International Journal of Computer Vision (2024) 132:1012–1028
https://doi.org/10.1007/s11263-023-01915-6

Towards a Unified Network for Robust Monocular Depth Estimation: Network Architecture, Training Strategy and Dataset

Mochu Xiang[1] · Yuchao Dai[1] · Feiyu Zhang[1] · Jiawei Shi[1] · Xinyu Tian[1] · Zhensong Zhang[2]

Received: 17 December 2022 / Accepted: 15 September 2023 / Published online: 23 October 2023

Abstract

Robust monocular depth estimation (MDE) aims at learning a *unified* model that works across diverse real-world scenes, which is an important and active topic in computer vision. In this paper, we present `Megatron_RVC`, our winning solution for the monocular depth challenge in the Robust Vision Challenge (RVC) 2022, where we tackle the challenging problem from three perspectives: network architecture, training strategy and dataset. In particular, we made three contributions towards robust MDE: (1) we built a neural network with high capacity to enable flexible and accurate monocular depth predictions, which contains dedicated components to provide content-aware embeddings and to improve the richness of the details; (2) we proposed a novel mixing training strategy to handle real-world images with different aspect ratios, resolutions and apply tailored loss functions based on the properties of their depth maps; (3) to train a unified network model that covers diverse real-world scenes, we used over 1 million images from different datasets. As of 3rd October 2022, our unified model ranked consistently first across three benchmarks (KITTI, MPI Sintel, and VIPER) among all participants.

Keywords Monocular depth estimation · Robust · Unified network · Multi-dataset training

1 Introduction

Given a single input RGB image, monocular depth estimation (MDE) (Zhao et al., 2020; Ming et al., 2021; Masoumian et al., 2022) aims at estimating the corresponding depth map. As a fundamental yet challenging task in computer vision, MDE has various downstream applications such as autonomous driving (Geiger et al., 2013; Cordts et al., 2016), visual odometry (Zhan et al., 2018), special effects (Luo et al., 2020), and 3D reconstructions (Kopf et al., 2021; Xu et al., 2023; Yin et al., 2023); application scenarios span from indoor (Ji et al., 2021; Li et al., 2021; Wu et al., 2022) to outdoor (Vyas et al., 2022). Recently, the task has been greatly advanced thanks to the development of deep neural network architectures, evolving from the convolutional neural networks (Alhashim & Wonka, 2018; Lee et al., 2019) to the vision transformer (Ranftl et al., 2021; Yuan et al., 2022).

However, the success of current deep learning-based MDE methods is generally limited to a single dataset due to the significant domain shifts across different datasets. For example, the KITTI dataset (Geiger et al., 2013) and the VIPER dataset (Richter et al., 2017) concentrate on real-world urban driving scenarios while the Sintel dataset (Butler et al., 2012) contains synthetic movies. Therefore, some methods that are state-of-the-art (SOTA) on one dataset often cannot achieve comparable performance on another dataset without substantial adaptation. In practice, the deep MDE network models can overfit in scene contents, focal length, image sizes or depth sources (Facil et al., 2019).

Towards real-world applications across diverse scenes, a robust MDE model should generalize well across different scenarios without adaptation. Thus, we need to push methods to be robust and perform well across different datasets with fixed model parameters and hyperparameters. A unified network to solve more real-world monocular depth estimation problems is of high practical value and is in urgent need.

Recently, robust MDE has gained a lot of attention. The ability of a robust MDE model makes it applicable in diverse situations, which provides out-of-the-box MDE capability

Communicated by Oliver Zendel.

✉ Yuchao Dai
daiyuchao@gmail.com

[1] School of Electronics and Information, Northwestern Polytechnical University and Shaanxi Key Laboratory of Information Acquisition and Processing, Xi'an, China

[2] Huawei Noah's Ark Lab, Shenzhen, China

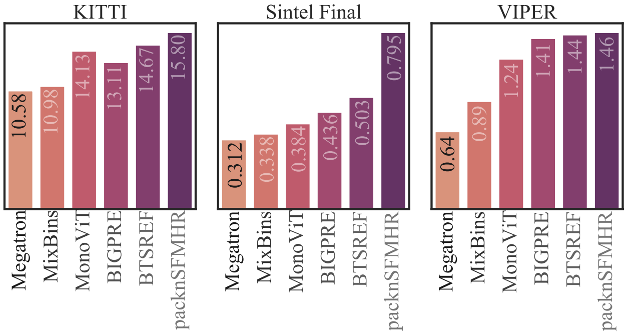


Fig. 1 Performance comparison between our method and SOTA methods on three datasets in terms of the SILog metric. Our method "Megatron_RVC" consistently outperforms all the competing methods and wins the challenge

for the community. These methods mainly seek solution in vision backbones (Ranftl et al., 2021) or diverse data collections, whether from large-scale web stereo data (Xian et al., 2018; Wang et al., 2019), human annotations (Chen et al., 2016) or Structure-from-Motion reconstructions (Li et al., 2018, 2019).

To foster the development of vision systems that are robust and consequently perform well on a variety of datasets with different characteristics, the Robust Vision Challenge (RVC) has been established (Fig. 1). The performance is measured across a number of challenging benchmarks with different characteristics, e.g., indoors versus outdoors, real versus synthetic, sunny versus bad weather, and different sensors (http://www.robustvision.net).

A robust monocular depth estimation model can not only be applicable to diverse real-world applications, achieving stronger transfer ability when be adapted to a specific domain, but also empower various downstream depth-related tasks (Xu et al., 2023; Yin et al., 2023; Zhan et al., 2018; Luo et al., 2020; Wang et al., 2020). However, practices towards training a robust MDE model have not been fully explored, which are mainly due to the difficulties in designing unified network architecture and effective training strategies. Furthermore, given massive amount of available open-sourced datasets, how to better explore their potentials is a problem that deserves in-depth studies.

In this paper, we rethink the essential ingredients towards a robust deep learning system and propose to tackle the task of robust and unified MDE from three perspectives: *network architecture, training strategy and dataset*. We name our method `Megatron_RVC`, an MDE model that performs constantly well on different benchmarks and is applicable in daily life. We explain the above three perspectives in details.

First, we exploit current SOTA vision backbones and present a unified network architecture, which not only is of high capacity, but also consists of components tailored for robust applications. We adopt a VQVAE module to provide content-aware embeddings. Furthermore, a convex upsampling module is utilized to improve the richness of the depth prediction details.

Second, we propose a novel multi-dataset mix training strategy called "Random Iterator Selection", which supports the native resolution and tailored loss functions for each of the datasets used. This strategy can avoid data bias among multiple datasets and is of high efficiency, especially when training in parallel with multiple GPUs.

Third, we collect millions of publicly available samples from multiple sources to provide supervision for our model, where the depth maps are either from depth sensors, stereo matching, multi-view reconstructions, synthetically rendered, or distilled from state-of-the-art MDE methods.

With such a large amount of data coupled with our mix training strategy, our unified network architecture with a large capacity of parameters can achieve robust performance across diverse scenarios. Our unified model ranked consistently first across three benchmarks (KITTI, MPI Sintel, and VIPER) among all participants and won the MDE track at the RVC challenge 2022.

Our main contributions are summarized as:

(1) We presented a network architecture that contains components tailored for robust MDE. We adopt a VQVAE module to provide content-aware embeddings, and a convex upsampling module to improve the richness of details of the depth prediction.
(2) We proposed a multi-dataset mix training strategy called "Random Iterator Selection", which supports the native resolution and tailored loss function for each dataset.
(3) We collected millions of publicly available samples from multiple sources to provide supervision for our model.
(4) Our method outperforms competing SOTA ones across different datasets and wins the monocular depth estimation track at the RVC challenge 2022.

2 Related Work

In this section, we briefly review related work in monocular depth estimation, large scale depth datasets and domain adaptation for robust MDE.

2.1 Monocular Depth Estimation

The task of Monocular Depth Estimation (MDE) (Zhao et al., 2020; Ming et al., 2021; Masoumian et al., 2022; Vyas et al., 2022) aims to predict the metric depth or relative distance given an input image. According to the source of supervision, monocular depth estimation can be roughly divided into three categories: supervised, self-supervised, and weakly supervised. The supervised MDE (Fu et al., 2018; Bhat et al., 2021; Yuan et al., 2022; Abdulwahab et al., 2022) methods directly learn the image-to-depth mapping from ground truth depth maps. Without needing ground truth depth, the self-supervised MDE methods (Masoumian et al., 2023; He et al., 2022; Zhou & Dong, 2022; Zhao et al., 2022a, b) learn to predict depth from left-right consistency (Godard et al., 2017) in stereo images or monocular video sequences (Zhou et al., 2017; Godard et al., 2019), where the supervision is achieved through view synthesis. The weakly-supervised MDE methods (Chen et al., 2016; Ren et al., 2020) learn the relative distance relationships from human annotations. In this paper, we confine our discussions to supervised MDE methods while the principle could be extended to other settings.

The supervised MDE methods generally outweigh the self-supervised and weakly-supervised methods due to the following aspects: (1) empirically, the supervised methods usually outperform the unsupervised and weakly-supervised ones in accuracy (Zhao et al., 2020); (2) theoretically, the supervised methods can make the model more robust in terms of achieving metric depth estimation and dealing with dynamic objects (Ming et al., 2021); (3) practically, the supervised methods are more parameter-effective and can incorporate training data from more diverse sources.

