SD-MTCNN: Self-Distilled Multi-Task CNN

Ankit Jha[1]
ankitjha16@gmail.com
Awanish Kumar[1]
awanishk389@gmail.com
Biplab Banerjee[1]
getbiplab@gmail.com
Vinay Namboodiri[2]
vinaypn@iitk.ac.in

[1] Indian Institute of Technology, Bombay
[2] Indian Institute of Technology, Kanpur

Abstract

Multi-task learning (MTL) using convolutional neural networks (CNN) deals with training the network for multiple correlated tasks in concert. For accuracy-critical applications, there are endeavors to boost the model performance by resorting to a deeper network, which also increases the model complexity. However, such burdensome models are difficult to be deployed on mobile or edge devices. To ensure a trade-off between performance and complexity of CNNs in the context of MTL, we introduce the novel paradigm of self-distillation within the network. Different from traditional knowledge distillation (KD), which trains the Student in accordance with a cumbersome Teacher, our self-distilled multi-task CNN model: SD-MTCNN aims at distilling knowledge from deeper CNN layers into the shallow layers. Precisely, we follow a hard-sharing based MTL setup where all the tasks share a generic feature-encoder on top of which separate task-specific decoders are enacted. Under this premise, SD-MTCNN distills the more abstract features from the decoders to the encoded feature space, which guarantees improved multi-task performance from different parts of the network. We validate SD-MTCNN on three benchmark datasets: CityScapes, NYUv2, and Mini-Taskonomy, and results confirm the improved generalization capability of self-distilled multi-task CNNs in comparison to the literature and baselines.

1 Introduction

The deep convolutional networks (CNN) have showcased superlative performance for a wide range of computer vision tasks, thanks to their ability towards learning a data-driven feature hierarchy. By and large, the CNNs designed to handle individual tasks require the availability of an ample amount of labeled training examples for training [14]. However, obtaining annotations is non-trivial in many resource-constrained scenarios. To this end, several notable endeavors exist to tackle the problems due to the dearth of annotations, including transfer learning, active learning, and multi-task learning [19, 29].

Multi-task learning (MTL) accounts for jointly training semantically related tasks within a unified framework so that the tasks can cooperate among themselves to enhance the overall learning. This can be regarded as more effectual not only in terms of memory and response time, but MTL also assists in getting rid of the label shortage problem since the tasks can collectively regularize each other. Amongst the variety of applications [4, 12], MTL is especially of interest in computer vision given the mutual dependence of several inference tasks like segmentation, depth-estimation, and many more. Nonetheless, the key issue in framing out deep CNNs for MTL arises from the design protocol to be adhered to. Amidst multiple possibilities, the notion of hard-sharing based approaches provides reasonable insights where an encoder-decoder based framework is broadly followed. The encoder is shared among the tasks, whereas the decoders are designed for each task autonomously, thus disentangling the generic to specific feature learning [17].

From a different point of view, any deep learning system's performance is expected to increase with expanding model complexity in terms of network depth or width. This severely hinders the deployment of CNN models for resource-constrained mobile or edge devices, which instigates research in the area of deep model compression without permitting significant performance drop. Amid different prospects, the notion of distilling the inherent knowledge of a very deep CNN model into a shallow network: knowledge distillation (KD), has demonstrated immense potential [9]. This leads to the construction of shallow CNN models (Student) that possess generalization proficiency at par with very deep models (Teacher) but offer extensive deployment scenarios due to their less-complex design.

Despite its success for single-task driven CNN models, the paradigm of KD suffers from two inherent bottlenecks: i) it is ideally incomprehensible to distill any Teacher to any Student, i.e., finding the optimal Student network for a given Teacher is a tedious job, ii) since the Student tries to mimic the final predictions of the Teacher, obtaining a distinguished Student model which can overtake the Teacher is practically occasional. In order to combat these issues, the idea of self-distillation [28] has recently emerged, which aims at distilling knowledge from the deeper layers which are supposed to contain more high-level features into the shallow network layers through an end-to-end training strategy. The resultant model not only requires less training time but can accomplish high accuracy. Although this paradigm clearly shows benefits for single-task CNN models, deployment of this idea for multi-task CNN is non-trivial given that the overall loss function for MTL combines variants of classification and regression loss measures the unique design strategies usually followed for the multi-task CNN models. As a result, *what to distill* and *between which pair of layers* designate the two critical aspects for inducing self-distillation in MTL.

