input images to their respective depth representation. This network consists of three sub-models: a generator as the first player, a discriminator as the second

player, and an additional refiner sub-model that we refer to as the third player. The refiner will be responsible for fine-tuning the locally generated depth prediction with the global scene information.

The proposed technique updates the generator weight by back-propagating through the discriminator during adversarial training. Meanwhile, the refiner combines the updated weight of the generator and then forwards it to the discriminator model for each mini-batch training. Further details are discussed in the succeeding subsections.

1) THE 1st PLAYER: GENERATOR

We reconfigured the residual network (ResNet) [32] structure in the generator as our backbone model, which has been demonstrated effective in improving the accuracy of depth prediction from a single image [15], [30], [33]. Then, we stacked some block layers to receive input from the previous layer; the first block is a convolution layer with 1×1 convolution kernels to capture a global view of the scene. The remaining blocks consist of transpose-convolution (up-conv-activation), followed by regular convolution block (conv-batch-activation) with {1024, 512, 256, 128}, and 64 filters, respectively. We utilized bilinear interpolation for our up-sampling, while Leaky ReLU activation was employed to minimize the vanishing gradient. The final depth extraction output layer has a linear activation function. The specifics of our generator model are depicted in Fig. 2 (a).

2) THE 2nd PLAYER: DISCRIMINATOR

Figure 2 (b) shows the detail of our discriminator model. This structure is encouraged by the work of Isola et al. [18], implemented as a patch GAN, which looks at the structure of local image patches and classifies each patch in an image as real or fake in the $N \times N$ output vector. Since the generator output is conditioned on the input, it is important to maintain the discriminator input image in the mix. Our discriminator, a conditional adversarial model, comprises pair of images as input: the RGB image and its ground truth depth and the RGB image and its corresponding generated image depth. Each of which is size 48×64 for NYU and 40×128 for KITTI data.

We concatenate the RGB with its depth before fusing them into the network. We modify parameter values using 4×4 kernel size and strides-2 except in the last two layers with {64, 128, 256, 512, 512, 1} filters, respectively. Batch-normalization is applied in all layers but in the first and last layers. At that, in the last layer, the convolution is utilized to map to a one-dimensional output with a size of 3×4 pixels, followed by a sigmoid activation function. The model output will be a probability of classifying whether the input patch images come from training or generated data.

3) THE 3rd PLAYER: REFINER

As shown in Fig. 2 (c), the refiner model in our architecture is a sequence of six block layers. The first five blocks are a stack of convolution, batch normalization, ReLU activation, and dropout regularization (conv-batch-activation-dropout) to handle the overfitting problem with 64 filters. We use 7×7 kernel size and strides-2 to down-sample our input in the first and second blocks, while the following three blocks use 5×5 kernel size and stride-1. With a small enough kernel size relative to the input, the extracted feature will not depend on the value of the whole pixel in the input image. Since the receptive field is smaller than the size of the input image, extracted features will only depend on the local pixels. The last block is a convolution layer with a filter number of one and 5×5 kernel size, followed by a linear activation to capture the depth of local features.

B. DEPTH RECONSTRUCTION LOSS AND LOSS FUNCTION

The discriminator is trained to maximize the predicted probability of real images and the inverted probability of deceptive images throughout training. The generator, on the other hand, works to maximize the log of the predicted probability of discriminator for counterfeit images. In addition, the refiner utilizes to improve the generator result as feedback from the discriminator. We set our depth reconstruction loss in Eq. (1).

$$\begin{aligned}\min_{G,R}\max_{D}(G,R,D) = {} & \mathbb{E}_{x,y}[\log D(x,y)] \\ & + \mathbb{E}_{x}[\log(1 - D(x, G(x)))] \\ & + \mathbb{E}_{x}[log(1 - D(x, R(x, G(x))))],\end{aligned} \tag{1}$$

where $D(x, y)$ is the discriminator from the input RGB image x with conditional target depth image y. $G(x)$ is the generator output when given input data x, and $R(x, G(x))$ is the refiner output that comes from the generator and real data x.

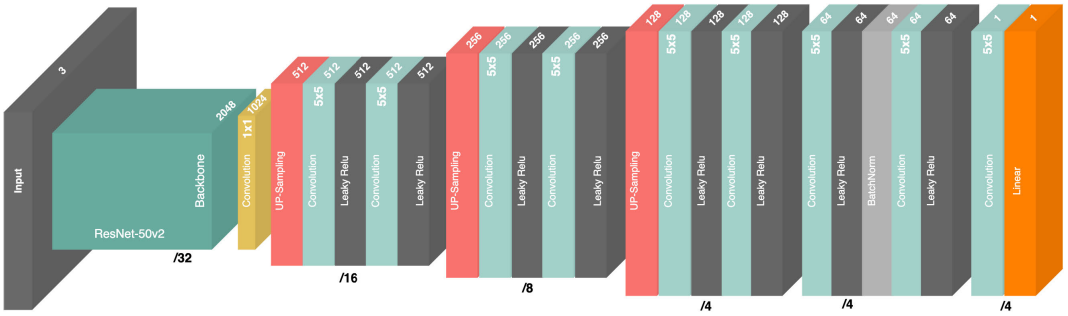
In general, the Mean Squared Error (MSE) or Mean Absolute Error (MAE) is taken as the standard loss for regression tasks to calculate the discrepancies between prediction and target outputs. Similar to MAE, MSE computes the error between two images by comparing pixel by pixel as defined in Eq. (2). On the other hand, the Structural Similarity Index (SSIM) measurement analyzes the structural difference between two images. This structural information signifies the idea that neighboring pixels have strong inter-dependencies with one another, which is a more effective strategy for image reconstruction tasks.

$$\mathrm{MSE}(y_t, y_p) = \frac{1}{N}\sum_{y_p \in |N|} |y_p - y_t|^2 \tag{2}$$

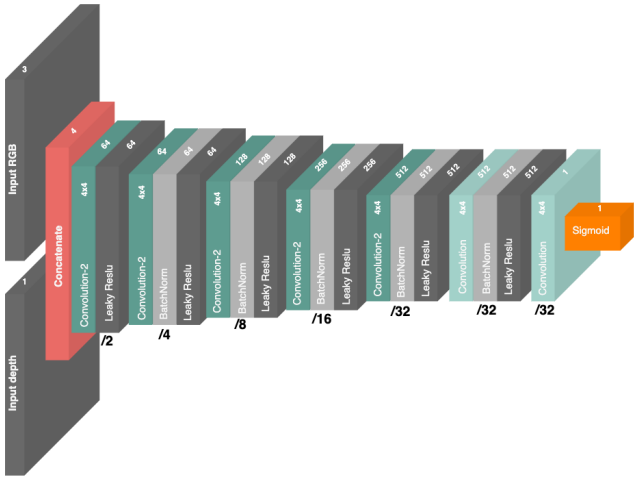
The SSIM formula, as expressed in Eq. (3), was introduced by [34], which comprises three parameter comparison measurements: luminance, contrast, and structure.

