MTAN and Cross-Stitch Networks

Mario De Simone

April 2025

Table of Contents

Introduction

Multi-task learning is a framework that allows the training of multiple tasks jointly on one model. In the following two main approaches will be analyzed:

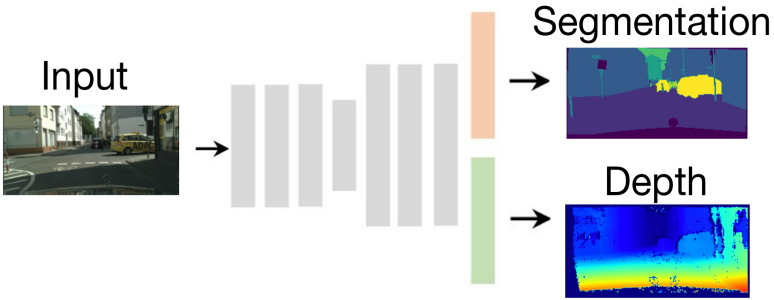
- **Attention Specific Networks (MTAN)** following the idea suggested by [3]
- **Cross-Stitch Networks** following the idea suggested by [4]

Moreover comparison with single-task and other multi-task architectures will be provided.

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019
[4] Misra et al., *Cross-stitch Networks for Multi-task Learning*, 2016

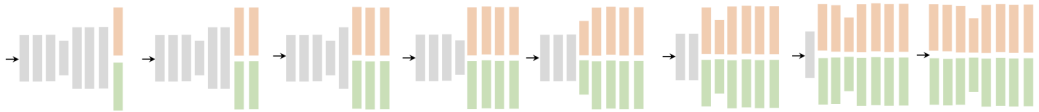
Past Works I

Idea: split the network into parts one for each task



Past Works II

Problem: where to split?



It is not smart to train all the possible splits, find a different way to implementa task specific networks
Note: the split impacts on the balance between task specific features and shared features

[4] Misra et al., *Cross-stitch Networks for Multi-task Learning*, 2016

SegNet

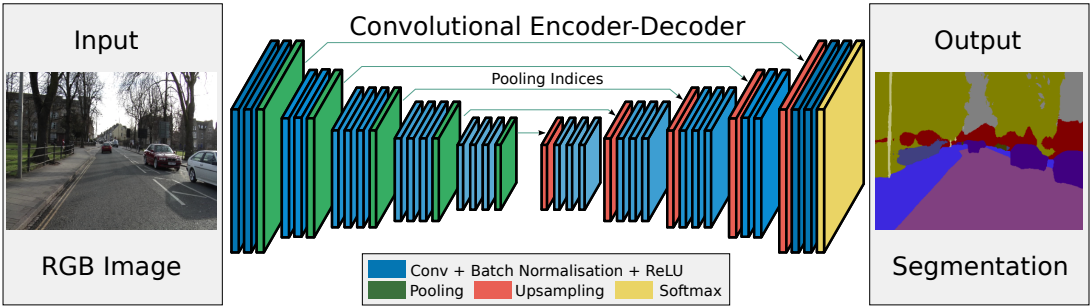


Figure: SegNet Model Architecture

[1] Badrinarayanan, Kendall, and Cipolla, *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*, 2016

SegNet as model for different tasks

Starting from SegNet the change of task is done by changing the last layer of the network.

- **Segmentation**: softmax layer
- **Depth Estimation**: regression layer
- **Surface Normals**: regression (normalized) layer

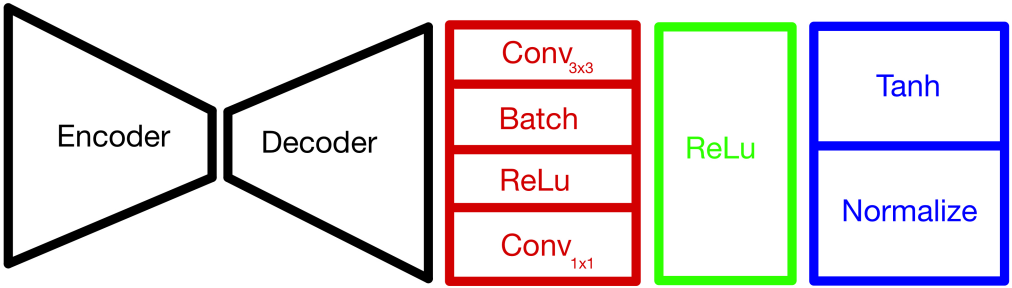


Figure: SegNet for different tasks

MTAN Idea

Idea: use attention modules to choose for each task the most important features from a shared representations

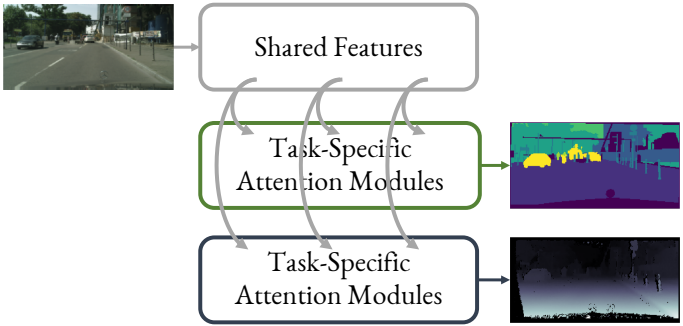


Figure: MTAN idea

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

MTAN Architecture

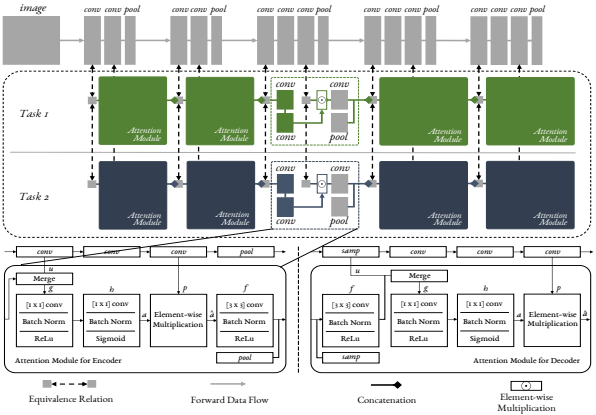


Figure: MTAN Architecture details

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

MTAN Attention Module

The shared features are built with an Encoder-Decoder architecture, shaped like a SegNet.

