electronics

MDPI

Article

Conditional Generative Adversarial Network for Monocular Image Depth Map Prediction

Shengang Hao [1], Li Zhang [2,*], Kefan Qiu [1] and Zheng Zhang [1]

[1] School of Computer Science, Beijing Institute of Technology, Beijing 100081, China
[2] School of Media Engineering, Communication University of Zhejiang, Hangzhou 310018, China
* Correspondence: nythhsg@163.com; Tel.: +86-178-1661-5955

Abstract: Deep map prediction plays a crucial role in comprehending the three-dimensional structure of a scene, which is essential for enabling mobile robots to navigate autonomously and avoid obstacles in complex environments. However, most existing depth estimation algorithms based on deep neural networks rely heavily on specific datasets, resulting in poor resistance to model interference. To address this issue, this paper proposes and implements an optimized monocular image depth estimation algorithm based on conditional generative adversarial networks. The goal is to overcome the limitations of insufficient training data diversity and overly blurred depth estimation contours in current monocular image depth estimation algorithms based on generative adversarial networks. The proposed algorithm employs an enhanced conditional generative adversarial network model with a generator that adopts a network structure similar to UNet and a novel feature upsampling module. The discriminator uses a multi-layer patchGAN conditional discriminator and incorporates the original depth map as input to effectively utilize prior knowledge. The loss function combines the least squares loss function and the L1 loss function. Compared to traditional depth estimation algorithms, the proposed optimization algorithm can effectively restore image contour information and enhance the visualization capability of depth prediction maps. Experimental results demonstrate that our method can expedite the convergence of the model on NYU-V2 and Make3D datasets, and generate predicted depth maps that contain more details and clearer object contours.

Keywords: autonomous mobile robot; conditional generative adversarial network; depth map prediction; intelligent manufacturing

Citation: Hao, S.; Zhang, L.; Qiu, K.; Zhang, Z. Conditional Generative Adversarial Network for Monocular Image Depth Map Prediction. *Electronics* **2023**, *12*, 1189. https://doi.org/10.3390/electronics12051189

Academic Editor: Mehdi Sookhak

Received: 19 January 2023
Revised: 6 February 2023
Accepted: 25 February 2023
Published: 1 March 2023

1. Introduction

Today, intelligent logistics has become an essential component of the promotion of "intelligent manufacturing". It is extensively used in production line assembly for discrete manufacturing industries and material access in enterprise warehouse rooms. Intelligent warehousing, a result of warehouse automation, can be achieved through various automation and interconnection technologies that work together to enhance the production efficiency of the production line and the distribution efficiency of the warehouse, minimize labor, and reduce errors.

In intelligent logistics and warehousing, Automated Guided Vehicles (AGVs) [1] and Autonomous Mobile Robots (AMRs) [2] play vital roles in handling materials such as raw materials, tools, products, and accessories. Unlike AGVs, which require preset guidance devices and simple programming instructions, AMRs can carry out more complex operations and processing and provide greater flexibility. They can realize more intelligent navigation functions such as map construction and autonomous obstacle avoidance, making them the best choice for realizing intelligent logistics and warehousing.

As the environmental complexity increases in intelligent manufacturing enterprises, two-dimensional maps are no longer sufficient for mobile robots' environmental perception.

Three-dimensional maps can provide more comprehensive environmental information and are a current research hotspot in the field of mobile robot map construction [3].

The depth map is a common way of expressing 3D scene information, where the value of each pixel in the map represents the distance between the corresponding point of the object and the collector in the scene. Image depth prediction, widely used in autonomous driving, robot obstacle avoidance, 3D map reconstruction, and object detection [4], is a classic problem in computer vision research. When using two or more cameras to predict the depth of the same object., the method is called binocular or binocular image depth prediction. The method of monocular image depth prediction only requires obtaining a large quantity of depth information from a single camera, which is low-cost and more widespread, and increasingly a current research focus in the field of computer vision.

With the rapid development of deep learning, much progress has been made in solving classical problems in computer vision. Deep learning has played an essential role in addressing computer vision tasks such as object recognition, object tracking, and image segmentation, resulting in significant improvements in efficiency and accuracy. The first monocular image depth prediction method that utilized a convolutional neural network was proposed by Eigen et al. [5] in 2014. Their approach, which employed an AlexNet-based network structure, consisted of two scales: one to capture global information and the other to capture local information. The global information capture was partly based on AlexNet. This method achieved promising results on both the NYU Depth and KITTI datasets. Since then, several monocular image deep models based on convolutional neural networks have been proposed, resulting in a range of outcomes [6–8]. Although existing models have shown effectiveness on standard public datasets, they rely too heavily on specific datasets and are vulnerable to security attacks.

In practical applications, recognized objects can vary greatly in shape, and the lighting of the environment can change. Intelligent warehousing scenarios pose particular challenges due to highly stacked objects, and the diverse shapes and sizes of goods, which can make it difficult for intelligent vehicles to accurately estimate depth information. Moreover, deep neural networks themselves are prone to security issues and can be vulnerable to security attacks.

To improve the robustness and generalization ability of the deep estimation algorithm, and to mitigate potential security threats, this article proposes an optimization algorithm for monocular image depth estimation, based on conditional generative adversarial networks.

The contributions to this article are as follows:

First, we present the conditional generative adversarial network (cGAN) structure as the fundamental framework for the monocular depth estimation algorithm. The cGAN can generate more realistic synthetic data, which increases the amount of available training data. The model uses conditional variables, such as the depth map and original image, as prior knowledge to enhance the accuracy of generated depth maps and the discrimination ability of the discriminator. Moreover, the cGAN training improves the learning of the mapping between input images and depth images, thereby enhancing the robustness of the system.

Second, we introduce a novel feature upsampling module in the generator that improves the resolution of the feature map. This is achieved by incorporating new deconvolution layers into the existing upsampling module, thereby improving the accuracy of the generated depth maps. We also use an improved loss function that combines the L1 norm loss with the least squares loss function. This resolves the issues of difficult convergence and mode collapse commonly encountered in generative adversarial networks. The improved loss function guides the model to generate more accurate depth maps.

The rest of this article is arranged as follows: Section 2 provides a brief overview of the current state-of-the-art in monocular image deep estimation methods, as well as adversarial generative networks based on convolutional neural networks. Section 3 delves into the key techniques and algorithmic framework design. Section 4 presents the implementation

results and their analysis. Finally, in Section 5, we draw conclusions from the results and discuss the next work.

2. Related Work

2.1. Deep Estimation Methods for Monocular Images Based on Deep Learning

The monocular image deep prediction methods based on deep learning can be broadly classified into three categories: supervised learning, unsupervised learning, and semi-supervised learning. Supervised learning, which involves training a model on labeled data, was pioneered by Eigen et al. in 2014 [5] and 2015 [6]. They used convolutional neural networks, including AlexNet and VGGNet-16, to estimate monocular image depth. In 2014, the author used AlexNet [5] as the fundamental model to produce an initial global depth map. To refine the depth map, a local fine network structure was used in conjunction with the original image information, which yielded favorable results at that time. However, due to the limited expressiveness of the AlexNet network, the depth prediction results were not satisfactory. Shortly after, in 2015, Eigen et al. [6] improved this work by incorporating deeper and more multi-scale convolutional neural networks. The authors employed VGGNet-16 for feature extraction and depth prediction, leading to better performance on standard datasets. Laina et al. [7] proposed a fully convolutional neural network structure based on deep residual networks to address the issue of excessive network parameters in monocular depth prediction. To enhance the depth prediction results, they introduced an up-projection module and utilized back-pooling to increase the depth map resolution.