$$\mathrm{SSIM}(y_t, y_p) = \frac{(2\mu_{y_t}\mu_{y_p} + c_1)(2\sigma_{y_t y_p} + c_2)}{(\mu_{y_t}^2 + \mu_{y_p}^2 + c_1)(\sigma_{y_t}^2 + \sigma_{y_p}^2 + c_2)} \tag{3}$$

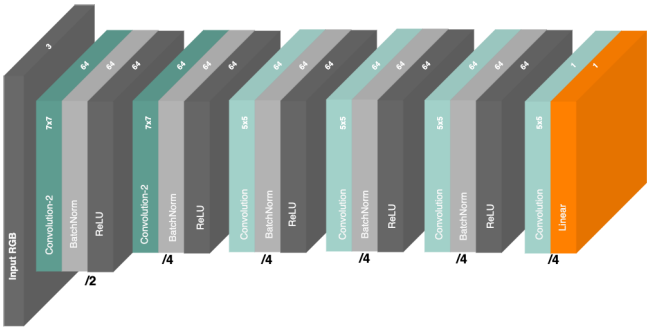
In contrast to the MSE or MAE, the SSIM score range from -1 and 1, with 1 indicating perfect similarity. We use SSIM loss (L_s) in Eq. (4) for our generator and refiner while training our adversarial network. Eventually, the SSIM loss will compute the perceptual difference based on the visible structure



(a) Layers in the generator (1st player) network.



(b) Layers in the discriminator (2nd player) network.



(c) Layers in the refiner (3rd player) network.

FIGURE 2. Layers for each sub-model in our proposed TP-GAN model.

of the ground truth and predicted image.

$$L_s = 0.5 - \frac{\text{SSIM}(y_t, y_p)}{2}, \tag{4}$$

With:
μ_{y_t} and μ_{y_p} are the mean of y_t and y_p, respectively.
$\sigma^2_{y_t}$ and $\sigma^2_{y_p}$ are the variance of y_t and y_p, respectively.
$\sigma_{y_t y_p}$ is the covariance of y_t and y_p.

IV. IMPLEMENTATION

This section describes the implementation of our proposed method in detail. First, we define the datasets used in this experiment and then explain the training process we conducted in detail. To confirm the efficacy of our proposed study, we present the standard assessment metric utilized in previous related studies.

A. DATASETS

We train our model using two popular publicly available depth datasets, indoor NYU Depth v2 [35] and outdoor KITTI data [36], commonly used in the area of depth estimation.

KITTI data contains outdoor scenes with images resolution of roughly 376 × 1241 captured by cameras and depth sensors in a driving car. It contains over 93K depth maps from 56 scenes with corresponding raw LiDAR scans and RGB images. We train our method on 25K images from the random scene and set depth upper bound of 80 meters. We test our model on the 697 images which are not included in the training, following the split by the work of Eigen et al. [7].

NYU Depth v2 is an indoor dataset gathered using a Microsoft Kinect camera with a resolution of 640 × 480. It contains about 120K raw RGB images and their corresponding depth from 464 different scenes. Only 50K images from random scenes are used for training our network. To validate the performance of our method, following the works of Eigen et al. [7], we test on 654 from 1449 available densely labeled pairs of aligned RGB and ground truth depth images in the maximum depth of 10 meters.

B. TRAINING DETAILS

We implement our adversarial depth estimation network based on deep learning Tensorflow [37] and Keras framework [38]. The training is done on Ubuntu 16.04 and an NVIDIA GeForce GTX 1080 GPU with 8 GB memory. The network architecture shown in Fig. 2 has been trained using initialized random weight. The training is performed using 16 mini-batches and load images and their corresponding depth using an online generator for GPU memory performance.

We practice on-the-fly data augmentation procedures to enrich the features of our inputs during training. Specifically, we randomize the channels of the input RGB images using a ratio of 0.5. We then apply a 0.25 ratio to our input RGB images to implement Poisson noise. We utilize mirroring techniques at a probability of 0.5 for both RGB images and depths. Another geometry-preserving affine transformation for RGB images and their corresponding depths, a horizontal flipping strategy, is also applied at a probability of 0.25.

Since our strategy is an adversarial model, the generator model was not trained independently and instead had its weight updated by the loss of the discriminator. On the other hand, the refiner model is updated by the previous generator weight as well as the discriminator feedback for every input batch.

In our approach, we train our model for 50 epochs using an adaptive moment estimation (Adam) optimizer with the exception of the discriminator, which uses Stochastic Gradient Descent (SGD) as encouraged in the works [39]. We started with a learning rate of 2×10^{-4} for the generator and refiner, while for the discriminator, we initialized 4×10^{-4} and periodically adjusted as the training progressed using an exponential rate decay of 0.5 and 0.999 for 1^{st} and 2^{nd} momentum, respectively.

C. EVALUATION

We validate the performance of our proposed depth estimation method on publicly available RGB-D NYU Depth v2 and KITTI datasets by evaluating our model compared with several relevant studies. In order to objectively assess the efficacy of our depth prediction model, we employ the following evaluation metrics, which have been widely employed in prior research. Specifically, we assess our method using metrics based on its error rate and accuracy in Eqs. (5), (6), (7), (8), (9), and (10).

- Root mean squared error (RMS):
  The standard deviation of the prediction errors to measure the difference between predicted (y_p) and the ground truth data (y_t).
$$\sqrt{\frac{1}{N}\sum_{y_p \in |N|} |y_p - y_t|^2}. \tag{5}$$
- Average $\log_{10}$ error (LOG10):
  The average of the absolute error of the log-transformed predicted (y_p) and log-transformed ground truth values (y_t).
$$\frac{1}{N}\sum_{y_p \in |N|} |\log_{10}(y_p) - \log_{10}(y_t)|. \tag{6}$$
- Average relative error (REL):
  The ratio of the absolute error of the predicted (y_p) to the ground truth (y_t).
$$\frac{1}{N}\sum_{y_p \in |N|} \frac{|y_p - y_t|}{y_t}. \tag{7}$$
- Root mean squared log error (RMS LOG):
  The Root Mean Squared Error of the log-transformed predicted (y_p) and log-transformed ground truth values (y_t).
$$\sqrt{\frac{1}{N}\sum_{y_p \in |N|} |\log y_p - \log y_t|^2}. \tag{8}$$
- Squared relative error (SQ REL):
  The ratio of the squared error of the predicted (y_p) to the ground truth (y_t).
$$\frac{1}{N}\sum_{y_p \in |N|} \frac{|y_p - y_t|^2}{y_t}. \tag{9}$$
- Accuracy with threshold (P_{th}): percentage (%) of y_p to $\max(\frac{y_t}{y_p}, \frac{y_p}{y_t}) = \delta < P_{\text{th}}$, where:
$$P_{\text{th}} \in \left\{1.25, 1.25^2, 1.25^3\right\}. \tag{10}$$

Here, y_p and y_t are the predicted and ground-truth depth, respectively, and N is the total number of pixels. With the exception of the accuracy with threshold, lower numbers indicate higher performance for all metrics.

V. EXPERIMENT

We carried out several experiments using our network on the publicly available NYU Depth v2 [35] and KITTI dataset [36] using Tensorflow [37] and Keras Framework [38] to demonstrate the performance of our proposed method. We conducted more experiments with various tasks to observe our proposed method's achievement.

