MTI-Net: Multi-Scale Task Interaction Networks for Multi-Task Learning Supplementary Materials

Simon Vandenhende[1], Stamatios Georgoulis[2], and Luc Van Gool[1,2]

KU Leuven/ESAT-PSI[1] ETH Zurich/CVL[2]

A Difference with Cross-Stitch Networks

Cross-stitch networks [4] also share features at multiple scales. However, the intended purpose, and implementation differ significantly from MTI-Net. We analyze the most notable points here.

- Cross-stitch networks model task interactions at the encoding stage by softly sharing features between task-specific encoders, before branching out to task-specific decoders without further interaction. Differently, MTI-Net operates at the decoding stage fusing task features close to the output that contain more disentangled task information. The latter is arguably better for distilling task information in structured output tasks with co-occurring patterns.
- Cross-stitch Networks distil task information sequentially layer-by-layer, greedily modeling interactions at a local scale. In contrast, MTI-Net models task interactions in parallel, globally fusing information across all scales (cf. FA module). The benefits are two-fold, i.e. enabling the modeling of 'long-term' relationships, and allowing for the use of sufficient context information which is crucial in dense prediction tasks [1].
- The model size of cross-stitch networks increases linearly with the number of tasks, thus scaling poorly to multiple tasks. Instead, MTI-Net provides a more computationally efficient alternative (see resource analysis), that is much closer to the single-task model size.

B Training setup

We include additional details of the training setup used for each experiment. We considered two different multi-scale backbone networks, i.e. HRNet and FPN. For HRNet, we use bilinear upsampling and concatenation followed by two convolutional layers to decode the multi-scale features in the feature aggregation unit. For FPN, the feature aggregation module decodes the multi-scale features as in panoptic feature pyramid networks [2]. In both cases, the non-linear function that produces the task attention mask in the FPM is implemented as two basic residual blocks – that aggressively reduce the number of channels – followed by a 1×1 convolutional layer.

Table S1: Multi-task learning on PASCAL using a ResNet-18 FPN backbone.

Method	Seg ↑	Parts ↑	Sal ↑	Edge ↑	Norm ↓	Δ_m ↑
Single task	64.49	57.43	66.38	68.20	14.77	+ 0.00
MTL (s)	54.51	55.12	64.76	-	-	- 7.32
MTL (a)	59.61	56.88	64.96	70.60	15.17	- 1.80
Ours (s)	65.47	61.32	66.37	-	-	+ 2.77
Ours (s)(E)	65.93	62.21	66.80	-	-	+ 3.61
Ours (s)(N)	64.99	61.09	66.80	-	-	+ 2.52
Ours (s)(E+N)	65.46	61.71	66.62	-	-	+ 3.06
Ours (a)	65.69	61.59	66.76	73.90	14.55	+ 3.84

B.1 NYUD-v2

We applied the data augmentation strategy of PAD-Net [5]. The RGB and depth images were randomly scaled with the selected ratio in $\{1, 1.2, 1.5\}$, and randomly horizontally flipped. The model was trained for 80 epochs with an Adam optimizer with initial learning rate 1e-4 and batches of size 6. We used a poly learning rate decay scheme.

B.2 PASCAL

We essentially plugged our model into the code base that was shared by [3]. In particular, the single-task models were trained with stochastic gradient descent with momentum 0.9. We used batches of size 8 and a poly learning rate decay scheme. The initial learning rate was 0.01. We applied weight decay $\lambda = 1e-4$. These hyperparameters are the same as the ones used in [3], ensuring fair comparison. The multi-task baseline models were trained using the same hyperparameters. The multi-task loss weighing was taken from [3]. We also tested the use of an Adam optimizer, but this did not yield better results.

Our MTI-Net was trained under the same settings as the single-task models, but we used an Adam optimizer with initial learning rate 1e-4. We re-used the loss weights from before to weight the losses from the initial task predictions.