Monocular image depth prediction is a complex task that involves calculating the depth value for each pixel in an image. Typically, this is treated as a high-dimensional regression problem where the model estimates the difference between the predicted depth value and the actual depth value, which is then used as the basis for the loss function. However, a more efficient approach is to transform the problem into a classification problem by dividing depth values into intervals and grouping pixels into corresponding bins, similar to a histogram. Cao et al. [8] applied this technique to extract features using deep residual networks, which were then fused using fully connected conditional random fields. The resulting model was trained using cross-entropy loss in the classification model. Liu et al. [9] used isolated conditional random fields for monocular image depth prediction. SENet-154 [10] introduced a new Squeeze-and-Excitation (SE) network module, which can adaptively learn the correlations between feature channels, thereby enhancing the network's representation and generalization capabilities. Meanwhile, the DenseDepth algorithm [11] proposed a transfer learning-based method that fine-tunes pre-trained models from large datasets like ImageNet for depth estimation. To further enhance the robustness and precision of depth estimation, the AdaBins algorithm [12] presents an adaptive depth estimation technique that adjusts the depth range in different scenarios and employs a novel loss function. Finally, the GLPDepth [13] algorithm proposes a novel Vertical CutDepth depth estimation method that leverages vertical information in-depth images to improve accuracy and efficiency. The authors also suggest a global-local path network architecture that captures both global and local information in scenes, leading to more accurate depth estimation.

In the field of unsupervised and semi-supervised learning, several researchers have proposed innovative methods to improve the accuracy and robustness of depth prediction and camera motion estimation. Godard et al. [14] utilized left-right view consistency for unsupervised depth prediction, which improved robustness by leveraging parallax and optimizing performance. Kuznietsov et al. [15] proposed a semi-supervised approach that utilizes sparse deep images as labels to achieve better performance. Mahjourian et al. [3] proposed an end-to-end learning approach that uses view synthesis as a supervised signal, resulting in a video sequence-based unsupervised learning framework for monocular image depth and camera motion estimation. Bian et al. [16] leveraged geometric consistency constraints to achieve scale consistency between adjacent frames and used this to detect

and remove dynamic objects and masked regions. This approach outperforms previous algorithms trained on binocular video. Casser et al. [17] proposed a model that takes RGB image sequences as input and is supplemented by a pre-computed instance segmentation mask. Bhutani et al. [18] proposed a Bayesian inference-based method for monocular image depth estimation and confidence prediction. This method estimates the posterior distribution of each pixel's depth and confidence through a combination of a neural network that estimates the prior distribution of pixel depth and a noise model, and the pixel values of the input image. Almalioglu et al. [19] proposed a monocular visual odometry (VO) and depth estimation algorithm based on depth learning. This method trains an unsupervised monocular VO and depth estimation model using geometric constraints from binocular vision, allowing for motion estimation and scene depth estimation even in extreme environments. The method introduces a new "Persistent" loss function which enables the network to learn persistent estimation of optical flow and scene depth, while reverse depth estimation and optical flow prediction increase the loss function's robustness. The method also employs a pyramid depth network, designed to extract depth information from various scale feature maps, resulting in more accurate and robust depth estimation.

Although the results may not always be outstanding, these methods and their practical applications are still worth exploring.

2.2. Current Status of Generative Adversarial Networks

In 2014, Ian J. Goodfellow [20] introduced Generative Adversarial Networks (GANs), which consist of a generator and a discriminator. The generator takes a high-dimensional noise vector as input and generates data that is fed into the discriminator. The discriminator then determines whether the input is a real sample or a fake sample generated by the generator. GANs use an unsupervised learning approach and reach an equilibrium point through a two-player game. at which point the generator can produce data that the discriminator cannot effectively distinguish as fake.

However, early GANs faced issues with training stability and the lack of control over the output. To address these problems, researchers introduced conditional GANs (cGANs) [21] in 2014, which incorporate additional conditional information during training to ensure the generator produces specific content. Despite these efforts, GANs are still challenging to train. The literature proposes various modifications to improve training stability, including Deep Convolutional Generative Adversarial Networks (DCGANs [22]), least squares loss functions (LSGANs [23]), Wasserstein loss functions (WGANs [24]), gradient normalization (WGAN-up [25]), and proportional control (BEGAN [26]).

In the context of monocular image depth prediction, GANs can partially solve the problem of over-smooth or under-detailed depth prediction. By training a GAN to measure the similarity between the predicted depth graph and the original depth label, the visualization of the depth estimate can be improved. Lsola et al. [27] propose a general model based on conditional GANs to solve image-to-image translation problems, including the monocular image depth prediction problem.

3. Methods

Similar to the conditional generative adversarial network structure proposed in the literature [27], we utilize an enhanced conditional generator and conditional discriminator for our GAN model. Specifically, we incorporate a generator structure based on deep residual networks, which includes a new up-sampling module (labeled as Up-Decon). Our discriminator classifier structure is based on the conditional patchGAN classifier introduced in literature [27], but with modifications to the loss function to enhance the performance of the generative adversarial networks.

3.1. Network Structure

The original GAN structure generates images by processing random noise through a neural network, which can lead to uncontrolled output content. To overcome this limitation,

additional constraints must be imposed on the original GAN. This paper proposes an optimized conditional GAN (op-cGAN) for monocular image depth estimation, which is a visual task performed at the image-to-image level. The specific model structure is depicted in Figure 1. The generator (G) takes the original image (x), the depth map (y), and random noise (z) as inputs, and outputs the predicted depth map (y′). The discriminator (D) takes the original image (x) and depth map as inputs and determines whether the depth map is from the training dataset (y) or generated by the generator (y′). A "fake" output from the discriminator indicates that the depth map is generated; while a "real" output indicates that the depth map is from the training dataset. By including the original image (x) as a constraint, the discriminator has access to additional priori knowledge, resulting in a more accurate and detailed depth map generation.