Based on the above deliberations, we are convinced about the possibility of self-distilling CNN models for MTL to be utilized in conjunction with low-latency devices, besides enhancing the overall model's performance. In this line, we introduce the novel notion of Self-Distilled Multi-Task CNN (SD-MTCNN) for multi-view visual scene understanding using hard-sharing based multi-task CNNs. Specifically, we majorly solve the problems of semantic image segmentation, depth-estimation, and surface-normal prediction from monocular images. SD-MTCNN distills the features from the task-specific decoders to the task-generic encoder-space within a unified training regime. The full network with both the encoder and decoder layers acts as the Teacher, while the sub-network consisting of the shared encoder and direct output branches specifies the Student model. Hence, if high-level task-oriented features from the Teacher can influence the learning of Student, the network layers can, in turn, learn more discriminative features for the tasks, thus revamping the overall knowledge embedded in the model. Since the Student is expected to mimic the teacher's performance at optimality, either of the models can well be utilized during inference. We introduce two training strategies for SD-MTCNN where i) one considers the Teacher/Student loss and the distillation loss, and ii) the other considering an ensemble of the Teacher and Student together with the distillation loss. Below, we summarize the novel contributions of this work:

- We introduce the novel paradigm of self-distillation for multi-task CNN and propose a novel framework called SD-MTCNN. Ours is the first method to employ self-distillation for MTL, to the best of knowledge.

- SD-MTCNN aims at distilling knowledge from the task-specific decoder modules to the task-generic sub-network, leading to a more discriminative feature embedding for the tasks. In this regard, our backbone encoder is defined in terms of standard CNN, while the decoders are equipped with task-specific attention learning. We validate our model on both Segnet [1] and U-Net [21] based architectures.

- We perform rigorous experiments on the benchmark CityScapes [5], NYUv2 [23], and Mini-Taskonomy [27] datasets for up to five concurrent tasks and find the self-distillation paradigm improves the MTL performance by a substantial margin.

2 Related Works

Multi-task learning: MTL [2, 7, 15] has been regarded as one of the cost-effective solutions for executing several related tasks together. Consequently, MTL aims to improve the learning capabilities for each of the tasks by judiciously exploring the complementary and common information available in all of them. Before the inception of deep learning, MTL was largely dealt with by utilizing traditional feature transformation based approaches such as latent support vector machine (SVM) [30], Bayesian matrix factorization [25], task clustering [11], matrix decomposition [3] and many more.

Lately, the traditional ad-hoc approaches are succeeded by deep learning techniques given their prominence in performing data-driven feature learning. MTL approaches developed in conjunction with the deep CNN models have successfully been implemented for solving several related visual perception tasks together [6]. From the architecture point of view, the feature extractor can either be soft or hard-shared among the tasks. While in soft-sharing, separate feature extractors are considered for the tasks with certain constraints on the network parameters. The hard-sharing-based approaches utilize a common encoder for all the tasks [13, 18]. For the soft-sharing based models, accurate segregation of the task-specific features from the generic feature space plays a crucial role. The usage of task-specific attention learning is helpful in this aspect [17].

Knowledge distillation: Inspired by the notion of Teacher-Student modeling for pairwise knowledge transfer, the knowledge distillation (KD) paradigm [9] is regarded as one of the promising remedies for deep model compression. The central idea behind KD is to simulate an over-parameterized Teacher model into a condensed Student network. The Student model is further capable of exploiting the knowledge encapsulated in the Teacher and doing so, high compression and acceleration can be achieved without compromising much on the accuracy front. There are various endeavors proposed to excel the potential of Student's learning such as attention transfer [26], hint learning in FitNet [20], generative adversarial learning for KD [16, 22], better test data generalization [8], to name a few. To circumvent the need to train separate Teacher and Student models, recently [28] introduces the idea of self-distillation with the network. Ideally, self-distillation mainly concentrates on boosting the model performance rather than model compression and acceleration by discarding the consideration of any external Teacher model.

As expected, handling multiple tasks within the realm of a single network is onerous than the typical single-task learning frameworks. This is mainly due to the specialized model architecture followed for MTL and the combination of heterogeneous loss functions, something the self-distillation strategy of [28] which solves a classification problem, cannot explicitly support. In contrast, SD-MTCNN is designed specifically keeping the multi-task learning setup in mind and follows a more streamlined approach to combine the loss measures, apart from augmenting the generic feature space by high-level task information.

