AdaMT-Net: An Adaptive Weight Learning Based Multi-Task Learning Model For Scene Understanding

Ankit Jha[‡]* Awanish Kumar[†]* Biplab Banerjee[‡] Subhasis Chaudhuri[†]

[†]Deptt. of Electrical Engineering [‡]Centre of Studies in Resources Engineering
IIT Bombay, Mumbai, India

{ankitjha16, awanishk389, getbiplab}@gmail.com, sc@ee.iitb.ac.in

Abstract

We tackle the problem of deep end-to-end multi-task learning (MTL) for visual scene understanding from monocular images in this paper. It is proven that learning several related tasks together helps in attaining improved performance per-task than training them independently. This is due to the fact that related tasks share important feature characteristics among themselves, which the MTL techniques can effectively explore for improved joint training. Based on this premise, we are interested in generic to specific feature extraction given the different tasks within a common framework. To this end, we propose a typical U-Net based encoder-decoder architecture called AdaMT-Net, where the densely-connected deep convolutional neural network (CNN) based feature encoder is shared among the tasks while the soft-attention based task-specific decoder modules produce the desired outcomes. One major issue in MTL is to assign the weights for the task-specific loss-terms in the final cumulative optimization function. As opposed to the manual approach, we propose a novel adaptive weight learning strategy by carefully exploring the loss-gradients per-task over the training iterations. Experimental results on the benchmark CityScapes, NYUv2, and ISPRS datasets confirm that AdaMT-Net achieves state-of-the-art performance on most of the evaluation metrics.

1. Introduction

The recent times have witnessed great strides in visual inference tasks with the application of deep CNN models. This can predominantly be attributed to the availability of large-scale labeled training samples for a given inference task [14]. However, as opposed to the traditional single task-specific CNN frameworks, the notion of learning multiple tasks together within a common CNN based foundation (aka multi-task learning or MTL in short) by exploring their similarities has secured much potential [1, 24]. At its core, this idea is inspired by human learning abilities, where one often applies the knowledge across subjects for better overall learning purposes.Theoretically, the inductive bias is provided by one of the competing tasks (generally referred to as the auxiliary task), which in turn causes the model to prefer hypotheses that are able to explain multiple tasks together. The idea of MTL is particularly eminent in the area of visual computing where there exist several inference tasks (dense pixel segmentation, dense depth estimation, etc.), which can be sensibly modeled together by exploring the homogeneity of local image patches. Needless to mention, such an MTL approach is capable of performing multi-view scene understanding by harnessing both the 2D semantics and 3D depth information. In this paper, we focus on introducing a novel CNN based framework for joint learning of three highly similar visual inference tasks, namely, semantic segmentation, depth estimation, and surface normal estimation.

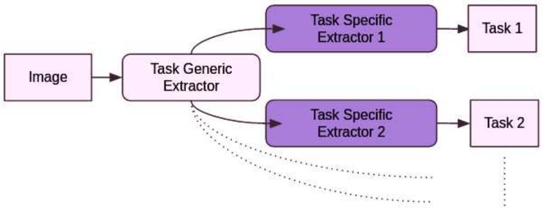

Figure 1. A high-level schematic of AdaMT-Net where the task-generic to task-specific feature learning stages are dissociated using an encoder-decoder framework.

Nonetheless, the design of efficient MTL architecture with respect to the deep CNN models has some inherent obstacles: i) how to intuitively share the knowledge among different tasks, ii) how to model the overall optimization cost taking all the competing tasks into account. Generally speaking, we note that tasks like semantic segmentation or depth estimation are usually carried out through

*Authors contributed equally

an encoder-decoder based CNN structure, where a latent feature space is first modeled in the encoder-end whereas the decoder performs the desired reconstruction. A similar principle has been extended in MTL as well recently [5, 16, 17]. While some of the deep MTL models prefer separate feature-encoder per-task, shared feature-encoder is considered in other cases (Figure 1). The second option is usually favoured considering that sharing the encoder across the tasks helps in regularizing the training process, besides bringing the number of trainable parameters down significantly.

At the same time, we foresee two major concerns in the architecture design of Figure 1: i) as we deal with dense pixel-wise prediction tasks, the standard vanilla CNN based encoder may not be able to capture the more low-level image information precisely. This is driven by the fact that while the shallow CNN layers capture low-level image features, the deeper layers are oriented towards extracting more abstract high-level feature representations, and ii) given the shared feature-encoder, how to channelize the task-specific decoders is another critical issue. It is important to sunder the features tailored for each task explicitly from the encoded feature space further to be fed to the individual decoder modules. One possible solution in this respect could be to learn distinct attention modules for the decoders.

From a different perspective, another apprehension in training the MTL models in general arises from the choice of the combined loss function for all the tasks. While semantic segmentation being a classification problem utilizes a cross-entropy based loss measure, depth-estimation and surface-normal estimation are typically dealt with under a regression framework. Hence, the mere addition of these loss-terms is not adequate as the proper balance among the tasks is not maintained. An alternative way would be to resort to a weighted summation of these loss-terms. It can be clearly envisioned that a manual weighting is imprecise in this case, and an adaptive weight learning paradigm is highly endorsed.