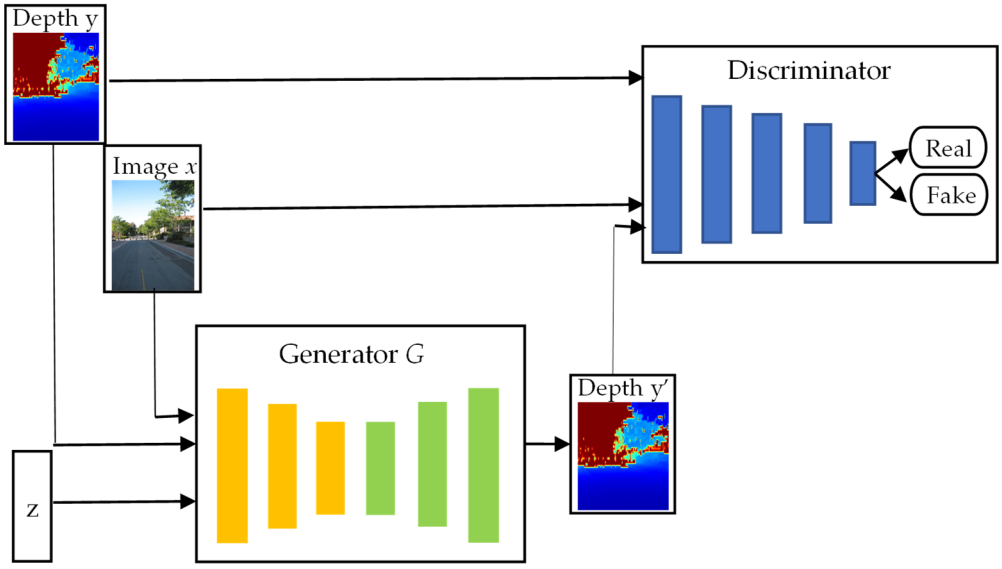

Figure 1. Network Structure of op-cGAN.

3.2. Generator

The generator section of the model follows an encoder-decoder structure, where the encoder extracts features and the decoder transforms these features into the final output. As shown in Figure 2, the generator employs a network structure based on deep residuals. The feature extraction process is aligned with the ResNet And begins with a stride-convolution and max-pooling to reduce the input image's resolution and minimize the number of parameters. The feature map then passes through four ResBlock, which reduces the feature map's resolution by half after each ResBlock while doubling the number of feature map layers. This process results in a feature map resolution that is 1/32 of the input resolution.

After extracting the upper feature map, a 1×1 convolutional kernel integrates the features. Next, the feature map passes through four Up-Decon modules, each consisting of three parts. The first part is a convolutional layer, followed by a concatenation layer that directly concatenates features of the same resolution extracted from the previous feature extraction stage. Lastly, a deconvolution layer is used to increase the feature map's resolution. The convolutional layer uses a 1×1 kernel to integrate cross-channel information and adjust the number of feature map channels. The concatenation layer uses concatenation or bitwise addition to combine the features and the deconvolution layer uses a 4×4 kernel with a stride of 2 and padding of 1 to double the feature map's resolution. After the Up-Decon modules, the feature map is processed by two convolutional modules to generate a depth prediction map with half the input resolution. Each pixel in the depth prediction map represents a predicted depth value in meters and is stored as a 32-bit floating-point number.

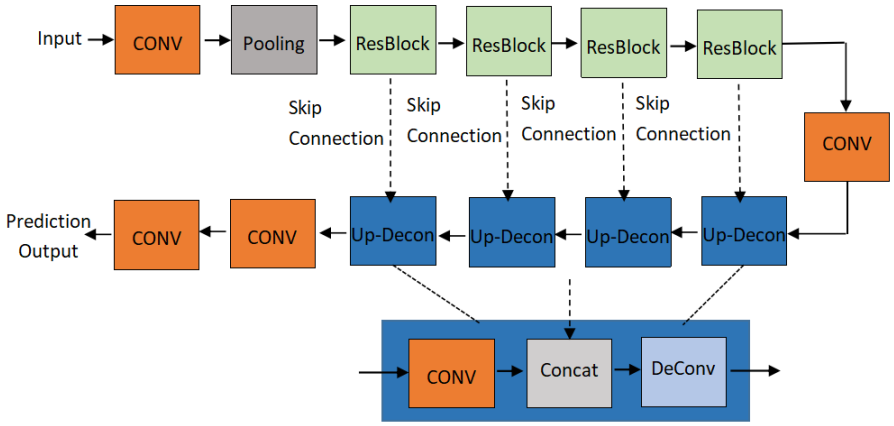

Figure 2. Network Structure of Generator.

3.3. Discriminator

To extract local image information, we adopt a conditional patchGAN discriminator, similar to the one proposed in reference [27]. As shown in Figure 3, the discriminator comprises a 5-layer full convolution network, which takes a concatenation of the depth map and the original image as input, without the sigmoid function on the last layer, since we use the least square function for loss calculation in this work. The original image serves as a conditional vector to guide the discriminator's classification. During training, the predicted depth map and the original image are concatenated as the negative samples, while the depth map from the training dataset and the original image are the positive samples. PatchGAN partitions the image into multiple patches and computes the classification results for each patch, thus treating the image as a Markov random field and assuming the independence of pixels across different patches. The final output of the discriminator is obtained by averaging the output of each patch. The loss function is calculated using convolution, enabling the use of smaller block sizes.

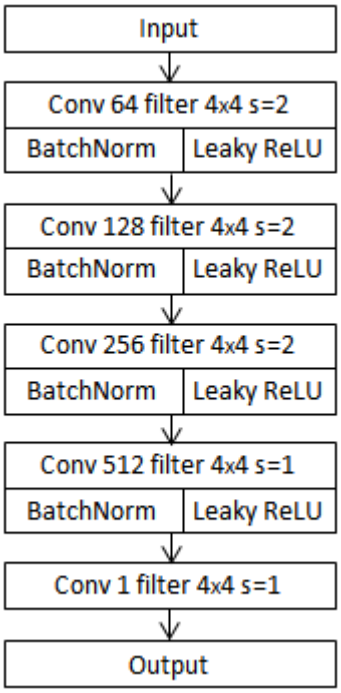

Figure 3. The structure of discriminator.

3.4. Loss Function

The loss function of traditional GAN is:

$$\min_G \max_D V_{GAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (1)$$

in which G is a generator, D is a discriminator, z is a noise variable sampled from a normalized or Gaussian distribution, $p_{data}(x)$ represents the probability distribution of real data x, and $p_z(z)$ represents the probability distribution of z, $\mathbb{E}_{x\sim p_{data}(x)}$ is the expectation value of x and $\mathbb{E}_{z\sim p_{data}(z)}$ is the expectation value of z. The goal is to train G to generate samples that are indistinguishable from real data, while D tries to correctly distinguish between real and fake samples. However, a major issue with traditional GAN training is that as D gets better, the gradient signal that G receives becomes weaker, which leads to poor sample quality. To address this problem, LSGAN (Least Squares GAN) [23] replaces the binary classification objective of D with a least squares regression objective, which removes the sigmoid activation function from its final layer. This change has two main benefits: (1) LSGAN assigns a penalty to samples based on their distance from the decision boundary, which ensures that G generates samples that are closer to the boundary, and (2) LSGAN generates stronger gradients for samples that are far from the boundary, which mitigates the gradient vanishing problem in traditional GAN. The optimal loss functions proposed in this paper are as follows:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}\left[(D(x)-1)^2\right] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z)))^2\right] \quad (2)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z))-1)^2\right] \quad (3)$$

According to reference [21], experiments have shown that adding the L1 loss function to the original loss function during the training of adversarial networks can lead to the generation of more realistic images. The L1 loss function is defined as follows:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\,||y - G(x,y,z)||_1\,] \quad (4)$$

in which y refers to the depth map from the training datasets that correspond to the real image x. As a result, the final loss function used in this paper can be expressed as follows:

$$\min_D V_{cGAN}(D) = \frac{1}{2}\mathbb{E}_{x,y}\left[(D(x,y)-1)^2\right] + \frac{1}{2}\mathbb{E}_{x,z}\left[(D(x,G(x,y,z)))^2\right] \quad (5)$$

$$\min_G V_{cGAN}(G) = \frac{1}{2}\mathbb{E}_{x,z}\left[(D(x,G(x,y,z))-1)^2\right] + \lambda\mathcal{L}_{L1}(G) \quad (6)$$

4. Experimental Results and Analysis

This section describes the experimental process and results of the monocular image depth prediction algorithm proposed in this study, which is based on op-cGAN.