3 Formulation of SD-MTCNN

The objective is to develop a multi-task CNN which incorporates the notion of knowledge distillation within the network. In this regard, we consider a typical encoder-decoder based CNN architecture for MTL (Segnet or U-Net) where the encoder is shared among the tasks on top of which separate task-specific decoders are placed. Decoders are further supplemented with attention learning modules to ensure the extraction of diligent task-oriented features. As already mentioned, the Teacher is represented by the entire network while the sub-network up to the shared-encoder specifies the Student. Hence, the Teacher follows a symmetric architecture where the same number of layers are considered both in the encoder and decoders. While for the Student, task-specific output branches are generated from the encoder bottleneck through the application of deconvolution. In a way, the Student can be considered as a naive network model where multiple loss measures are defined from the same feature embedding space (Figure 1). We aim at distilling the high-level features from the deepest decoder layers of the Teacher into the penultimate layer of the Student, e.g., the final shared-encoder layer. Through iterative training, this will simultaneously enhance the representation abilities of both the shared feature space as well as the decoders. Besides uplifting the Teacher's performance, the student's accuracy, which is generally poor

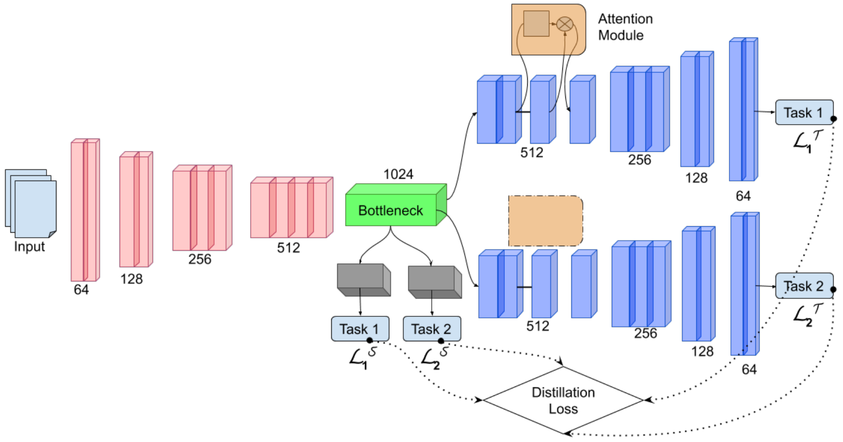
Attention Module
512
256
128
64
Task 1
$\mathcal{L}_1^{\mathcal{T}}$
1024
Bottleneck
Input
64
128
256
512
Task 1
Task 2
$\mathcal{L}_1^{\mathcal{S}}$
$\mathcal{L}_2^{\mathcal{S}}$
512
256
128
64
Task 2
$\mathcal{L}_2^{\mathcal{T}}$
Distillation Loss

Figure 1: The proposed encoder-decoder based SD-MTCNN model. (Pink) The shared-encoder, (Green) The encoder bottleneck, (Blue) The decoders for the Teacher, (Grey) The decoders for the Student, and (Orange) The attention module. As a whole the Teacher is represented by the Pink, Green, and the Blue parts of the model. On the other hand, the Student is defined in terms of the Pink, Green, and the Grey parts.

compared to the Teacher due to its shared nature, is expected to be enhanced considerably. Once trained, the Student or the Teacher can be deployed into different environments during inference as per the requirement.

Formally, let us consider a multi-task 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 majority of our experiments, we consider three dense structured prediction tasks: semantic segmentation, depth estimation, and surface-normal prediction, respectively, from the input image. Further, the outputs corresponding to the Teacher and Student for the t^{th}-task are referred to as y_T^t and y_S^t for the sake of clarity. The encoder network is denoted as $f_E(;,\theta_E)$ with parameters θ_E. On the other hand, the decoder networks for the Teacher can be denoted by $f_t^{TD}(;,\theta_t^{TD})$ with task-specific parameters θ_t^{TD} while we use $f_t^{SD}(;,\theta_t^{SD})$ with parameters θ_t^{SD} to denote the task-specific output branches of the Student. The predictions of the network for the t^{th}-task can be put forward as,

$$\hat{x} = f^E(x, \theta^E),\ \hat{y}_S^t = f_t^{SD}(\hat{x}, \theta_t^{SD}),\ \hat{y}_T^t = f_t^{TD}(\hat{x}, \theta_t^{TD}) \tag{1}$$