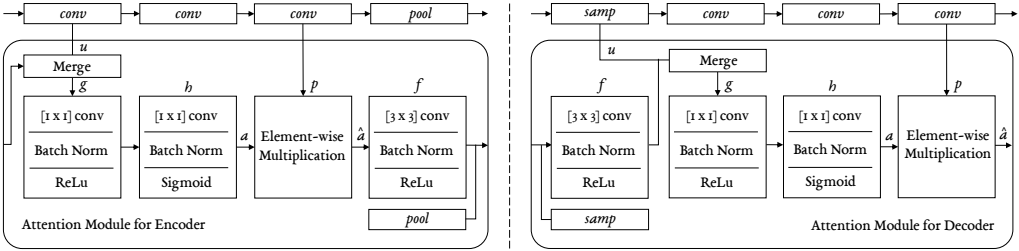


Figure: MTAN Attention Module details

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

Cross-Stitch Architecture

Idea: use a dynamic combination of network deployed for different task

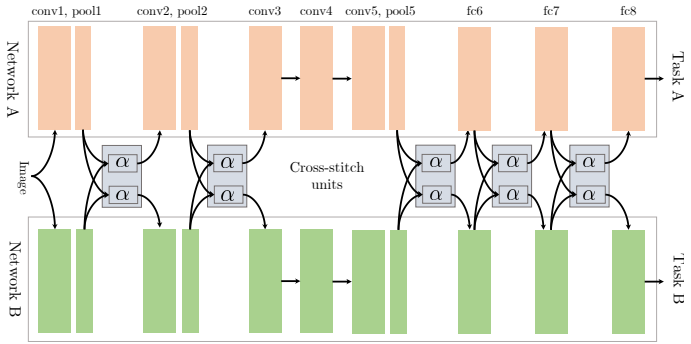


Figure: Example of a network with Cross-Stitch Units

[4] Misra et al., *Cross-stitch Networks for Multi-task Learning*, 2016

Cross-Stitch Units

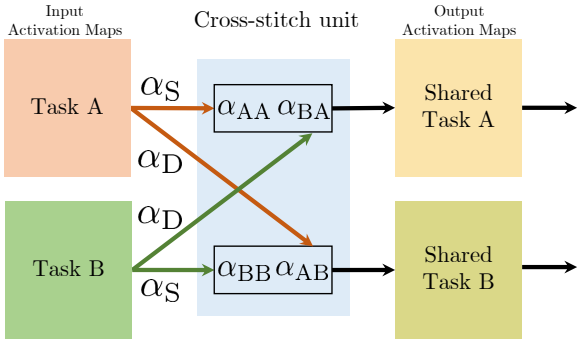


Figure: Cross-Stitch detailed operations

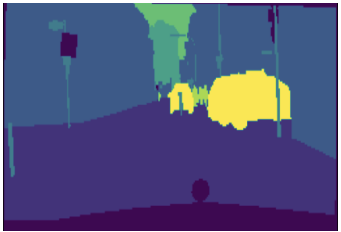
Initialization: [4] suggests $\alpha_i \in [0, 1]$ to preserve magnitude
Note: formulation can be extended to an arbitrary number of tasks

[4] Misra et al., *Cross-stitch Networks for Multi-task Learning*, 2016

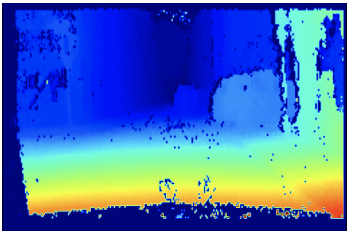
Cityscapes

Outdoor images resized to be [128×256] for segmentation and depth estimation.
Segmentation with 2, 7 or 19 classes (in the experiments were used the 7-classes instances).

(a) Original



(b) Segmentation



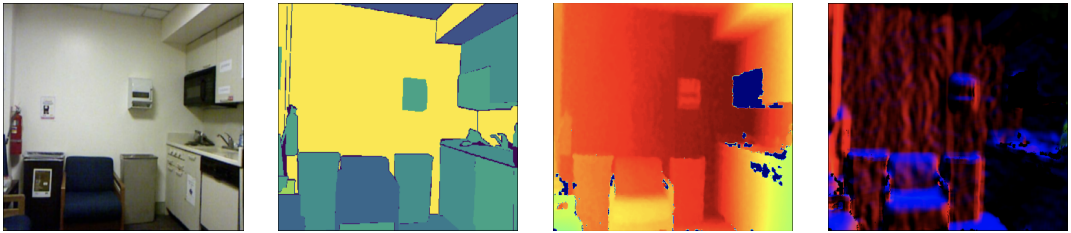
(c) Depth Estimation

Figure: Examples images of Cityscapes Dataset

[2] Cordts et al., *The Cityscapes Dataset for Semantic Urban Scene Understanding*, 2016

NYUv2

Indoor images [288×384] for segmentation, depth and surface normals estimation.
Segmentation with 13 classes