4.1. Experimental Design

To take into account the complex and unique convergence process of cGAN training, we conducted our experiments in two stages. In the first stage, we compared our op-cGAN algorithm with several monocular image depth prediction algorithms based on classical deep learning models. This is because cGANs are generative models, and the generator in a cGAN can be trained to generate the predicted depth map y′ from an observed image x and a random noise vector z. In the second stage, we compared the monocular image depth prediction algorithm based on op-cGAN with the one based on the original cGAN. We evaluated the performance of these algorithms from both quantitative and qualitative perspectives. To conduct a comprehensive evaluation, we used two datasets: NYU-V2 for

indoor scenes and Make3D for indoor and outdoor scenes. We utilized industry-standard valuation metrics to assess the effectiveness of depth prediction from different viewpoints. Assuming that y_i is the actual depth value, y_i^* is the predicted depth value, and T represents the number of effective pixels, the evaluation metrics are described as follows:

$$\begin{aligned} \text{Absolute Relative Difference(rel)} &= \frac{1}{|T|}\sum_{y\in T}|y - y^*|/y^* \\ \text{Root Mean Squared Error(rmse)} &= \sqrt{\frac{1}{|T|}\sum_{y\in T}\left|\left|y - y^*\right|\right|^2} \\ \text{Root Mean Squared Log-Error(rmse(log))} &= \sqrt{\frac{1}{|T|}\sum_{y\in T}\left|\left|\log y - \log y^*\right|\right|^2} \\ \text{Mean } \log 10 \text{ Error :} \ \frac{1}{|T|}\sum_{y\in T}&\left|\log_{10}(y) - \log_{10}(y^*)\right| \end{aligned} \tag{7}$$

4.2. Training Method of Model

Unlike the general deep learning network model, cGAN has its own training method. Algorithm 1 shows the pseudo-code for the training procedure of monocular image depth prediction based on the op-cGAN.

Algorithm 1: Pseudo-code of monocular image depth prediction based on the cGAN

For the number of training iterations, do:
For *k* steps do:

- sample minibatch of m images $\{x^{(1)}, \ldots, x^{(m)}\}$ and corresponding depths images $\{y^{(1)}, \ldots, y^{(m)}\}$
- sample minibatch of m noise images $\{z^{(1)}, \ldots, z^{(m)}\}$
- update discriminator by descending its stochastic gradient when fixed generator gradient: $\min_D V_{cGAN}(D)$

End for

- sample minibatch of m images $\{x^{(1)}, \ldots, x^{(m)}\}$ and corresponding depths images $\{y^{(1)}, \ldots, y^{(m)}\}$
- sample minibatch of m noise images $\{z^{(1)}, \ldots, z^{(m)}\}$
- update generator by descending its stochastic gradient when fixed discriminator: $\min_G V_{cGAN}(G)$

End for

Training a neural network from scratch can be extremely challenging. Therefore, to achieve better results, the academic community usually relies on pre-trained network models. In this study, we pre-trained our model on ImageNet. Pre-training on ImageNet offers two benefits: (1) it speeds up the training process as the pre-trained model has learned feature extraction methods on millions of training examples, and fine-tuning is sufficient to achieve better results on small datasets; (2) it improves the results on the training set, as deep networks are challenging to train, and the millions of training examples on ImageNet can enhance the network's expressive power. In our experiment, we set the K value to 1 because the generator's main body is pre-trained with ResNet-50, which provides it with a strong feature extraction ability.

Furthermore, Batch normalization is highly effective in aiding the flow of gradients flow within the network and reducing the impact of parameter initial values on the training process. This allows for a higher learning rate during training and also helps to regularize

the model, reducing the need for Dropout operations. The formula for batch normalization is as follows:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right]}} \tag{8}$$

where $x^{(k)}$ represents the output of each layer's linear activation function. Batch Normalization is applied to this output by subtracting the mean and standard deviation of the minibatch it belongs to. This normalization transforms $\hat{x}^{(k)}$ into a normal distribution with mean 0 and variance 1, effectively solving the "internal covariate shift" problem. However, this transformation can reduce the expressive power of the network. To address this issue, the authors of [28] introduced two learnable parameters, scale, and shift, to each layer's output. With the addition of these two parameters, the original normalization method is modified into $y^{(k)} = r^{(k)}\,\hat{x}^{(k)} + \beta^{(k)}$, in which $r^{(k)}$ and $\beta^{(k)}$ has a size equal to the original batch size.

However, a drawback of this approach is that during inference, when there is only an input instance (i.e., batch size of 1), the statistics used for normalization become meaningless. To address this, practical deep-learning frameworks do not use batch normalization during inference. During training, the batch mean and variance are computed in the same way, but additional variables that are independent of batch size are retained to calculate the global mean and variance. During inference, batch normalization uses the global mean and variance that was calculated during training.

4.3. Experimental Results and Conclusions

4.3.1. NYU-V2 Dataset

We demonstrate the efficacy of our proposed algorithm using the NYU-V2 dataset, which is one of the largest indoor depth datasets worldwide. The NYU-V2 dataset consists of video sequences captured by Microsoft's Kinect camera. The dataset is divided into groups of continuous frames containing image and depth information, with some images being manually annotated with pixel categories. The dataset includes:

1. 1449 densely annotated aligned image-depth pairs;
2. Data from 464 new scenes across 3 cities;
3. 407,024 unannotated frames.

We sampled around 5000 data pairs evenly from the original dataset, with a pixel resolution of 480 × 640. As the dataset was collected over an extended period, there are many invalid pixels in the surroundings with a depth value of less than 0. To mitigate the impact of these invalid pixels, we excluded them during data processing by determining the average range of invalid pixels in all images and subtracting it from the corresponding training pairs in the original dataset. We then downsampled the data to 224 × 256, and to increase the training data diversity and avoid overfitting, we employed two data augmentation methods:

1. Random noise addition: Add some noise to each random vector during each training epoch, where the noise is sampled from a Gaussian distribution with a mean of 0 and a variance of 1.
2. Conditional vector addition: Use room type, indoor furniture, lighting, and other information vectors as conditional inputs to the generator to generate realistic images.

Each training data pair was augmented with one of these methods, resulting in a final set of 150,000 training pairs.

In the first stage of the experiments, we evaluated the performance of our op-cGAN-based monocular depth estimation model in comparison with established models such as AlexNet [5], VGGNet [6], ResNet [7], DORN [29], and the SOTA algorithm PixelFormer [30]. We carried out quantitative and qualitative assessments, and Table 1 shows the quantitative results. All evaluation metrics in this paper are sourced from the original papers. The generator named ResNet-Up-Decon in our op-cGAN model used the following parameters:

the learning rate r was set to 0.01, the optimization algorithm used was momentum = 0.9, the model was trained for 10 epochs, and the loss function is L1.

