Medusa: Universal Feature Learning via Attentional Multitasking

Jaime Spencer, Richard Bowden, Simon Hadfield
Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey
{jaime.spencer, r.bowden, s.hadfield}@surrey.ac.uk

Abstract

Recent approaches to multi-task learning (MTL) have focused on modelling connections between tasks at the decoder level. This leads to a tight coupling between tasks, which need retraining if a new task is inserted or removed. We argue that MTL is a stepping stone towards universal feature learning *(UFL), which is the ability to learn generic features that can be applied to new tasks without retraining.*

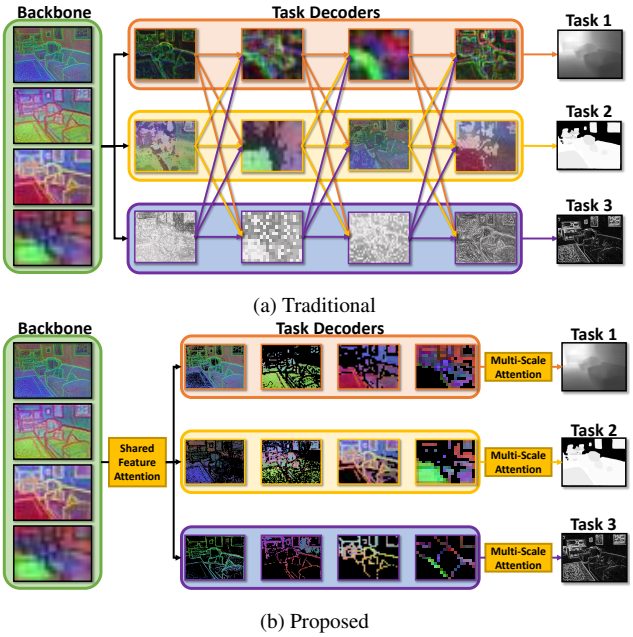
We propose Medusa *to realize this goal, designing task heads with dual attention mechanisms. The shared feature attention masks relevant backbone features for each task, allowing it to learn a generic representation. Meanwhile, a novel Multi-Scale Attention head allows the network to better combine per-task features from different scales when making the final prediction. We show the effectiveness of* Medusa *in UFL (+13.18% improvement), while maintaining MTL performance and being 25% more efficient than previous approaches.*

1. Introduction

Classical approaches to computer vision relied on handcrafted heuristics and features that encapsulated what researchers believed would be useful for a given task. With the advent of deep learning, features have become part of the learning process, leading to representations that would have never been developed heuristically. Unfortunately, most deep learning systems learn features that perform well on only one target task. Even if pretrained features are used, these require finetuning. Works that explored generic features [13, 15, 35] have focused on invariance to illumination and viewpoint changes, with the objective of establishing geometric correspondences. Whilst this is a useful step in many applications, these features are not suitable for a wider range of tasks.

Meanwhile, there has been a recent surge in multi-task learning (MTL), since training a network to solve multiple tasks simultaneously can provide a performance increase over training each task independently [23, 26]. Nonetheless, these works have focused on maximizing accuracy, not generality. At their core, they still try to learn features that perform well on a specific subset of tasks. It is often difficult or impossible to include new tasks into a previously trained model. Moreover, modern approaches [42, 43] have such tight connections between tasks that it becomes impossible to evaluate a single task without all other training tasks, as illustrated in Figure 1. It is difficult to argue that such features are truly generic.

This paper addresses the problem of universal feature learning (UFL), where a system is capable of learning



(a) Traditional

(b) Proposed

Figure 1. **Transferable Feature Learning.** Current high-performing approaches to MTL rely on connections between every combination of tasks, leading to a quadratic parameter complexity w.r.t. the number of tasks. We maintain independent task heads, making it possible to easily add/remove new tasks *a posteriori*. The dual spatial attention mechanisms (SFA & MSA) allow us to maintain performance while scaling linearly and learning highly reusable feature representations. See Figure 2 for a full overview.

generic features useful for all tasks. As discussed, MTL is evaluated on the same set of tasks used during training, *i.e*. the training and evaluation tasks are identical. In contrast, UFL aims to generalize *beyond* this training set. In other words, the training and evaluation tasks are not the same. As such, the resulting representations produced by the backbone are referred to as *universal features*. It is worth noting that, in order to insert a new task into the network, the layers corresponding to the task head still need training. However, the focus of UFL is on learning backbone feature representations that are left frozen while adding these new task heads. This results in shorter and more efficient training, as well as avoiding catastrophic forgetting in the shared backbone features.

