Mixing Datasets for Self-Supervised Monocular Depth Estimation

Jinsoo Kim
POSTECH
fusion4268@postech.ac.kr

Gaetan Herry
POSTECH
gaetan.herry@postech.ac.kr

Abstract

Self-supervised learning has emerged as an efficient method for training models to perform monocular depth estimation, without requiring per-pixel ground-truth depth data. On the KITTI dataset, Godard et al. *[5] has obtained state-of-the-art results, however the model does not cope well with other environments. With this paper, we show that their architecture is powerful, and can be used for predicting depth on wider environments if we train it with other datasets. Especially, we implement a dataset mixing strategy for multi-task learning in order to get more robust results on all environments. To determine which dataset choices yields better results, we run an ablation study.*

1. Introduction

We seek to automatically produce an accurate dense depth image from a single color input image of diverse environments. The ability to generate high-quality depth-from-color is appealing, it might complement LIDAR sensors used in self-driving cars while also enabling new single-photo applications like picture editing and AR compositing. Solving for depth is also a great approach to pretrain deep networks for downstream discriminative tasks using big unlabeled image datasets. Collecting big and diverse training datasets with correct ground truth depth for supervised learning, is itself a difficult challenge. As an alternative, several recent self-supervised methods approaches have shown great results on monocular depth estimation using only monocular video [16].

Monodepth2 from the paper of Godard *et al*. [5] has obtained state-of-the-art results on the KITTI dataset [9], using only self-supervised monocular video and with a simple model. However, KITTI is only containing sequences of street images, and as you can see in Fig. 1, monodepth2 performs poorly for indoor images. Our work aims at improving monocular depth estimation for a bigger variety of environments. We chose to use as a baseline the monodepth2 model, and to improve it by implementing dataset mixing. This method aims at training on a mix of datasets in order to obtain more robust results using multi-task learning. It has been proved to be efficient on the recent work of Ranfl *et al*. [10].

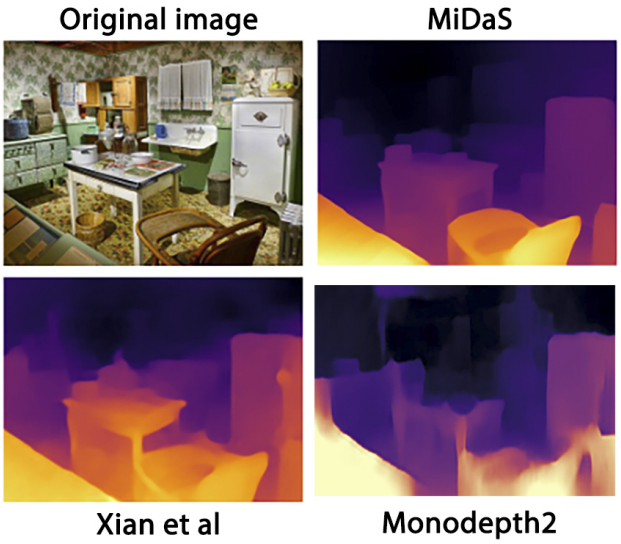

Figure 1. Monodepth2 [5] is obtaining poor results on an indoor image compared to recent models : MiDaS [10] and Xian *et al*. [15]

2. Related work

2.1. Monocular depth estimation

Learning based methods have shown significant advances in monocular depth estimation. They used MRF-based formulations [12] or simple geometric assumptions [8] to estimate the depth using self-supervised learning. However, fully supervised methods need large datasets with ground truth depth during training.

Godard *et al*. [5] introduced self-supervised depth estimation, which trains depth estimation models using image reconstruction as the supervisory signal. They proposed the depth network, which is based on the U-Net architecture [11]. The encoder is formed from a ResNet18 [7] and weights pretrained on ImageNet were used as a baseline. The model is trained and tested on KITTI dataset [4], and

it showed state-of-the-art result depth prediction on monocular depth estimation. However, they only showed performance on the KITTI dataset, and the result on Make 3D was relatively poor.