(a) Original (b) Segmentation (c) Depth Estimation (d) Normals Estimation

Figure: Examples images of NYUv2 Dataset

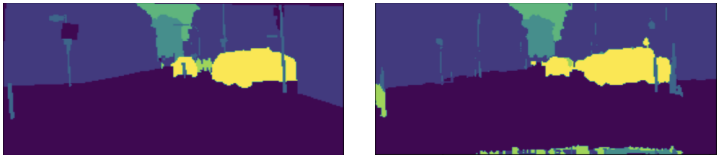
[5] Silberman et al., *Indoor Segmentation and Support Inference from RGBD Images*, 2012

Cityscapes vs NYUv2

NYUv2 is more complex (in fact worse results were achieved) than Cityscapes:

- **Segmentation**: the classes are more specific and their position in the images are more variable
- **Depth Estimation**: while in cityscapes the structure of the images make the depth map of different images similar, the same does not occur onto NYUv2

Cityscapes



NYUv2

Table: Examples of ground truth (left) and segmentation for Cityscapes and NYUv2 with Dense Model (right)

Total Loss

Problem: How to get one loss with respect to different goals of training?
Solution: Divide the loss into components, one for each task:

$$L_{tot} = \sum_{t \in Tasks} \lambda_t L_t(X, Y_t) \tag{1}$$

Problem: how to choose λ_t?
Idea: one way is to calculate them according to past losses (**Dynamic Weight Averaging**)

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

Segmentation Task

Cross Entropy Loss:

$$L(X, Y_s) = -\sum_{p,q} Y_s(p,q) \log \hat{Y}(p,q)$$

- X input data
- Y_s segmentation ground truth
- $\hat{Y}$ segmentation prediction of the network
- p, q spatial position of image $H \times W$

Statistics:

- Pixel Accuracy:

$$acc_{pix}(X, Y_s) = \frac{1}{HW} \sum_{p,q} \mathbb{1}_{Y_s(p,q)=\hat{Y}(p,q)}$$

- Mean Intersection Over Union:

$$miou(X, Y_s) = \frac{1}{K} \sum_{k=1\ldots K} \frac{TP}{TP + FP + FN}$$

- K number of classes
- TP true positives, FP false positives and FN false negatives of class k

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

Depth Estimation Task

L1 Loss:

$$L(X, Y_d) = \sum_{p,q} |Y_d(p,q) - \hat{Y}(p,q)|$$

- X input data
- Y_d depth ground truth
- $\hat{Y}$ depth prediction of the network
- p, q spatial position of image $H \times W$

Statistics:

- Mean Absolute Error:

$$mae(X, Y_d) = \frac{1}{HW} \sum_{p,q} |Y_d(p,q) - \hat{Y}(p,q)|$$

- Mean Absolute Relative Error:

$$mare(X, Y_d) = \frac{1}{HW} \sum_{p,q} \frac{|Y_d(p,q) - \hat{Y}(p,q)|}{Y_d(p,q)}$$

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

Surface Normals Estimation

Dot Product Loss:

$$L(X, Y_n) = -\sum_{p,q} Y_n(p,q) \cdot \hat{Y}(p,q)$$

where:

- X input data
- Y_d normals ground truth
- $\hat{Y}$ normals prediction of the network
- p, q spatial position of image $H \times W$

Statistics:

- Mean Angle Distance:

$$ad(X, Y_n) = \frac{1}{HW} \arccos(X \cdot Y_n)$$

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

Additional Details

- mIoU does not take into account background
- Pixel Accuracy, MAE, MARE and Angle Distance do not take into account pixel with ground truth -1 (pixel without class prediction)
- in addition to mean angle distance, median and the number of elements under some angles tolerances (11.25, 22.5, 30) are computed

Dynamic Weight Averaging

Idea: for each task t compute λ_t taking into account the previous loss value of that task L_t

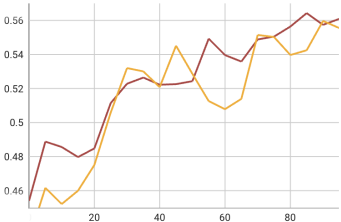
$$\lambda_t(k) = \frac{K \exp(w_t(k-1)/T)}{\sum_{i \in Tasks} \exp(w_i(k-1)/T)} \qquad w_t(k-1) = \frac{L_t(k-1)}{L_t(k-2)} \tag{2}$$

some notes:

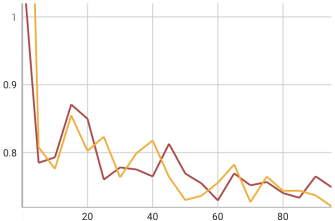
- $K = |Tasks|$
- T temperature ($T = 2$ used in [3] and in all the experiments)
- by construction $\sum_{t \in Tasks} \lambda_t = K$
- **initialization:** $w_t(k) = 1 \quad \forall k \leq 2$

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

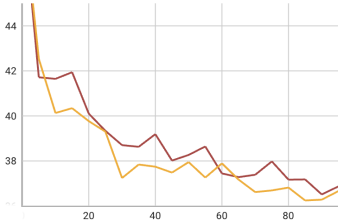
DWA vs Equal Weighting I



(a) Pixel Accuracy



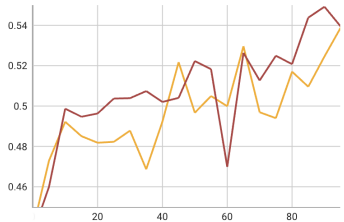
(b) Mean Absolute Error



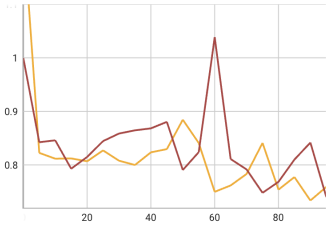
(c) Angle Distance (Mean)

Figure: Statistics of MTAN on NYUv2 validation set with Dynamic Weight Averaging (orange) and equal weighting (red)