The method proposed in this paper, dubbed *Medusa*, aims to learn this universal representation. We design an architecture with completely independent task heads, where the only shared component is the backbone. Each task head retains only the specific subset of relevant backbone features via a spatial attention mechanism. This allows the backbone to learn generic features, while reducing the likelihood of *negative transfer* between tasks. The model then makes initial predictions at each backbone resolution, which are further combined in a novel Multi-Scale Attention (MSA) head. By feeding back diverse training tasks, we encourage the learned features to encode a wide variety of information across scales. Furthermore, independent task heads result in an efficient feature extraction process that utilizes significantly less resources but maintains competitive performance, while having a flexible architecture where new task heads can be easily added.

Our contributions are summarized as:

1. We highlight the importance of *universal feature learning* in contrast to MTL. The main objective behind this is to learn a universal language for computer vision applications. This requires a system to learn features that require no additional finetuning to perform well in tasks they were not originally trained for. In practice, this means that the set of evaluation tasks is different from those used during feature training.

2. We present a novel Multi-Scale Attention task head and show how it can be used to develop an architecture capable of addressing the UFL problem.

3. Finally we show that *Medusa* can still be applied to traditional MTL, where it achieves competitive performance while requiring far fewer resources.

2. Related Work

Multi-task learning. At its core, MTL [3, 32, 41] aims to train a single network to accomplish multiple tasks. Through feature sharing, these models can reduce compute requirements while performing better than expert network counterparts. Initial approaches consisted of multiple task encoders with additional feature sharing layers. The seminal UberNet [21] introduced a multi-scale, multi-head network capable of performing a large number of tasks simultaneously. Cross-stitch networks [27] introduced soft feature sharing by learning linear combinations of multiple task features. In practice, this requires first training each task separately and then finetuning their features. Sluice networks [33] extended this idea by incorporating subspace and skip-connection sharing. Meanwhile, NDDR-CNNs [17] replaced the linear combination of features with a dimensionality reduction mechanism.

Kokkinos *et al*. [21] and Zhao *et al*. [46] showed that feature sharing in unreleated tasks results in a degradation in performance for both tasks, known as *negative transfer*. To account for this, MTAN [23] used convolutional attention to build task specific decoders from a shared backbone. Other methods learn where to branch from the backbone and what layers to share. Vandenhende *et al*. [40] decide what layers to share based on precomputed task affinity scores [16]. FAFS [25] begins with a fully shared model, optimizing the separation between dissimilar tasks while minimizing model complexity. BMTAS [2] and LTB [19] instead use the Gumbel softmax to represent branching points in a tree structure.

More recent approaches introduce additional refinement steps prior to making the final prediction. PAD-Net [43] was the first of these networks, using simple task heads to make intermediate predictions. Each possible pair of tasks were then connected via spatial attention, from which a final prediction was made. MTI-Net [42] extended this approach to multiple scales, incorporating feature propagation modules between them. PAP-Net [45] instead learned per-task pixel affinity matrices, estimating the pixel-wise correlation between each combination of tasks. Zhou *et al*. [47] additionally incorporated inter-task patterns. Due to the connections between all possible tasks, these approaches suffer from a quadratic growth of network parameters, leading to intractable compute requirements.