Inspired by these arguments, we propose a novel U-Net [19] based MTL model with an adaptive weight learning scheme for the individual loss-terms: **AdaMT-Net**. Seemingly, AdaMT-Net consists of a shared feature-encoder for all the tasks and separate decoder networks for the individual tasks, as illustrated in Figure 1. However, in order to combat the aforementioned issues, we introduce the following measures. i) We follow a dense-block architecture for the shared-encoder, which can efficiently combine low to high-level images features, thus aiding in the dense structured prediction tasks. The task-specific decoders, in contrast, reasonably utilize the notion of attention learning to highlight the task-oriented feature-components from the shared space. ii) In order to avoid manual weight assignment to the loss-terms, we subsequently propose to explore the gradient magnitudes of the individual loss-term with respect to the respective decoder parameters over the training iterations to calculate the loss-specific weights adaptively. In this respect, we hypothesize that if a particular task is being trained well, the loss-gradient magnitude is expected to be low. This, in turn, suggests that the respective loss-weight can be reduced. We ensure to follow a convex combination of the loss-weights for stability. Our novel contributions can be summarized as follows:

i) We propose an adaptive CNN based model for performing multi-task learning called AdaMT-Net. ii) AdaMT-Net can separate the shared features given a set of competing tasks from the task-specific features through novel network architecture. In this regard, we follow a dense-block framework in the encoder, while attention-based decoder models are considered for the tasks. We also introduce a novel weight learning scheme for the task-specific loss-terms by following the loss-gradients of the individual tasks. iii) Extensive experimental analysis is carried out on the CityScapes [4], NYUv2 [21], and ISPRS [20] datasets where improved performance is observed consistently.

2. Related works

Multi-task learning: MTL [1, 7, 15] has been regarded as one of the cost-effective solutions for deciphering several inference tasks simultaneously. Precisely, MTL aims to improve the learning for each of the tasks by efficiently exploring the complementary and shared information jointly present in all the tasks. Earlier, MTL was primarily solved using traditional feature transformation based approaches such as latent support vector machine (SVM) [25], Bayesian matrix factorization [23], task clustering [10], matrix decomposition [2], to name a few. Subsequently, the traditional ad-hoc approaches have been replaced by the deep learning techniques, which are greatly benefited by adhering to their feature learning capabilities. MTL approaches developed in conjunction with the deep CNN models have successfully been implemented in problems concerning joint semantic segmentation, depth prediction, and surface normal estimation [6, 17], preferably from monocular images. In this respect, the majority of the CNN based frameworks are designed in encoder-decoder fashion [17, 13]. Another compelling aspect of MTL is to learn the weights for the individual loss-terms. Manual tuning of weights leads to sub-optimal model training. In this regard, several works have focused on the Bayesian approach [11], learning-based model [16] for adaptive weight learning.

Attention models: The attention modules in general help in highlighting visually appealing regions in images. Typically, attention modules are deployed in either of two ways: soft or hard attention. In soft attention, relative weights are learned for different feature dimensions. On the other hand, hard attention processes each feature dimension and either selects or rejects the same. For image data, the conventional deep CNN models can be extended to support the attention modeling within the end-to-end training setup. In this regard, convolutional block attention module (CBAM) [22], local attention masks [8], attention U-Net [18], multi-task attention network (MTAN) [16] are some of the popular variants.

The existing method most similar to ours is MTAN [16], which follows the segnet based encoder-decoder setup for MTL. Instead, we postulate to follow the U-Net driven architecture since U-Net offers better feature space exploration by connecting the encoder layers to the respective decoder layers in the up-sampling stage. Next, we follow a dense-block model in the encoder, which can efficiently combine the low to high-level image features in the encoded feature space, a paradigm that is important for structured prediction tasks. Finally, we propose a simple yet intuitive gradient-based adaptive weight learning strategy for the individual loss-terms. Experimentally, we find AdaMT-Net sharply outperforms MTAN along with other comparative methods for all the datasets.

3. Proposed methodology

The objective is to develop a deep U-Net [19] based encoder-decoder model, which is capable of simultaneously learning multiple different but related tasks together from a given input. Formally, let us consider a multi-task learning dataset $\mathcal{X} = \{x_i, \{y_i^t\}_{t=1}^{\mathcal{T}}\}$ equipped with $\mathcal{T}$ tasks where $x \in \mathcal{X}$ denotes the input image and $y^t \in \mathcal{Y}^t$ is the output corresponding to the t^{th} task. In our experiments, we fix $\mathcal{T} = 2$ or $\mathcal{T} = 3$ given three different structured prediction tasks: semantic segmentation, depth estimation, and surface normal estimation, respectively. We further note that ours is a homogeneous MTL setup since all the tasks are trained on the same training images. Under this setup, our proposed AdaMT-Net follows a hard feature sharing based MTL framework by implementing a shared feature encoder for all the tasks on top of which separate attention-driven task-specific decoder modules are enacted.