Recent years have witnessed tremendous progress in MDE, where various kinds of deep neural networks have been proposed. After Eigen et al.'s (2014) seminal work in utilizing the deep neural network for MDE, the ever-improving backbone networks have nourished many MDE models. For example, ResNet (He et al., 2016) has been exploited in (Laina et al., 2016) by Laina et al. (2017) while DenseNet has been exploited by DenseDepth (Alhashim & Wonka, 2018) and BTS (Lee et al., 2019). Recently, the network architecture EfficientNet (Tan et al., 2019) has been utilized by AdaBins (Bhat et al., 2021) to achieve accurate depth estimation. Very recently, with the development of vision transformers, the accuracy of MDE methods has been further improved and remarkable performance has been achieved. The Vision Transformer backbone ViT (Dosovitskiy et al., 2021) has been extended to MDE by DPT (Ranftl et al., 2021) while the Swin transformer (Liu et al., 2021) backbone has been utilized in NeWCRFs (Yuan et al., 2022). The backbone network plays a significant role in the ever-increasing performance of state-of-the-art MDE methods. It is worth noting that most of the above success is limited to a single dataset, i.e., different network models should be trained for each dataset separately.

Robust MDEs are usually achieved from the aspect of data. MiDaS (Ranftl et al., 2020) utilizes nearly 2 million samples to train an MDE model that can produce scale-invariant inverse depth predictions. Such ability is further improved in Ranftl et al. (2021) by switching to using vision transformer (Dosovitskiy et al., 2021) as the backbone network. The Mannequin Challenge (Li et al., 2019) and MegaDepth (Li et al., 2018) are large-scale depth datasets reconstructed from internet videos and image collections using the Structure-from-Motion technique, which mainly focus on the depth of human and buildings. The Red-Web (Xian et al., 2018) and WSVD (Wang et al., 2019) are two large-scale datasets consisting of stereo images and videos, where ordinal relationships can be extracted. Some methods utilize large-scale internet photo and video collections, obtain the depth information through Structure-from-Motion (Li et al., 2019, 2018) or stereo-matching (Xian et al., 2018; Wang et al., 2019), and train an MDE model towards diverse scenes. The pre-trained models of these methods provide the out-of-the-box capability of estimating depth from a single image, which greatly facilitate the prosperity of the 3D vision field. However, among these works, the strategies of efficiently mixing multiple datasets are not fully explored, especially when facing datasets with characteristics that vary greatly.

2.2 Large Scale Depth Datasets

A robust MDE model is expected to provide consistently good predictions under different scenarios, thus a dataset that contains diverse scenes is helpful for building MDE models with high generalization ability. Most models trained on specific datasets are difficult to generalize to unconstrained scenarios due to the strong data bias (Torralba & Efros, 2011). Commonly used datasets target a single scenario or topic. For example, the NYU dataset (Silberman et al., 2012) includes 1,449 indoor scenes, where the ground truth depth maps are captured by the Kinect sensor. The KITTI dataset (Geiger et al., 2013) concentrates on the urban autonomous driving scenarios, which mainly contains road scenes captured by cameras and a Lidar sensor mounted on the car. Similarly, Cityscapes (Cordts et al., 2016) consists of street scenes, and the depth information comes from stereo cameras. Make3D (Saxena et al., 2008) mainly consists of outdoor scenes of university campuses and DIODE (Vasiljevic et al., 2019) consists of more indoor and outdoor data, using Lidar to obtain dense depth maps.

Large-scale datasets containing more diverse scenarios obtain data from the Internet. Chen et al. (2016) presented the Depth in the Wild (DIW) dataset consisting of 495k web images annotated with relative depth pairs. Megadepth (Li et al., 2018) consists of 129K outdoor images collected from the Internet, where the depth maps are reconstructed using SfM (Schonberger & Frahm, 2016) techniques and are up-to-scale. ReDWeb (Xian et al., 2018), Holopix50k (Hua et al., 2020) and HRWSI (Xian et al., 2020) are stereo datasets containing more diverse daily scenes. Because these datasets cannot provide the ground truth depth captured by the sensors, they usually need pre- and post-processing steps to obtain reliable depth.

Restricted by imaging conditions, sensor application scopes and limited filming scenarios, real-world captured data could have its limitations. Therefore, some datasets (Hurl et al., 2019; Richter et al., 2017; Gaidon et al., 2016) contain synthetically rendered data from games and virtual engines are proposed. These datasets ease the problem of collecting data, but may suffer from the Synthetic to real gap (Zhao et al., 2019) when generalizing to real-world data.

2.3 Domain Adaptation

Due to the difference in probability distribution between training data and testing data, the performance of the model is often deteriorated due to domain distribution gaps (Quinonero-Candela et al., 2008). Therefore, it is important to improve the model's generalization ability.

Domain adaptation is to maximize the performance of the model in a known target domain by using the existing source domain. For domain adaptation methods for MDE, (Atapour-Abarghouei & Breckon, 2018) proposed a two-stage method, the first stage uses synthetic data to train an MDE model, and the second stage trains the model to transfer the style of synthetic data to real-world data. A twin pipeline training framework named T^{2}Net is proposed in Zheng et al. (2018), where a synthetic-to-realistic translation network and a task network for MDE learn jointly. Zhao et al. (2019) trained two symmetric style translation networks and two MDE networks in an end-to-end framework, learning from the ground truth labels in the synthetic domain and epipolar geometry in the synthetic domain.

Furthermore, by broadening the scope of the training data to include more scenarios, it is possible to increase the model's generalization ability, such as training on large-scale and diverse datasets. Existing work (Yin et al., 2021; Xu et al., 2022; Yin et al., 2021) train MDE by mixing high, medium and low quality data in the same proportion in each batch. DeMoN (Ummenhofer et al., 2017), CAM-Convs (Facil et al., 2019), MiDaS (Ranftl et al., 2020) utilized training data from multiple datasets to train an MDE network, where the mixing training strategy is essential toward an unbiased MDE model.

3 Overview

In this section, we present our solution "Megatron_RVC" for unified and robust MDE. We tackle the challenge from three perspectives, namely *network architecture, training strategy and dataset*, which are widely recognized as the essential ingredients towards a practical deep learning system.

First, in Sect. 4, we exploit current SOTA vision backbones and present a unified network architecture, which consists of components tailored for robust applications. We adopt a VQVAE module to provide content-aware embeddings and a convex upsampling module to improve the richness of the depth prediction.

Second, we present our novel multi-dataset mix training strategy "Random Iterator Selection" in Sect. 5, which enables us to train a robust single model with different datasets. To resolve the issue of *forgetting* in iteratively training multiple datasets, we propose to use the mix training strategy that supports various resolutions, assigns the proper loss function to use, and can reduce the dataset bias (Torralba & Efros, 2011).

Third, we discuss our diverse dataset that covers as rich contents as possible to provide a wider range of knowledge in Sect. 6. We have collected over 1 million samples from multiple datasets to guarantee a training dataset with high diversity.

4 Network Architecture

To achieve robust MDE, the design of the network architecture deserves extra consideration. Modern backbone networks could benefit downstream tasks to a large extent, and the entire network needs a large number of parameters to produce accurate depth prediction when applied to different scenarios.

We adopt the architecture of NeWCRFs (Yuan et al., 2022) as our baseline: Swin transformer V1 (Liu et al., 2021) is adopted as the backbone, a pyramid pooling module (Zhao et al., 2017) processes features from the bottleneck layer, and the NeWCRFs decoder predicts the depth map. We add a VQVAE module (Van Den & Vinyals, 2017) at the bottleneck of the network to provide content-aware embeddings. Furthermore, a convex upsampling module like (Teed & Deng, 2020) is used to upsample the final depth map to match the size of the input images. The convex upsampling module is adopted to replace the 4× bilinear sampling, which produces depth predictions with richer details. The network architecture is illustrated in Fig. 2. Below, we introduce each module in detail.

4.1 Encoder Network

We adopt the Swin transformer V1 as our encoder network. As shown in Fig. 2b, the encoder consists of four stages. At the beginning of each stage, the feature map is gradually downsampled through a linear embedding layer, followed by several Swin transformer blocks. Each transformer block contains two consecutive modules, called the Window-based Self-Attention and the **S**hifted **Win**dow

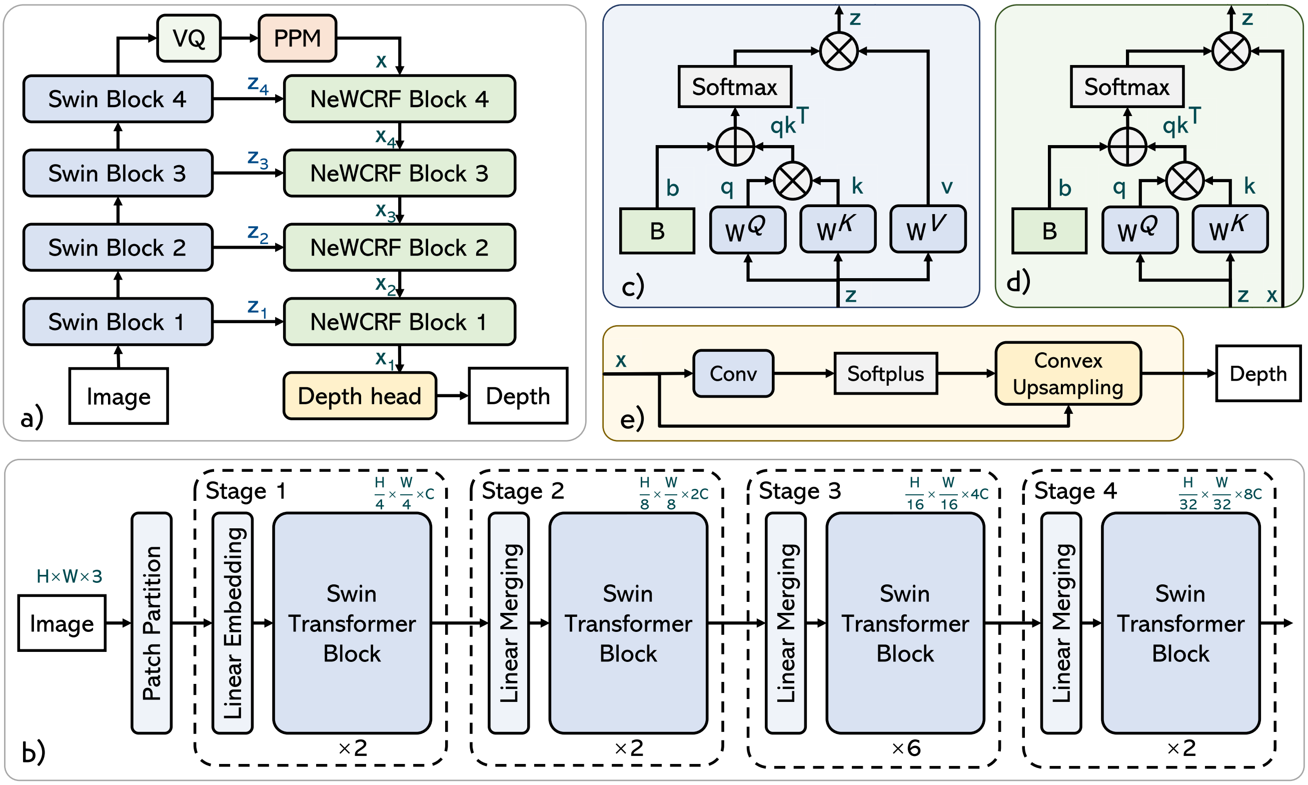


Fig. 2 The network architecture of our method. **a** overview of our network model; **b** the encoder architecture; **c** the attention module in Swin blocks; **d** the attention module in NeWCRFs blocks; **e** the architecture of the depth head. Figures are adapted from Liu et al. (2021, 2022) and Yuan et al. (2022)