3.1 Loss functions used for the tasks

We note that while semantic segmentation is regarded as a supervised classification task, depth estimation and surface-normal prediction are envisaged as regression tasks in general. In the same line as of the literature [17], we follow i) a cross-entropy based loss for segmentation, ii) ℓ_1-distance based loss for depth estimation, and iii) element-wise dot-product between the ground-truth and the model predictions for surface-normal evaluation, respectively. In this regard, the losses are to be evaluated at both the Teacher and Student ends. For brevity, we use $\mathcal{L}_t^S$ and $\mathcal{L}_t^T$ to denote the loss incurred for the Student and the Teacher for the t^{th} task where $t = 1, 2$ and 3 for segmentation, depth perception, and surface-normal prediction. The loss functions are reported in the following for $m \in \{S, T\}$,

$$\left.\begin{aligned}
\mathcal{L}_1^m(y^1, \hat{y}_m^1) &= \mathbf{E}[-\frac{1}{HW} \Sigma_{j=0}^{H-1} \Sigma_{k=0}^{W-1} \hat{y}^1(j,k) \log y_m^1(j,k)] \\
\mathcal{L}_2^m(y^2, \hat{y}_m^2) &= \mathbf{E}[\frac{1}{HW} \Sigma_{j=0}^{H-1} \Sigma_{k=0}^{W-1} \mid\mid y^2(j,k) - \hat{y}_m^2(j,k) \mid\mid_1^1] \\
\mathcal{L}_3^m(y^3, \hat{y}_m^3) &= \mathbf{E}[-\frac{1}{HW} \Sigma_{j=0}^{H-1} \Sigma_{k=0}^{W-1} y^3(j,k) \cdot \hat{y}_m^3(j,k)]
\end{aligned}\right\} \tag{2}$$

3.2 Self-distillation loss

The primary aim of the distillation loss is to equip the Student with the teacher model's capabilities. However, it is not straightforward to define the distillation loss in the MTL context since it includes both the classification and regression loss functions. In traditional KD, the distillation for the classification task is typically attained by training the Student using the soft-labels from the already trained Teacher. However, in our case, both the Student and the Teacher are concurrently trained, besides the fact that both the models share the feature extraction sub-network. Hence, we constrain the Student to follow the evolution of the teacher model's outputs over the iterations. As a result, we consider the Kullback-Leibler (KL) divergence between the Student and the Teacher's outputs as the distillation loss for the segmentation task,

$$\mathcal{L}_{KD}^1 = \mathbf{E}[KL(\hat{y}_S^1, \hat{y}_T^1)] \tag{3}$$

In Equation 3, it is assumed that the outputs of both models are interpreted as softmax scores. In the same line, we resort to the ℓ_2-distance as the distillation loss measures corresponding to depth estimation and surface-normal prediction, respectively, as follows.

$$\mathcal{L}^2_{KD} = \mathbf{E}[|\hat{y}^2_S - \hat{y}^2_T|_2] + \mathbf{E}[|\hat{y}^3_S - \hat{y}^3_T|_2] \tag{4}$$

3.3 Overall loss functions

We propose two different overall loss functions for training the SD-MTCNN model. They are specified below,

- In the first approach, we train the model with the task-specific loss functions for the Teacher plus Student and the distillation losses of Equation 3-4 as below.

$$\underset{\theta^E,\theta^{SD},\theta^{TD}}{\operatorname{argmin}} \; \mathcal{L}^T_1 + \mathcal{L}^T_2 + \mathcal{L}^T_3 + \mathcal{L}^S_1 + \mathcal{L}^S_2 + \mathcal{L}^S_3 + \lambda(\mathcal{L}^1_{KD} + \mathcal{L}^2_{KD}) \tag{5}$$

  The Teacher network, being deeper, is expected to learn more high-level features than the Student given only the task-specific loss functions. On the other hand, the inclusion of the distillation loss measures in this respect directs the Student to learn features comparable to the Teacher. Since the representation ability of the shared feature encoder enhances due to the self-distillation paradigm, sharp performance improvements for both the Student and Teacher can be observed as a consequence. λ defines the weighting parameter.

- While the model of Equation 5 has separate task-specific losses for both the Student and Teacher, our second model is based on the idea of ensemble learning. Precisely, the task-specific loss functions are defined by aggregating the predictions of both the networks and then comparing them with the respective ground-truths. In this case, the overall loss corresponds to the summation of the new task-specific loss measures and the distillation loss. Let us consider $\tilde{y}^t = \frac{\hat{y}^t_T + \hat{y}^t_S}{2}$ for $t \in \{1,2,3\}$, then we define the cumulative cost function as where $\mathcal{L}_{1-3}$ are defined as per Equation 2,

$$\underset{\theta^E,\theta^{SD},\theta^{TD}}{\operatorname{argmin}} \; \mathcal{L}_1(y^1,\tilde{y}^1) + \mathcal{L}_2(y^2,\tilde{y}^2) + \mathcal{L}_3(y^3,\tilde{y}^3) + \lambda(\mathcal{L}^1_{KD} + \mathcal{L}^2_{KD}) \tag{6}$$