3.1. Model architecture

As mentioned, AdaMT-Net consists of two major modules, the shared global feature learning network ($f^E(;,\theta^E)$) with parameters θ^E and separate task-specific decoders ($f_t^D(;,\theta_t^D)_{t=1}^{\mathcal{T}}$) corresponding to $\mathcal{T}$ tasks where θ_t^D defines the parameters for the t^{th} decoder. We note that f^E and f_t^D are realized in terms of CNNs where each of the corresponding encoder and decoder blocks (for all the decoder streams separately) are connected by bridge connections, following the design protocol for U-Net. Such a bridge is used to directly transfer the encoder-block feature maps to the respective decoder-end. A block diagram for AdaMT-Net can be found in Figure 2. Precisely, for a given input x, the encoded representation $\hat{x}$ and t^{th} decoder output $\hat{y}_t$ are obtained as follows:

$$\left.\begin{aligned} \hat{x} &= f^E(x, \theta^E) \\ \hat{y}^t &= f_t^D(\hat{x}, \theta_t^D) \end{aligned}\right\} \tag{1}$$

In the following, we discuss about the design of f^E and $\{f_t^D\}_{t=1}^{\mathcal{T}}$ together with the loss functions considered.

Shared feature encoder f^E: We choose to follow a dense-block CNN based framework for designing the feature encoder f^E where dense forward connections exist between all pairs of convolution blocks. Each block consists of a small feed-forward CNN model consisting only of convolution layers. The final encoder block is subsequently attached to the bottleneck layer, which represents the shared feature space ($\hat{x}$ for a given x). As already stated, we note that the convolutional kernels corresponding to the initial convolution layers of a typical CNN tend to learn more low-level feature constructs while the feature abstraction increases proportionately with the network depth. We feel that proper amalgamation of such a feature hierarchy is expected to be fruitful for dense prediction tasks. From another perspective, the use of extensive skip-connections helps to handle the problems related to vanishing gradient by offering multiple paths for gradient-flow during the backward propagation stage of the training iterations. Learning the encoder means obtaining the optimal values for θ^E minimizing the final multi-task loss function.

Task-specific decoders $\{f_t^D\}_{t=1}^{\mathcal{T}}$: We consider $\mathcal{T}$ decoder networks where each of the decoder streams follows symmetric network structure with respect to the shared encoder f^E. Due to the bridge connections between the respective encoder and decoder blocks, the input to the first convolution layer within a decoder block is derived from the concatenation of the feature maps of the previous decoder block and the respective encoder block. Further, the decoder modules are equipped with a sophisticated attention learning scheme to highlight the task-specific features explicitly. We detail the loss functions used for each of the decoders first, which is followed by the discussions about the attention scheme and the task-specific weight learning strategy. For simplicity, we consider decoder f_1^D for depth estimation, decoder f_2^D for semantic segmentation, and decoder f_3^D for surface normal prediction, respectively.

a) Decoder loss for depth estimation: We follow the standard ℓ_1-norm based distance to define the depth loss ($\mathcal{L}_D$) given their efficacy in the depth estimation literature [6].

$$\mathcal{L}_D(y^1, \hat{y}^1) = \frac{1}{HW} \sum_{j=0}^{H-1} \sum_{k=0}^{W-1} \mid\mid y^1(j,k) - \hat{y}^1(j,k) \mid\mid_1^1 \tag{2}$$

where y^1 and $\hat{y}^1 = f_1^D(f^E(x))$ define the ground-truth

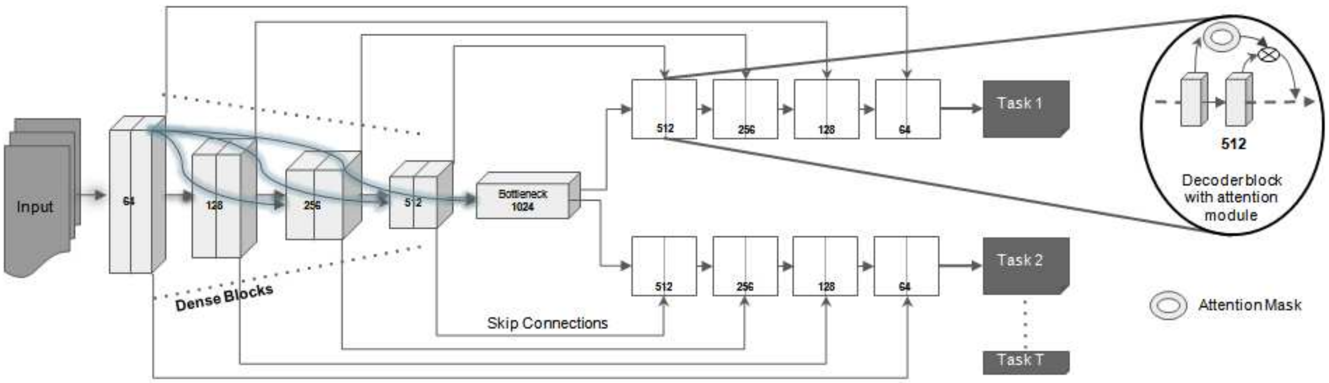

Figure 2. The overall architecture of the U-Net based AdaMT-Net with dense-block architecture followed in the encoder network and separate soft-attention based task-specific decoders. Also, note that each decoder has a symmetric structure with respect to the encoder. We also depict the architecture of the attention learning process. The number of feature maps mentioned for each block remains the same for all the dataset.