C Extra experiments on PASCAL

We perform an additional ablation experiment using a ResNet-18 FPN backbone in Table S1. The conclusions are similar to the ones reported for the HRNet-18 in Table 2b of the main paper. This shows that our method can be used in combination with various backbone architectures. Again, our model improves over the single-tasking models, both for the small (+2.77%) and complete (+3.84%) set of tasks. We also consider the effect of adding additional auxiliary tasks when predicting the small task set. When adding edge detection as an auxiliary task, the results are further improved (+2.77% to +3.61%). However, this is not the case when we add surface normals prediction as an auxiliary task (+2.77% to 2.52%). We observe a similar effect when including both edge detection and surface normals prediction (3.61% to 3.06%). We believe that this is due to the approximate nature of the surface normals in this dataset, as the latter were obtained through distillation, and as such they are rather noisy.

Table S2: Ablation studies on NYUD-v2 using an HRNet18-V2 backbone. Auxiliary tasks are indicated in brackets.

(a) Results on the depth estimation task.

Method	rmse ↓	rel ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Single task	0.667	0.186	0.731	0.931	0.981
MTL	0.668	0.193	0.717	0.927	0.980
PAD-Net	0.660	0.189	0.726	0.930	0.981
PAD-Net (N)	0.658	0.187	0.726	0.932	0.982
PAD-Net (N+E)	0.655	0.184	0.731	0.934	0.982
Ours - w/o FPM	0.640	0.181	0.747	0.937	0.982
Ours - w/o FPM (N)	0.642	0.175	0.753	0.940	0.983
Ours - w/o FPM (N+E)	0.637	0.174	0.757	0.939	0.984
Ours - w/ FPM	0.620	0.161	0.781	0.946	0.986
Ours - w/ FPM (N)	0.600	0.162	0.788	0.947	0.985
Ours - w/ FPM (N+E)	0.607	0.166	0.783	0.945	0.985

(b) Results on the semantic segmentation task.

Method	pixel-acc ↑	mean-acc ↑	IoU ↑
Single task	65.04	45.07	33.18
MTL	64.61	43.55	32.09
PAD-Net	65.00	44.61	32.80
PAD-Net (N)	64.77	46.28	33.85
PAD-Net (N+E)	65.05	44.79	32.92
Ours - w/o FPM	65.52	45.98	34.38
Ours - w/o FPM (N)	65.27	46.63	34.49
Ours - w/o FPM (N+E)	66.15	46.97	34.68
Ours - w/ FPM	66.30	47.85	35.12
Ours - w/ FPM (N)	66.98	49.04	36.22
Ours - w/ FPM (N+E)	68.03	51.05	37.49

D Extra experiments on NYUD-v2

This section contains additional results on the NYUD-v2 dataset. Section D.1 gives a more detailed view on the ablation studies that we performed on NYUD-v2 using an HRNet-18 backbone. Note that the main results of this experiment were already discussed in the experiments section of the paper. In Section D.2, we perform an additional experiment using an FPN backbone based on ResNet-18.

D.1 HRNet18-V2

Table S2 contains additional metrics for the depth estimation and semantic segmentation task on the NYUD-v2 dataset, when using an HRNet-18 backbone. This is an extension to the metrics shown in Table 2a of the paper.