2.2. Mixing datasets

Multi-task learning sometimes shows better performance than learning on one task [2]. Ranfl *et al*. [10] applied multi-task learning on monocular depth estimation to provide robust performance on testing diverse datasets. Learning on each dataset is considered as a separate task on multi task learning. They explored two different mixing strategies in their experiment. First, they mixed datasets in equal parts in each minibatch. Second, according to the loss over datasets, they sought an approximate Pareto optimum [13]. It showed remarkable performance on estimating the depth of "image in the wild."

3. Method

3.1. Dataset choices

In order to train the Monodepth2 model to perform better on various datasets, it is crucial to find the right datasets to train it with. Monodepth2 is built to receive sequences of images as input, however we do not need them to have ground truth data. The only thing we need besides the sequences are the camera intrinsic matrices composed of the principal point coordinates and the focal length values, and a validation set.

First, we found TUM-RGB-D [14], a novel benchmark for the evaluation of visual odometry and visual SLAM systems. We took all the sequences from Freiburg 3, that are undistorted. These are various indoor images with various lengths, and their size is 640x480. Together with it, we used the ICL-NUIM dataset [6], similar to TUM with the same image size and undistorted. Two different scenes are provided, a living room and an office room scene. The third dataset we used for training was the EuRoC MAV Dataset [1], it contains several sequences of 752x480 images, from a machine hall and a vicon room.

All these three datasets differ a lot from the outdoor data from KITTI, so it may be very useful for training, and at the same time very challenging, so we would potentially have low testing results at first. For the datasets without a validation set provided, we attributed to that purpose some sequences at the ratio 80% training for 20% validation. It was also an important factor that some of these datasets contain ground truth so they can be used for testing.

3.2. Implementation

Monodepth2 uses the data split of Eigen *et al*. [3] on KITTI. For training, they follow Zhou *et al*. [16] preprocessing to remove static frames. This results in a random mix of the sequence frames as the input, and during the forward pass, we treat triplets of adjacent frames in the sequence : $[I_{-1}, I, I_{+1}]$. In order to retrieve adjacent frames of an input, the KITTI data is named by its sequence order. We modified our datasets to follow these requirements, generating new index files to access them. To make each dataset's dataloader, we adapted the KITTI one, changing the camera intrinsic matrices and input sizes.

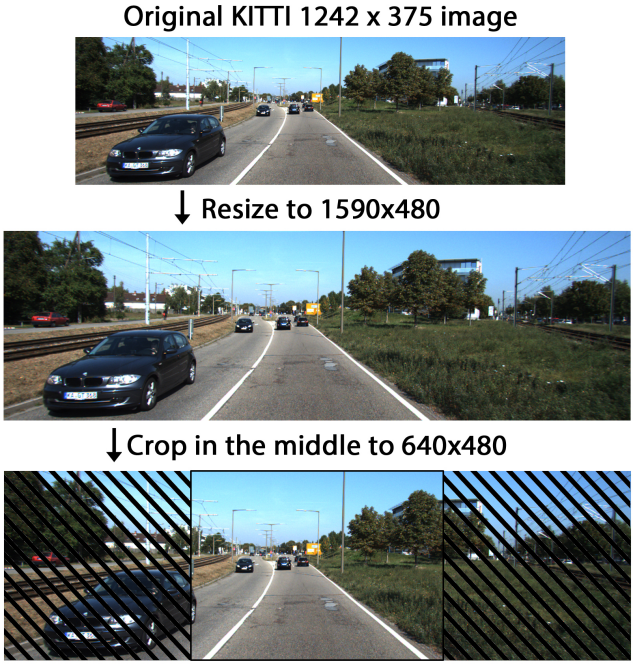

Figure 2. We chose to resize and crop in the middle for the KITTI images. Cropping else may confuse the model.