Table 1. Comparison of the proposed approach against other methods on the NYU-V2 dataset.

NYU V2	REL	RMSE	RMSE (log)	δ<1.25	δ<1.25^2	δ<1.25^3
Eigen et al. [5]	0.215	0.907	0.285	0.611	0.887	0.971
Eigen and Fergus [6]	0.158	0.641	0.214	0.769	0.95	0.988
Laina et al. [7]	0.127	0.573	0.195	0.811	0.953	0.988
DORN [29]	0.115	0.509	-	0.828	0.964	0.992
PixelFormer [30]	0.090	0.322	-	0.929	0.992	0.998
ours	0.115	0.492	0.167	0.878	0.972	0.992

Our proposed method outperforms all previous algorithms except for the latest algorithm PixelFormer, as measured by all metrics. While literature [5,6] use AlexNET and VGGNet as the model backbone, our method, which benefits from ResNet's stronger network representation, far exceeds the methods presented in these two papers in terms of results. Additionally, our method produces depth estimation images with higher resolution. It is worth mentioning that compared to [7], which also uses pre-trained ResNET-50 as the model backbone, our proposed method achieves a decrease of 0.11 in the rel metric, 0.081 in the rmse metric, and 0.28 in the rmse(log) metric. Most importantly, our method increases by 0.067 on the metric $\delta < 1.25$, which means that 5% more pixels fall within this range of estimated depth values than in [7]. This demonstrates the effectiveness of our proposed method and the improved depth estimation prediction resolution module. Compared with the DORN algorithm [29], which uses the spacing-increasing discretization (SID) strategy, our method still outperforms it in the rmse, log10, $\delta < 1.25$, and $\delta < 1.25^2$ metrics.

The latest algorithm, PixelFormer [30], uses an improved attention module (Skip Attention Module) and Bin Center Predictor (BCP) module. Based on the experimental results of the original paper, PixelFormer outperforms the proposed algorithm across all metrics. As the actual training data used by our algorithm is not exactly the same as the standard NYU-V2 dataset, the absolute difference in evaluation metrics between the two algorithms has little reference value. Nonetheless, PixelFormer still exhibited superior performance. In future work, we will integrate the Skip Attention Module and Bin Center Predictor module into the conditional generative adversarial network framework, and compare it to PixelFormer to explore the impact of the conditional generative adversarial network framework on depth estimation algorithms.

Figure 4 displays the visual results of two depth prediction algorithms based on the VGGNet [6] and ResNet model [7], both implemented by the authors and with model parameters provided in the published parameter files. The visualization shows that the method proposed in [6] can generate relatively good depth map estimations. However, the limited expressive power of the VGGNeT model used for feature extraction, results in many predicted depth values being significantly different from the actual values. Finally, our proposed method not only achieves more accurate depth estimation results, but also resolves the issue of overly smooth depth estimations to a certain extent.

In the second stage of the experiment, we compared the performance of the standalone generator model proposed in this paper with the op-cGAN model as a whole.

During the training of the standalone generator model, we used the momentum optimization algorithm with momentum set to 0.9, a batch size of 8, and an L1 loss function. The learning rate was set to 0.01, and we did not use a learning rate decrease method. The model was trained for 10 epochs, and the best-performing model on the test set was selected from the 10 trained models.

The cGAN training is different from traditional deep learning networks. After numerous experiments, we obtained a set of relatively good training parameters. The generator's learning rate was set to 0.0001, and we used the Adam optimization algorithm. In the loss function, we set the λ value to 10. For the discriminator, we set the learning rate to 5×10^{-4}, the batch size to 8, and used the Adam optimization algorithm.

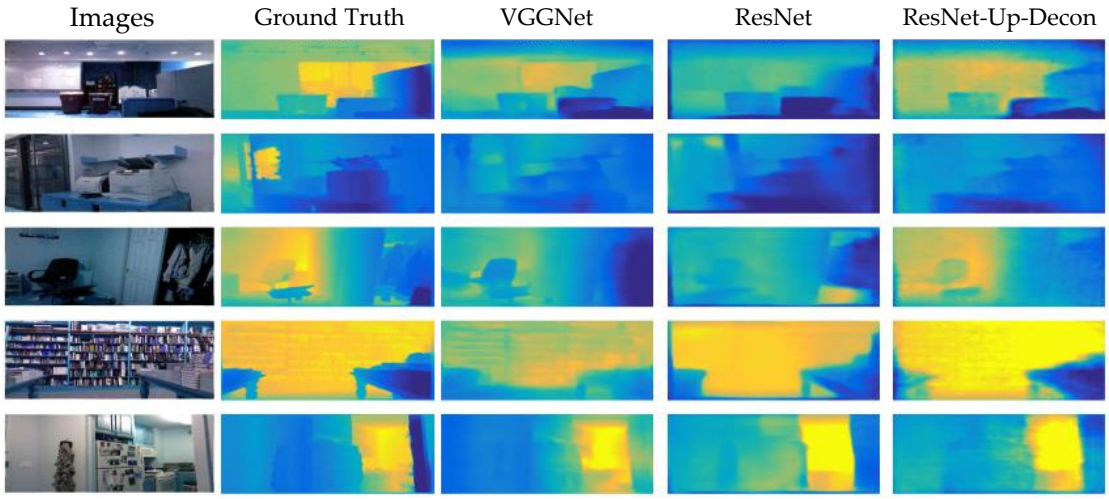

Figure 4. Experimental results on the NYU-V2 dataset of monocular image depth prediction algorithms based on different models.

Figures 5–7 show the experimental results of the two models under different evaluation metrics and training epochs.

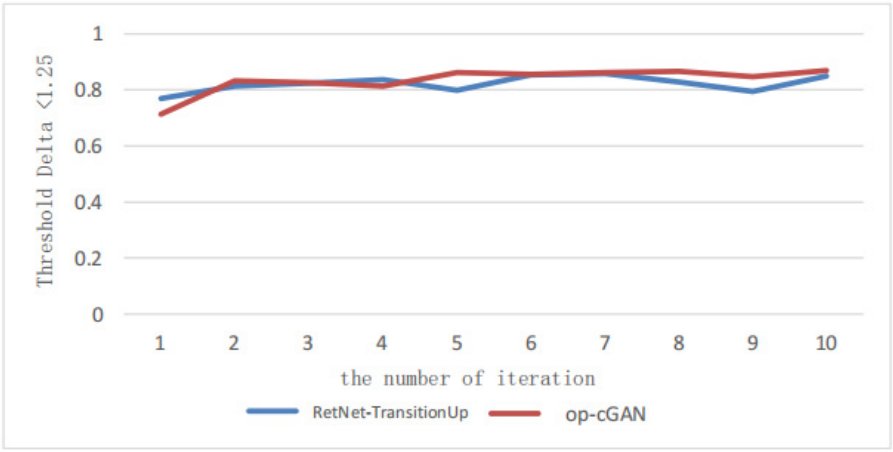

Figure 5. Experimental results of $\delta < 1.25$ for the two different models in each training iteration on the NYU-V2 dataset.