and predicted depth masks for image x with (H, W) denoting the number of rows and columns of x, respectively.

b) Decoder loss for semantic segmentation: We note that decoder f_2^D is very much similar to f_1^D except, it deals with a multi-class prediction problem given the image pixels. Hence, the cross-entropy loss is deployed for the same. Typically, considering that the pixels can take any of the $\{1, 2, \cdots, \mathcal{C}\}$ semantic labels, the segmentation loss can be mentioned as,

$$\mathcal{L}_S(y^2, \hat{y}^2) = -\frac{1}{HW} \sum_{j=0}^{H-1} \sum_{k=0}^{W-1} y^2(j,k) \log \hat{y}^2(j,k) \quad (3)$$

where, y^2, $\hat{y}^2 = f_2^D(f^E(x))$ are the ground-truth labels and predicted segmentation maps for x.

c) Decoder loss for surface normal estimation: Surface normal estimation is very important to learn the spatial characteristics of the surrounding environment, and they are essentially computed from the 3D mesh structure. We follow the element-wise dot $(\cdot)$ product between the normalized pixels and the ground truth map for designing the loss corresponding to surface normal prediction as follows,

$$\mathcal{L}_N(y^3, \hat{y}^3) = -\frac{1}{HW} \sum_{j=0}^{H-1} \sum_{k=0}^{W-1} y^3(j,k) \cdot \hat{y}^3(j,k) \quad (4)$$

where, y^3, $\hat{y}^3 = f_3^D(f^E(x))$ are the ground-truth labels and predicted surface normal maps for x.

Attention learning framework followed for each decoder: In order to learn more focused task-specific features at the decoders, we further consider introducing attention learning modules for the decoder blocks. These attention learning modules are inspired from MTAN [16]. Assuming the presence of two convolution layers within each decoder block, let $\mathcal{M}_l^1$ and $\mathcal{M}_l^2$ be the feature maps corresponding to each of the layers of the l^{th} block, respectively. We learn the attention mask $\mathcal{A}_l$ from $\mathcal{M}_l^1$ using a small network and subsequently perform a dot-product of the same with $\mathcal{M}_l^2$. Henceforth, the set of feature maps to be propagated from the l^{th} to the $(l+1)^{th}$ block is represented as the concatenation of $\mathcal{M}_l^2$ and $\mathcal{A}_l \odot \mathcal{M}_l^2$. Since the first convolution layer of the decoder block has its input from the previous decoder block and the corresponding encoder block, $\mathcal{A}_l$ learns important feature weights and propagates the same to the second convolution layer which by itself is expected to learn more complex feature abstractions. Hence, the decoders ensure that the lower level important features are not neglected in the deeper layers. It is also to be noted that the attention learning module has its own set of parameters that are trained in a self-supervised fashion along with other parameters of AdaMT-Net.

3.2. Overall loss function for AdaMT-Net

In MTL, the cumulative loss for all the tasks can be defined in various ways, such as just adding all the loss-terms with manual weights per loss or weighted sum of loss-terms with learnable weights. Here we consider the magnitude of the loss gradient to be the measure of importance for a given task in each iteration and accordingly decide on the weight for the respective loss-term. Apparently, if the gradient magnitude is less, it signifies that the task is being learned towards optimality, and the corresponding weight may be decreased. On the other hand, a large gradient magnitude suggests abrupt training for the task and requires further attention. We highlight that our approach is very different from the GradNorm method of [3] in the sense that GradNorm automatically tweaks the gradient magnitude of the loss-specific gradient magnitudes to regularize the multi-task training whereas we explore gradient magnitude to perform adaptive loss-specific weight learning. In this regard, we consider the normalized average gradient magnitude of a given loss-term with respect to the corresponding decoder's parameters as the relative weight for the specific task. Mathematically, the weight $\mathcal{W}$ and the total loss $\mathcal{T}_{Total}$ for $i, j \in \{D, S, N\}$ can be mentioned as,

$$\mathcal{W}_i = \frac{\overline{\nabla_{\vec{\theta}_i^D} \mathcal{L}_i}}{\sum\limits_{j=1}^{\mathcal{L}} \overline{\nabla_{\vec{\theta}_j^D} \mathcal{L}_j}} \tag{5}$$

$$\mathcal{L}_{Total} = \mathcal{W}_S \mathcal{L}_S + \mathcal{W}_D \mathcal{L}_D + \mathcal{W}_N \mathcal{L}_N \tag{6}$$

where $\overline{\nabla_{\vec{\theta}_i^D} \mathcal{L}_i}$ is the average gradient magnitude of $\mathcal{L}_i$ with respect to the decoder's parameters $\vec{\theta}_i^D$ for the $i^{th} \in \{D, S, N\}$ task. The normalization strategy of Equation 5 further ensures a convex combination for the weights: $\mathcal{W}_D + \mathcal{W}_S + \mathcal{W}_N = 1$. From Bayesian point of view, it can also be presumed that a small $\mathcal{W}$ refers to high certainty whereas a large $\mathcal{W}$ denotes uncertainty for learning the task.