Transfer learning. A topic closely related to UFL is transfer learning [31, 37, 39, 50]. However, these works typically focus on solving domain shift at the input level, performing the same task with a different input modality. In other cases, the target is a closely related task, *e.g*. classification on a different set of labels. More closely related to *Medusa* are feature- and network-based techniques for transfer learning. Feature-based approaches aim to transform the source feature representations into new representations for the target domain. This includes approaches such as feature augmentation [9, 14, 20], mapping [24, 29, 30], clustering [7, 8] and sharing [11, 18, 22]. Meanwhile, network-based techniques have instead focused on parameter sharing. Some

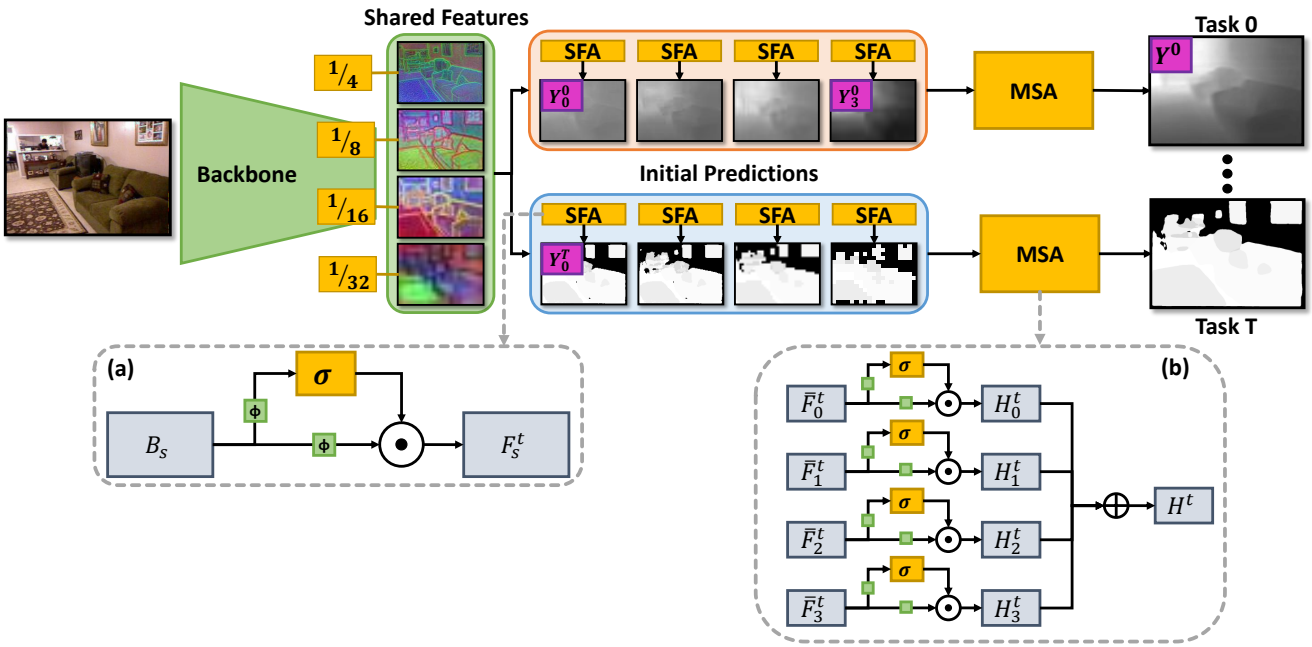


Figure 2. **Proposed *Medusa* Architecture.** We focus on building independent task heads. This allows us to efficiently scale to a larger number of tasks thanks to the dual attention mechanisms. **(a)** Shared Feature Attention, selecting relevant backbone features for each task and scale through per-channel spatial attention. **(b)** Novel Multi-Scale Attention head, combining task features at different scales to generate the final predictions.

notable examples include matrix factorization [48, 49] and parameter reuse from a network pretrained on a source domain [28, 38]. However, the focus lies mainly on the performance after finetuning on a specific new target task, rather than the overall performance on a wide range of tasks, which is the focus of MTL and UFL.

More recently, [16,44] were proposed to learn and model the relationship between tasks. However, these approaches do not truly solve the UFL problem since it is still necessary to train a separate network on each task. Instead they follow a brute force approach to find the best possible source to transfer for a given target task. In contrast, *Medusa* learns a single representation which can generalize well across future target tasks.