A. ABLATION STUDIES

We perform an ablation study to examine the proposed three-player adversarial with a non-adversarial model counterpart to discover the effectiveness of our proposed approach. We report the quantitative result in terms of accuracy in Tab. 1 on the outdoor KITTI dataset. We observe that the presence of the third sub-model improves the depth performance of the standard GAN model. Further improvement is found by utilizing Stochastic Gradient Descent (SGD) optimizer in the discriminator compared adaptive moment estimation (Adam) to all sub-models. Our proposed TP-GAN achieve greater improvement by utilizing the Structural Similarity Index Measure (SSIM) loss rather than the standard Mean Squared Error (MSE). The TP-GAN-ADAM-SGD-SSIM improves the standard GAN-ADAM-MSE accuracy by 3%, 1%, and 0.5% for the threshold $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$, respectively. These further demonstrate that the performance of our depth estimation is positively impacted by the presence of the refiner sub-model along with the SSIM loss in our TP-GAN. Additional ablation research to evaluate the influence of SSIM loss function is included in our supplementary materials.

TABLE 1. Ablation study on the outdoor KITTI data.

	Optimizers*				Accuracy Thresholds**		
	(G)	(D)	(R)	Loss	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Standard-GAN-ADAM-MSE	ADAM	ADAM	—	MSE	0.854	0.963	0.987
TP-GAN-ADAM-MSE	ADAM	ADAM	ADAM	MSE	0.869	0.969	0.990
TP-GAN-ADAM-SGD-MSE	ADAM	SGD	ADAM	MSE	0.880	0.971	0.991
TP-GAN-ADAM-SGD-SSIM	ADAM	SGD	ADAM	SSIM	**0.884**	**0.973**	**0.992**

*(G) generator, (D) discriminator, (R) refiner.
**The higher the better.

B. COMPARISON WITH EXISTING METHODS

We compare our proposed TP-GAN with some of the most notable single image depth estimation methods and report the accuracy and error rate results in Tabs. 2 and 3 for NYU Depth v2, and Tabs. 4 and 5 for KITTI data, respectively. To confirm an adequate and meaningful evaluation, we analyze the effectiveness of our model using the same dataset split validation technique as Eigen et al. [7].

1) NYU DEPTH v2

We examine our model with several previous adversarial networks and non-adversarial methods on NYU Depth v2, as shown in Tab. 2 and Tab. 3. In the adversarial approach, our accuracy performs slightly lower than [28] in the first threshold ($\delta < 1.25$) but perform better in other metrics with a significant margin. Compared to the non-adversarial based methods, our TP-GAN outperforms the preceding works [7], [10], [14], [40], [41], [42], [43] and achieves comparative performance, even better than [15], [44]. Here, ours achieves better than [44] on the thresholds ($\delta < 1.25$ and $\delta < 1.25^2$), and error rate performances (root mean square error (RMS) and the average log error (LOG10)). Whereas [15] performs better only in the first threshold ($\delta < 1.25$) with a small margin, our TP-GAN consistently improves performance in the other two thresholds ($\delta < 1.25^2$ and $\delta < 1.25^3$) and performs the lowest RMS with a large margin.

2) KITTI DATASET

We report the performance of comparison with several similar strategies on the KITTI dataset, both adversarial and non-adversarial. In terms of accuracy, as demonstrated in Tab. 4, our technique surpasses all nine previous adversarial works [19], [20], [21], [22], [23], [24], [25], [26], [27] as well as non-adversarial methods [7], [11], [12], [16], [45], [46], [47], [48], [49], [50] by significant margins for all the three thresholds $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$, but performs lower than the work in [51] with a small margin. In Tab. 5, we show our TP-GAN reliability in comparison to the previous related works in terms of the average relative error (REL), the squared relative error (SQ REL), and the root mean square log error (RMS LOG).

VI. DISCUSSION

We evaluate depth estimation on indoor NYU Depth v2 up to a maximum distance of 10 and 80 meters for the outdoor KITTI dataset. To show that our proposed adversarial approach benefits from the additional player in the GAN model, we specifically compare the performance with previous related works on adversarial methods.

TABLE 2. Accuracy comparison on NYU Depth v2. The best results are in bold and second best are underlined.

		Accuracy*		
	range [m]	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Adversarial Methods				
Zheng *et al.* 2018 [27]	1–10	0.540	0.832	0.948
Kwak *et al.* 2020 [28]	—	**0.834**	0.941	0.976
Non-adversarial Methods				
Eigen *et al.* 2014 [7]	0–10	0.611	0.887	0.971
Eigen *et al.* 2015 [40]	0–10	0.769	0.950	0.988
Wang *et al.* 2015 [41]	—	0.605	0.890	0.970
Roy *et al.* 2016 [42]	0–10	—	—	—
Chakrabarti *et al.* 2016 [43]	—	0.806	0.958	0.987
Li *et al.* 2019 [44]	—	0.788	0.958	**0.991**
Zhao *et al.* 2020 [50]	—	0.701	0.912	0.987
Gur *et al.* 2020 [14]	0–10	0.772	0.942	0.984
Bian *et al.* 2021 [15]	0–10	<u>0.820</u>	<u>0.956</u>	<u>0.989</u>
Ye *et al.* 2022 [10]	—	—	—	—
Ours	0–10	0.819	**0.960**	<u>0.989</u>

*the higher the better.

TABLE 3. Error rate comparison on NYU Depth v2. The best results are in bold and second best are underlined.

		Error Rate*		
	range [m]	RMS	LOG10	REL
Adversarial Methods				
Zheng *et al.* 2018 [27]	1–10	0.915	—	0.257
Kwak *et al.* 2020 [28]	—	0.652	—	—
Non-adversarial Methods				
Eigen *et al.* 2014 [7]	0–10	0.907	—	0.215
Eigen *et al.* 2015 [40]	0–10	0.641	—	0.158
Wang *et al.* 2015 [41]	—	0.824	—	0.220
Roy *et al.* 2016 [42]	0–10	0.774	—	0.187
Chakrabarti *et al.* 2016 [43]	—	0.620	—	0.149
Li *et al.* 2019 [44]	—	0.635	0.063	<u>0.143</u>
Zhao *et al.* 2020 [50]	—	0.686	0.079	0.189
Gur *et al.* 2020 [14]	0–10	0.546	0.063	0.149
Bian *et al.* 2021 [15]	0–10	0.532	**0.059**	**0.138**
Ye *et al.* 2022 [10]	—	<u>0.518</u>	—	—
Ours	0–10	**0.509**	<u>0.060</u>	<u>0.143</u>

*the lower the better.