For naive mixing, as described in [10], the strategy is to mix datasets in equal parts in each minibatch. We sample B/L training samples from each dataset for a minibatch of size B, where L specifies the number of datasets. This technique assures that all datasets, regardless of size, are equally represented in the effective training set. To that purpose, we concatenated the dataloaders, and made a random sampler to take the same number of training samples from each dataset for each minibatch.

Since every dataset presents different sizes, input size should be considered. An input size of 640x480 seemed an obvious choice since already two of our datasets shared this size, and it is also a close size to the other datasets. As explained in [10], keeping the aspect ratio of the images is crucial. For the EuRoC MAV data, we decided to crop symmetrically 56 pixels on each side, and for KITTI we chose to first resize and then crop symmetrically. There are several options to adapt KITTI size which is pretty large, to a 640x480 ratio, such a reflection padding for instance. Resizing seemed better, but for further use, it would be inter-

	KITTI	TUM	ICL	EuRoC
SINGLE1	O			
SINGLE2		O		
MIX1	O	O	O	
MIX2	O	O	O	O

Table 1. Combinations of datasets used for training

esting to try other methods. In the end, the smallest dataset will impose the length of all datasets, in order to have an equal amount of each used for training. This restricts quite a lot the model performances, when using dataset mixing with image sequences.

Our models are implemented in PyTorch the same way they are in the original paper [5], except for some differences. We trained for 20 epochs using Adam, with a learning rate of 10^{-4} for the first 15 epochs which is then dropped to 10^{-5} for the remainder. The smoothness term λ is set to 0.001. Due to memory limitations, we use a batch size of 6 for TUM and 4 for naive mixing, since the batch size needs to be a multiple of the number of datasets. The input/output sizes are always 640x480. Training takes 11, and 15 hours on a single Titan Xp, for TUM, and naive mixing.

4. Experiments

We trained four models to analyze the performance of mixing datasets Table. 1. Model SINGLE1 and SINGLE2 are only trained with KITTI and TUM. For naive mixing, we implemented two models, MIX1 and MIX2. MIX1 is a combination of KITTI, TUM, and ICL. For MIX2, we added EuRoC on MIX1 to see the difference in adding more datasets.

We used the evaluation metric that was used in Monodepth2 [5]. For each KITTI, TUM, Sintel, and NYU, about 600 images are used for testing. To exploit the effect of dataset mixing, we analyze four models on each test dataset. Quantitative results of our models are shown in Table. 2.

4.1. Quantitative results

KITTI
SINGLE1, which is trained on KITTI, has the best score among the four models. As we expected, SINGLE 2, which is only trained on TUM, has the worst score among the four models. MIX1 and MIX2 show better performance than SINGLE1 but worse than SINGLE2. From this result, we can find that the Monodepth2 is well-designed for the KITTI dataset.

TUM
Similar results with testing on KITTI are seen in testing on TUM. SINGLE2 shows the best performance among all models. However, MIX1 and MIX2 show similar performance with SINGLE1 even if they used the TUM model while training. It means dataset mixing in an inept way can worsen the performance of the model.

Sintel & NYU
Sintel and NYU are unseen datasets while training. Our goal was to increase the performance of the model by mixing datasets compared to a model trained with a single dataset. However, in both test datasets, SINGLE1 and SINGLE2 show similar or superior results to MIX1 and MIX2.

In almost all test datasets, MIX1 shows slightly better performance than MIX2. It means that EuRoCMAV is not appropriate for mixing datasets in monocular depth estimation. Furthermore, MIX1 and MIX2 do not outperform SINGLE1 and SINGLE2.

The KITTI dataset, originally used to train Monodepth2, is generated on outdoor car driving. However, we trained on other datasets, TUM, ICL-NUIM, and EuRoC MAV, which are indoor datasets. Indoor datasets have fewer feature points compared to outdoor datasets. Self-supervised training on indoor data is more challenging and needs extra processing to handle. Mixing datasets could be beneficial in training with outdoor datasets such as KITTI.

