Supplementary Materials of SUW-Learn: Joint Supervised, Unsupervised, Weakly Supervised Deep Learning for Monocular Depth Estimation

Haoyu Ren*, Aman Raj**, Mostafa El-Khamy* and Jungwon Lee*

*SOC R&D, Samsung Semiconductor, Inc., California, USA
**University of San Diego, California, USA
{haoyu.ren, mostafa.e, jungwon2.lee}@samsung.com amraj@ucsd.edu

1. Network architecture

In this section, we give the implementation details of the encoder-decoder network and the hourglass network discussed in Section 3.1 of our paper.

1.1. Encoder-decoder network

As shown in Figure 1, our encoder-decoder network consists of two parts, the encoding part, and the decoding part. The encoding part contains several depth-wise separable convolutional layers [2] to extract the discriminative features from the input image. Each depth-wise separable convolutional layer consists of a depth-wise convolutional layer, and a point-wise convolutional layer. The output feature map of the encoding module is $\times 4$ downsampled compared to the input image. The decoding part consists of three decoding blocks. The first and second decoding block consists of two plain convolutional layers. The third decoding block is implemented by stacked atrous multi-scale (SAM) module proposed in [3]. Inspired by the U-Net architecture [6], we concatenate the feature maps from the encoding block to the decoding blocks. The exact design of the encoding and the decoding blocks can be found in Table 1.

During the training, we uniformly quantize the input continuous depth x with depth-range $[\alpha, \beta]$ into B bins in the log scale. As shown in Eq. 1, the input continuous depth x is quantized to discrete value b, B is the number of bins, and q is the width of each bin.

$$\begin{aligned} b &= round(\log_{10}(x) - \log_{10}(\alpha))/q) \\ q &= (\log_{10}(\beta) - \log_{10}(\alpha))/B. \end{aligned} \tag{1}$$

The soft classification loss $\mathcal{L}_{cls}$ mentioned in our paper at Section 3.1 is calculated based on this quantized depth. We use Adam optimizer with initial learning rate = 0.0001 to train our encoder-decoder network. The learning rate is set to 0.00001 after 10 epochs.

1.2. Hourglass network

Our hourglass network follows a similar architecture as [1]. It consists of a series of convolutions based on inception [7] module and downsampling, followed by a series of convolutions and upsampling, interleaved with skip connections that add back features from high resolutions. The symmetric shape of the network resembles a 'hourglass'. In Figure 2, we visualize the network design of our hourglass network. Only the Block H is a convolution with 3×3 filters, while all other blocks are inception blocks, as given in Figure 3. Each inception block consists of four branches with different filter size. The outputs of these four branches are concatenated together. The exact design of these inception blocks are detailed in Table 2.

We use use Adam optimizer with initial learning rate = 0.0005 to train our hourglass network. The learning rate decays 10 times every 5 epochs.

2. Additional implementation details

In this section, we give more implementation details of training our network with SUW-Learn on M&M dataset, as well as on KITTI dataset.

2.1. KITTI datset

In Section 4.1 of our paper, we mentioned that the training of SUW-Learn framework on KITTI dataset utilized the object-motion based unsupervised learning and patch-based weakly supervised learning. Here we give more details of these two modules.

2.1.1 Unsupervised learning with object motion

KITTI dataset doesn't provide the extrinsic matrices for all frames, so we use a pose estimation network to calculate the camera motion and the object motion. When estimating the camera motion, the whole input frames I_{t-1}, I_t, I_{t+1}