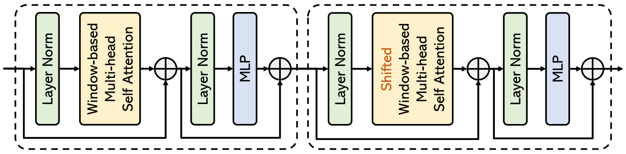

Layer Norm
Window-based Multi-head Self Attention
Layer Norm
MLP
Layer Norm
Shifted Window-based Multi-head Self Attention
Layer Norm
MLP

Fig. 3 The structure of the window-based self-attention and the shifted one

based Self-Attention, as shown in Fig. 3. The architecture of the self-attention module is illustrated in Fig. 2c.

We chose the Swin-L as our backbone network, which is initialized using the the parameters pre-trained on ImageNet 22K (Deng et al., 2009) with the resolution of 224×224.

4.2 Decoder Network

The structure of the decoder network highly resembles that of the encoder network. The basic mechanism of the attention operation is shown in Fig. 2, where the feature from the last decoder layer takes the place of *value*, while *query* and *key* are processed from the feature from the encoder layer.

4.3 The VQVAE Module

To provide content-aware embeddings without explicitly training the network to identify images from different domains, we adopt the VQVAE (Van Den & Vinyals, 2017) module at the bottleneck of the network, which is followed by a PPM head (Zhao et al., 2017) in Yuan et al. (2022).

The features c_i in the codebooks are initialized randomly. For each of the feature vector z_i' in the encoded feature map, the closest feature c_i' is selected to replace it, and we train our network to narrow down the distance between z_i' and c_i', and we also encourage diversity among features in the codebook c_i. Details can be found in Sect. 5.6.

4.4 Depth Prediction Head

We use SoftPlus as the final activation function to produce positive values representing metric depth estimations. The advantages of adopting SoftPlus as the activation function over using Sigmoid is that the network can produce positive predictions without upper bounds. This is beneficial under the multi-dataset setting, since their depth can have different maximum values.

5 Training Strategy

Most MDE methods focus on achieving accurate depth predictions on a *single* dataset, where images and depth maps are usually captured with a single set of devices, and the scenes are with similar contents. However, when training with data from multiple datasets, the situations are quite different. Given data from multiple domains, images could differ in aspect ratios, imaging styles, contents and themes while depth maps could also differ in sparsity and scales. In this section, we provide our training strategy towards robust MDE.

5.1 The Training Cycle

During our early experiments, we found that training a network with each dataset one after another will lead to the forgetting issue (Zhang et al., 2020). A network can achieve the best accuracy on a certain dataset after training with it, but after switching to training on another one that contains less similar scenes to the former one, the performance of validation on the first dataset will decrease dramatically. In other words, the model cannot achieve consistently good performance on all datasets.

To resolve this problem, an intuitive solution is to switch more frequently between datasets, and an ideal situation is that we can randomly sample images from different datasets in every single batch. However, by doing so, a prerequisite is that images should have the same sizes, thus they can be formed into a batch. This is feasible if we resize and crop the image and depth map into patches with the same size, and then randomly mix multiple datasets. However, the ill-posedness of the MDE task poses higher requirements on the resolution of images, i.e., it may bring ambiguities and hinder the training process.

To conduct analysis, we train our network on two datasets, KITTI and NYU. After training on a dataset for one epoch, we switch to training on the other one. Evaluations on both datasets are done periodically to monitor the training process. The absolute relative error on the testing dataset is reported in Fig. 4. Our network fits on the two datasets at different time steps, and after starting the training process on another dataset, the performance drops dramatically. This indicates that the network can forget the learned dataset-specific knowledge very soon, especially for a heavily data-dependent task like MDE.

In order to prevent the network from forgetting too fast, we increase the switching frequency, and show the evaluation results on the KITTI dataset in Fig. 5. As the switching frequency increases, we observe that the upper bound of the evaluation error decreases as the training process goes on, which indicates that the network *forgets* less when being reminded more often.

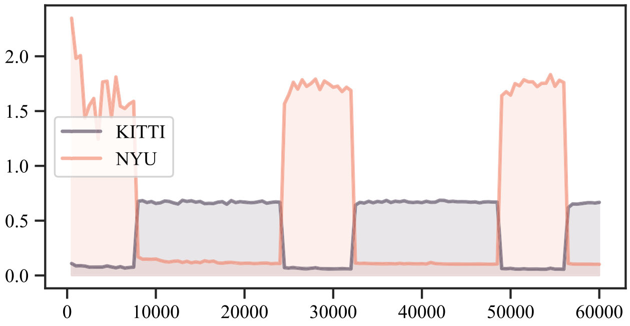


Fig. 4 Evaluation error on KITTI and NYU when training iteratively between two datasets, the switching frequency is every 1 epoch

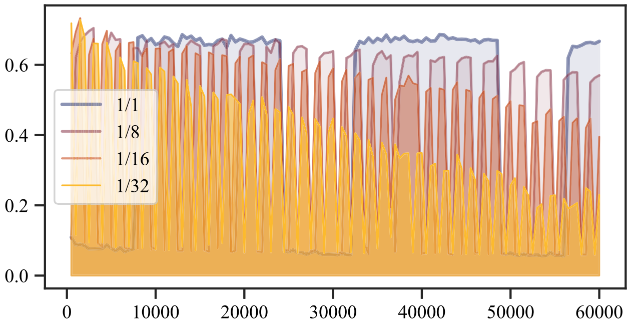


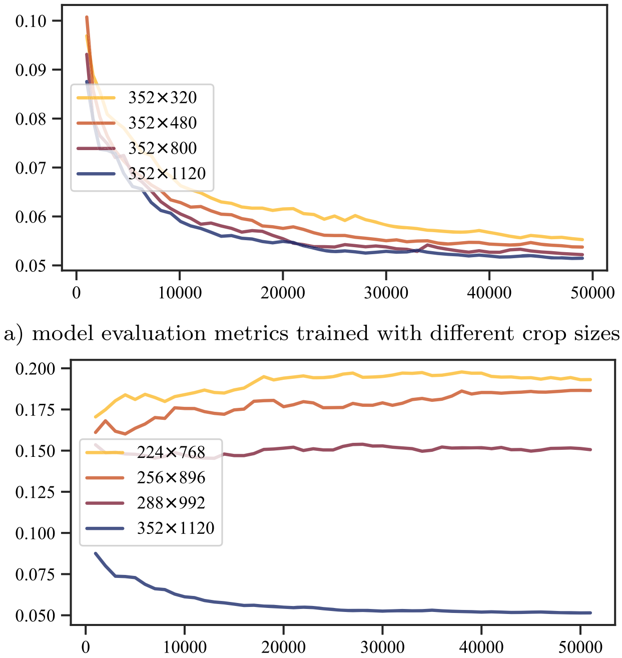
Fig. 5 Evaluation error of KITTI when training iteratively between two datasets with different switching frequencies, e.g., every 1/8 epoch

5.2 Resolution Overfitting

As discussed in Miangoleh et al. (2021), the MDE networks are sensitive to changes in the resolution of input images. The diversity of image resolution in multiple datasets also requires training in the native size, e.g., the images in the KITTI dataset (Geiger et al., 2013) are almost 3 times wide as images from the NYU dataset (Silberman et al., 2012).

In Fig. 6, we demonstrate several toy examples to illustrate the effect brought by resizing and cropping. We train the network with different (a) cropping and (b) resizing configurations on KITTI, and show the evaluation results periodically. Figure 6a shows that training with the native resolution will lead to the highest accuracy under certain time steps, but samples after random croppings without changing the aspect ratios of image contents can make the model eventually converge to similar degrees of accuracy. In Fig. 6b, we can observe that resizing the image to lower resolutions can harm the model accuracy severely.

We can draw a conclusion that for the most of the efficiency and accuracy, it is best to train MDE models with images of native sizes.



a) model evaluation metrics trained with different crop sizes;

b) model evaluation metrics trained with different resized samples.

Fig. 6 Absolute relative error evaluated on KITTI periodically when training with different cropping and resizing configurations

5.3 Random Iterator Selection

Discussions in Sect. 5.1 reveal that to train an MDE network on a mixed dataset, the ideal solution is to equally have images from each dataset in one batch. Section 5.2 reveals that it is better to train on a dataset with the native resolution and aspect ratio, however, this puts us in a dilemma where images with different resolutions cannot form a batch.

Switching between datasets more frequently is an alternative, and the extreme condition is to randomly choose another dataset at every step. When training with multiple GPUs, this randomness can be further extended to each process, which can be an approximation to having samples with different sizes in one batch.