4 Experimental evaluations

We consider three benchmark datasets for evaluating SD-MTCNN. **NYUv2**: We consider the NYUv2 [23] dataset for the joint segmentation, surface normal prediction, and depth estimation tasks, respectively. This dataset consists of RGB-D indoor scene images from 13 semantic categories. It is challenging to handle primarily due to variations in camera viewpoint, scene occlusion, differences in lighting conditions, etc. **CityScapes**: The CityScapes [6] dataset contains high-resolution street-view images to be deployed for semantic segmentation and depth estimation. In this regard, we consider the standard 7 semantic classes for evaluating the segmentation performance of SD-MTCNN. **Mini Taskonomy**: The Taskonomy [27] dataset consists of over 4.5 million images for 26 visual inference tasks as a whole. However, we consider a subset of the dataset called Mini-Taskonomy [24] with five tasks: semantic segmentation (for 18 classes), depth estimation, surface-normal prediction, 2D keypoints estimator, and edge prediction, respectively.

4.1 Model architecture and training protocols

The architecture of SD-MTCNN consists of the encoder-decoder based Segnet model for all the datasets primarily. However, we show experiments with respect to the U-Net model as well. The shared feature encoder consists of four CNN blocks. As shown in Figure 1, the first and second blocks have two CNN layers each, while the third and fourth blocks have three and four CNN layers with a kernel size of 3×3 is considered for all the blocks. A max-pool layer follows each convolution block with a kernel stride of 2. To ensure stability during training, ReLU non-linearity and Batch-normalization are used after each block. In essence, the encoder module computes the feature maps of depth 64, 128, 256, 512, respectively. The bottleneck layer or the shared features space consists of 1024 feature-maps. With respect to the shared feature encoder, a symmetric architecture is followed for the decoder modules. In addition, the decoders' CNN blocks are equipped with attention learning modules, each making up two convolution layers with a kernel size of 3×3. Batch-normalization follows each convolution layer, and Sigmoid non-linearity is appended finally to ensure the mask's values to lie within the range $[0,1]$. In addition to this arrangement for the full SD-MTCNN model or the so-called Teacher network, the Student network is built by implementing separate deconvolutional layers to deduce the task-specific outputs directly from the bottleneck. The training is iterated for 200 epochs with an initial learning rate of $1e-4$ and using the Adam optimizer. A batch size of 2, 4, 8, are used for the NYUv2, Mini-Taskonomy, and CityScapes, respectively. The λ parameter of Equation 5 and 6 is set to 1 in all the experiments. Also, we note that equal weights are considered for the task-specific loss functions, given that our major focus is to showcase the efficacy of the self-distillation approach. We consider the standard evaluation metrics for all the tasks as detailed in [17].

4.2 Baselines and comparison to the literature

We initially validate the usage of the distillation loss measures for two single-task scenarios for NYUv2, namely, segmentation and depth perception. Ideally, the networks follow a single encoder and single decoder setup. They are trained with a combination of the respective task loss and the distillation loss (either the KL-loss or the ℓ_2-loss) where the Teacher/Student is defined in the same way as of SD-MTCNN. As per Table 1, sharp enhancements can be observed in all the metrics for both the Teacher and Student in comparison to the standalone encoder-decoder based single-task CNNs using both the traditional and ensemble-based training strategies of Equation 5 and 6. We also note that the teacher and student's performance are very much comparable for both the tasks. This confirms that the shared feature encoder indeed can inherit the discriminative characteristics from the deeper decoder layers.

Furthermore, we consider four baselines to compare our full SD-MTCNN models as follows, i) **Vanilla Segnet model**: We train a vanilla Segnet model, which mimics the structure of our Teacher (Baseline-1). ii) **MTL CNN without any distillation loss**: In this case, we train the network given the combined task-specific losses of both the Teacher and the Student but without the distillation losses (Baseline-2): we report results according to Equation 5 and 6 but setting $\lambda = 0$. iii) **Traditional knowledge distillation**: We follow the traditional KD approach where the Teacher model is trained first, and then a distillation stage is carried out to train the Student. Precisely, the Student is trained with a combination of the task-specific losses and the distillation losses (Baseline-3). We follow the architectures of the Teacher and Student for traditional KD in accordance with the SD-MTCNN model. iv) **SD-MTCNN without attention in the decoders**: To showcase the effectiveness of the task-specific attention learning at the decoders, we train both the models (Equation 5 and 6) but without any attention learning (Baseline-4). For Baseline-2 and 4, we report the performance of both the Teacher and the Student, while the Student's performance is mentioned for Baseline-3. In Table 2, we report the baseline analysis for the three-tasks setup of the NYUv2 dataset [1]. As can be observed, the proposed self-distillation strategies considerably improve the performance of both the Teacher and the student. Specifically, self-distillation performance outperforms that of the traditional distillation strategy for all the tasks without the burden of training two separate networks. Figure 2 shows the qualitative assessment on the segmentation and depth-estimation tasks of both the Teacher and Student as per Equation 5 with respect to Baseline-3 on CityScapes. We also highlight the efficacy of self-distillation in the U-Net model (Table 2), where the self-distillation strategies of Equation 5 and 6 sharply extend the performance of the vanilla U-Net.