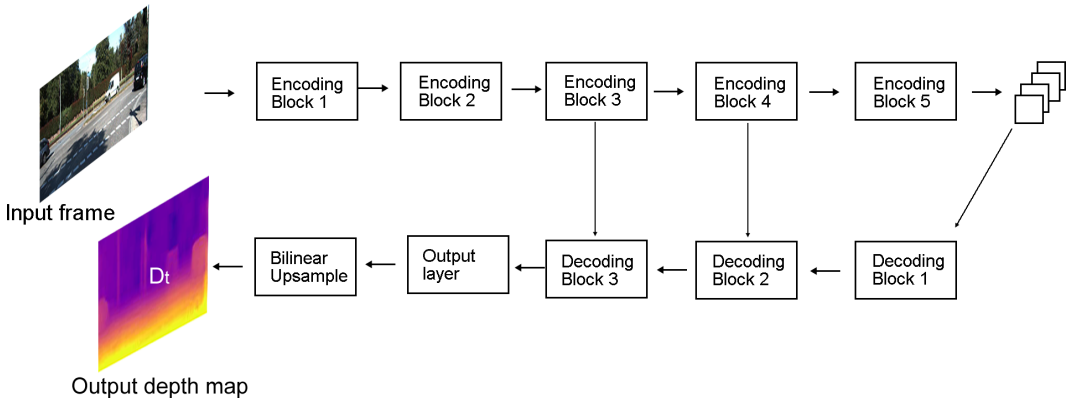

Figure 1. Network architecture of the encoder-decoder network.

Table 1. Network architecture of the encoder-decoder network.'s' means stride, 'd' means dilation rate, 'dw' means depth-wise separable convolutions. 'convbn' layer includes a convolutional layer, followed by a batch normalization layer. 'SAM' stands for 'stacked atrous multi-scale' module [3].

	Layers	Output feature map size
Input		$3 \times W \times H$
Encoding block 1	$convbn, 3 \times 3, 32, s2, d1$ $convbn, 3 \times 3, 64, s1, d1$	$64 \times \frac{1}{2}W \times \frac{1}{2}H$
Encoding block 2	$[convbn, 3 \times 3, 64, s1, d1, dw] \times 3$ $convbn, 3 \times 3, 128, s2, d1, dw$	$128 \times \frac{1}{4}W \times \frac{1}{4}H$
Encoding block 3	$[convbn, 3 \times 3, 128, s1, d1, dw] \times 3$ $convbn, 3 \times 3, 256, s1, d1, dw$	$256 \times \frac{1}{4}W \times \frac{1}{4}H$
Encoding block 4	$[convbn, 3 \times 3, 32, s1, d2, dw] \times 12$ $convbn, 3 \times 3, 256, s1, d1, dw$	$256 \times \frac{1}{4}W \times \frac{1}{4}H$
Encoding block 5	$convbn, 3 \times 3, 256, s1, d1, dw$ $convbn, 1 \times 1, 128, s1, d1$	$128 \times \frac{1}{4}W \times \frac{1}{4}H$
Decoding block 1	$convbn, 3 \times 3, 32, s1, d1$ $convbn, 3 \times 3, 32, s1, d1$	$32 \times \frac{1}{4}W \times \frac{1}{4}H$
Decoding block 2	$convbn, 3 \times 3, 32, s1, d1$ $convbn, 3 \times 3, 32, s1, d1$	$32 \times \frac{1}{4}W \times \frac{1}{4}H$
Decoding block 3	SAM	$32 \times \frac{1}{4}W \times \frac{1}{4}H$
Output layer	$convbn, 3 \times 3, B, s1, d1$ $softmax, weighted\ sum$	$1 \times \frac{1}{4}W \times \frac{1}{4}H$

are utilized as input. When estimating the object motion, the object-masked input frames $I_{t-1,m}, I_{t,m}, I_{t+1,m}$ are utilized as input. $I_{t,m}$ is the RGB image of frame t masked by aligned mth object. To obtain such object mask, we use Mask-RCNN to generate the instance segmentation map of KITTI images, and only consider persons, bicycles, and cars since these objects will have different motion vectors as the camera. In each of the 3-frame video clip used for KITTI training, we align these objects among these 3 frames based on IoU (Intersection over Union). An object is considered as aligned if

- There are three object masks (generated by Mask-RCNN) with same object category appearing in frame $t-1, t, t+1$
- The IoU between the object mask of frame $t-1$ and frame t is higher than 0.3
- The IoU between the object mask of frame t and frame $t+1$ is higher than 0.3

This mask generation and alignment procedure is applied offline, so that it will not increase the training time.

When implementing the pose estimation network, we also use an encoder-decoder network. The encoding part shares the same weights as the depth estimation network[1] given in Table 1. The decoding part consists of four $32 \times 3 \times 3$ plain convolutional layers, and an additional output layer to make the output size of the pose estimation network to 6 dimensions (3 angles for rotation and 3 dis-

[1]The input channel of the first layer of the encoding part is different since the pose estimation requires multiple time stamps as input.