Since we train our model with multiple GPUs in parallel, each GPU receives a batch of data of the same size, but samples in different batches do not necessarily need to be of the same size. Thus for each GPU process, we randomly choose a batch of images from a dataset. For the most of the efficiency, we assign different batch sizes, e.g., if the images are small, we increase the batch size as long as they fit into the memory.

We name our method as *Random Iterator Selection*. In practice, we randomly select data iterators per GPU process when training using the Data Distributed Parallel model on multiple GPUs. Using this technique, each GPU is randomly assigned a batch of images with the same size from a dataset independently, and multiple GPUs can have different data allocations, which is equivalent to having inputs from randomly chosen datasets, with different sizes considering all batches when accumulating gradients.

The above pipeline also allows us to effectively choose the appropriate loss function to apply, which we will introduce in Sect. 5.4. We store these iterators as *values* in a dictionary, and their names are the *keys*. We randomly sample one key-value pair for each step, so that we can identify them using the keys and fetch data from the iterator. Since each GPU receives samples from one dataset, the attributes of their depth maps are identical, then each process only needs to choose one loss function and apply it to all samples in one GPU batch.

Since we do not want datasets with huge capacities to dominate the training process, we manually reduce the probability that a large dataset can be selected, shown as *Sub* in Table 1. The intention of our multi-resolution mixing strategy resembles that of the multi-scale sampler in Mehta and Rastegari (2021), but we provide a simpler implementation, which supports sample rate adjustments and dedicated loss function allocation.

5.4 Multi-Loss Function Training

Existing MDE datasets provide depth supervision in different forms (*cf.* Table 1), thus we have to assign different loss functions for different datasets based on the property of their depth maps. For datasets with absolute scale (denoted as *Metric* in Table 1), we use the SILog loss $\mathcal{L}_{Log}$ as in Yuan et al. (2022),

$$\mathcal{L}_{\text{Log}} = \sqrt{\frac{1}{K}\sum_i \Delta d_i^2 - \frac{\lambda}{K^2}\left(\sum_i \Delta d_i\right)^2}. \quad (1)$$

For depth prediction $\{\hat{d}_i\}^K$ and ground truth depth $\{d_i^*\}^K$, where K is the total number of pixels in an image, $\Delta d_i = \log \hat{d}_i - \log d_i$ is the per-pixel log difference, and $\lambda = 0.85$ makes the SILog loss not absolutely scale invariant. For datasets with affine ambiguity (denoted as *Affine*), we use affine invariant loss $\mathcal{L}_{\text{Affine}}$ as in Ranftl et al. (2021),

$$\mathcal{L}_{\text{Affine}} = \frac{1}{K}\sum_i \rho_i \left|\hat{d}_i^* - d_i^*\right|, \quad (2)$$

where

$$d_i^* = \frac{d_i - \text{median}(\{d_i\})}{1/K|\sum_i (d_i - \text{median}(\{d_i\})|}, \quad (3)$$

is the normalized ground truth depth, so as for depth prediction $\hat{d}_i^*$, and ρ_i makes the loss function ignores the top 20%

Table 1 Datasets used for network training, which cover both indoor and outdoor scenes

Dataset	Length	Image contents	Depth source	Depth accuracy	Subsample rate	Loss function
MegaDepth (Li et al. 2018)	129K	Outdoor	Multi-View	Up-to-scale	1/10	Affine
NYU (Silberman et al. 2012)	50K	Indoor	Kinect	Metric	1/10	Metric
KITTI (Geiger et al. 2013)	43K	Outdoor	Lidar	Metric		Metric
DIODE (Vasiljevic et al. 2019)	25K	In/Outdoor	Lidar	Metric	1/10	Metric
HRWSI (Xian et al. 2020)	20K	In/Outdoor	Stereo	Affine		Affine
Cityscapes (Cordts et al. 2016)	20K	Outdoor	Stereo	Metric	1/10	Metric
ReDWeb (Xian et al. 2018)	3.6K	In/Outdoor	Stereo	Affine		Affine
DIML (Kim et al. 2016)	3.1K	In/Outdoor	Kinect+Stereo	Metric		Metric
PreSIL (Hurl et al. 2019)	45K	Synthetic	Synthetic	Metric		Metric
VKITTI (Cabon et al. 2020)	21K	Synthetic	Synthetic	Metric	1/5	Metric
EDEN (Le et al. 2021)	13K	Synthetic	Synthetic	Metric	1/10	Metric
Sintel (Butler et al. 2012)	2.9K	Synthetic	Synthetic	Metric		Metric
GTAV (https://github.com/gta5-vision/GTA5-depth-estimation)	0.8K	Synthetic	Synthetic	Metric		Metric
ImageNet (Deng et al. 2009)	1.3M	In/Outdoor	DPT (Ranftl et al. 2021)	Inverse-depth prediction	1/20	Inverse-Affine
MegaDepth (Li et al. 2018)	129K	Outdoor	DPT (Ranftl et al. 2021)	Inverse-depth prediction	1/10	Inverse-Affine
VIPER (Richter et al. 2017)	13K	Synthetic	DPT (Ranftl et al. 2021)	Inverse-depth prediction		Inverse-Affine

items. For inverse-depth predictions $\{p_i\}^K$ from DPT (Ranftl et al., 2021) (denoted as *Inv-affine*), we convert our depth into inverse-depth ($\hat{p}_i = 1/\hat{d}_i$), and $\mathcal{L}_{\text{InvAffine}}$,

$$\mathcal{L}_{\text{InvAffine}} = \frac{1}{K} \sum_i \left| \hat{p}_i^* - p_i^* \right|, \tag{4}$$

where the difference between the two normalized inverse-depth maps are measured.

5.5 Training Steps

We use 6 RTX 3090 GPUs to train our model in parallel, and the entire training process requires approximately 600 GPU hours.

We first verify the effectiveness of the network architecture by training and evaluating on the KITTI (Geiger et al., 2013) Eigen split (Eigen et al., 2014). During this period, we follow the settings of Yuan et al. (2022) and train the network for 50 epochs, the model with the best SILog is kept for the next step. This process takes around 100 GPU hours.

Then, we implement a large-scale pretraining, where images are randomly cropped into smaller and fixed sizes. We use all datasets in Table 1 except KITTI, and load the parameters from the last step. The performance is tracked by evaluating on the KITTI Eigen split periodically. The model that achieves the best SILog metric on KITTI is kept for the next step, and this process takes around 500 GPU hours.

Finally, we finetune our model using KITTI, Sintel (Butler et al., 2012) and PreSIL (Hurl et al., 2019), each with their own sizes and aspect ratios. Empirically, we first split the Sintel dataset into training and testing parts. The mix-training process only uses the training set, and the performance on the testing set is monitored periodically. We notice that the evaluation performance no longer decreases after around 10K steps, so we assume that the network fits on all three datasets well. Finally, we use all available samples in the datasets and manually stop training at 10K steps. This process takes less than 20 GPU hours.

5.6 Loss Functions

Apart from the loss terms measured between the predicted and the ground truth depth, we need another two losses to train the VQVAE module. $\mathcal{L}_{Sim}$ encourages similarities between the encoded features z and the features c in the codebook, and $\mathcal{L}_{Dis}$ encourages dissimilarities among features in the codebook:

$$\begin{aligned} \mathcal{L}_{\text{Sim}} &= \sum_i \|z_i - c_i\|, \\ \mathcal{L}_{\text{Dis}} &= \sum_{i,j} \exp\left(-\|c_i - c_j\|\right). \end{aligned} \tag{5}$$

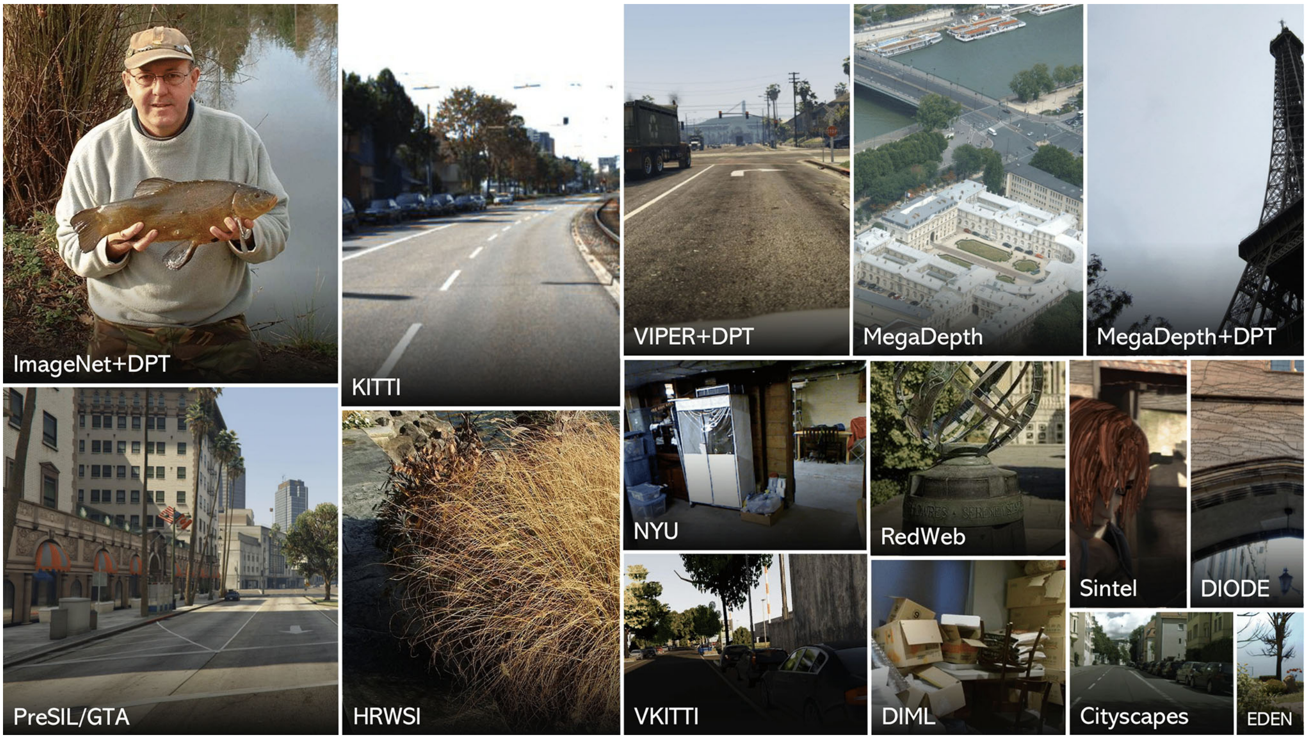


Fig. 7 Overview of the datasets used for training our unified network model. The area of each block corresponds to the equivalent capacity after selection probability reducing introduced in Sect. 5.3 in log-scale