4.2. Qualitative results

In Fig. 3, we estimate the depth from not only our test dataset but also the image "in the wild" to check the performance of our model on an unseen outdoor dataset. For indoor images, SINGLE2 shows the best performance. SINGLE2 only distinguishes the object in image 1 and 2. On KITTI images, SINGLE1 and MIX show almost similar accuracy. In contrast, SINGLE2 has many blurry parts compared to others. Finally, MIX1 shows the best performance on unseen outdoor data, image 5 and 6. SINGLE1 usually recognizes the depth of the sky, similar to the structure on the side. However, MIX1 has a better borderline on the sky and the structure. Given the qualitative results, we believe that Monodepth2 can be improved for outdoor images if using a wiser dataset mixing. We chose indoor datasets to mix with KITTI, which weakens the performance of our model.

5. Conclusion

In this work, we presented mixing datasets to propose a robust monocular depth estimation model. The key idea is to train with various datasets to have decent performance on unseen data. However, the quantitative result was quite different from what we expected. MIX1 and MIX2 have similar performance on unseen data with SINGLE1 and SINGLE2.

From the qualitative result, we conclude that We used too many different datasets for training and testing. In training,

Model	Test	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
SINGLE1	KITTI	**0.115**	**0.902**	**4.86**	**0.193**	**0.877**	**0.959**	**0.981**
SINGLE2	KITTI	0.346	3.286	9.972	0.462	0.412	0.702	0.854
MIX1	KITTI	0.159	1.139	5.803	0.241	0.77	0.927	0.972
MIX2	KITTI	0.164	1.235	5.879	0.244	0.761	0.922	0.97
SINGLE1	TUM	0.509	0.702	1.117	0.562	0.304	0.542	0.773
SINGLE2	TUM	**0.412**	**0.545**	**0.904**	**0.448**	**0.484**	**0.72**	**0.816**
MIX1	TUM	0.52	0.993	1.185	0.571	0.376	0.622	0.772
MIX2	TUM	0.53	1.063	1.14	0.541	0.378	0.637	0.779
SINGLE1	Sintel	**0.462**	**1.147**	**1.537**	**0.494**	0.47	**0.715**	**0.843**
SINGLE2	Sintel	0.513	1.401	1.631	0.518	0.442	0.688	0.828
MIX1	Sintel	0.477	1.186	1.604	0.51	0.452	0.684	0.828
MIX2	Sintel	0.511	1.416	1.608	0.511	**0.488**	0.689	0.825
SINGLE1	NYU	0.345	0.656	1.157	0.374	0.508	0.778	0.904
SINGLE2	NYU	**0.279**	**0.288**	**0.762**	**0.316**	**0.541**	**0.846**	**0.949**
MIX1	NYU	0.347	0.62	1.216	0.382	0.474	0.774	0.912
MIX2	NYU	0.35	0.664	1.152	0.381	0.485	0.78	0.913

Table 2. **Quantatitative results.** Best results in each test dataset are in **bold**; second best are underlined.

all datasets except KITTI are indoor datasets. In testing, NYU is an indoor dataset, and Sintel is generated from the animation movie. We expect that we could have a better result if we only used one type of scene while training.

Furthermore, different kinds of mixing strategies or mode designs can be considered. A mixing strategy such as seeking Pareto-optimum can perform better than naive mixing. Our loss function is based on a disparity map which requires a sequence of data. If the loss function is designed to handle both outdoor and indoor datasets, a more powerful model can be generated.

References

[1] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[2] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[3] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. 2015.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[5] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. 2019.

[6] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. Hong Kong, China, May 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005.

[9] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. pages 3061–3070, 2015.

[10] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[12] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[13] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *arXiv preprint arXiv:1810.04650*, 2018.

[14] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[15] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018.