4. Experimental evaluations

This section deals with the evaluation of AdaMT-Net on three datasets: CityScapes [4], NYUv2 [21], and ISPRS [20]. While the major tasks consider for all the dataset being semantic segmentation and depth estimation, we evaluate AdaMT-Net for the additional task of the surface normal prediction on the NYUv2 dataset. As a whole, we follow the dataset design protocols followed in [16].

4.1. Datasets

CityScapes: The CityScapes dataset contains high-resolution street-view images to be deployed for the purpose of semantic segmentation and depth estimation. In this regard, we consider the standard 7 semantic classes for evaluating the segmentation performance of AdaMT-Net. As a pre-processing step, the images are re-sized to $[128, 256]$ prior to feeding to the network.

NYUv2: Our model is evaluated on the NYUv2 dataset for the joint segmentation, surface normal prediction, and depth estimation tasks. This dataset consists of RGB-D indoor scene images from 13 semantic categories and is more challenging compared to the CityScapes dataset. It is primarily due to variations in camera viewpoint, scene occlusion, differences in lighting conditions, etc. Similar to CityScapes; the images are re-sized to $[128, 256]$.

ISPRS: Finally, the ISPRS dataset is considered for semantic segmentation and depth estimation. This dataset contains the aerial scenes of the ground surface as well as the digital surface model (DSM) of Potsdam city of Germany, where the pixels are mapped to 6 semantic classes. The DSM data is considered for depth estimation. We consider non-overlapping tiles of size $[256, 256]$ from the bigger scenes for training and evaluation.

4.2. Architecture and training protocols

The feature encoder of AdaMT-Net consists of four CNN blocks, each of them being made up of two convolution layers with kernel size 3×3 for each layer. The convolution blocks of feature encoder are further interconnected in a similar way as the dense blocks in a typical DenseNet model [9]. Additionally, each of the convolution blocks is followed by a dropout and a max-pool layer with a kernel size of 2×2 and stride 2. ReLU non-linearity and Batch-normalization are used to ensure stable training. In gist, the convolution blocks compute the feature maps of depth 64,128, 256, and 512, respectively. The bottleneck layer, which also represents the shared feature space, consists of 1024 feature maps. The decoder modules follow symmetric architecture with respect to the shared feature encoder. Besides, the input to each of the convolution blocks at the decoder-ends is the output of concatenated feature maps of the corresponding encoder block and the immediate previous convolution block in the decoder. For the decoder blocks, the respective attention modules consist of two 3×3 convolution layers, each followed by Batch-normalization and Sigmoid layers. Sigmoid non-linearity ensures that the masks' values are squeezed within the range $[0, 1]$.

The model is trained for 200 epochs using Adam optimizer [12] with an initial learning rate of $1e-4$. We further consider a batch size of 8, 4, and 2 for CityScapes, ISPRS, and NYUv2 datasets, respectively.

4.3. Evaluation metrics

To evaluate the performance, we rely on following standard metrics: intersection over union (IoU), and mean intersection over union (mIoU) for semantic segmentation, absolute error, and relative error for depth estimation, and mean, median, and angular distance of the percentage of pixels whose predictions lie within the angular deviations of $11.25°$, $22.5°$ and $30°$ to the ground truth for surface normal prediction.

4.4. Comparison to the literature

We compare the performance of AdaMT-Net with a number of techniques from the literature: STAN [16], DenseNet [9], Cross-Stitch network [17], and MTAN [16], respectively. In addition, we analyze the performance of MTL with respect to learning each of the tasks separately: ST-Net, which follows a single encoder-single decoder setup out of AdaMT-Net. As far as the weight learning for the losses is concerned, we consider two cases: manual weighting with equal weights to all the loss terms and the proposed weighting scheme of Equation 5.

Method	Segmentation		Depth error		Surface Normal				
					Angle Distance		Within t°		
	(Higher)		(Lower)		(Lower)		(Higher)		
	IoU	mIoU	Absolute	Relative	Mean	Median	11.25	22.5	30
ST-Net (Ours)	58.44	20.98	0.6657	0.2746	29.67	25.08	20.94	45.52	58.74
STAN [16]	52.89	15.73	0.6935	0.2891	32.09	26.32	21.49	44.38	56.51
Dense [9]	52.73	16.06	0.6488	0.2871	33.58	28.01	20.07	41.50	53.35
Cross-Stitch [17]	52.73	14.71	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
MTAN [16]	55.32	17.72	**0.5906**	0.2577	31.44	25.37	23.17	45.65	57.48
AdaMT-Net (Ours)∗	59.55	22.36	0.6247	0.2558	29.41	24.05	23.72	47.42	59.83
AdaMT-Net (Ours)†	**60.35**	**21.86**	0.5933	**0.2456**	**27.74**	**21.85**	**26.58**	**51.63**	**63.88**

Table 1. 3-task multi-task validation results on NYUv2 dataset for 13-class semantic segmentation, depth estimation and surface normal prediction. †Gradient-based weight learning, * Equal weights.