A. NON-ADVERSARIAL MODELS

On the NYU Depth v2 dataset, our method achieves less accurate depth than the work of [15] in $\delta < 1.25$ by a small margin of 0.001, but performs better in the second threshold ($\delta < 1.25^2$) by a bigger margin (0.004). In terms of error performance, our TP-GAN obtains the lowest RMS error by a margin of 0.009 when compared with a very close

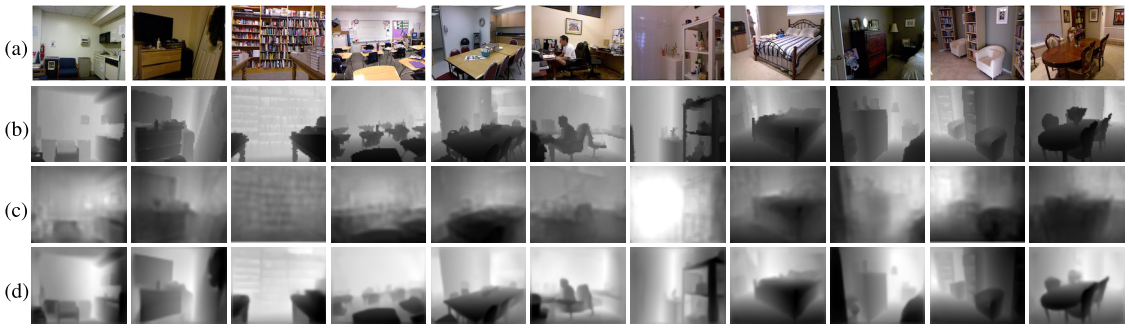


FIGURE 3. Depth Prediction on NYU Depth v2 qualitative results from top to bottom: (a) RGB image, (b) ground truth, (c) Eigen et al. [7], (d) Our TP-GAN.

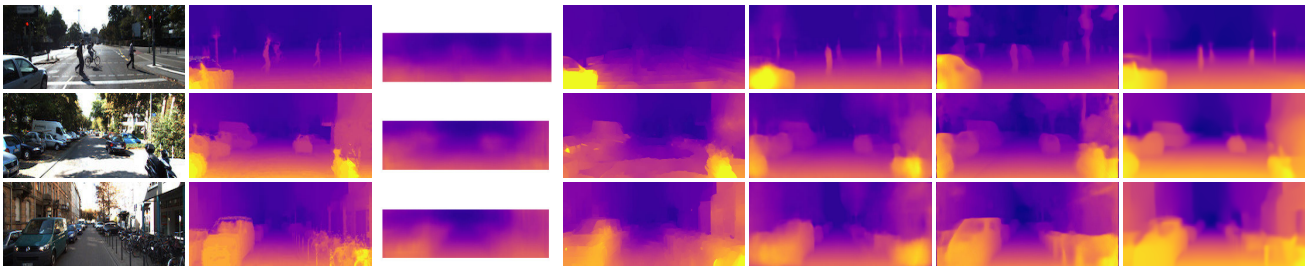


FIGURE 4. Qualitative comparison result on KITTI data. (a) RGB image, (b) ground truth, (c) Eigen et al. [7], (d) Liu et al. [45], (e) Kutznizov et al. [46], (f) Godard et al. [11], (g) Our TP-GAN. Our method can determine missing depth in the upper part of the image.

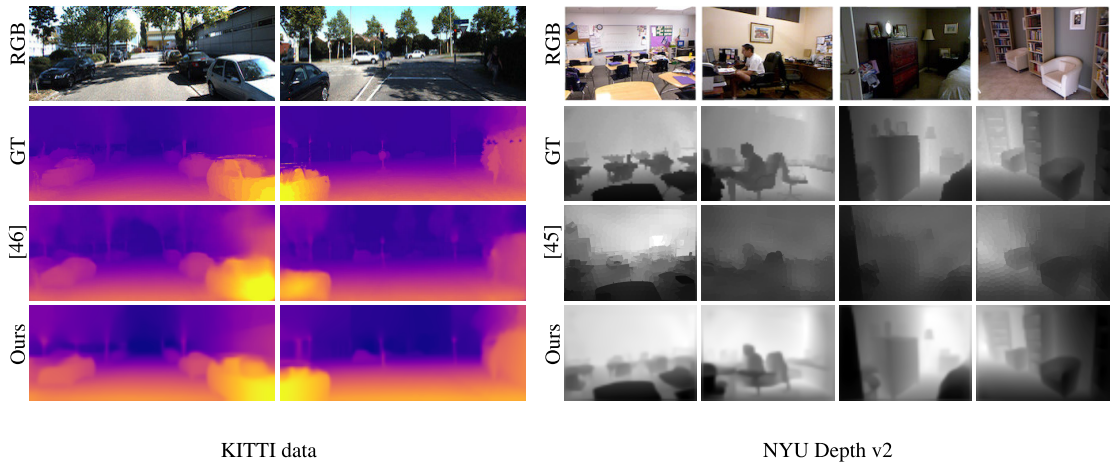


FIGURE 5. Additional qualitative comparison result with Kutznizov et al. [46] on KITTI data (left) and with Liu et al. [45] on NYU Depth v2 (right).

competitive method of [10]. While our quantitative results are not as good as to those of [51] on the KITTI data, ours outperforms a vast majority of previous methods [7], [11], [12], [16], [45], [46], [47], [48], [49], [50]. Among the approaches, our TP-GAN achieves the best on SQREL and RMSE LOG scores. In fact, the transformer-based model of [51] employs a far more complex and much deeper network with two distinct encoders and a single decoder. Consequently, the model will have significantly larger parameters and demand much more GPU RAM to train. It can be stated that our method performs comparably along with the previous non-adversarial works in the metrics of interest. The global performance of our proposed method revealed adequate depth prediction.

B. ADVERSARIAL MODELS

On the NYU Depth v2 dataset, we present a comparison to similar previous works. However, there are only a few GAN implementations in this dataset. Even though our depth estimation appears less accurate depth than that of [28] in the $\delta < 1.25$, we reported significantly higher performance in the

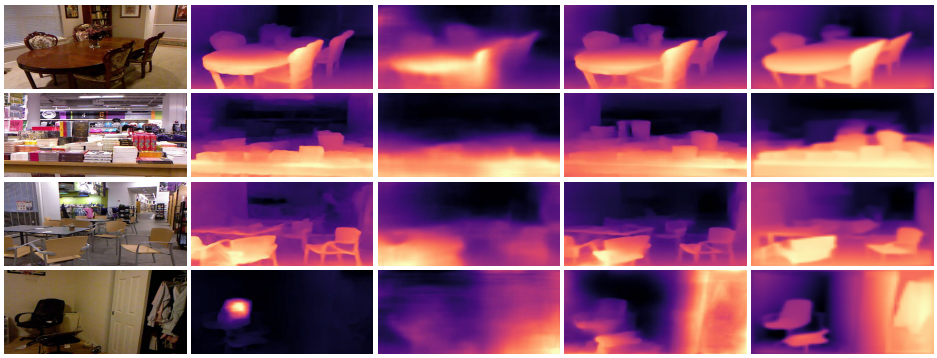


FIGURE 6. Qualitative comparison result on NYU Depth v2 from left to right: RGB images, Zhao et al. [50], Godard et al. [12], Bian et al. [15], and our TP-GAN.

TABLE 4. Accuracy comparison on KITTI data. The best results are in bold and second best are underlined.