Method	Segmentation ↑		Depth error ↓	
	IoU	mIoU	Abs.	Rel.
Single Task	49.84	15.35	0.7102	0.2863
SD-Single Task^T ($\star$)	**56.83**	**22.80**	**0.6211**	**0.2627**
SD-Single Task^S	56.48	22.22	0.6430	0.2755
SD-Single Task^T ($\star\star$)	**56.58**	**21.93**	**0.6361**	**0.2666**
SD-Single Task^S	56.39	21.66	0.6413	0.2697

Table 1: Validation of the distillation losses for classification and regression (only for depth-estimation) for single-task setups for NYUv2 dataset as per ($\star$) by Eq. 5 and ($\star\star$) by Eq. 6 and following the Segnet based models. ^{T} Teacher, ^{S} Student.

Table 3-5 depicts the comparative analysis of SD-MTCNN for CityScapes (two tasks), NYUv2 (both two and three tasks), and Mini-Taskonomy (five tasks), respectively, with respect to a number of recent techniques. While we find that the Teacher modules for both the models (Equation 5-6) sharply outperform the literature in most of the performance metrics, the performance of the respective Students is at par or even better than most of the considered approaches. This signifies the effectiveness of the self-distillation model in jointly promoting the efficiency of both the deep Teacher and shallow Student in parallel. Further, the model of Equation 5 is found to be better than the model of Equation 6.

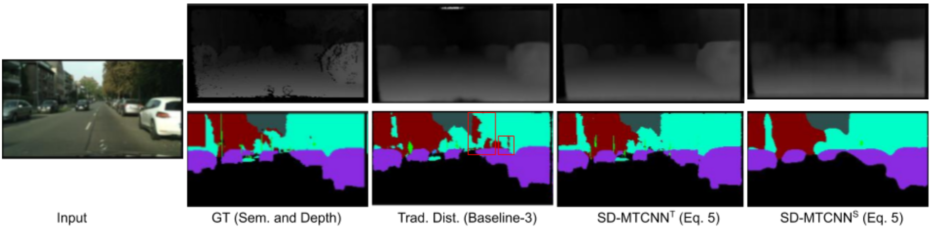

Figure 2: Semantic seg. and depth visualization on CityScapes dataset. From left to right: input, ground truth, Student network from Baseline-3, and the Teacher and Student of SD-MTCNN of Eq. 5, respectively. Wrong predictions shown by red boxes.

[1] Baseline analysis for CityScapes and Mini-Taskonomy are mentioned in the supplementary

Method	Segmentation		Depth error		Surface Normal				
	↑		↓		Angle Distance ↓		Within t° ↑		
	IoU	mIoU	Abs.	Rel.	Mean	Median	11.25	22.5	30
Vanilla Segnet†	51.88	15.59	0.6177	0.2511	32.08	26.93	21.16	43.01	55.00
SD-Vanilla Seg.^{T} ($\star$)‡	56.14	20.45	0.5922	0.2510	30.45	25.92	22.55	45.09	57.26
SD-Vanilla Seg.^{S}	53.63	16.60	0.6374	0.2822	33.70	29.37	16.10	38.22	51.24
SD-Vanilla Seg.^{T} ($\star\star$)‡	55.88	17.54	0.5874	0.2416	33.02	28.95	14.44	38.05	52.07
SD-Vanilla Seg.^{S}	26.71	7.92	1.3621	0.5582	33.47	29.10	16.24	38.74	51.70
Trad. K.D.∓	52.55	15.54	0.6136	0.2542	31.94	26.47	21.96	43.72	55.48
SD-MTCNN^T ($\star$)#	55.56	21.48	0.6019	0.2597	29.92	25.32	21.50	45.27	58.01
SD-MTCNN^S	55.27	21.31	0.6114	0.2612	31.61	27.65	18.04	41.06	54.13
SD-MTCNN^T ($\star\star$)#	55.23	21.11	0.6057	0.2597	31.34	26.68	18.57	42.63	55.76
SD-MTCNN^S	55.01	20.87	0.6213	0.2996	31.95	27.65	17.88	40.35	52.44
SD-MTCNN^T ($\star$)$full$	**56.90**	**22.44**	**0.5857**	0.2483	**29.66**	**25.02**	**22.66**	**45.84**	**58.39**
SD-MTCNN^S	56.43	21.90	0.5986	0.2532	30.76	26.98	19.52	44.20	57.32
SD-MTCNN^T ($\star\star$)$full$	**56.61**	22.31	**0.5864**	**0.2458**	**30.04**	**25.24**	21.36	**45.40**	**58.20**
SD-MTCNN^S	56.11	**22.59**	0.5899	0.2506	30.42	25.63	21.22	44.79	57.21
Ablation on U-Net									
U-Net	60.15	22.47	0.6871	**0.2599**	29.93	25.73	22.45	46.28	60.12
U-Net^T ($\star$)$full$	**61.34**	**23.91**	**0.6792**	0.2610	**28.51**	**24.05**	**23.67**	**47.36**	**60.56**
U-Net^S	60.16	22.84	0.7188	0.2935	30.87	26.88	19.23	45.05	58.59
U-Net^T ($\star\star$)$full$	**61.04**	**23.30**	**0.6808**	0.2664	**29.38**	**25.22**	**23.07**	**46.77**	**60.35**
U-Net^S	60.03	22.72	0.7110	0.2984	30.76	26.11	19.91	45.31	57.79