Method	Segmentation (Higher Better)		Depth error (Lower Better)	
	IoU	mIoU	Abs.	Rel.
ST-Net (Ours)	58.44	20.98	0.6657	0.2746
STAN [16]	55.07	16.65	0.6935	0.2891
Dense [9]	55.59	17.22	0.6002	0.2654
Cross-Stitch [17]	53.99	17.01	0.6095	0.2671
MTAN [16]	56.24	18.32	**0.5931**	0.2562
AdaMT-Net (Ours)∗	58.42	21.74	0.6320	0.2551
AdaMT-Net (Ours)†	**58.91**	**20.61**	0.6136	**0.2547**

Table 2. 2-task multi-task validation results on NYUv2 dataset for 13-class semantic segmentation and depth estimation. †Gradient-based weight learning, * Equal weights.

Method	Segmentation (Higher Better)		Depth error (Lower Better)	
	IoU	mIoU	Abs.	Rel.
ST-Net (Ours)	93.71	61.58	0.0131	24.21
STAN [16]	90.87	51.90	0.0145	27.46
Dense [9]	90.89	51.91	0.0138	27.21
Cross-Stitch [17]	90.33	50.08	0.0154	34.49
MTAN [16]	91.11	53.04	0.0144	33.63
AdaMT-Net (Ours)∗	94.01	61.91	0.0129	23.82
AdaMT-Net (Ours)†	**94.16**	**62.53**	**0.0125**	**22.23**

Table 3. Multi-task validation results on CityScapes dataset for 7-class semantic segmentation and depth estimation. †Gradient-based weight learning, * Equal weights.

Method	Segmentation (Higher Better)		Depth error (Lower Better)	
	IoU	mIoU	Abs.	Rel.
ST-Net (Ours)	81.52	43.27	0.1731	1.3843
STAN [16]	78.87	35.82	0.1926	1.4721
Dense [9]	78.92	35.92	0.1746	1.3936
Cross-Stitch [17]	78.69	34.83	0.1794	1.504
MTAN [16]	79.02	35.98	0.1867	**1.3072**
AdaMT-Net (Ours)∗	81.82	43.66	0.1698	1.3595
AdaMT-Net (Ours)†	**83.04**	**45.34**	**0.1642**	1.3676

Table 4. Multi-task validation results on ISPRS dataset for 6-class semantic segmentation and depth estimation. †Gradient-based weight learning, * Equal weights.

The results for three tasks (semantic segmentation, depth estimation, and surface normal prediction) on the NYUv2 dataset are shown in Table 1. Compared to Cross-Stitch, Dense, and MTAN networks, AdaMT-Net outperforms all the other methods at least by $+8.33\%$ in IoU, $+18.98\%$ in mIoU, -4.9% in relative error, -13.33% and -15.97% in mean and median angle distances, $+12.83\%$, $+11.58\%$, and $+9.7\%$ in the percentage of pixels within $11.25°$, $12.5°$, and $30°$ respectively. From another perspective, we consider the semantic segmentation and the depth estimation as the two tasks for the model evaluation on NYUv2. Table 2 confirms that our model tops in three out of four metrics among all the comparative techniques in this regard. Further, while comparing Table 1 and Table 2, we can observe that the performance on segmentation improves when three tasks are considered. Similar trends can be observed for both the CityScapes (Table 3) and ISPRS (Table 4) datasets. In all the cases, we observe that the multi-task performance exceeds that of the single-task networks (ST-Net), besides the fact that the adaptive weight learning outperforms the manual weighting scheme.

4.5. Critical analysis

Consideration of an extra self-supervised task: To further assess the scalability of AdaMT-Net, we consider a scenario with four tasks for NYUv2. Apart from the three tasks of segmentation, depth estimation, and surface normal prediction, we consider the auxiliary self-supervised task of image reconstruction. For the reconstruction task, we use PSNR as the performance metric. The rationale behind including a self-supervised task is that such an auxiliary task helps in modeling a better latent space by highlighting important image properties that may be overlooked by the different supervised inference tasks. We compare the performance of the 4-task setup with the MTAN [16] model trained with these four tasks. From Table 5, we can observe sharp improvements in the performance of AdaMT-Net with respect to MTAN for all the tasks. This clearly establishes the robustness of AdaMT-Net in terms of extracting more focused feature as the number of tasks grows.

Ablation on network design: Table 6 emphasizes the specific architecture choice for AdaMT-Net with respect to a number of baselines. In this respect, we consider vanilla

Method	Segmentation		Depth error		Surface Normal					Reconstruction
	(Higher)		(Lower)		Angle Distance (Lower)		Within t° (Higher)			(Higher)
	IoU	mIoU	Absolute	Relative	Mean	Median	11.25	22.5	30	PSNR
MTAN [16]	53.38	16.66	**0.6173**	0.2692	32.56	26.39	23.02	44.17	55.52	23.25
AdaMT-Net (Ours)†	**59.22**	**20.93**	0.6519	**0.2618**	**31.69**	**25.13**	**23.16**	**45.80**	**57.61**	**30.68**

Table 5. AdaMT-Net's 4-tasks performance on NYUv2 dataset for 13-class semantic segmentation, depth estimation, surface normal prediction, and reconstruction. † Gradient-based weighting.