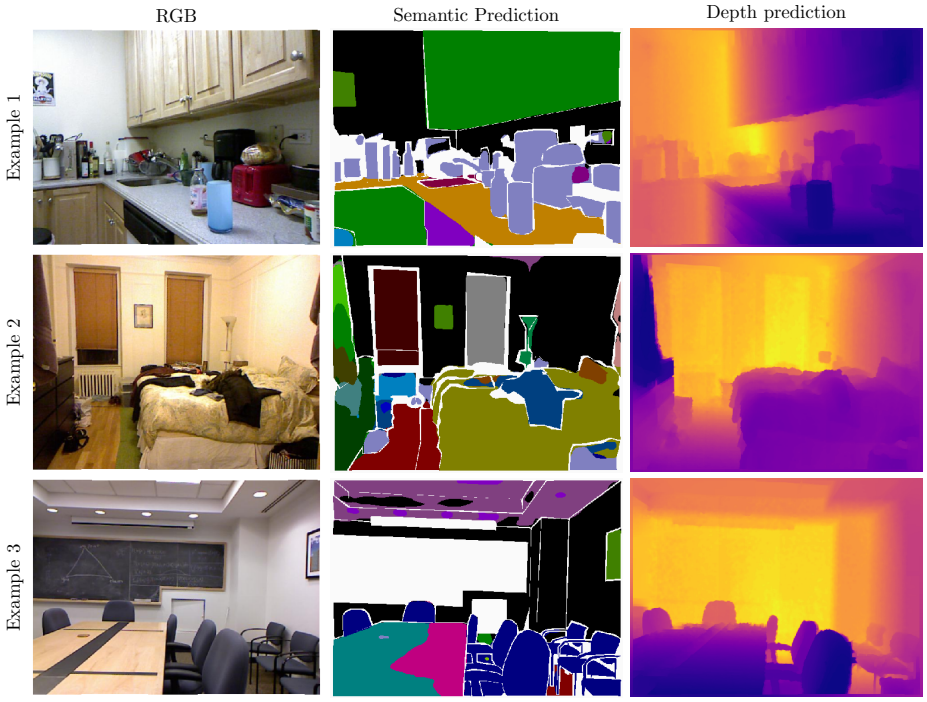

Fig. S1: **Qualitative results on NYUD-v2:** Semantic and depth predictions made by our HRNet-48 model.

D.2 FPN - ResNet-18

We repeated a smaller version of our ablation studies on the NYUD-v2 dataset when using an FPN backbone based on ResNet-18. Table S3 contains the results. We end up at similar findings compared to the model based on HRNet-18. Again, we see a significant improvement over the set of single-task models. Additionally, we find that the use of auxiliary tasks can help to improve the quality of the predictions.

E Qualitative results on NYUD-v2

Figure S1 shows predictions made by our HRNet-48 model on images from the NYUD-v2 test set. The quantitative results were already reported in Table 6 of the paper.

Table S3: Additional results on NYUD-v2 when using an FPN backbone based on ResNet-18. Similarly to Table 2, auxiliary tasks are indicated in brackets.

(a) Multi-task learning performance.

Method	Seg (IoU) ↑	Dep (rmse) ↓	$\Delta_m\%$
Single task	34.46	0.659	+0.00
MTL	33.52	0.665	-1.82
PAD-Net	34.15	0.662	-0.69
PAD-Net (N)	34.18	0.657	-0.23
PAD-Net (N+E)	34.60	0.668	-0.45
Ours	36.01	0.630	+4.43
Ours (N)	36.81	0.628	+5.74
Ours (N+E)	36.65	0.618	+6.27

(b) Results on the depth estimation task.

Method	rmse ↓	rel ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Single task	0.659	0.183	0.730	0.935	0.982
MTL	0.665	0.190	0.726	0.930	0.980
PAD-Net	0.662	0.188	0.731	0.931	0.979
PAD-Net (N)	0.657	0.185	0.735	0.934	0.980
PAD-Net (N+E)	0.668	0.185	0.729	0.933	0.980
Ours	0.630	0.173	0.767	0.939	0.981
Ours (N)	0.628	0.180	0.755	0.939	0.982
Ours (N+E)	0.618	0.169	0.768	0.944	0.984

(c) Results on the semantic segmentation task.

Method	pixel-acc ↑	mean-acc ↑	IoU ↑
Single task	65.51	46.50	34.46
MTL	64.85	45.33	33.52
PAD-Net	65.23	45.65	34.15
PAD-Net (N)	65.07	45.80	34.18
PAD-Net (N+E)	65.68	46.77	34.60
Ours	66.44	49.03	36.01
Ours (N)	66.89	50.50	36.81
Ours (N+E)	67.23	49.93	36.65

References

1. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018)
2. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR. pp. 6399–6408 (2019)
3. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: CVPR. pp. 1851–1860 (2019)
4. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: CVPR (2016)
5. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: CVPR. pp. 675–684 (2018)