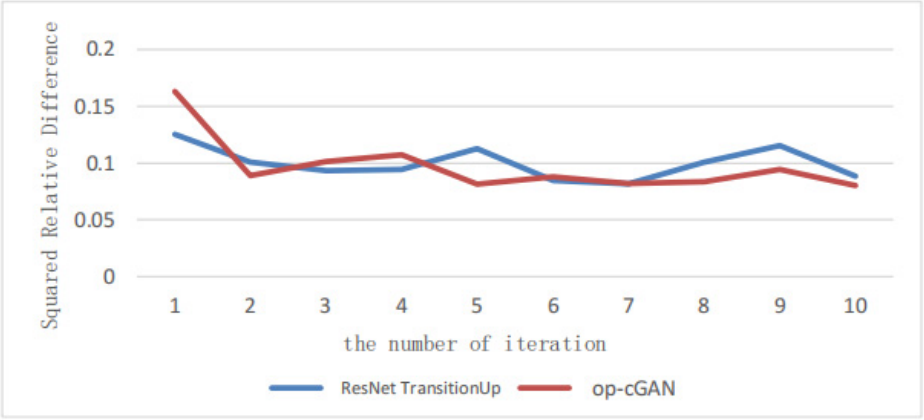

Figure 6. Experimental results of squared relative difference for the two different models in each training iteration on the NYU-V2 dataset.

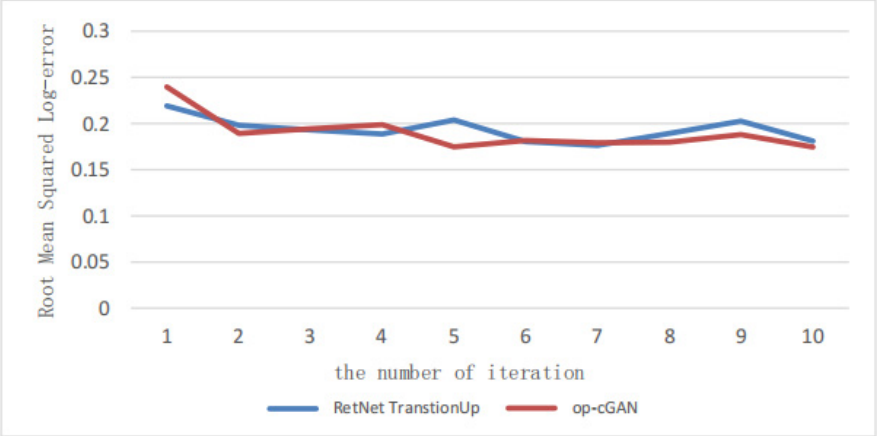

Figure 7. Experimental results of root mean squared log error for the two different models in each training epoch on the NYU-V2 dataset.

Figure 5 depicts the models' performance under different training epochs with a particular evaluation metric $\delta < 1.25$. The op-cGAN model significantly outperformed the standalone generator model after the first epoch and gradually improved in the following epochs. Moreover, the op-cGAN model exhibited greater stability and minimal fluctuations. After ten epochs, the op-cGAN model achieved a depth error value of 0.85, indicating that over 85% of the pixels' depth estimation values were smaller than the actual depth value. These results demonstrated the effectiveness of the op-cGAN model.

Figures 6 and 7 show the performance of the op-cGAN model compared to the standalone generator model in the evaluation metrics rmse and rmse (log) under different training epochs. The figures indicate that the op-cGAN model has a faster convergence rate and higher stability in these two evaluation metrics and outperformed the generator model significantly after the 5th epoch.

As this paper aims to address the issue of blurry depth maps generated by existing monocular image depth prediction algorithms, the visualization results are crucial. We saved the depth map visualization results to a file, adding a "0" as a separator between each depth map. Figure 8 shows selected depth estimation results from the test set, including the ground truth of the depth map, the depth map generated by the standalone generator, and that generated by the op-cGAN model from left to right. The visualization results indicate that the op-cGAN model generates clearer and more accurate depth maps, as demonstrated by the clear display of the windows in the first image's ground truth, object contours in the second image's ground truth, and the door and window in the third image's ground truth. These results confirm the effectiveness of our proposed op-cGAN model.

4.3.2. Make3D Dataset

Next, we will evaluate the performance of different depth estimation algorithms on the Make3D dataset. This dataset contains depth maps of indoor and outdoor scenes obtained from LIDAR scans. The official split includes 400 aligned image-depth pairs for training and 134 images for testing. Due to the age of this dataset, the resolution of the depth map is only 305×55, while the resolution of the images is 1704×2272. Therefore, during preprocessing, we first adjust the resolution of all training data to 256×192 using bilinear interpolation to serve as input to the model. Furthermore, because 400 image-depth pairs are insufficient for training a neural network, we use the following offline data enhancement methods to expand the training dataset.

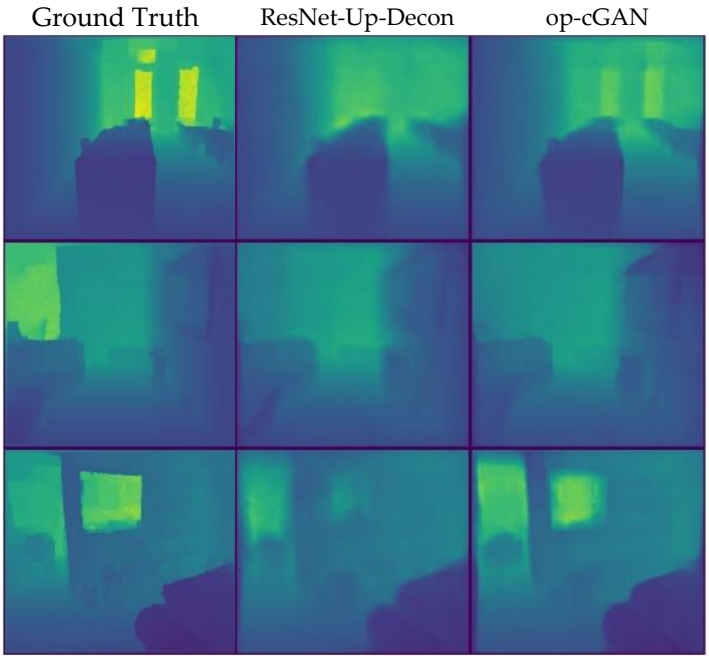

Figure 8. Comparison of visualization effects of different models on NYU-V2 dataset.

1. Scaling: the Input and target images are scaled with the corresponding depth data divided by $s \in [1, 1.5]$.
2. Rotation: the input and target images are rotated by $r \in [-5, 5]$ degrees.
3. Color adjustment: the Input image is multiplied by a random RGB value $c \in [0.8, 1.2]$.
4. Flips: the Input and target images are horizontally flipped with a 0.5 probability
5. Adding conditional vector: information vectors such as city or rural, lighting, and roads can be added as conditional vectors to the generator to generate realistic images.

Specifically, we applied the first four data enhancement methods to each of the original 400 training pairs to generate new training data pairs, which we repeated 10 times. Finally, one of the conditional information from the 5th method is chosen to obtain 50 K training data pairs.