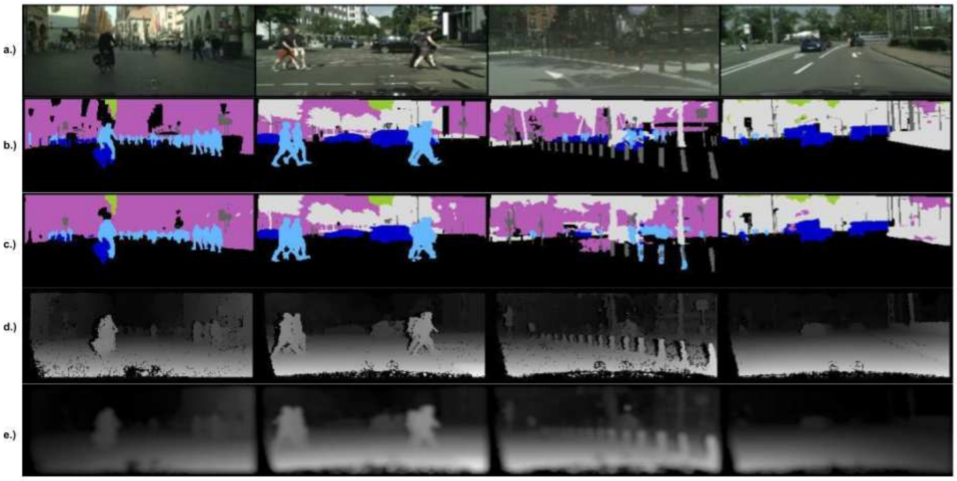

Figure 3. Qualitative results of semantic segmentation and depth estimation on CityScapes dataset (7 categories). From top to bottom: a.) Input image, b.) Semantic true, c.) Semantic predicted, d.) Depth true, and e.) Depth predicted.

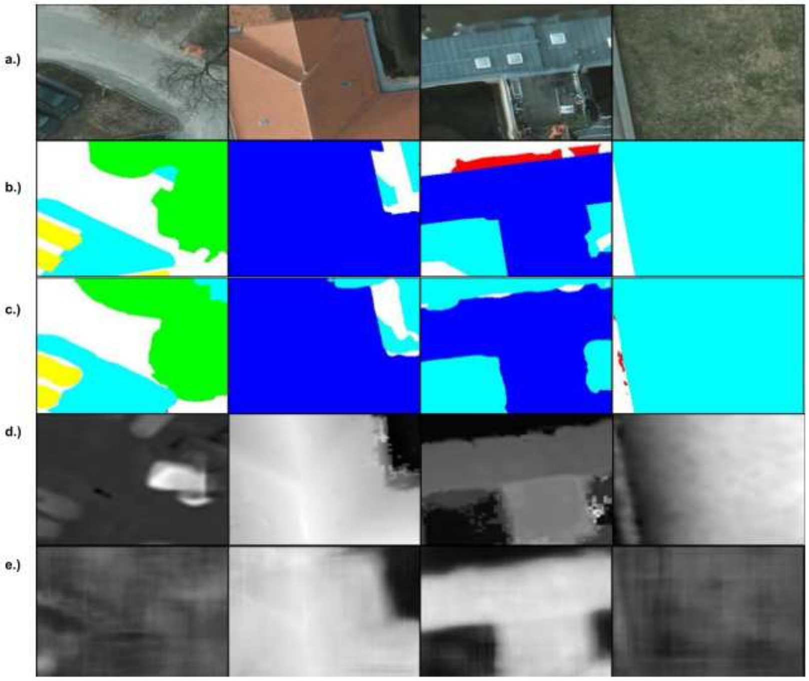

Figure 4. Qualitative results of semantic segmentation and depth estimation on ISPRS dataset (6 semantic classes). From top to bottom: a.) Input image, b.) Semantic true, c.) Semantic predicted, d.) Depth true, and e.) Depth predicted.

U-Net based MTL model, U-Net equipped with dense encoder but without attention learning in the decoder, and the full AdaMT-Net model, respectively. As can be seen, the performance gradually increases from vanilla U-Net to full

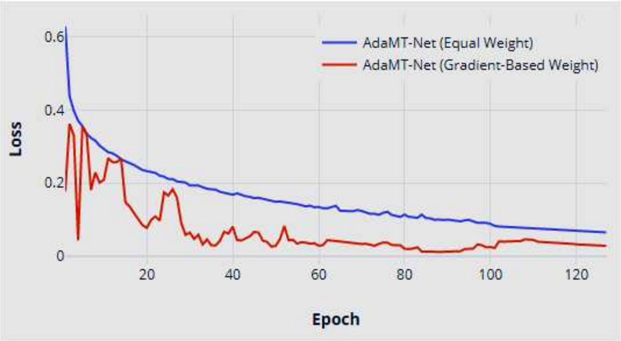

Figure 5. The evolution of the loss during training for i) AdaMT-Net with proposed weight learning of Equation 5, ii) with equal weight to the loss terms for the CityScapes dataset.