3. Methodology

The aim of this work is to introduce an architecture capable of learning universal features that perform well in multiple different tasks. As shown in Figure 2, *Medusa* consists of two main components: a shared backbone and individual task heads. One key design feature is that each task head is independent from the rest. This allows us to add new task heads *a posteriori*, which can be trained in conjunction or separately from the existing tasks.

3.1. Shared Feature Attention

The only part of the architecture common to all tasks is the shared backbone. The backbone produces the shared features $\mathbf{B}_s \in \mathbb{R}^{C_s}$ at multiple scales S, where C_s is the number of channels per-scale. Our goal is to learn universal features useful in a wide range of tasks, which may not be known at training time. In order to let the backbone learn a broad range of features whilst allowing tasks to pick their own specific subsets, we introduce spatial attention between the backbone and each task head. We define the process of applying spatial attention SA to a generic feature map $\mathbf{F}$ as

$$SA\left(\mathbf{F}\right) = \sigma\left(\phi_1\left(\mathbf{F}\right)\right) \odot \phi_2\left(\mathbf{F}\right), \tag{1}$$

where σ is the sigmoid operation, $\odot$ the Hadamard product and ϕ a convolution operation followed by batch normalization and a ReLU activation. Note that the concept of spatial attention is also known as the GLU activation [10] and has previously been used in MTL [23, 42, 43]. In *Medusa*, the convolution weights for each scale and task are independent from each other. Therefore, $\mathbf{F}_s^t = SA_s^t\left(\mathbf{B}_s\right)$ represents the initial task features for scale s and task t.

The shared backbone can now learn a generic feature representation that suits a much wider range of tasks. Through the per-channel spatial attention $\sigma\left(\phi_1\left(\mathbf{F}\right)\right)$, each task/scale retains only the specific subset of backbone fea-

tures relevant to it. This alleviates the possibility of *negative transfer*, where sharing features between unrelated tasks can degrade the performance of both tasks. Whilst previous approaches also make use of spatial attention, they place a larger focus on modeling the connections between each pair of tasks. By creating an information bottleneck, *Medusa* places more importance on learning features common to all tasks that therefore provide better transfer capabilities. Additionally, our multi-scale approach provides the subsequent task features with a wide variety of information which, combined with the proposed MSA head, helps to provide optimal features for the final prediction.

3.2. Multi-Scale Task Predictions

Rather than building a per task sequential decoder such as [23], we build parallel task heads by using each scale of backbone features to make an initial prediction for each task. This results in S predictions per task, used as additional supervision during training. The initial task features $\mathbf{F}_s^t$ are refined through

$$\bar{\mathbf{F}}_s^t = \psi_2\left(\psi_1\left(\mathbf{F}_s^t\right)\right), \tag{2}$$

where $\bar{\mathbf{F}}_s^t$ are the refined task features and $\psi\left(\mathbf{F}\right) = \phi\left(\mathbf{F}\right) + \mathbf{F}$ is a residual convolutional block. The initial predictions are given by $\mathbf{Y}_s^t = \phi_s^t\left(\bar{\mathbf{F}}_s^t\right)$, where ϕ_s^t is the convolution mapping from C_s channels provided by the backbone to those required by the task. These predictions are used as intermediate supervision exclusively during training, while the refined task features are combined in the MSA heads.

3.3. Multi-Scale Attention Task Heads

The final step is combining task features from multiple scales to make the final prediction for each task. In the naïve case, one could simply upsample all task features to the same resolution, concatenate channel-wise and process them together [42]. We refer to this task head as HRHead in the results further on. This assumes that the predictions from each scale are equally valid and important. However, due to the varying resolution and subsequent receptive field of each scale, this is not typically the case. In practice, higher resolution predictions can help to provide more accurate and sharp edges for some tasks. On the other hand, lower resolutions with more channels provide more descriptive features with a larger receptive field, making predictions more consistent on a global scale.