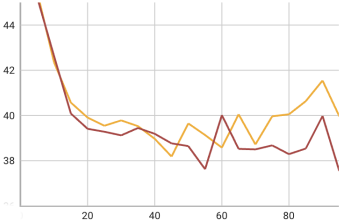
DWA vs Equal Weighting II



(a) Pixel Accuracy



(b) Mean Absolute Error



(c) Angle Distance (Mean)

Figure: Statistics of Cross-Stitch Network on NYUv2 validation set with Dynamic Weight Averaging (orange) and equal weighting (red)

DWA vs Equal Weighting III

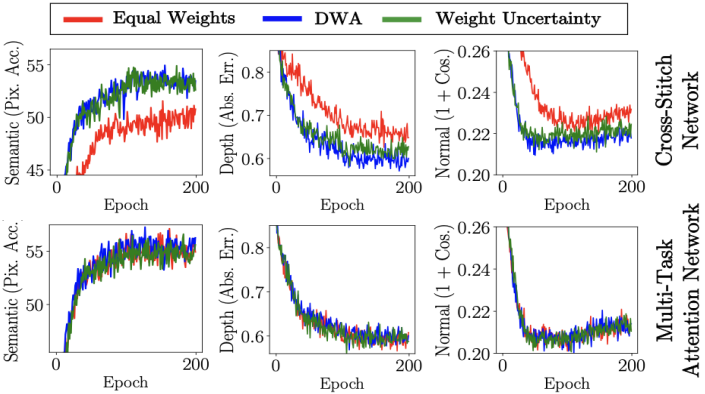
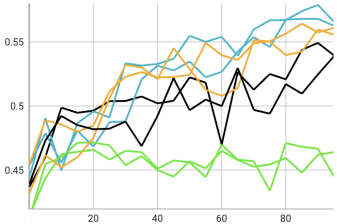


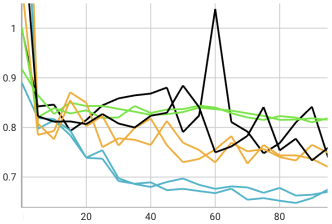
Figure: Paper results about different weighting schemes on NYUv2 validation set

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

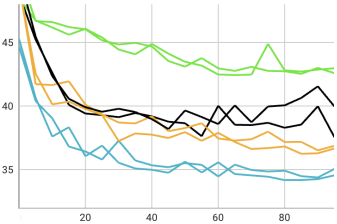
DWA vs Equal Weighting IV



(a) Pixel Accuracy



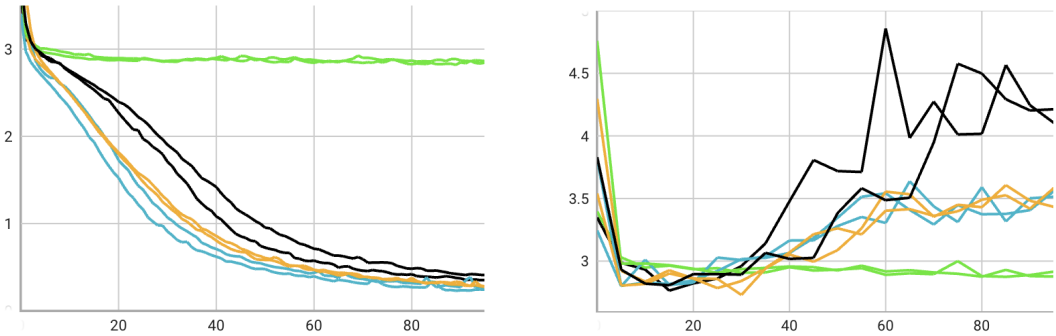
(b) Mean Absolute Error



(c) Angle Distance (Mean)

Figure: Statistics of MTAN (orange), Cross-Stitch Network (black), DenseNet (blue) and SplitNet (green) on NYUv2 validation set

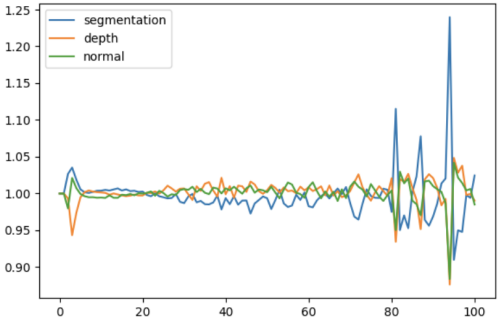
DWA vs Equal Weighting V



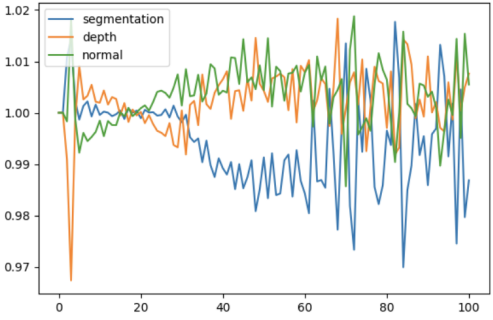
(a) Train (b) Validation

Figure: Total losses of MTAN (orange), Cross-Stitch Network (black), DenseNet (blue) and SplitNet (green) on NYUv2 validation set

DWA vs Equal Weighting VI



(a) MTAN lambdas



(b) Cross-Stitch Network lambdas

Figure: Time evolution of λ_t according to DWA on NYUv2 dataset

Experiments

Training details:

- **Adam** optimizer, learning rate 0.0001
- All models trained for 100 **epochs** (on both datasets)
- **Batch size** of 2 for NYUv2 and 8 for Cityscapes