		Accuracy*		
	range [m]	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Adversarial Methods				
Kumar *et al.* 2018 [26]	—	0.732	0.897	0.959
Pilzer *et al.* 2018 [25]	0–80	0.789	0.918	0.965
Aleotti *et al.* 2018 [24]	0–80	0.808	0.939	0.975
Zheng *et al.* 2018 [27]	1–50	0.867	0.960	0.986
Almalioglu *et al.* 2019 [23]	0–50	0.867	0.970	0.983
Li *et al.* 2019 [22]	0–80	0.823	0.936	0.974
Puscas *et al.* 2019 [21]	0–80	0.828	0.933	0.967
Groenendijk *et al.* 2020 [20]	—	0.847	0.945	0.975
Zhao *et al.* 2021 [19]	0–80	0.821	0.942	0.978
Non-adversarial Methods				
Eigen *et al.* 2014 [7]	0–80	0.692	0.899	0.967
Liu *et al.* 2015 [45]	—	0.647	0.882	0.961
Godard *et al.* 2017 [11]	0–50	0.861	0.949	0.976
Kutznietsov *et al.* 2017 [46]	0–80	0.862	0.960	0.986
Zhan *et al.* 2018 [47]	0–80	0.820	0.933	0.971
Zou *et al.* 2018 [48]	—	0.806	0.933	0.973
Godard *et al.* 2019 [12]	0–80	0.876	0.958	0.980
Ranjan *et al.* 2019 [49]	0–80	0.826	0.941	0.975
Zhao *et al.* 2020 [50]	—	0.871	0.961	0.984
Bian *et al.* 2021 [16]	0–80	0.873	0.960	0.982
Manimaran *et al.* 2022 [51]	0–80	**0.926**	**0.986**	**0.997**
Ours	0–80	0.884	0.973	0.992

*the higher the better.

TABLE 5. Error rate comparison on KITTI data. The best results are in bold and second best are underlined.

		Error Rate*		
	range [m]	ABS REL	SQ REL	RMSE LOG
Adversarial Methods				
Kumar *et al.* 2018 [26]	—	0.211	1.979	0.264
Pilzer *et al.* 2018 [25]	0–80	0.152	1.388	0.247
Aleotti *et al.* 2018 [24]	0–80	0.150	1.414	0.216
Zheng *et al.* 2018 [27]	1–50	0.114	0.627	0.178
Almalioglu *et al.* 2019 [23]	0–50	0.137	0.892	0.201
Li *et al.* 2019 [22]	0–80	0.150	1.127	0.229
Puscas *et al.* 2019 [21]	0–80	0.135	1.1815	0.235
Groenendijk *et al.* 2020 [20]	—	0.122	0.928	0.215
Zhao *et al.* 2021 [19]	—	0.139	1.034	0.214
Non-adversarial Methods				
Eigen *et al.* 2014 [7]	0–80	0.190	1.515	0.270
Liu *et al.* 2015 [45]	—	0.217	1.841	0.289
Godard *et al.* 2017 [11]	0–50	0.114	0.898	0.206
Kutznietsov *et al.* 2017 [46]	0–80	0.113	0.741	0.189
Zhan *et al.* 2018 [47]	0–80	0.135	1.132	0.229
Zou *et al.* 2018 [48]	—	0.150	1.124	0.223
Godard *et al.* 2019 [12]	0–80	0.106	0.806	0.193
Ranjan *et al.* 2019 [49]	0–80	0.140	1.070	0.217
Zhao *et al.* 2020 [50]	—	0.113	0.704	0.184
Bian *et al.* 2021 [16]	0–80	0.114	0.813	0.191
Manimaran *et al.* 2022 [51]	0–80	**0.082**	—	—
Ours	0–80	0.103	**0.624**	**0.156**

*the lower the better.

$\delta < 1.25^2$ and $\delta < 1.25^3$ by 0.019 and 0.013, respectively. Notably, [28] does not publish the depth range information of their evaluation data, which significantly impacts the quantitative result. Nonetheless, we achieve the best error rate performance compared with all the related methods [27], [28]. Next, we provide a comparison to nine previous related adversarial works on the KITTI data. Despite the fact that some of them use a smaller depth range [23] and [27], and some do not show their depth range [20] and [26], our approach outperforms a series of the previous methods by a big margin in all metrics of interest. The results show that our proposed TP-GAN outperforms the previous monocular depth estimation on adversarial model architectures.

C. QUALITATIVE RESULTS

We provide qualitative visualization results for more analysis of our proposed method. We compare our predicted depth with the work of [7] in Fig. 3, [45] in Fig. 5, and the works of [12], [15], and [50] in Fig. 6 on the NYU Depth v2 dataset. To ensure a reasonable visualization comparison, we use similar sample images adopted from their paperworks. It is clear in Fig. 3, Fig. 5, and Fig. 6 that our proposed approach is sufficient to generate more reproducible image depth estimation performance, in which some results are close to their ground truths.

Meanwhile, the performance of our depth estimation on the KITTI data along with the works [7], [11], [45], and [46] are shown in Fig. 4 and in Fig. 5 to that of [45]. Compared to their output depth, it can be seen that our method yields more visually satisfying predictions with more visible transitions that correlate with local depth information. We show that our strategy is more proficient at detecting the major depth structure of the image for both datasets.

D. ADDITIONAL QUALITATIVE RESULTS

Supplementary, we provide Fig. 7, Fig. 8, Fig. 9, and Fig. 10 to demonstrate additional qualitative results of our proposed approach on 15 unseen random NYU and KITTI data. Fig. 7