We capture this information by introducing a novel Multi-Scale Attention task head. Given the processed task features $\bar{\mathbf{F}}_s^t$ the network is able to select the important information from each scale using the spatial attention SA previously defined in (1). This results in

$$\mathbf{H}_s^t = SA_s^t\left(\bar{\mathbf{F}}_s^t\right), \tag{3}$$

$$\mathbf{H}^t = \mathbf{H}_0^t \oplus \mathbf{H}_1^t \oplus \ldots \oplus \mathbf{H}_S^t, \tag{4}$$

where $\oplus$ represents channel-wise concatenation of the attended per task per scale features $\mathbf{H}_s^t$. Note that the spatial attention weights are independent from those previously used to extract $\mathbf{F}_s^t$. The final per task features $\mathbf{H}^t$ are used to obtain the final predictions as $\mathbf{Y}^t = \phi^t\left(\mathbf{H}^t\right)$, where ϕ^t maps the final number of channels $\sum C_s$ to the required task channels.

Thanks to the design of the system it becomes trivial to attach new task heads to the shared backbone. These task heads are able to choose relevant features from the shared backbone and adapt the multiple scales to the needs of new tasks. Furthermore, since the task heads are independent, the number of parameters increases only linearly with the number of tasks. Because these heads are lightweight, the resulting system is highly efficient. This is contrary to approaches such as [42, 43], where each task requires connections to every other task, resulting in a quadratic parameter-complexity with regards to the number of tasks.

4. Results

Dataset. We use the NYUD-v2 dataset [34], containing labels for depth estimation, semantic segmentation, edge detection and surface normal estimation. Following existing benchmarks [26, 42], we focus on evaluating depth and semantic segmentation, leaving edges and surface normals as auxiliary tasks for use during training. Depth is evaluated through the Root Mean Squared Error (RMSE), while semantic segmentation uses the mean-Intersection over Union (m-IoU).

Implementation details. We use HRNet-18 [36] pretrained on ImageNet [12] as the backbone, due to its suitability for dense prediction tasks. This produces features at downsampling scales of $\{4, 8, 16, 32\}$ with $\{18, 36, 72, 144\}$ channels, respectively. We use the Adam optimizer, with a base LR=1e-4 and a polynomial decay [4]. Experimentally, we found that training the shared backbone with a lower learning rate than the heads (typically LR*0.1) produced better results. Models are trained for 100 epochs. Regarding the losses, we use the L_1 loss for depth and surface normal estimation, cross-entropy for semantic segmentation and a binary cross-entropy (with positive weighting of 0.95) for edge detection.

4.1. Multi-task Evaluation

Performance. We first evaluate *Medusa*'s performance in a traditional MTL setting, following the procedure in [26]. As mentioned, the main tasks evaluated are depth estimation and semantic segmentation. However, during training we make additional use of **E**dge detection and surface **N**ormal estimation to show the network a varied set of tasks. Following [26], we define multi-task learning performance as

Table 1. **Multi-task Evaluation.** When performing MTL on NYUD-v2 *Medusa* is on par with the current SotA [42], whilst using less resources (see Figure 4). This is due to the novel lightweight MSA task head. We highlight the **best** and next best performing techniques.

	Backbone	Head	N+E	Seg ↑	Depth ↓	$\Delta_m\%$ ↑
ST Baseline	ResNet-18	DeepLab-v3+		35.77	0.600	+0.00
MT Baseline	ResNet-18	DeepLab-v3+		35.74	0.597	+0.12
Cross-stitch [27]	ResNet-18	DeepLab-v3+		36.01	0.600	+0.30
NDDR-CNN [17]	ResNet-18	DeepLab-v3+		34.72	0.611	-2.47
MTAN [23]	ResNet-18	DeepLab-v3+		36.00	0.594	+0.79
ST Baseline	HRNet-18	HRHead		34.57	0.606	+0.00
MT Baseline	HRNet-18	HRHead		33.21	0.614	-2.63
MTAN [23]	HRNet-18	DeepLab-v3+		35.25	0.581	+3.02
MTAN	HRNet-18	DeepLab-v3+	✓	36.19	0.567	+5.57
PAD-Net [43]	HRNet-18	HRHead		34.39	0.617	-1.23
PAD-Net	HRNet-18	HRHead	✓	35.46	0.604	+1.43
MTI-Net [42]	HRNet-18	HRHead		36.94	0.559	+7.26
MTI-Net	HRNet-18	HRHead	✓	37.40	**0.540**	**+9.48**
***Medusa* (ours)**	HRNet-18	**MSA (ours)**		36.99	0.573	+6.19
Medusa	HRNet-18	**MSA**	✓	**37.48**	0.545	+9.24