Table 2 Performance comparison between our solution and competing methods across different metrics

	KITTI		Sintel Final		Sintel Clean		VIPER	
Method	SIlog	iRMS	SIlog	iRMS	SIlog	iRMS	SIlog	iRMS
Megatron	**10.58**	**11.57**	**0.312**	**0.269**	**0.222**	0.280	**0.64**	**42.22**
MixBins	10.98	11.77	0.338	0.272	0.333	**0.269**	0.89	82.71
MonoViT	14.13	14.85	0.384	0.276	0.410	0.281	1.24	162.9
BIGPRE	13.11	26.39	0.436	0.317	0.453	0.315	1.41	187.42
BTSREF	14.67	16.84	0.503	0.322	0.550	0.337	1.44	201.36
packnSFMHR	15.80	17.96	0.795	0.389	0.765	0.386	1.46	201.94

Best performance in bold

Our final loss for each sample is reached as

$$\begin{aligned}\mathcal{L} = & (\mathcal{L}_{\text{Log}} \text{ or } \mathcal{L}_{\text{Affine}} \text{ or } \mathcal{L}_{\text{InvAffine}}) \\ & + \omega_1 \mathcal{L}_{Sim} + \omega_2 \mathcal{L}_{Dis},\end{aligned} \tag{6}$$

where $\omega_1 = 0.5$ and $\omega_2 = 0.2$. We use $\mathcal{L}_{\text{Log}}$ or $\mathcal{L}_{\text{Affine}}$ or $\mathcal{L}_{\text{InvAffine}}$) based on the characteristics of each dataset.

6 Dataset

The task of MDE is heavily data-dependent, to train a unified network model that works well across diverse real-world scenes, we use over 1 million images from different datasets. We use publicly available datasets (Fig. 7) to train our network, whose details are reported in Table 1.

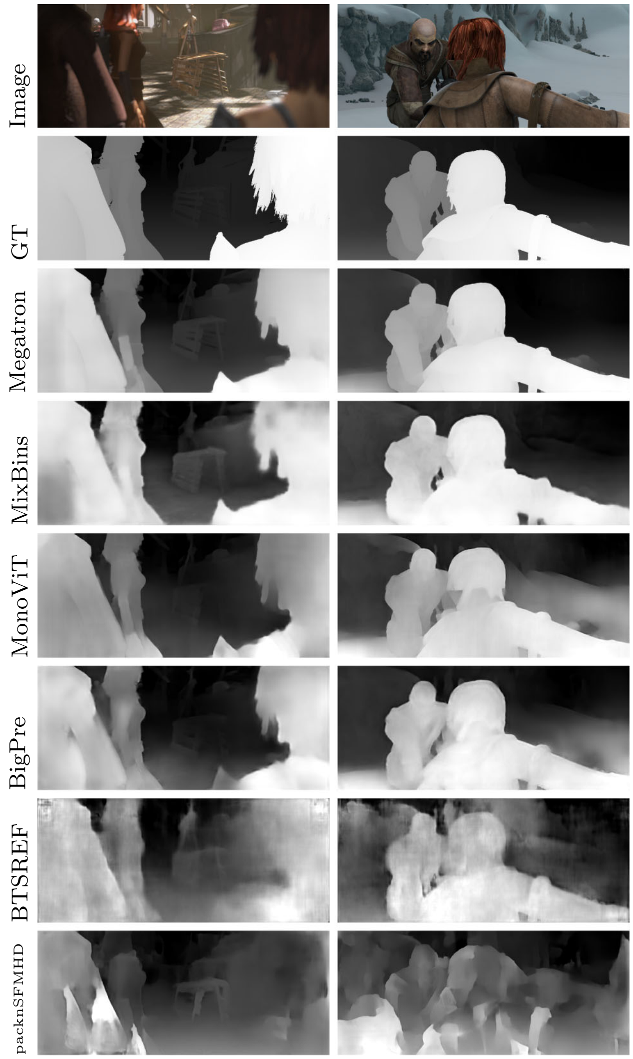


Fig. 8 Visual comparisons between methods of participants. Figures are taken from official benchmark of Sintel-depth

According to the manner how the depth maps are captured, existing monocular depth datasets can be roughly classified into the following categories: (1) Captured using active depth sensors: NYU (Silberman et al., 2012), KITTI (Geiger et al., 2013), DIODE (Vasiljevic et al., 2019), DIML (Kim et al., 2016); (2) Computed by stereo matching: HRWSI (Xian et al., 2020), Cityscapes (Cordts et al., 2016), ReDWeb (Xian et al., 2018), DIML (Kim et al., 2016); (3) Computed by structure-from-motion: MegaDepth (Li et al., 2018); (4) Synthetically rendered PreSIL (Hurl et al., 2019), GTA (https://github.com/gta5-vision/GTA5-depth-estimation), VKITTI (Cabon et al., 2020), Eden (Le et al., 2021), Sintel (Butler et al., 2012) and (5) Predicted by using state-of-the-art monocular depth estimation methods DPT (Ranftl et al., 2021) on ImageNet 1K (Deng et al., 2009), MegaDepth (Li et al., 2018), VIPER (Richter et al., 2017). These datasets cover urban autonomous driving scenarios (Geiger et al., 2013; Cabon et al., 2020; Cordts et al., 2016; Hurl et al., 2019; Richter et al., 2017) (https://github.com/gta5-vision/GTA5-depth-estimation), indoor daily life (Silberman et al., 2012; Vasiljevic et al., 2019; Kim et al., 2016) and synthetic contents (Hurl et al., 2019; Cabon et al., 2020; Le et al., 2021; Butler et al., 2012) (https://github.com/gta5-vision/GTA5-depth-estimation).

Our dataset collections contain images with different resolutions, aspect ratios and field-of-views. Images in the PreSIL dataset (Hurl et al., 2019) have resolutions up to 1080 × 1920, NYU (Silberman et al., 2012) contains images of smaller sizes at 480×640. Images in KITTI (Geiger et al., 2013)/VKITTI (Cabon et al., 2020) are extremely wide, with 3.45:1 aspect ratios, while ImageNet has images vertically shot. Datasets whose depth maps are actively captured usually contain samples with fixed focal length as the same devices are used. The datasets whose depth maps are from stereo matching, multi-view reconstructions or from state-of-the-arts generally contain images with various field-of-views.

Based on the characteristics of each dataset, appropriate loss functions should be used by considering the properties of each dataset. We show the corresponding loss functions in Table 1, where each loss function has been introduced in Sect. 5.4.

7 Experimental Results

7.1 The RVC Leaderboard

The Robust Vision Challenge held at ECCV 2022 requires competitors to use a single model and achieve good performance on multiple benchmarks. For MDE, the benchmarks include KITTI (Geiger et al., 2013), Sintel (Butler et al., 2012) and VIPER (Richter et al., 2017).

As of 3rd, October 2022, our model ranked first on all the three benchmarks. Detailed metrics are reported in Table 2, depth predictions on the Sintel dataset are visualized in Fig. 8.

7.2 Wider Applications

Trained on diverse data shown in Fig. 7, our model learns to predict accurate depth map given diverse scenes (shown in Fig. 9). Our model not only achieves top metrics in the benchmark, but also can be applied well in daily life scenarios. We show model inference results on unseen diverse scenes from the DIW (Chen et al., 2016) dataset in Fig. 10 to demonstrate the strong generalization ability of our robust MDE model.