Architecture	Cityscapes	NYUv2
Single Task (SegNet)	$\approx 65M$	$\approx 65M$
STAN	$\approx 64M$	$\approx 64M$
Split	$\approx 65M$	$\approx 65M$
Dense	$\approx 78M$	$\approx 108M$
Cross-Stitch	$\approx 50M$	$\approx 75M$
MTAN	$\approx 43M$	$\approx 56M$

Table: Number of (learnable) parameters of different models

My Quantitative Results (Cityscapes)

Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err
One Task	n.a.	21.80	76.23	0.0299	37.59
STAN	n.a.	39.15	89.51	0.0171	31.78
Split	Equal Weights	**29.56**	**84.48**	**0.0287**	36.72
	DWA, $T = 2$	23.68	78.83	0.0318	**36.58**
Dense	Equal Weights	44.75	91.78	**0.0159**	**26.98**
	DWA, $T = 2$	**45.23**	**92.15**	**0.0159**	31.08
Cross-Stitch	Equal Weights	45.11	92.00	0.0185	31.95
	DWA, $T = 2$	**45.81**	**92.56**	**0.0165**	**28.95**
MTAN	Equal Weights	**43.16**	**91.12**	0.0184	30.23
	DWA, $T = 2$	42.67	91.07	**0.0177**	**26.99**

Table: Quantitative results on Cityscapes dataset

Liu, Johns, and Davison Quantitative Results (Cityscapes)

Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err
One Task	n.a.	51.09	90.69	0.0158	34.17
STAN	n.a.	51.90	90.87	0.0145	27.46
Split, Wide	Equal Weights	50.17	90.63	0.0167	44.73
	Uncert. Weights	**51.21**	**90.72**	**0.0158**	44.01
	DWA, $T = 2$	50.39	90.45	0.0164	**43.93**
Split, Deep	Equal Weights	**49.85**	88.69	0.0180	43.86
	Uncert. Weights	48.12	88.68	**0.0169**	**39.73**
	DWA, $T = 2$	49.67	**88.81**	0.0182	46.63
Dense	Equal Weights	**51.91**	90.89	0.0138	27.21
	Uncert. Weights	51.89	**91.22**	**0.0134**	**25.36**
	DWA, $T = 2$	51.78	90.88	0.0137	26.67
Cross-Stitch	Equal Weights	50.08	90.33	0.0154	34.49
	Uncert. Weights	50.31	90.43	**0.0152**	**31.36**
	DWA, $T = 2$	**50.33**	**90.55**	0.0153	33.37
MTAN	Equal Weights	53.04	**91.11**	**0.0144**	**33.63**
	Uncert. Weights	**53.86**	91.10	0.0144	35.72
	DWA, $T = 2$	53.29	91.09	0.0144	34.14

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

My Quantitative Results (NYUv2)

Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err	Surface Normal Angle Distance (Lower Better) Mean	Surface Normal Angle Distance (Lower Better) Median	Surface Normal Within t° (Higher Better) 11.25	Surface Normal Within t° (Higher Better) 22.5	Surface Normal Within t° (Higher Better) 30
One Task	n.a.	8.53	44.51	0.8036	0.3464	40.60	37.91	11.45	28.77	39.61
STAN	n.a.	12.62	50.33	0.7751	0.3393	36.73	31.20	16.38	37.08	49.17
Split	Equal Weights	8.76	46.42	**0.8226**	**0.3752**	43.54	41.08	6.93	22.87	34.43
	DWA, $T = 2$	**9.27**	**46.61**	0.8289	0.3803	**42.59**	**40.05**	**8.36**	**26.40**	**37.58**
Dense	Equal Weights	**16.75**	**58.34**	**0.6472**	**0.2779**	34.52	**27.46**	**21.35**	**43.17**	**54.55**
	DWA, $T = 2$	16.18	57.32	0.6726	0.2906	**34.46**	27.54	20.84	42.95	54.43
Cross-Stitch	Equal Weights	**16.05**	**55.07**	**0.7422**	0.3020	**37.57**	**31.78**	**16.77**	**36.95**	**48.34**
	DWA, $T = 2$	14.57	53.35	0.7599	**0.2835**	39.25	33.55	15.74	35.05	46.05
MTAN	Equal Weights	**15.74**	**56.55**	**0.7393**	0.2818	36.74	30.94	**17.74**	38.09	49.47
	DWA, $T = 2$	15.63	54.15	0.7468	**0.2725**	**36.00**	**30.45**	17.18	38.09	**50.11**

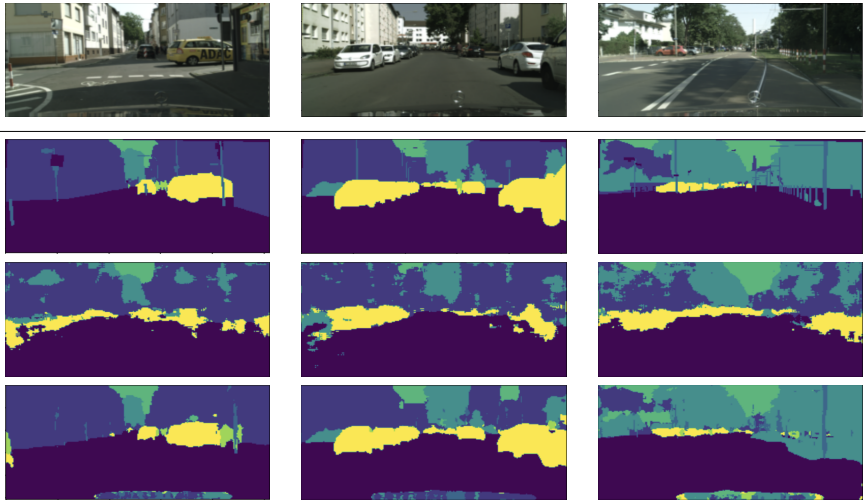
Table: Quantitative results on NYUv2 dataset