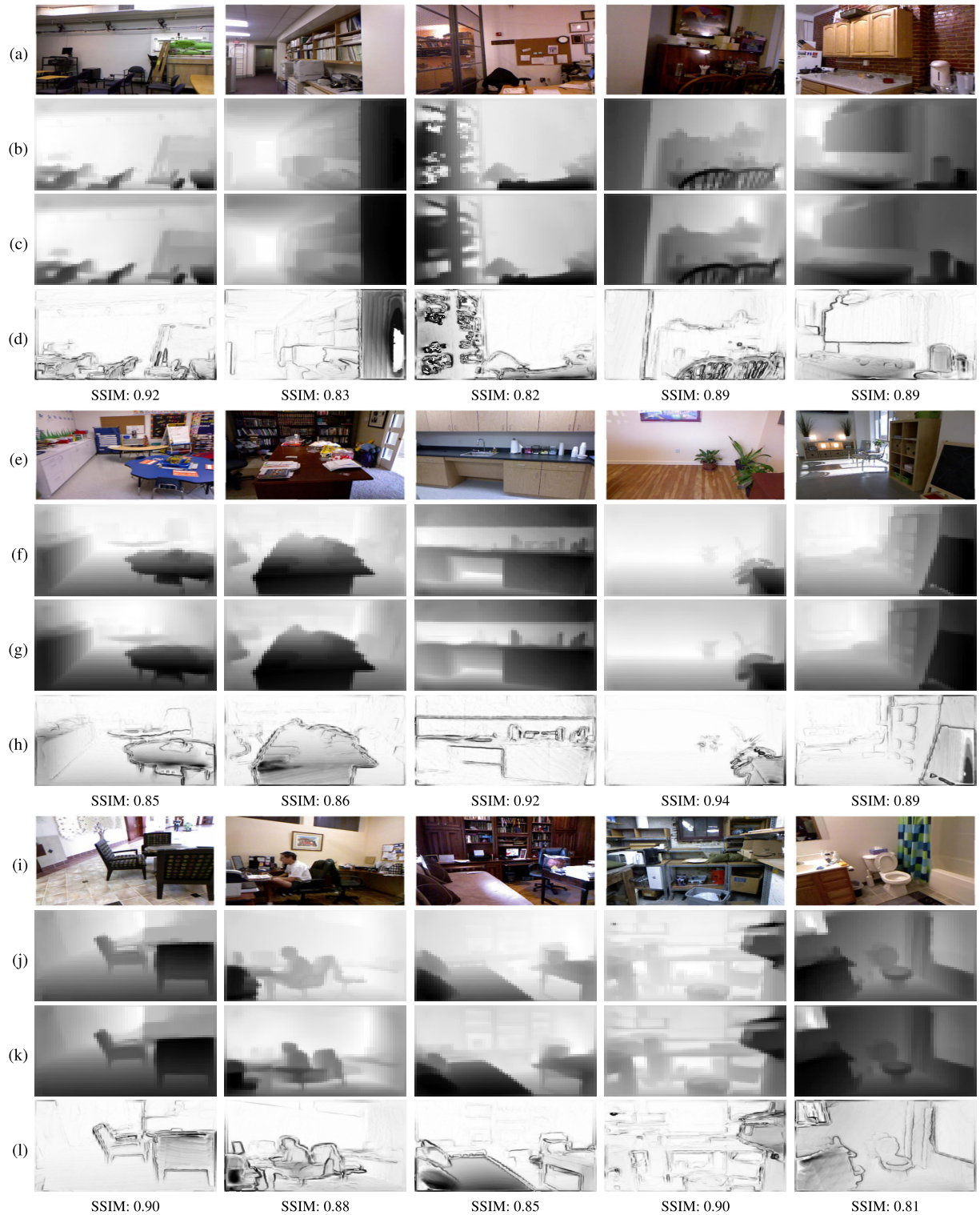


FIGURE 7. Image depth prediction from random images on NYU Depth v2: (a), (e), (i) RGB image, (b), (f), (j) ground truth, (c), (g), (k) our TP-GAN method. (d), (h), (l) SSIM reconstruction error.

and Fig. 9 show further qualitative results of our predicted depths, visualizing their SSIM error reconstruction images, and calculating the SSIM scores. We demonstrate the effectiveness of our proposed method in generating consistent better depth visualization. The dark portion of the SSIM reconstruction images represents the depth dissimilarity between the predicted and the ground truth depth. Our method prediction achieves a good performance in which some

FIGURE 8. Depth value histogram from random images on NYU Depth v2: (a), (e), (i) RGB image, (b), (f), (j) GT histogram, (c), (g), (k) Predicted histogram. (d), (h), (l) Combination histogram.

results are relative to the ground truth, as represented by their SSIM error scores being close to 1.

As illustrated in Fig. 8 and Fig. 10, we study the depth value distribution by analyzing the histogram from the ground truths and its predicted images. The histograms of the ground truth data show that the depth distribution corresponding to different RGB images can vary to a large extent. Despite some obscured deviations of the predicted histograms from the ground truth, the overall depth value data distributions of our predicted images are more visually appealing for the random unseen data. In addition, we also demonstrate 3-D point cloud visualizations of the NYU sample data In Fig. 11. Additional qualitative results, including random images from the internet, can be found in the supplementary materials.

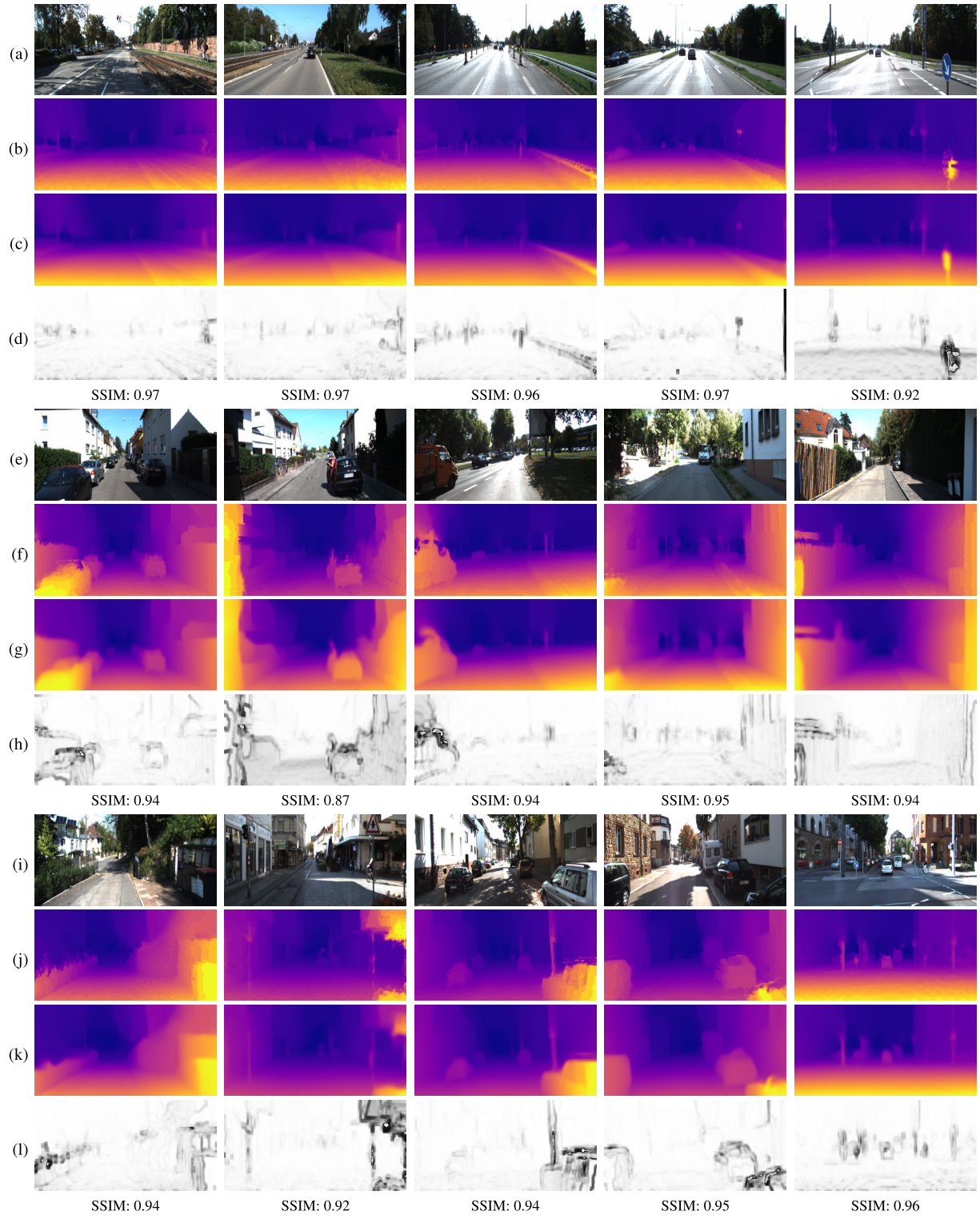


FIGURE 9. Image depth prediction from random images on KITTI dataset: (a), (e), (i) RGB image, (b), (f), (j) ground truth, (c), (g), (k) our TP-GAN method. (d), (h), (l) SSIM reconstruction error.