Table 2: 3-task validation results on the NYUv2 dataset for 13-class semantic seg., depth, and surface normal prediction with various baselines on Segnet based models and ablation analysis on U-Net model. ^{T} Teacher, ^{S} Student. † Baseline-1, ‡ Baseline-2, ∓ Baseline-3, # Baseline-4, ($\star$) by Eq. 5, ($\star\star$) by Eq. 6. We compare SD-MTCNN($\star$) against all the ($\star$) baselines and similar for ($\star\star$).

Method	Segmentation		Depth error		Surface Normal				
	↑		↓		Angle Distance ↓		Within t° ↑		
	IoU	mIoU	Abs.	Rel.	Mean	Median	11.25	22.5	30
Single Task	49.84	15.35	0.7102	0.2863	32.49	28.81	20.47	42.14	52.39
Vanilla Segnet†	51.88	15.59	0.6177	0.2511	32.08	26.93	21.16	43.01	55.00
Dense [□]	52.73	16.06	0.6488	0.2871	33.58	28.01	20.07	41.50	53.35
Cross-Stitch [□]	52.73	14.71	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
Split (Wide) [□]	51.19	15.89	0.6494	0.2804	33.69	28.91	18.54	39.91	52.02
Split (Deep)	41.17	13.03	0.7836	0.3326	38.28	36.55	9.50	27.11	39.63
MTAN [□]	55.32	17.72	0.5906	0.2577	31.44	25.37	**23.17**	45.65	57.48
SD-MTCNN^T ($\star$)$full$	**56.90**	**22.44**	**0.5857**	**0.2483**	**29.66**	**25.02**	22.66	**45.84**	**58.39**
SD-MTCNN^S	**56.43**	**21.90**	0.5986	0.2532	**30.76**	26.98	19.52	44.20	57.32
SD-MTCNN^T ($\star\star$)$full$	**56.61**	**22.31**	**0.5864**	**0.2458**	**30.04**	**25.24**	21.36	45.40	**58.20**
SD-MTCNN^S	**56.11**	**22.59**	**0.5899**	**0.2506**	**30.42**	25.63	21.22	44.79	57.21

Table 3: 3-task task validation results on the NYUv2 dataset (13-class) semantic seg., depth and surface normal prediction on Segnet based models. ^{T} Teacher, ^{S} Student. † Baseline-1, ($\star$) by Eq. 5, ($\star\star$) by Eq. 6. We highlight the Teacher and Student whichever outperforms the above-referred literature.

5 Conclusions

We introduce the novel paradigm of self-distillation within a network for the purpose of multi-task learning. Given the hard-sharing based multi-task CNN architecture, we specifically aim at distilling knowledge from the task-specific decoders to the shared-encoder. Two training strategies are followed to train the network end-to-end where one consists of a combination of the task-specific losses of Teacher-Student together with the proposed distillation losses for the tasks, whereas the other defines an ensemble-based learning strategy for the task-specific losses along with the distillation losses. Experimentally, we find that the proposed self-distillation outperforms the traditional distillation strategies and many benchmark MTL approaches. We are currently interested in performing a more principle continuous