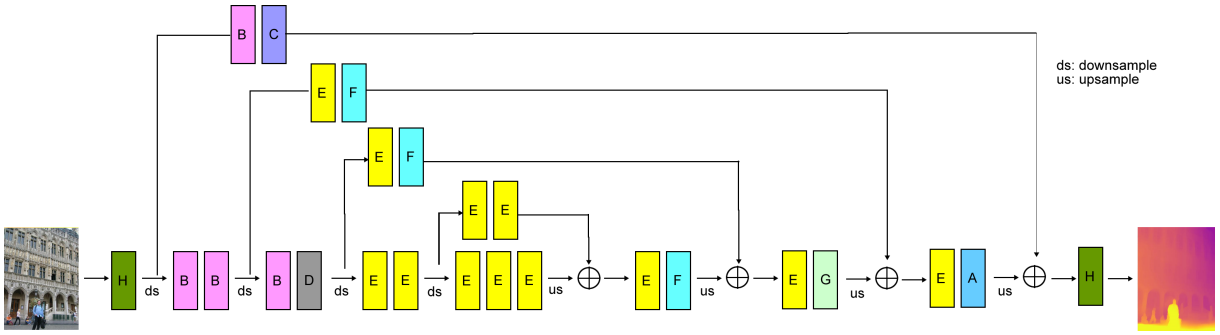

Figure 2. Network design of our hourglass network. The $\oplus$ sign denotes the element-wise addition. Blocks sharing the same color are identical. The exact design of the blocks A to H can be found in Table 2.

Block Id	A	B	C	D	E	F	G
#In/#Out	128/64	128/128	128/128	128/256	256/256	256/256	256/128
Inter Dim	64	32	64	32	32	64	32
Conv1	1x1	1x1	1x1	1x1	1x1	1x1	1x1
Conv2	3x3	3x3	3x3	3x3	3x3	3x3	3x3
Conv3	7x7	5x5	7x7	5x5	5x5	7x7	5x5
Conv4	11x11	7x7	11x11	7x7	7x7	11x11	7x7

Table 2. Implementation of the inception blocks in our hourglass network. Conv1 to Conv4 correspond to the Conv1 to Conv4 in Figure 3. Conv2 to Conv4 share the same number of input and is specified in *Inter Dim*.

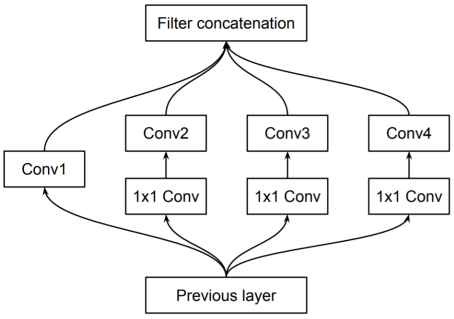

Figure 3. Inception blocks used in our hourglass network as shown in Figure 2.

tances for translation).

2.1.2 Generate the weak depth label from semantic mask

To generate the weakly depth label for KITTI dataset, we use PSPNet [8] to obtain the semantic mask of KITTI training images. We adopt the following heuristic rules to extract the weakly-labeled patch-pairs from semantic knowledge, including

- The sky is farther than any other objects (see patch pair A in Figure 4)

- The tree/building laying at the top-left/top-right of the image is farther than the road/fence/grass laying at the bottom-left/bottom-right of this tree/building (see patch pair B1 and B2 in Figure 4)

Figure 4. Using semantic knowledge to generate weakly labeled depth.

- The car/person/bicycle/traffic sign is farther than the road laying at the bottom of this car/person/bicycle/traffic sign (see pixel pair C in Figure 4)[2]

The above rules can be utilized in generating both the pixel-based label, and our proposed patch based label when the patch pair $\{x, y\}$ comes from different semantic classes. During the training, we randomly generate one pixel or patch-pair from the pre-generated semantic mask for each of the image in each iteration. The patch size $W \times H$ ranges from 3×3 to 9×9.

2.2. M&M dataset

When training our hourglass network on this dataset, we rescale all images to fix resolution of 320×240, and correspondingly change its available ground truth (SfM depth) and camera parameters (intrinsic or extrinsic). Since Mannequin Challenge (MC) dataset doesn't provide ground

[2]The second and third rules are not correct in all the scenario. But these rules are correct for the driving scenes in KITTI.

truth depth maps, we use the COLMAP algorithm to generate SfM+MVS depth for MC images following all the depth refinement steps as outlined in [4].