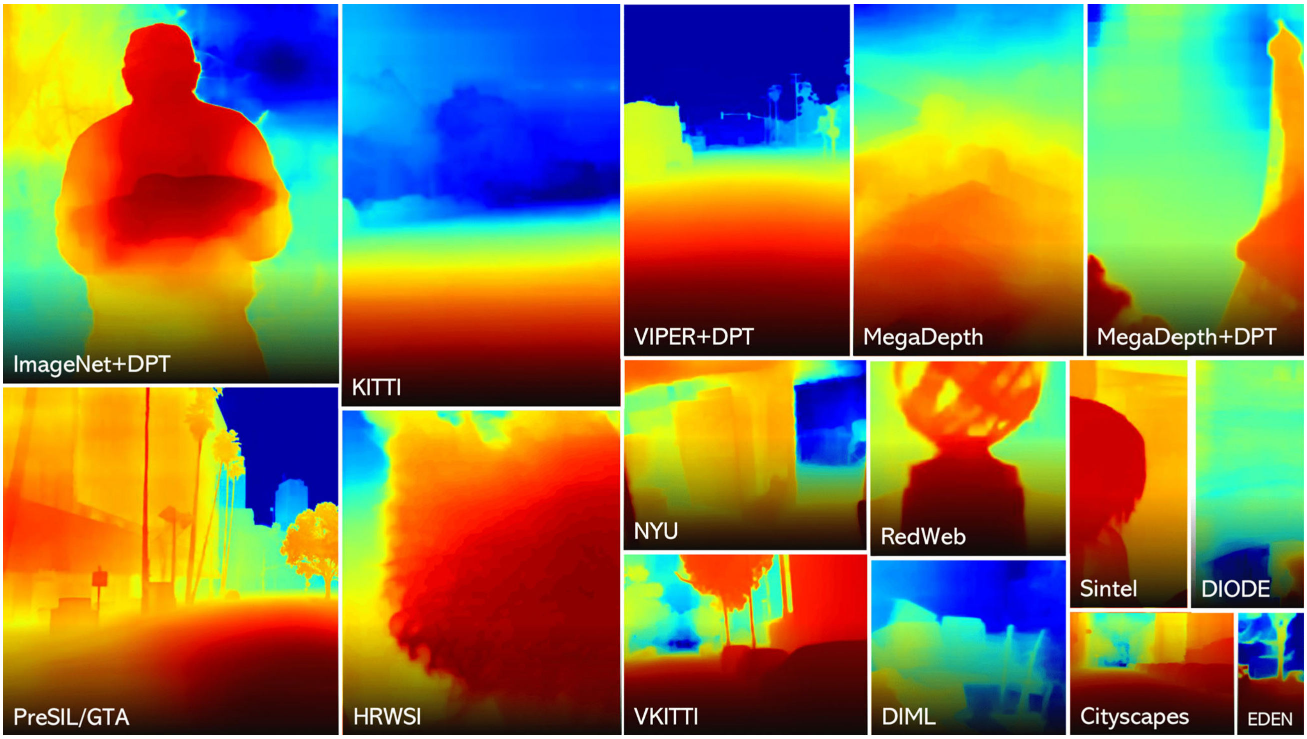


Fig. 9 Model predictions on datasets used for training

8 Ablations and Discussions

In this section, we conduct a series of ablation studies to analyze the contribution of each module of our unified network architecture. Furthermore, we report the inference time. We conclude this section with discussions on limitations and failure cases.

8.1 Backbone Networks

We conducted ablation studies on the KITTI dataset by switching to using other backbone networks, including Vision GNN (Han et al., 2022), ConvNext (Liu et al., 2022) and CSwin (Dong et al., 2022), the best performance of them is reported in Table 3.

We find that modern backbones with large-scale parameters achieve similar performance, but the Swin transformer stands out, considering that it can bring out the most accuracy in KITTI.

8.2 The VQVAE Module

As introduced in Sect. 4.3, we added a VQVAE module at the bottleneck layer of the network. We hope that such a module can provide content-related embeddings, which serves as a data-specific guidance but without explicitly identifying images from different domains.

We conduct an experiment where we adopt the same settings in Fig. 5, but we build two models with and without the VQVAE module. We show in Fig. 11 that after adding the VQVAE module, the network achieves better performance when not training on the corresponding dataset. This indicates that the VQVAE module can provide embeddings that are helpful for robust MDE.

Table 4 shows the effectiveness of the VQVAE module when the model is trained with fully-mixed NYU and KITTI. The model with the VQVAE module can achieve better performance simultaneously on two datasets than the model without.

8.3 Depth Regression

Table 5 demonstrates the effectiveness of the convex upsampling module and the activation function. A visual comparison of convex upsampling and bilinear upsampling is shown in Fig. 12. The convex upsampling module helps the network produce depth estimations with sharper boundaries, while the softplus activation function helps further improve the accuracy, while providing more flexibility in terms of metric depth range.

8.4 Depth Scale Over-Fitting

We show in Fig. 5 the absolute relative error when trained iteratively on two datasets, and the error shows extreme

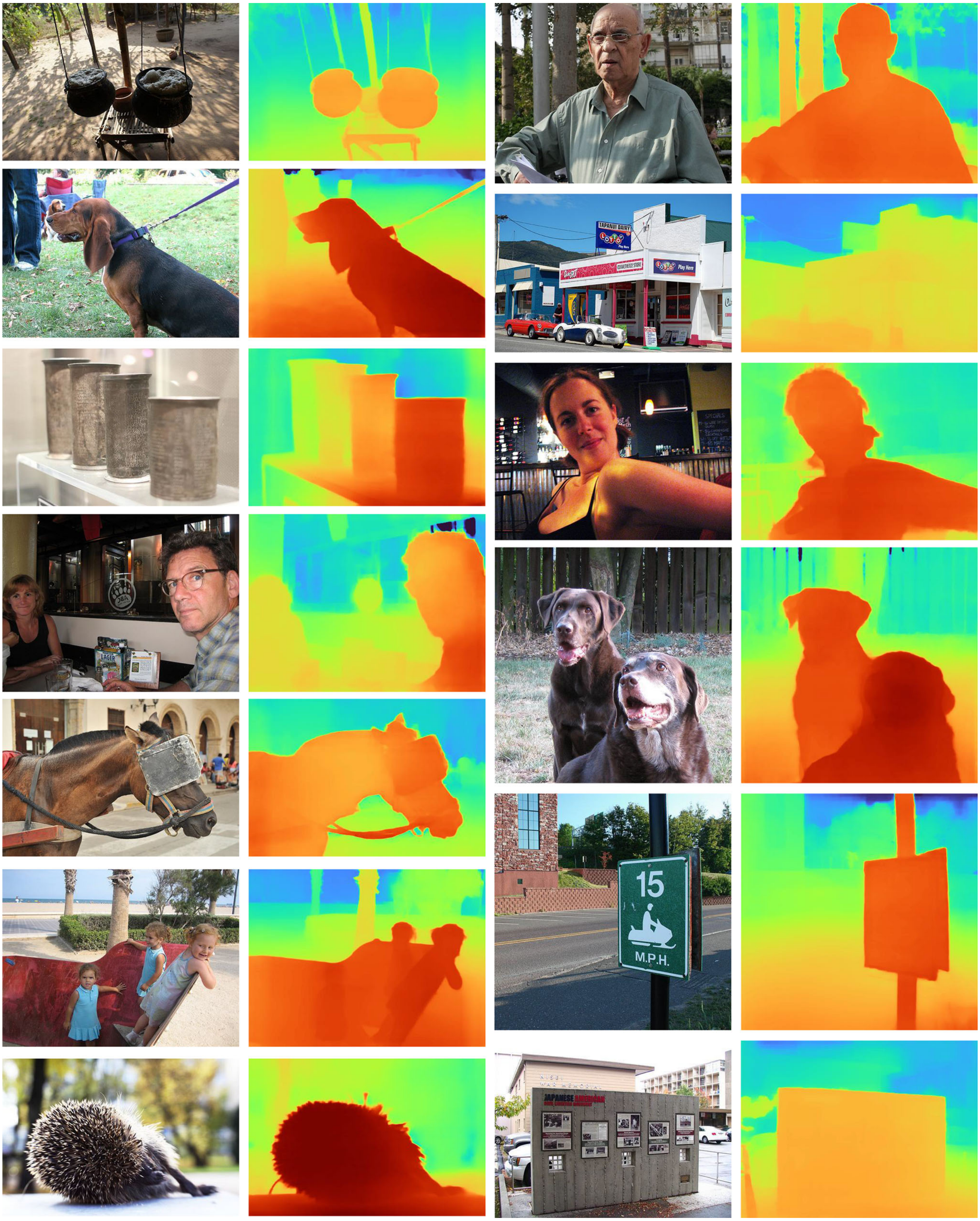


Fig. 10 Qualitative results on the DIW dataset, our unified MDE model owns excellent generalization ability

Table 3 Performance comparison of MDE models adopting different backbone networks

Encoder	SILog	Rel	Log10	RMS	δ_1
SwinV1 224 (Liu et al. 2021)	7.0393	0.0518	0.0225	2.0626	0.9758
ViG-B 224 (Han et al. 2022)	9.2040	0.0643	0.0285	2.5990	0.9504
ConvNext (Liu et al. 2022)	7.2793	0.0550	0.0243	2.1898	0.9728
CSwin (Dong et al. 2022)	7.5797	0.0551	0.0240	2.2261	0.9706

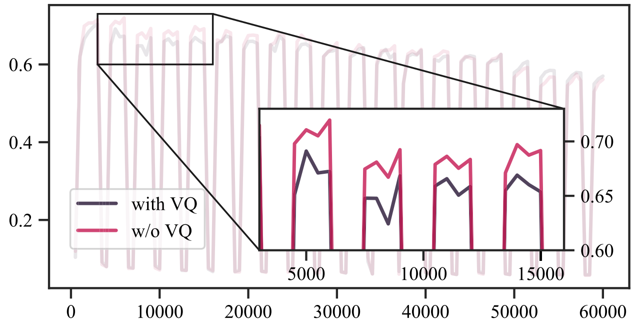


Fig. 11 Evaluation error on the KITTI dataset with and without the VQVAE module

Table 4 The effectiveness of the VQVAE module on the joint training of KITTI and NYU

Dataset	VQ	Rel	RMS	Log10	δ_1
NYU	✓	**0.1017**	**0.3752**	**0.0434**	**0.9075**
		0.1053	0.3800	0.0442	0.9023
KITTI	✓	**0.0538**	2.1572	**0.0235**	**0.9736**
		0.0554	**2.1500**	0.0246	0.9728

Best performance in bold

Table 5 The effectiveness of the depth regression module

Up	Act	Rel	RMS	Log10	δ_1
Bil	σ	0.0516	2.1071	0.0226	0.9750
Cvx	σ	**0.0515**	2.0981	**0.0224**	0.9755
Cvx	SP	0.0518	**2.0626**	0.0225	**0.9758**

Up and Act denote the upsampling method and activation function. Bil and Cvx represent bilinear upsampling and convex upsampling, respectively; σ and SP represents sigmoid and softplus, respectively
Best performance in bold

deviation on different training steps. Since this metric is scale-sensitive, we report the SILog metric, which is scale-invariant, in Fig. 13.

Compared with scale-sensitive errors, the scale invariant error shows less severe deviations. This indicates that predicting metric depth is harder than predicting the structural relationships, but the *forgetting* issue still exists, and the scale-invariant metric gradually deteriorates when not trained on the specific dataset.