Method	Segmentation ↑		Depth error ↓		SN ↑	Key ↓	Edge ↓
	IoU	mIoU	Abs.	Rel.	CS	Abs.	Abs.
Single Task	86.93	43.45	0.0340	0.4979	0.8657	0.0172	0.0190
Vanilla Segnet†	89.04	50.38	0.0336	0.3607	0.8805	0.0153	0.0103
Dense [10]	87.93	44.76	0.0379	0.6329	0.8677	0.0459	0.0329
Cross-Stitch [18]	86.87	35.66	0.0375	0.5481	0.8601	0.0488	0.0396
Split (Wide) [17]	88.72	49.26	0.0447	0.5125	0.8631	0.0455	0.0411
Split (Deep)	88.89	47.52	0.0453	0.4909	0.8674	0.0408	0.0390
MTAN [17]	88.38	48.91	0.0361	0.6440	0.8707	0.0481	0.0211
SD-MTCNN^T $(\star)\,full$	**89.36**	**55.36**	**0.0290**	**0.3572**	**0.9058**	**0.0138**	**0.0097**
SD-MTCNN^S	**89.02**	**54.78**	0.0370	**0.4010**	0.8608	0.0327	**0.0118**
SD-MTCNN^T $(\star\star)\,full$	**89.21**	**55.26**	**0.0327**	**0.4318**	**0.8907**	**0.0147**	**0.0101**
SD-MTCNN^S	88.93	**53.68**	0.0419	0.5221	0.8616	0.0340	**0.0129**

Table 4: 5-task validation results on the Mini-Taskomomy dataset for semantic seg., depth, surface normal, key-points, and edge on Segnet based models. ^{T} Teacher, ^{S} Student. ($\star$) by Eq. 5, ($\star\star$) by Eq. 6. We highlight the Teacher and Student whichever outperforms the above-referred literature.

Dataset	NYUv2				CityScapes			
Method	Segmentation ↑		Depth error ↓		Segmentation ↑		Depth error ↓	
	IoU	mIoU	Abs.	Rel.	IoU	mIoU	Abs.	Rel.
Single Task	49.84	15.35	0.7102	0.2863	88.39	48.19	0.0167	33.52
Vanilla Segnet†	50.64	14.90	0.6244	0.2612	89.73	49.71	0.0161	35.91
Dense [10]	55.59	17.22	0.6002	0.2654	90.89	51.91	0.0138	**27.21**
Cross-Stitch [18]	53.99	17.01	0.6095	0.2671	90.33	50.08	0.0154	34.49
Split (Wide) [17]	55.83	18.13	0.6126	0.2584	90.63	50.17	0.0167	44.73
Split (Deep)	46.39	13.40	0.7321	0.3057	88.69	49.85	0.0180	43.86
MTAN [17]	56.24	18.32	0.5931	0.2562	91.11	53.04	0.0144	33.63
SD-MTCNN^T $(\star)\,full$	**57.18**	**23.01**	**0.5847**	**0.2466**	**92.54**	**56.70**	**0.0131**	27.68
SD-MTCNN^S	55.80	**23.19**	0.6033	0.2588	**91.60**	**54.09**	0.0150	33.23
SD-MTCNN^T $(\star\star)\,full$	**56.29**	**22.63**	0.6042	**0.2544**	**91.57**	**54.63**	**0.0134**	31.11
SD-MTCNN^S	56.09	**22.50**	0.6103	0.2674	90.99	**53.27**	0.0151	34.82

Table 5: 2-task validation results on the NYUv2 and CityScapes datasets for 13 and 7 class semantic seg. and depth estimation on Segnet based models. ^{T} Teacher, ^{S} Student. † Baseline-1, ($\star$) by Eq. 5, ($\star\star$) by Eq. 6. We highlight the Teacher and Student whichever outperforms the above-referred literature.

self-distillation by introducing a number of pseudo-Teachers from intermediate decoders between the Student and the Teacher.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. URL http://arxiv.org/abs/1511.00561.

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

[3] 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.

[4] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[5] 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.

[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] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks, 2018.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[10] 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.

[11] 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.

[12] 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.

[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] Peiye Liu, Wu Liu, Huadong Ma, Tao Mei, and Mingoo Seok. KTAN: knowledge transfer adversarial network. *CoRR*, abs/1810.08126, 2018.

[17] 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.

[18] 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.

[19] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[20] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2014.

[21] 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.

[22] Zhiqiang Shen, Zhankui He, and Xiangyang Xue. MEAL: multi-model ensemble via adversarial learning. *CoRR*, abs/1812.02425, 2018. URL http://arxiv.org/abs/1812.02425.

[23] 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.

[24] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv preprint arXiv:1905.07553*, 2019.

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

[26] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *CoRR*, abs/1612.03928, 2016. URL http://arxiv.org/abs/1612.03928.

[27] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[28] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *CoRR*, abs/1905.08094, 2019. URL http://arxiv.org/abs/1905.08094.

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

[30] 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.