Liu, Johns, and Davison Quantitative Results (NYUv2)

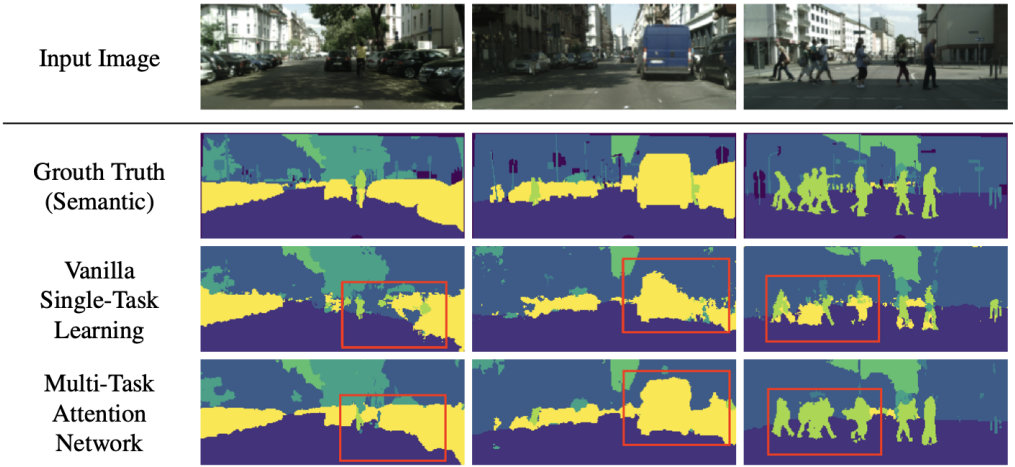
Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err	Surface Normal: Angle Distance (Lower Better) Mean	Surface Normal: Angle Distance (Lower Better) Median	Surface Normal: Within t° (Higher Better) 11.25	Within t° 22.5	Within t° 30
One Task	n.a.	15.10	51.54	0.7508	0.3266	31.76	25.51	22.12	45.33	57.13
STAN	n.a.	15.73	52.89	0.6935	0.2891	32.09	26.32	21.49	44.38	56.51
Split, Wide	Equal Weights	15.89	51.19	0.6494	0.2804	33.69	28.91	18.54	39.91	52.02
	Uncert. Weights	15.86	51.12	**0.6040**	0.2570	**32.33**	**26.62**	**21.68**	**43.59**	**55.36**
	DWA, $T = 2$	**16.92**	**53.72**	0.6125	**0.2546**	32.34	27.10	20.69	42.73	54.74
Split, Deep	Equal Weights	13.03	41.47	0.7836	0.3326	38.28	36.55	9.50	27.11	39.63
	Uncert. Weights	**14.53**	43.69	0.7705	0.3340	**35.14**	**32.13**	**14.69**	**34.52**	**46.94**
	DWA, $T = 2$	13.63	**44.41**	**0.7581**	**0.3227**	36.41	34.12	12.82	31.12	43.48
Dense	Equal Weights	16.06	52.73	0.6488	0.2871	33.58	28.01	20.07	41.50	53.35
	Uncert. Weights	**16.48**	**54.40**	0.6282	0.2761	**31.68**	**25.68**	**21.73**	**44.58**	**56.65**
	DWA, $T = 2$	16.15	54.35	**0.6059**	**0.2593**	32.44	27.40	20.53	42.76	54.27
Cross-Stitch	Equal Weights	14.71	50.23	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
	Uncert. Weights	15.69	52.60	0.6277	0.2702	32.69	27.26	21.63	42.84	54.45
	DWA, $T = 2$	**16.11**	**53.19**	**0.5922**	**0.2611**	**32.34**	**26.91**	**21.81**	**43.14**	**54.92**
MTAN	Equal Weights	**17.72**	55.32	**0.5906**	0.2577	31.44	**25.37**	**23.17**	45.65	57.48
	Uncert. Weights	17.67	**55.61**	0.5927	0.2592	**31.25**	25.57	22.99	**45.83**	**57.67**
	DWA, $T = 2$	17.15	54.97	0.5956	**0.2569**	31.60	25.46	22.48	44.86	57.24

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

My Qualitative Results (Segmentation)

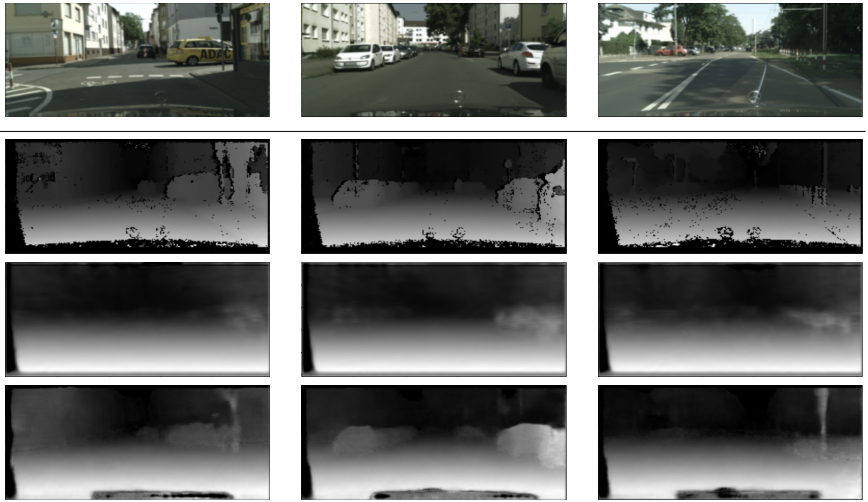


Liu, Johns, and Davison Qualitative Results (Segmentation)