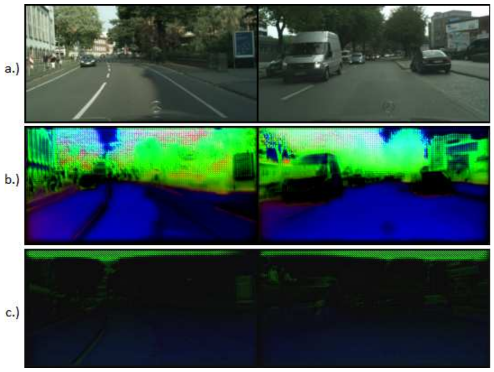

Figure 6. The attention masks learned by the task-specific decoders on CityScapes dataset. From top to bottom: a.) Input image, b.) Semantic attention mask, and c.) Depth attention mask.

Method	Segmentation (Higher Better)		Depth error (Lower Better)	
	IoU	mIoU	Abs.	Rel.
Simple U-Net [19]	91.27	53.55	0.0196	26.19
Dense encoder without decoder attention	92.07	55.67	0.0187	24.67
AdaMT-Net (Ours)*	94.01	61.91	0.0129	23.82
AdaMT-Net (Ours)†	**94.16**	**62.53**	**0.0125**	**22.23**

Table 6. Ablation on the proposed network design of AdaMT-Net for CityScapes dataset. $*$ Equal weight, †Gradient-based weight learning.

AdaMT-Net in all the metrics. Moreover, the performance on AdaMT-Net on both the manual weighting and adaptive weight learning schemes outperform the other baselines.

Comparison of different weight learning techniques: In Figure 5, we show the evolution of training for two models based on the AdaMT-Net: i) model trained with our weight learning scheme, and ii) model with equal weight to all the loss terms where a weight value of 1 is considered. It can be seen that the proposed weight learning scheme induces lower empirical loss than the manual weighting scheme. Besides, it can be observed that the convergence is obtained rapidly with the proposed weight learning scheme. Additionally, we compare the proposed weight learning strategy with a number of very recent weight learning schemes used in MTL in Table 7. In this regard, we compare our gradient-based weight learning with Bayesian uncertainty based weighting [11] and loss value-based weighting [16], respectively, where we train our model with these different weight learning strategies for the CityScapes dataset. It is found that the proposed weight learning produces the best performance in this respect.

Weight Type	Segmentation (Higher Better)		Depth error (Lower Better)	
	IoU	mIoU	Abs.	Rel.
Equal Weights	91.11	53.04	0.0144	33.63
Weights Uncertainty [11]	91.10	53.86	0.0144	35.72
DWA [16]	91.09	53.29	0.0144	34.14
Gradient-based Weights	**94.16**	**62.53**	**0.0125**	**22.23**

Table 7. Results comparison between different weighting scheme and our proposed gradient-based scheme for the CityScapes dataset.

Visualization: Figure 3 and 4 depict sample segmentation and depth estimation outputs for the CityScapes and the ISPRS datasets. The qualitative analysis confirms that the predicted segmentation and depth maps largely resemble the ground-truth maps. For example, in Figure 3, the person in different depth are found to be well segmented. A similar pattern can be observed for the ISPRS dataset.

Although, by design, the decoders are the same, yet they learn different attention masks for different tasks. Figure 6 shows the semantic and depth attention masks learned by the task-specific decoders on the CityScapes dataset. It is clear from the figure that the attention masks for different decoders focus on very different scene characteristics, which establishes the fact that learning task-specific attention in the decoders induces minimal redundancy.

5. Conclusions

In this paper, we propose a novel U-Net based multi-task learning framework called AdaMT-Net for jointly carrying out multiple visual inference tasks from monocular images. Our model efficiently combines the notion of task-generic and task-specific feature learning. For this purpose, while we follow a dense-block convolution architecture model in the shared feature encoder, separate attention-driven task-specific decoder modules are deployed to perform the individual tasks. Additionally, we introduce a novel loss gradient-based weight learning scheme for the individual loss-terms. Our experimental results show sharp improvements on three benchmark datasets, both quantitatively and qualitatively. As a future endeavor, we plan to scale-up AdaMT-Net to several tasks and incorporate the notion of task-clustering for quicker model training.

References

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

[2] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 42–50. ACM, 2011.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[7] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[8] Adam W Harley, Konstantinos G Derpanis, and Iasonas Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5038–5047, 2017.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*, pages 745–752, 2009.

[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012.

[16] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[17] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, Matthew C. H. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999, 2018.

[19] 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.

[20] Franz Rottensteiner, Gunho Sohn, Markus Gerke, Jan Dirk Wegner, Uwe Breitkopf, and Jaewook Jung. Results of the isprs benchmark on urban object detection and 3d building reconstruction. *ISPRS journal of photogrammetry and remote sensing*, 93:256–271, 2014.

[21] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[22] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. *CoRR*, abs/1807.06521, 2018.

[23] Chao Yuan. Multi-task learning for bayesian matrix factorization. In *2011 IEEE 11th International Conference on Data Mining*, pages 924–931. IEEE, 2011.

[24] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[25] Jun Zhu, Ning Chen, and Eric P Xing. Infinite latent svm for classification and multi-task learning. In *Advances in neural information processing systems*, pages 1620–1628, 2011.