Table 2. **Spatial Attention Ablation Study.** The SFA column indicates the presence of spatial attention between the shared backbone and the task heads. Meanwhile, the MSA task head incorporates attention when combining each task's multi-scale features. Both types of attention lead to clear improvements. All models use the HRNet-18 backbone.

	SFA	Head	N+E	Seg ↑	Depth ↓	$\Delta_m\%$ ↑
ST Baseline		HRHead		34.57	0.606	+0.00
MT Baseline		HRHead		33.21	0.614	-2.63
MT Baseline		**MSA**		35.58	0.598	+2.12
Medusa		HRHead	✓	36.50	0.558	+6.71
Medusa	✓	HRHead	✓	36.64	0.553	+7.31
Medusa		**MSA**	✓	37.14	0.555	+7.91
Medusa	✓	**MSA**	✓	**37.48**	**0.545**	**+9.24**

$$\Delta_m = \frac{1}{T}\sum_{t=0}^{T\text{-}1}(\text{-}1)^{l^t}\,\frac{M_m^t\text{-}M_b^t}{M_b^t}, \tag{5}$$

where $M^t_{\{m,b\}}$ is the per-task performance of the ***m***ultitask or ***b***aseline network and l^t indicates if a lower value means a better performance for the given task. As such, Δ_m represents the average increase (or drop) in performance for each task, relative to the single task baseline.

We obtain single task baselines (ST) for each backbone by training expert networks on each task separately, resulting in two completely separate models. ResNet models use Deeplab-v3+ ASPP [5] task heads, while HRNet-18 uses the naïve multi-scale task head, upsampling all scales and concatenating channel-wise (HRHead). Meanwhile, the multi-task baselines (MT) use a joint backbone with separate task heads. The baselines were obtained by retraining the code provided by the authors of [26, 42]. In order to make results more comparable, we also create and train a version of MTAN adapted to make use of the HRNet backbone. However, since MTAN builds a per task decoder, rather than making initial predictions at multiple scales, it still requires use of the DeepLap-v3+ head.

The results can be found in Table 1, where the column (N+E) indicates the presence of the auxiliary edges and surface normals tasks. It is interesting to note that some MT baselines and methods [17, 43] actually lead to a degrada-