E. CROSS-DATASET ADAPTATION

To demonstrate how effective our model performance generalizes to other datasets, we examine the cross-dataset adaptation capabilities by training on one dataset and testing on another, and vice versa. As shown in the supplementary materials, the accuracy of depth estimation decreases when training and testing across two datasets. This is likely due to different data collection environments (e.g., the maximum

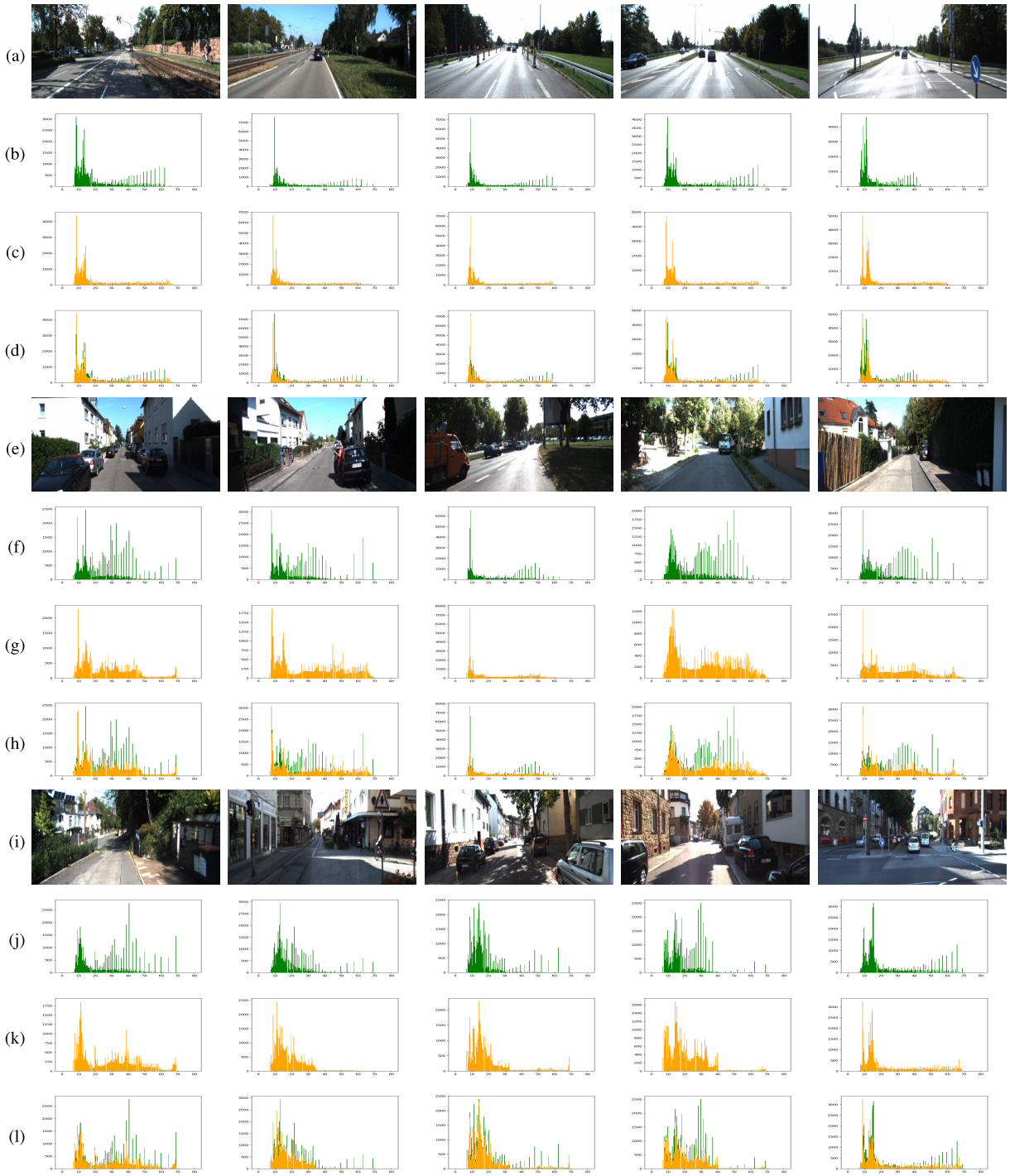


FIGURE 10. Depth value histogram from random images on KITTI data: (a), (e), (i) RGB image, (b), (f), (j) GT histogram, (c), (g), (k) Predicted histogram. (d), (h), (l) Combination histogram.

range in NYU is 10 meters and 80 meters in KITTI dataset). After training on the outdoor NYU dataset, the model has an issue estimating the distant objects on the indoor KITTI dataset. Likewise, the trained KITTI model has difficulty generating depth for some particular objects on the NYU indoor dataset. Nevertheless, despite the different data settings between the two datasets, we show that our model was able to generalize in learning scene variations across both datasets and confirms reliable results, especially when trained indoors and tested that use the outdoor dataset.

F. MODEL PERFORMANCE

We demonstrated that the model we proposed in this research is rather concise, yet its performance is reliable. In fact, our

FIGURE 11. 3-D point cloud from sample indoor NYU Depth v2.

TP-GAN is comprised of only three simple sub-models, the first of which is a ResNet50V2-based generator sub-model. The second sub-model, a discriminator, consists of a stack of six convolution layers constructed as a patch GAN model. The final sub-model consists of a series of six convolution blocks which the first five blocks comprising convolution, batch normalization, Relu activation, and dropout. The last block contains a convolution layer following a linear activation.

In particular, our TP-GAN model has around 59.2M training parameters, which 51.7M for the generator, 7M for the discriminator, and 520K for the refiner sub-model. During testing, only the refiner parameter is taken into account. It takes about 39.5 minutes and 50 minutes to finish one epoch for training KITTI and NYU data, respectively, measured in a single 8GB NVIDIA GeForce GTX 1080. All the results presented in this paper, the training process typically takes around 36 epochs for the NYU and 28 epochs for the KITTI dataset to converge with a batch size of 16.

Another benefit of using an adversarial network is the ability to train the network using a single batch. This strategy enables our model to train with fewer input data while still

maintaining to show reliable performance. In comparison to Eigen et al. work [7], we trained our model utilizing $50K$ vs. $120K$ for the NYU v2 depth data and around $25K$ vs. $40K$ for the KITTI data. Whereas 25K vs. 39K training data on KITTI compared with the works [11], [46].