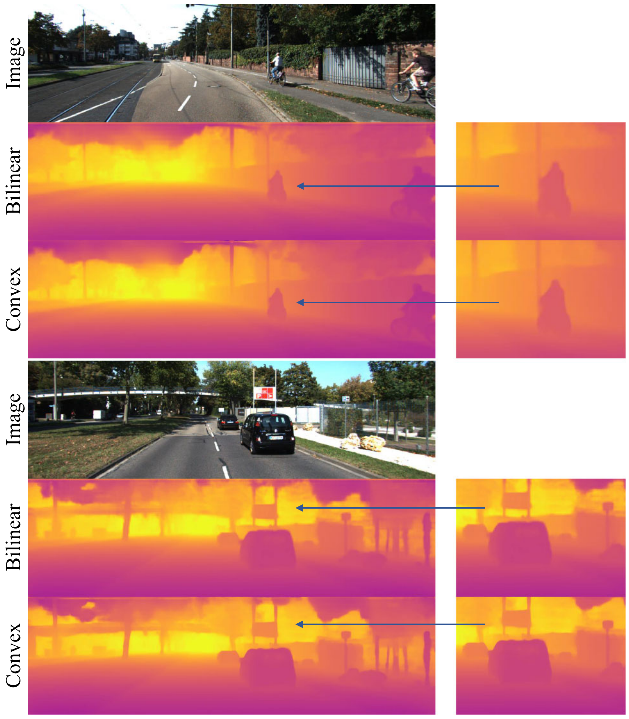


Fig. 12 A visual comparison of upsampling methods

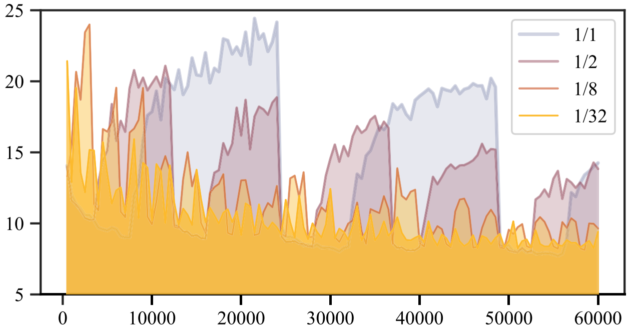


Fig. 13 The scale invariant error of KITTI with different iterating frequency

8.5 Benefits for Zero-Shot MDEs

MiDaS (Ranftl et al., 2020) proposes to measure the error after the per-image scale and shift alignment in the inverse-depth space. Following this, we measure the performance of our model on TUM after applying the optimal affine transfor-

Table 6 Evaluation metrics on the TUM Dynamic Object subset

Method	SILog	Rel	Log10	RMS	δ_1
NeWCRFs	34.3105	0.3056	0.1212	1.1989	0.5186
+data	19.4656	0.1498	0.0638	0.5958	0.7768
Ours	24.9534	0.1914	0.0805	0.8291	0.7303
+data	17.1665	0.1251	0.0542	0.5209	0.8480

All metrics are measured between predictions after the optimal affine transformation, where parameters (shift and scale per image) are obtained through least square fitting

Table 7 The inference speed of our method and DPT given inputs with different sizes

Size	224×224	384×384	512×512	576×1024
Ours(ms)	46.67	47.43	47.93	49.60
DPT(ms)	35.56	37.82	42.70	57.57

mation in the depth space. Likely-wise, the two parameters are obtained by least-square fitting.

We report the metrics of our model and NeWCRFs (Yuan et al., 2022) on the *Dynamic Objects* subset of TUM in Table 6. We also train both networks on the *3D Object Reconstruction* category of TUM for 3 epochs, for the domain information of TUM, and report the evaluation results on the Dynamic Objects category, in Table 6 (see +*data* rows).

The results in Table 6 indicate that our model owns excellent generalization ability, which can save training time. When the data from a target domain is available, higher performance can be achieved.

8.6 Inference Speed

Our model contains 372 M parameters, and we show in Table 7 the inference speed on a single NVIDIA 3090 GPU given input images with different sizes. The results are averaged over 1000 runs. Our model can provide robust MDE results with a reasonable speed, which further proves the practicability of our method. Thanks to the windowed attention mechanism in both the encoder and decoder, our model is with almost identical speed given inputs with different resolutions, which can support faster inference with input of larger sizes.

8.7 Limitation and Failure Cases

We show our failure cases in Fig. 14, where samples are from the DIW (Chen et al., 2016) dataset. Our method can fail to generate depth predictions under specific conditions where images may contain too complex or too blurry scenes. Our model can make common mistakes where it can produce depth predictions with counter-intuitive ordinal relationships.

As mentioned in Sect. 5.4, our model directly produces depth estimations. This works fine for evaluation on benchmarks, but may reduce the robustness when predicting depth for far-away objects, since some datasets cannot provide valid supervision in the sky region, and they can vary on the maximum depth value.

9 Conclusion

In this paper, we proposed `Megatron`, our winning solution to the monocular depth estimation track in the Robust Vision Challenges 2022. We tackled the challenging task from three different perspectives, namely, *network architecture, training strategy and dataset*. Our network with tailored components, trained using diverse dataset show strong performance across all three benchmarks, and can produce plausible MDE predictions on various scenes. Our proposed mix training strategy *Random Iterator Selection* supports various image resolutions and tailored loss functions. Our solution towards robust and unified MDE is not limited to the task of MDE, but can also be transferred to other tasks.

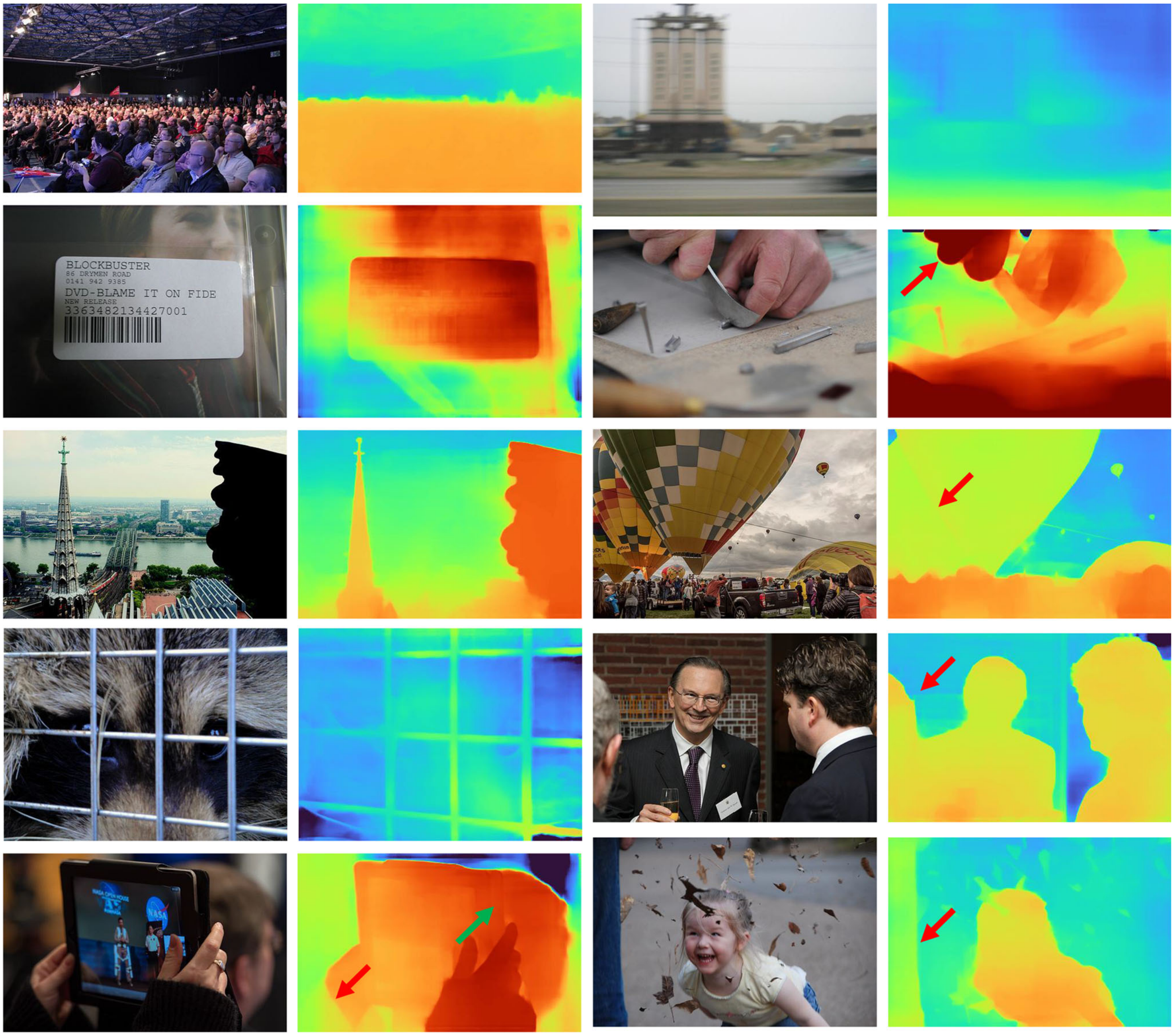


Fig. 14 Failure cases. Our method can be less robust when facing too complex structures, too blurry images, close-up photo shooting scenarios, and may make counter-intuitive depth estimations which are indicated by arrows

Acknowledgements This research was supported in part by the National Natural Science Foundation of China (62271410) and the Fundamental Research Funds for the Central Universities.

Data Availability All data generated or analysed during this study are included in articles or links in Table 1.

References

Abdulwahab, S., Rashwan, H. A., Garcia, M. A., Masoumian, A., & Puig, D. (2022). Monocular depth map estimation based on a multiscale deep architecture and curvilinear saliency feature boosting. *Neural Computing and Applications, 34*(19), 16423–16440.