[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

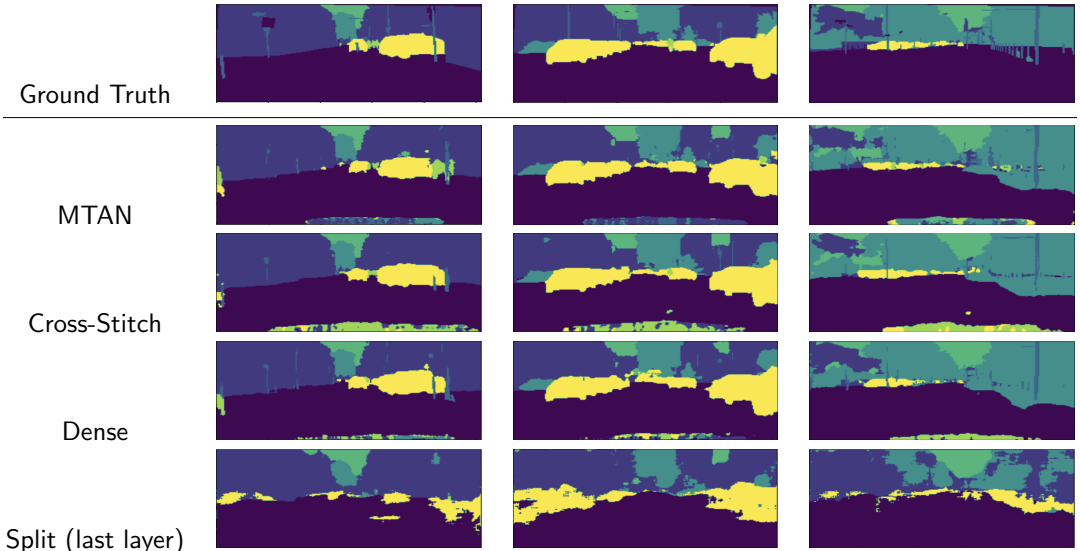
My Qualitative Results (Depth Estimation)



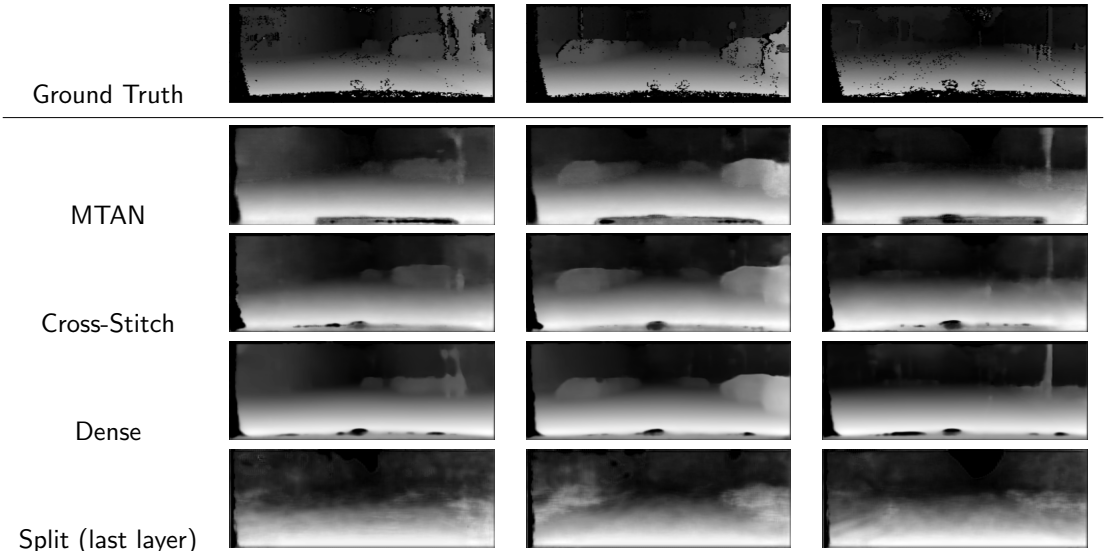
Liu, Johns, and Davison Qualitative Results (Depth Estimation)

[3] Liu, Johns, and Davison, *End-to-End Multi-Task Learning with Attention*, 2019

My Multi-task Models Comparison (Segmentation)



My Multi-task Models Comparison (Depth Estimation)



Conclusion

Final observations:

- Qualitative and quantitative results suggest the effectiveness of a multi-task approach
- DWA does not seem so useful (although not detrimental) for the tasks in this paper
- Depth estimantion and segmentation tasks seem to be more correlated than surface normal estimation

Future Work:

- Explore other architectures
- Explore other datasets
- Explore other weighting strategies

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2016. arXiv: 1511.00561 [cs.CV]. URL: https://arxiv.org/abs/1511.00561.

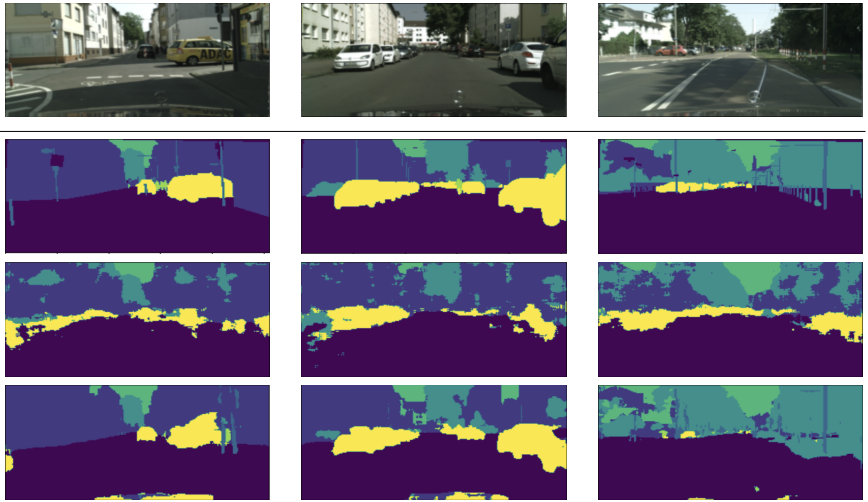
[2] Marius Cordts et al. *The Cityscapes Dataset for Semantic Urban Scene Understanding*. 2016. arXiv: 1604.01685 [cs.CV]. URL: https://arxiv.org/abs/1604.01685.