MegaDepth (MD) dataset only provides single frame images, so it is not possible to extract the camera and object motion for unsupervised learning. Most of the Mannequin Challenge (MC) images are indoor scenarios, which lacks of the consistency ordinal depth relationship from the semantic knowledge. When training with our SUW-Learn framework on M&M dataset, we only calculate the unsupervised loss for MC images, and the weakly supervised loss for MD images. Since all the intrinsic and extrinsic matrices of MC images are given, we may calculate the camera relative pose from these parameters of adjacent frames directly to calculate the unsupervised loss $\mathcal{L}_u$ when training the SUW. In MC images, all the objects are frozen, so that the object motion is consistent with camera motion. Thus our unsupervised loss function is only calculated based on the ego-motion as $\mathcal{L}_u = \mathcal{L}_{u,ego}$. We follow a same way as the MD paper [5] to calculate our weakly supervised loss $\mathcal{L}_w = \mathcal{L}_{w,pix}$ based on the pixel labels, since the MD dataset already provides the the official foreground/background maps to generate these pixel-based weak labels.

3. Additional experimental results

3.1. Ablation study on generating the weak depth label

In this section, we give more analysis of our proposed weakly supervised learning based on patch labels. Our proposed patch-based label can be extracted based on two stragies, from the same semantic class, or different semantic classes. In Table 3, we train our encoder-decoder networks with supervised learning and weakly supervised learning (SW) by using different ways to generate the weak labels. It can be seen that if we generate the patch-based label from one of the above two strategies (row 4, row 5), the accuracy is still better than using supervised learning only (row 2), as well as the S+W with pixel-based weak label (row 3). If we generate the patch-based label using two strategies together, the accuracy is further improved, as given in row 6.

Another way to generate the weak depth label is randomly sampling from the ground-truth depth (this is not weakly supervised learning anymore). Since KITTI's ground-truth is very sparse, it is hard to extract two patches with exact same number of labeled pixels. So we only generate the pixel-based labels with the closer/farther relationship measured by the ground-truth absolute depth. It can be seen that the accuracy (row 7) is still lower than our patch-based label generated from semantic knowledge (row 6). The reason is that randomly sampling does not capture any semantic information. The generated weak label will be less discriminative. The above results validate the effectiveness of our weak label generation method based on semantic knowledge.

3.2. Visualizations of our depth outputs

We give more visualizations of our estimated depth from SUW-KITTI model in the supplementary video[3], which demonstrates that the more learning strategies we use, the better depth estimation accuracy we may get. Using all the supervised, unsupervised, and weakly supervised learning together (SUW) can achieve the best accuracy.

In Figure 5, we give more results of our SUW-model trained on M&M datasets when evaluating on wild images downloaded from google. It can be seen that our network also generalizes well on real-world scenarios. This indicates that training on M&M dataset is an effective way to obtain a depth estimation network with good generalization ability.

References

[1] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NIPS*, pages 730–738, 2016.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017.

[3] Xianzhi Du, Mostafa El-Khamy, and Jungwon Lee. Amnet: Deep atrous multiscale stereo disparity estimation networks. *arXiv preprint arXiv:1904.09099*, 2019.

[4] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, pages 4521–4530, 2019.

[5] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018.

[6] 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.

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[8] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[3] https://www.youtube.com/watch?v=fWhvc6OClVg

Table 3. Depth estimation accuracy of our encoder-decoder network with different ways to generate the weak labels on KITTI Eigen's split. 'same' means that the two patches in our weak depth label come from same semantic class. The depth cap is 80m. The official ground-truth is utilized for evaluation.

Method	Learning	Weak label	Generation	REL (%)	RMSE (in meter)
Ours	S	-	-	6.61	3.058
Ours	SW	pixel	Diff	5.67	2.802
Ours	SW	patch	Same	5.61	2.699
Ours	SW	patch	Diff	5.49	2.601
Ours	SW	patch	Same+Diff	5.17	2.478
Ours	SW*	pixel	Random from gt	5.51	2.796

Figure 5. Some qualitative results on images from the wild. The sky region is masked out since sky doesn't have any meaningful depth.