In the first stage of the experiment, we compare the performance of the depth estimation algorithms based on the DCNN model and CRF model [31,32], ResNet model [7], and the generator in the op-cGAN model proposed in this paper, from both quantitative and visual perspectives. Because there are limited evaluations on this dataset, we used results reported in the literature for the quantitative comparison. For the visual results, we could not access the authors' visualization results, so we only present the results of the generator in the op-cGAN model proposed in this paper.

During training, we used the L1 loss function and momentum optimization algorithm with a value of 0.9. The generator was trained for 40 epochs, with the learning rate halved every 20 epochs. The quantitative results are shown in Table 2, where we observed that the proposed method in this paper has improved in all evaluation metrics except for the rmse metric, which is lower than the method proposed in [7]. The reason could be that the dataset is too small and of low quality, making it difficult to train such a large network.

Table 2. Comparison results of different algorithms in the Make3D dataset.

Make 3D	REL	RMSE	Log10
Li et al. [31]	0.335	9.39	0.137
Liu et al. [32]	0.278	7.19	0.092
Laina et al. [7]	0.223	4.89	0.089
ResNet-Up-Decon	0.214	6.99	0.083

The visual results are shown in Figure 9, from which we can see that the proposed method can predict the contours of the depth map well and has no scale prediction error, validating the effectiveness of our method. Due to the limitation of the original training set, the resolution of the image in the training dataset is much higher than that of the depth images, leading to many mismatches.

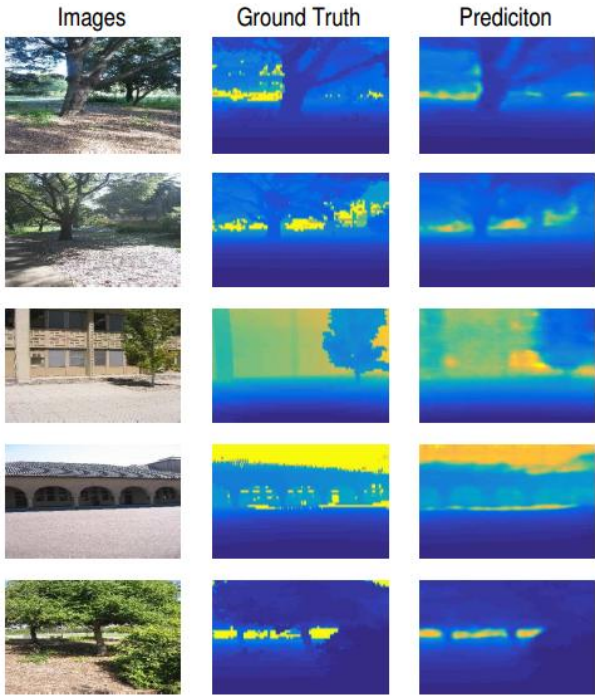

Figure 9. Visualization of Depth Prediction on Make3D.

In the second stage of the experiment, we compared the performance of a generator in the op-cGAN model with the full op-cGAN model for monocular depth estimation.

During the experiment, we trained the ResNet-Up-Decon model on the Make3D dataset, with all hyperparameters the same as those used for the NYU-V2 dataset, except for the number of training epochs, which was set to 20. Similarly, for the full op-cGAN model, we used the same hyperparameters as those used for NYU-V2, except for the number of training epochs, which was also set to 20.

Figures 10 and 11 display the experimental results of these two models under different training iterations. Specifically, Figure 10 presents the performance of the absolute relative difference metric for different training epochs. It can be observed that the depth estimation algorithm based on the cGAN model is superior to the ResNET-Up-Decon model in terms of stability and convergence speed. The former achieved a very low error rate in the first epoch and continued to reduce the error in subsequent epochs.

Figure 11 shows the performance of the log10 error evaluation metric across different training epochs. Invalid values (represented by 0) are caused by negative predicted depth values, which result in an invalid log10 error. The ResNet-Up-Decon model displays fewer invalid values than the op-cGAN model, indicating greater stability. Regarding the error values, the op-cGAN model has low errors, thus providing some validation of the proposed algorithm's effectiveness.

Figure 12 presents the visual results of the two models. For a fair comparison, all final depth values were scaled to the same scale. Values closer to the original depth pixel values are indicative of more accurate results. The op-cGAN model proposed in this paper performs better at recovering contour information from the images, which was attributed to the use of original images in the discriminator's input and the high pixel quality of the original images in the Make3D dataset, leading to the deep neural network learning the details of the original images.

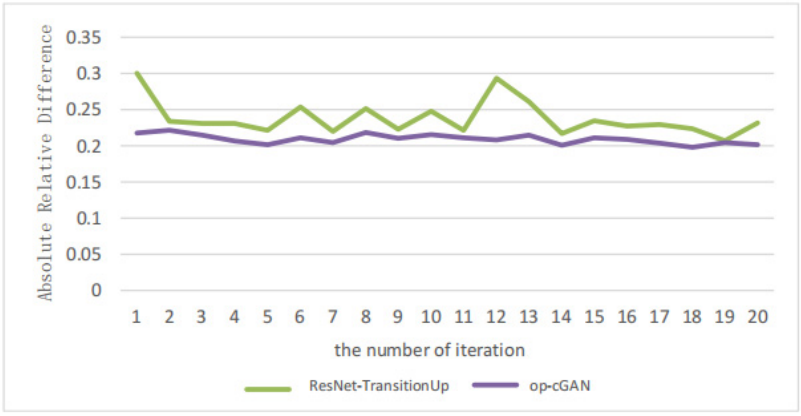

Figure 10. Experimental results of absolute relative difference for the two different models in each training iteration on the Make3D dataset.

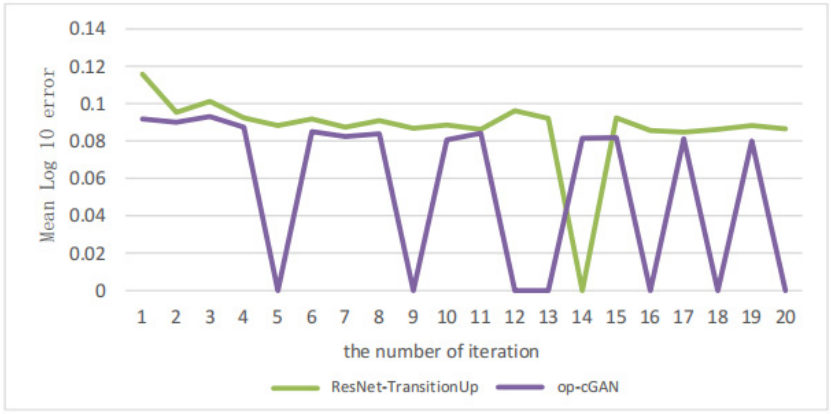

Figure 11. Experimental results of log10 error for the two different models in each training iteration on the Make3D dataset.