[3] Shikun Liu, Edward Johns, and Andrew J. Davison. *End-to-End Multi-Task Learning with Attention*. 2019. arXiv: 1803.10704 [cs.CV]. URL: https://arxiv.org/abs/1803.10704.

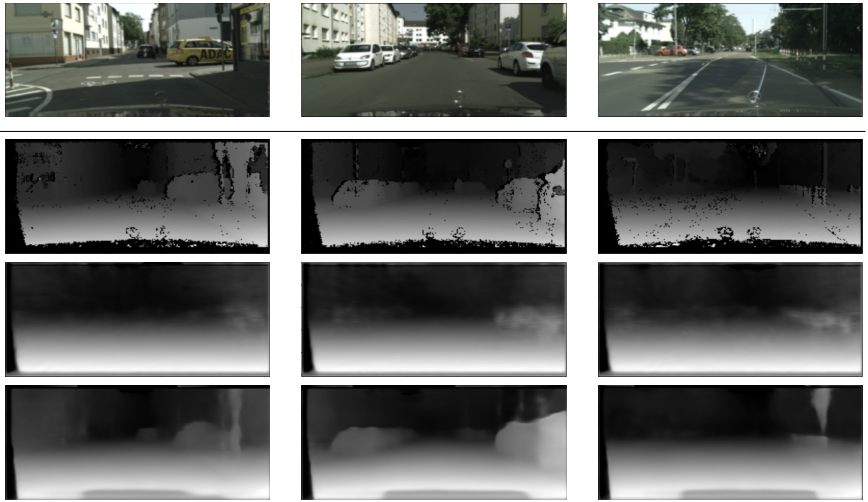
[4] Ishan Misra et al. *Cross-stitch Networks for Multi-task Learning*. 2016. arXiv: 1604.03539 [cs.CV]. URL: https://arxiv.org/abs/1604.03539.

[5] Nathan Silberman et al. *Indoor Segmentation and Support Inference from RGBD Images*. 2012. URL: https://cs.nyu.edu/~fergus/datasets/indoor_seg_support.pdf.

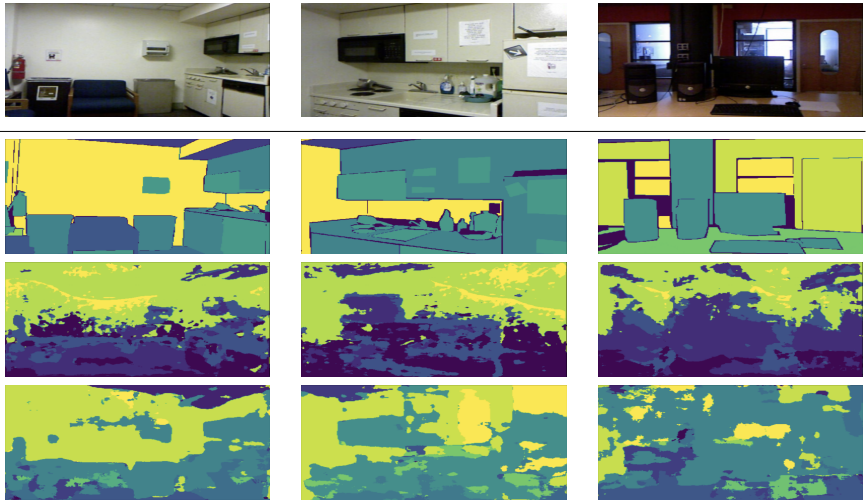
Appendix (STAN vs Single Task I)



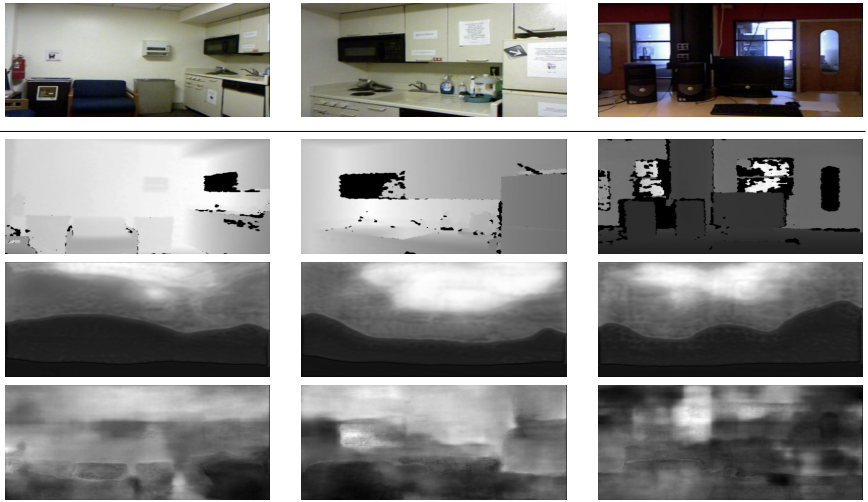
Appendix (STAN vs Single Task II)



Appendix (NYUv2 Qualitative Results I)



Appendix (NYUv2 Qualitative Results II)



Appendix (NYUv2 Qualitative Results III)