Alhashim, I., & Wonka, P. (2018). High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941

Atapour-Abarghouei, A., & Breckon, T. P. (2018). Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2800–2810).

Bhat, S. F., Alhashim, I., & Wonka, P. (2021). Adabins: Depth estimation using adaptive bins. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4009–4018).

Butler, D. J., Wulff, J., Stanley, G. B., & Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision (ECCV)* (pp. 611–625).

Cabon, Y., Murray, N., & Humenberger, M. (2020). Virtual KITTI 2. arXiv preprint arXiv:2001.10773

Chen, W., Fu, Z., Yang, D., & Deng, J. (2016). Single-image depth perception in the wild. In *Advances in neural information processing systems (NeurIPS)* (vol. 29).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE confer-*

ence on computer vision and pattern recognition (CVPR) (pp. 3213–3223).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 248–255).

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., & Guo, B. (2022). CSWin transformer: A general vision transformer backbone with cross-shaped windows. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp 12124–12134).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems (NeurIPS)* (vol. 27).

Facil, J. M., Ummenhofer, B., Zhou, H., Montesano, L., Brox, T., & Civera, J. (2019). CAM-convs: Camera-aware multi-scale convolutions for single-view depth. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 11826–11835).

Fu, H., Gong, M., Wang, C., Batmanghelich, K., & Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2002–2011).

Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4340–4349).

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research, 32*(11), 1231–1237.

Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 270–279).

Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *IEEE international conference on computer vision (ICCV)* (pp. 3828–3838).

Gta5-depth-estimation, Retrieved July 26, 2022. https://github.com/gta5-vision/GTA5-depth-estimation

Han, K., Wang, Y., Guo, J., Tang, Y., & Wu, E. (2022). Vision GNN: An image is worth graph of nodes. arXiv preprint arXiv:2206.00272

He, M., Hui, L., Bian, Y., Ren, J., Xie, J., & Yang, J. (2022). RA-depth: Resolution adaptive self-supervised monocular depth estimation. In *European conference on computer vision (ECCV)* (pp. 565–581).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).

Hua, Y., Kohli, P., Uplavikar, P., Ravi, A., Gunaseelan, S., Orozco, J., & Li, E. (2020). Holopix50k: A large-scale in-the-wild stereo image dataset. arXiv preprint arXiv:2003.11172

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4700–4708).

Hurl, B., Czarnecki, K., & Waslander, S. (2019). Precise synthetic image and LiDAR (PreSIL) dataset for autonomous vehicle perception. In *IEEE intelligent vehicles symposium (IV)* (pp. 2522–2529).

Ji, P., Li, R., Bhanu, B., & Xu, Y. (2021). MonoIndoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *IEEE international conference on computer vision (ICCV)* (pp. 12787–12796).

Kim, Y., Ham, B., Oh, C., & Sohn, K. (2016). Structure selective depth superresolution for RGB-D cameras. *IEEE Transactions on Image Processing (TIP), 25*(11), 5227–5238.

Kopf, J., Rong, X., & Huang, J. B. (2021). Robust consistent video depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1611–1621).

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. In *International conference on 3D vision (3DV)* (pp. 239–248).

Le, H. A., Mensink, T., Das, P., Karaoglu, S., & Gevers, T. (2021) EDEN: Multimodal synthetic dataset of enclosed garden scenes. In *IEEE winter conference on applications of computer vision (WACV)* (pp. 1579–1589).

Lee, J. H., Han, M. K., Ko, D. W., & Suh, I. H. (2019). From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326

Li, Z., & Snavely, N. (2018). MegaDepth: Learning single-view depth prediction from internet photos. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2041–2050).

Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., & Freeman, W. T. (2019). Learning the depths of moving people by watching frozen people. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4521–4530).

Li, B., Huang, Y., Liu, Z., Zou, D., & Yu, W. (2021). StructDepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *IEEE international conference on computer vision (ICCV)* (pp. 12663–12673).

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., & Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 12009–12019).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE international conference on computer vision (ICCV)* (pp. 10012–10022).

Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 11976–11986).

Luo, X., Huang, J. B., Szeliski, R., Matzen, K., & Kopf, J. (2020). Consistent video depth estimation. *ACM Transactions on Graphics (ToG), 39*(4), 71–1.

Masoumian, A., Rashwan, H. A., Abdulwahab, S., Cristiano, J., Asif, M. S., & Puig, D. (2023). GCNDepth: Self-supervised monocular depth estimation based on graph convolutional network. *Neurocomputing, 517*, 81–92.

Masoumian, A., Rashwan, H. A., Cristiano, J., Asif, M. S., & Puig, D. (2022). Monocular depth estimation using deep learning: A review. *Sensors, 22*(14), 5353.

Mehta, S., & Rastegari, M. (2021). MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International conference on learning representations (ICLR)*.

Miangoleh, S. M. H., Dille, S., Mai, L., Paris, S., & Aksoy, Y. (2021). Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9685–9694)

Ming, Y., Meng, X., Fan, C., & Yu, H. (2021). Deep learning for monocular depth estimation: A review. *Neurocomputing, 438*, 14–33.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset shift in machine learning*. MIT Press.

Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision transformers for dense prediction. In *IEEE international conference on computer vision (ICCV)* (pp. 12179–12188).

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 44*(3), 1623–1637.

Ren, H., Raj, A., El-Khamy, M., & Lee, J. (2020). SUW-Learn: Joint supervised, unsupervised, weakly supervised deep learning for monocular depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR) workshop* (pp. 750–751).

Richter, S. R., Hayder, Z., & Koltun, V. (2017). Playing for benchmarks. In *IEEE international conference on computer vision (ICCV)* (pp. 2232–2241).

Saxena, A., Sun, M., & Ng, A. Y. (2008). Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 31*(5), 824–840.

Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4104–4113).

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. In *European conference on computer vision (ECCV)* (pp. 746–760).

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning (ICML)* (pp. 6105–6114).

Teed, Z., & Deng, J. (2020). RAFT: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision (ECCV)* (pp. 402–419).

The robust vision challenge (2022). http://www.robustvision.net

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1521–1528).

Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., & Brox, T. (2017). DeMoN: Depth and motion network for learning monocular stereo. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5038–5047).

Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. In *Advances in neural information processing systems (NeurIPS)* (vol. 30).

Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F. Z., Daniele, A. F., Mostajabi, M., Basart, S., & Walter, M. R., Shakhnarovich, G. (2019). Diode: A dense indoor and outdoor depth dataset. arXiv preprint arXiv:1908.00463

Vyas, P., Saxena, C., Badapanda, A., & Goswami, A. (2022). Outdoor monocular depth estimation: A research review. arXiv preprint arXiv:2205.01399

Wang, C., Lucey, S., Perazzi, F.,& Wang, O. (2019). Web stereo video supervision for depth prediction from dynamic scenes. In *International conference on 3D vision (3DV)* (pp. 348–357).

Wang, X., Yin, W., Kong, T., Jiang, Y., Li, L., & Shen, C. (2020). Task-aware monocular depth estimation for 3d object detection. In *AAAI conference on artificial intelligence (AAAI)* (vol. 34, pp. 12257–12264).

Wu, C.Y., Wang, J., Hall, M., Neumann, U., & Su, S. (2022). Toward practical monocular indoor depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3814–3824).

Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., & Luo, Z. (2018). Monocular relative depth perception with web stereo data supervision. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 311–320).

Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., & Cao, Z. (2020). Structure-guided ranking loss for single image depth prediction. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 611–620).

Xu, G., Yin, W., Chen, H., Cheng, K., Zhao, F., & Shen, C. (2022). Boosting monocular depth estimation with sparse guided points. arXiv preprint arXiv:2202.01470

Xu, G., Yin, W., Chen, H., Shen, C., Cheng, K., & Zhao, F. (2023). Pose-free 3d scene reconstruction with frozen depth models. In *IEEE international conference on computer vision (ICCV)*.

Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., & Shen, C. (2023). Metric3D: Towards zero-shot metric 3d prediction from a single image. In *IEEE international conference on computer vision (ICCV)*.

Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., & Shen, C. (2021). Learning to recover 3d scene shape from a single image. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 204–213).

Yin, W., Liu, Y., & Shen, C. (2021). Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 44*(10), 7282–7295.

Yuan, W., Gu, X., Dai, Z., Zhu, S., & Tan, P. (2022). Neural window fully-connected CRFs for monocular depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3916–3925).

Zhan, H., Garg, R., Weerasekera, C. S., Li, K., Agarwal, H., & Reid, I. (2018). Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 340–349).

Zhang, Z., Lathuiliere, S., Ricci, E., Sebe, N., Yan, Y., & Yang, J. (2020). Online depth learning against forgetting in monocular videos. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4494–4503).

Zhao, S., Fu, H., Gong, M., & Tao, D. (2019). Geometry-aware symmetric domain adaptation for monocular depth estimation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9788–9798).

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2881–2890).

Zhao, C., Zhang, Y., Poggi, M., Tosi, F., Guo, X., Zhu, Z., Huang, G., Tang, Y., & Mattoccia, S. (2022). MonoViT: Self-supervised monocular depth estimation with a vision transformer. In *2022 international conference on 3D vision (3DV)* (pp. 668–678). IEEE

Zhao, C., Sun, Q., Zhang, C., Tang, Y., & Qian, F. (2020). Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences, 63*(9), 1612–1627.

Zhao, C., Tang, Y., & Sun, Q. (2022). Unsupervised monocular depth estimation in highly complex environments. *IEEE Transactions on Emerging Topics in Computational Intelligence, 6*(5), 1237–1246.

Zheng, C., Cham, T. J., & Cai, J. (2018). T2Net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *European conference on computer vision (ECCV)* (pp. 767–783).

Zhou, Z., & Dong, Q. (2022). Self-distilled feature aggregation for self-supervised monocular depth estimation. In *European conference on computer vision (ECCV)* (pp. 709–726).

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017) Unsupervised learning of depth and ego-motion from video. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1851–1858).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.