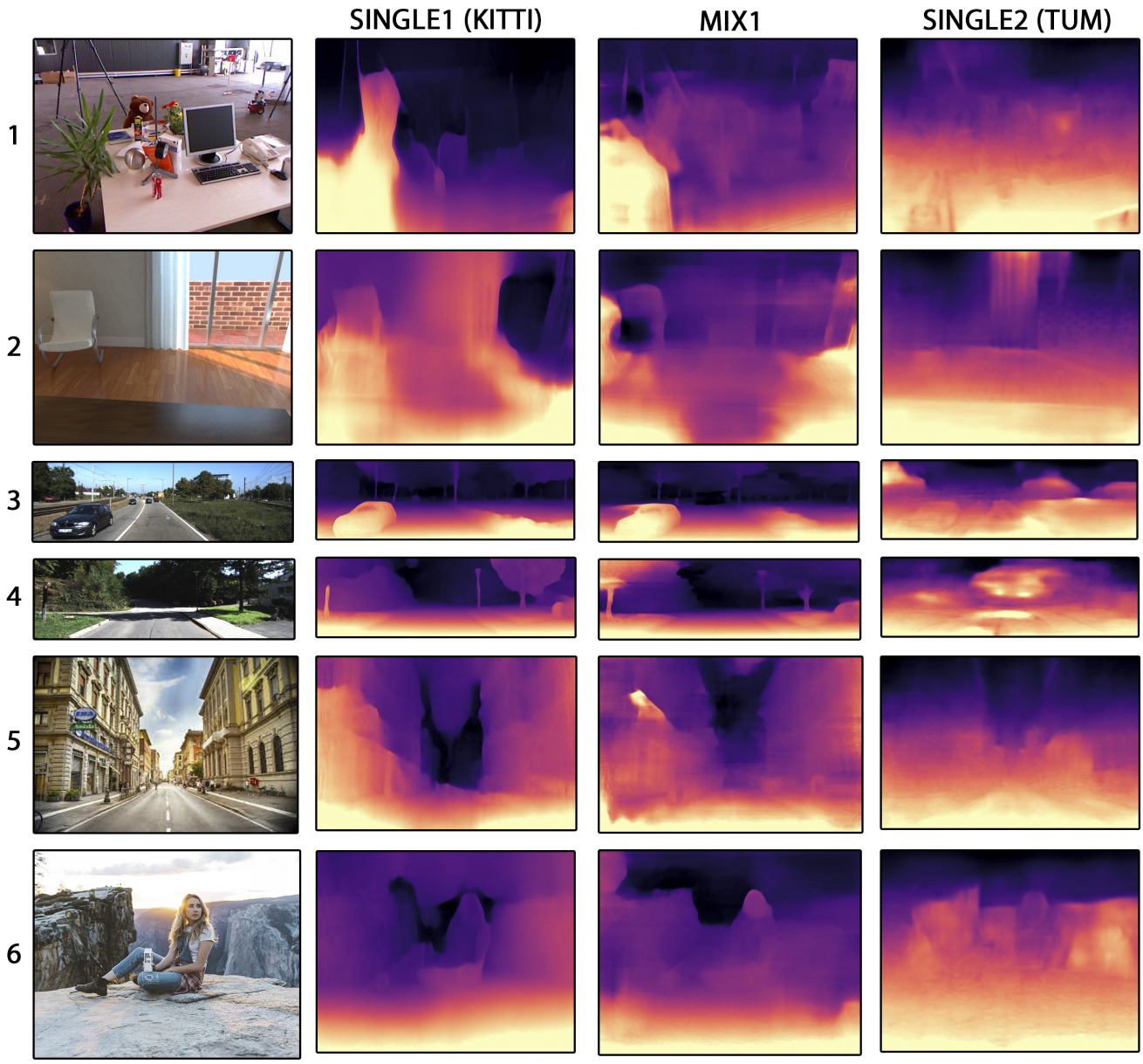

Figure 3. **Qualitative result**. Our model (MIX1) shows better performance on unseen outdoor images. (top to bottom) TUM, ICL-NUIM, KITTI, KITTI, "in the wild", "in the wild"

[16] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. 2017.