VII. CONCLUSION AND FUTURE WORKS

The use of an additional sub-model to integrate global scene structure and local scene information in a generative adversarial network (GAN) has been successfully demonstrated for single image depth estimation. We confirmed that regardless of its simple structure, the presence of the third player (TP) in adversarial learning effectively improves the overall depth prediction performance of the model. Extensive experimental results demonstrate that employing a third player along with the SSIM loss is beneficial in a single image depth estimation. Our proposed TP-GAN-SSIM improves the standard GAN-MSE accuracy by 3%, 1%, and 0.5% for the threshold $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$ respectively, and shows competitive performance compared with state-of-the-art on the outdoor dataset KITTI and indoor dataset NYU depth v2. Furthermore, we demonstrated that our proposed model required less training time to converge compared with the aforementioned related methods regardless of the GPU device.

Our future work is encouraging to develop a robust single image depth estimation to be applied not only for indoor or outdoor data, but also will be applicable for such a complex environment e.g. underwater. We will also consider to add some scene data generated from Carla simulator for greater generalization capability across different datasets.

REFERENCES

[1] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Depth from familiar objects: A hierarchical model for 3D scenes," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2410–2417.

[2] E. Coupeté, F. Moutarde, and S. Manitsaris, "Gesture recognition using a depth camera for human robot collaboration on assembly line," *Proc. Manuf.*, vol. 3, pp. 518–525, Dec. 2015.

[3] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for obstacle detection with fully convolutional networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4296–4303.

[4] T. Kim, M. Motro, P. Lavieri, S. S. Oza, J. Ghosh, and C. Bhat, "Pedestrian detection with simplified depth prediction," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2712–2717.

[5] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2021, pp. 1709–1717.

[6] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of pedestrian movements near an amenity in walkways of public buildings," in *Proc. 8th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2022, pp. 394–400.

[7] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, 2014, pp. 2366–2374.

[8] X. Chen, X. Chen, and Z.-J. Zha, "Structure-aware residual pyramid network for monocular depth estimation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 694–700.

[9] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen, "Learning to recover 3D scene shape from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 204–213.

[10] H. Ye and D. Xu, "Inverted pyramid multi-task transformer for dense scene understanding," in *Proc. Eur. Conf. Comput. Vision.*, 2022, pp. 1–10.

[11] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[12] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 567–675.

[13] A. J. Amiri, S. Yan Loo, and H. Zhang, "Semi-supervised monocular depth estimation with left-right consistency using deep neural network," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 602–607.

[14] S. Gur and L. Wolf, "Single image depth estimation trained via depth from defocus cues," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[15] J.-W. Bian, H. Zhan, N. Wang, T.-J. Chin, C. Shen, and I. Reid, "Auto-rectify network for unsupervised indoor depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9802–9813, Dec. 2022.

[16] J.-W. Bian, H. Zhan, N. Wang, Z. Li, L. Zhang, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth learning from video," *Int. J. Comput. Vis.*, vol. 129, no. 9, pp. 2548–2564, Sep. 2021.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–5.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with condiitonal adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nov. 2017, pp. 5967–5976.

[19] C. Zhao, G. G. Yen, Q. Sun, C. Zhang, and Y. Tang, "Masked GAN for unsupervised depth and pose prediction with scale consistency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5392–5403, Dec. 2021.

[20] R. Groenendijk, S. Karaoglu, T. Gevers, and T. Mensink, "On the benefit of adversarial training for monocular depth estimation," *Comput. Vis. Image Understand.*, vol. 190, Jan. 2020, Art. no. 102848.

[21] M. M. Puscas, D. Xu, A. Pilzer, and N. Sebe, "Structured coupled generative adversarial networks for unsupervised monocular depth estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 134–145. [Online]. Available: https://arxiv.org/pdf/1908.05794.pdf

[22] S. Li, F. Xue, X. Wang, Z. Yan, and H. Zha, "Sequential adversarial learning for self-supervised deep visual odometry," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2851–2860.

[23] Y. Almalioglu, M. R. U. Saputra, P. P. B. D. Gusmao, A. Markham, and N. Trigoni, "GANVO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5474–5480.

[24] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018. [Online]. Available: https://openaccess.thecvf.com/content_ECCVW_2018/papers/11129/Aleotti_Generative_Adversarial_Networks_for_unsupervised_monocular_depth_prediction_ECCVW_2018_paper.pdf

[25] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 587–595. [Online]. Available: https://arxiv.org/pdf/1807.10915.pdf

[26] A. C. Kumar, S. M. Bhandarkar, and M. Prasad, "Monocular depth prediction using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 413–421.

[27] C. Zheng, T.-J. Cham, and J. Cai, "T2Net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 767–783.

[28] D.-H. Kwak and S.-H. Lee, "A novel method for estimating monocular depth using cycle GAN and segmentation," *Sensors*, vol. 20, no. 9, p. 2567, Apr. 2020.

[29] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5162–5170.

[30] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[33] A. Hendra and Y. Kanazawa, "Smaller residual network for single image depth estimation," *IEICE Trans. Inf. Syst.*, vols. E104–D, no. 11, pp. 1991–2001, Nov. 2021.
[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 7546, Oct. 2012, pp. 746–760.
[36] 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.
[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, and D. Mané, "TensorFlow: Large-scale machine learning on heterogeneous systems," Tech. Rep., 2015. [Online]. Available: https://www.tensorflow.org/about/bib
[38] F. Chollet. *Keras*. Accessed: Aug. 28, 2022. [Online]. Available: https://keras.io
[39] A. C. Wilson, R. Roelofs, S. Mitchell, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4151–4161.
[40] 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.
[41] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2800–2809.
[42] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proc. CVPR*, Jun. 2016, pp. 5506–5514.
[43] A. Chakrabarti, J. Shao, and G. Shakhnarovich, "Depth from a single image by harmonizing overcomplete local network predictions," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 2566–2674.
[44] J. Li, C. Yuce, R. Klein, and A. Yao, "A two-streamed network for estimating fine-scaled depth maps from single RGB images," *Comput. Vis. Image Understand.*, vol. 186, pp. 25–36, Sep. 2019.
[45] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2015.
[46] Y. Kutznietzov, J. Stuchler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. CVPR*, 2017, pp. 190–201.
[47] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1238–1245.
[48] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 456–564.
[49] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3452–3460.
[50] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without PoseNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9151–9161.
[51] G. Manimaran and J. Swaminathan, "Focal-WNet: An architecture unifying convolution and attention for depth estimation," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2022, pp. 786–790.

ANDI HENDRA received the M.S. degree in information engineering from the Sepuluh Nopember Institute of Technology, Surabaya, Indonesia, in 2009. He is currently pursuing the Ph.D. degree in computer science and engineering with the Toyohashi University of Technology, Aichi, Japan. His research interests include image processing and computer vision.

YASUSHI KANAZAWA (Member, IEEE) received the B.E. and M.S. degrees in information engineering from the Toyohashi University of Technology, Aichi, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in information and computer science from Osaka University, in 1997. After engaging in research and development of image processing systems with Fuji Electric Company, Tokyo, Japan, and as a Lecturer in information and computer engineering with the Gunma College of Technology, Gunma, Japan, he is currently an Associate Professor in computer science and engineering with the Toyohashi University of Technology. His research interests include image processing and computer vision.