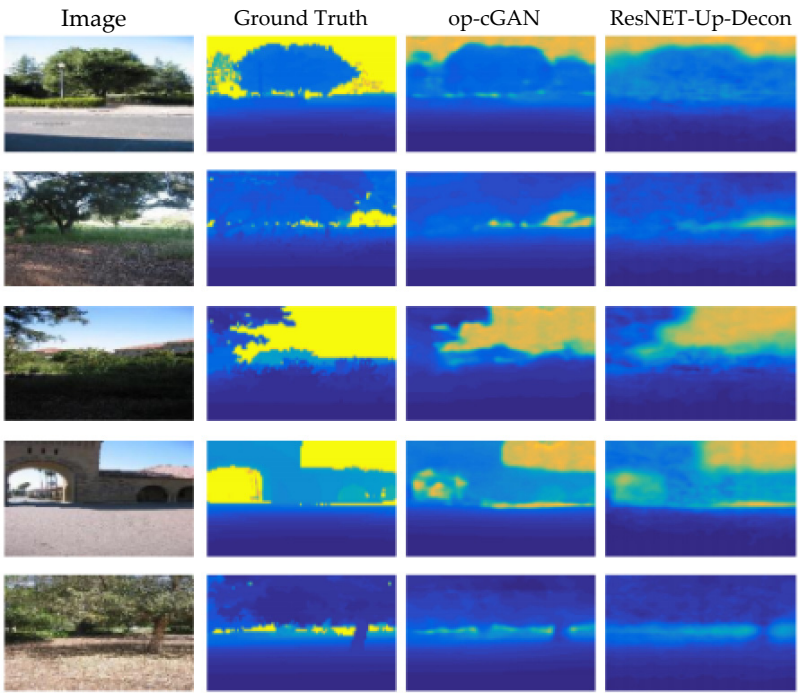

Figure 12. Visualization Comparison of depth prediction on Make3D dataset.

5. Limitation

The limitations of our approach primarily consist of two aspects:

1. The primary drawback of using GANs is that they can be challenging to train. Despite the implementation of various empirical tricks to improve efficiency (such as using batch normalization in our proposed method), GANs remain difficult to train.
2. Compared with the latest monocular image depth estimation algorithms, the performance of our algorithm is not outstanding enough, possibly because we have not optimized the generator structure optimization well, especially since the attention module has not been added. Experiments have shown that the attention mechanism can significantly improve the accuracy and detail extraction of image depth estimation.

6. Conclusions

Mobile robot plays an important role in the intelligent logistics and intelligent warehousing applications of the smart manufacturing industry. 3D map reconstruction is a core problem, which can help mobile robots achieve autonomous cruising and automatic obstacle avoidance in a complex environment. Monocular depth prediction is a fundamental method for understanding the 3D map's geometric information. This paper proposed an improved monocular image depth prediction method based on a conditional generative adversarial network to address the problem of insufficient diversity of training data and of overly blurry depth maps in monocular image depth prediction. Our method employed an improved monocular image depth estimation model based on depth residual networks as the generator of the conditional GAN, with a 5-layer patchGAN network as the discriminator. We combined the LSGAN loss function with the L1 loss function for the generator's loss function. Experimental results indicated that our proposed method can accelerate the convergence on the small Make3D dataset and can achieve a more optimized model on the larger NYN-V2 dataset, despite slower initial convergence. The visualization results show that our method can recover images with more detailed and clearer contours.

Although our proposed monocular depth estimation methods based on cGANs face difficulties in GAN network training and do not have the best performance on the evaluation metrics compared to the latest algorithm PixelFormer, their strong anti-interference with training sample and good model stability make these drawbacks acceptable. In the future, we will integrate the Skip Attention Module and Bin Center Predictor module into the conditional generative adversarial network framework, and compare it to PixelFormer again to explore the impact of the conditional generative adversarial network framework on depth estimation algorithms.

Author Contributions: Methodology, Investigation, Writing—original draft, S.H.; Writing—review & editing, L.Z.; software, validation, K.Q.; software, visualization, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2018YFB1701402, and the National Natural Science Foundation of China, grant numbers U1936218 and 62072037.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. NYU-V2 data can be found here: [https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html, accessed on 17 October 2022] and Make 3D data presented in this study are openly available in [Make3D] at [doi: 10.1109/TPAMI.2008.132].

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

References

1. Babic, B.; Miljkovic, Z.; Vukovic, N.; Antic, V. Towards Implementation and Autonomous Navigation of an Intelligent Automated Guided Vehicle in Material Handing Systems. *IJST-T Mech Eng.* **2012**, *36*, 25–40.
2. Jensen, L.K.; Kristensen, B.B.; Demazeau, Y. FLIP: Prototyping multi-robot systems. *Robot Auton. Syst.* **2005**, *53*, 230–243. [CrossRef]
3. Mahjourian, R.; Wicke, M.; Angelova, A. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
4. Chen, K.; Li, J.; Lin, W.; See, J.; Wang, J.; Duan, L.; Chen, Z.; He, C.; Zou, J. Towards accurate one-stage object detection with AP-loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 15–21 June 2019.
5. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multiscale deep network. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014.
6. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
7. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016.
8. Cao, Y.; Wu, Z.; Shen, C. Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 3174–3182. [CrossRef]
9. Liu, M.; Salzmann, M.; He, X. Discrete-Continuous Depth Estimation from a Single Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
10. Hu, J.; Ozay, M.; Zhang, Y.; Okatani, T. Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019.
11. Alhashim, I.; Wonka, P. High Quality Monocular Depth Estimation via Transfer Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
12. Bhat, S.F.; Alhashim, I.; Wonka, P. AdaBins: Depth Estimation Using Adaptive Bins. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
13. Kim, D.; Ka, W.; Ahn, P.; Joo, D.; Chun, S.; Kim, J. Global-Local Path Networks for Monocular 13. Depth Estimation with Vertical CutDepth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–23 June 2022.
14. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
15. Kuznietsov, Y.; Stuckler, J.; Leibe, B. Semi-supervised deep learning for monocular depth map prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
16. Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
17. Casser, V.; Pirk, S.; Mahjourian, R.; Angelova, A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In Proceedings of the AAAI Conference on Artificial Intelligence, Waikoloa, HI, USA, 27 January–1 February 2019.
18. Bhutani, V.; Vankadari, M.; Jha, O.; Majumder, A.; Kumar, S.; Dutta, S. Unsupervised Depth and Confidence Prediction from Monocular Images using Bayesian Inference. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
19. Almalioglu, Y.; Saputra, M.R.U.; Gusmão, P.P.B.d.; Markham, A.; Trigoni, N. GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
20. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
21. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
22. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the 33th International Conference on Machine Learning, San Juan, PR, USA, 2–4 May 2016.
23. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least Squares Generative Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
25. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

26. Berthelot, D.; Schumm, T.; Metz, L. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv* **2017**, arXiv:1703.10717.
27. Lsola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
28. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on International Conference on Machine Learning, Lile, France, 6–11 July 2015.
29. Fu, H.; Gong, M.M.; Wang, C.H.; Batmanghelich, K.; Tao, D. Deep Ordinal Regression Network for Monocular Depth Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake, UT, USA, 18–22 June 2018.
30. Agarwal, A.; Arora, C. Attention Attention Everywhere: Monocular Depth Prediction With Skip Attention. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 1–7 January 2023.
31. Li, B.; Shen, C.; Dai, Y.; Van Den Hengel, A.; He, M. Depth and surface normal estimation from monocular images using regressionon deep features and hierarchical crfs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
32. Liu, F.; Shen, C.; Lin, G. Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.