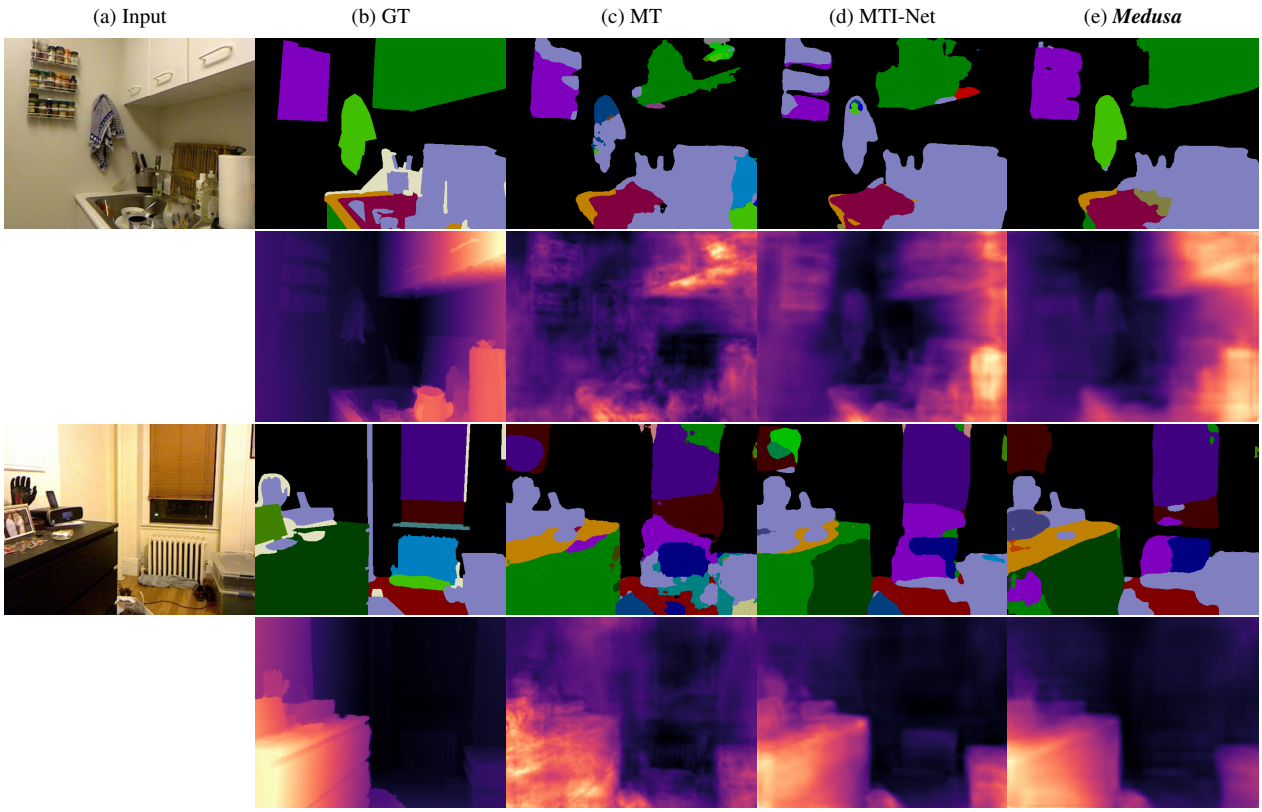


Figure 3. **Qualitative Evaluation.** Through the proposed MSA heads, *Medusa*'s predictions are both globally consistent and have well defined borders. Results are on par with the current State-of-the-Art (SotA) while using less resources.

tion in performance. This is likely due to a combination of *negative transfer* and task loss balancing during training. Meanwhile, despite not modelling task connections in the decoder, *Medusa* still shows improvements when incorporating the auxiliary (N+E) tasks. This demonstrates the ability of *Medusa* to learn generic features that complement all tasks, sharing only the useful information. To summarize, *Medusa* greatly outperforms all baselines with independent task heads [23] and is comparable to the current SotA [42] while using resources in a more efficient manner, as we will now discuss.

Ablation. We perform an ablation study to understand the importance of *Medusa*'s components, primarily focused on the uses of spatial attention. In the case of the Shared Feature Attention (SFA), we replace the spatial attention connecting the shared backbone to each task head with a convolutional block with BatchNorm and ReLU. On the other hand, we compare the proposed MSA head to the default HRNet, which does not contain spatial attention.

Table 2 shows that both attention components result in large benefits. Incorporating the SFA results in a consistent relative improvement across the different techniques of 8.94% and 16.81%. Meanwhile, the MSA head leads to even larger performance gains—from 17.88% to 180.60%. Most notably, incorporating the MSA head into the MT baseline results in an improvement over the ST baseline. Even across all *Medusa* variants, this novel task head leads to a consistent increase in accuracy. Similarly, incorporating the SFA between the backbone and task heads improves performance regardless of the task head used. Overall, the dual attention mechanisms used in *Medusa* lead to a relative improvement of 37.7% over the plain convolutional baseline. Once again, we believe that this is due to spatial attention providing an effective, yet efficient mechanism for routing information between different stages in the network. This allows it to easily decide what information should or should not be shared across either tasks and scales.

Visualizations. Figure 3 shows qualitative results based on the network predictions. As expected, the MT baseline shows the worst results. MTI-Net shows more spurious class predictions, especially in cluttered environments, as seen in the second image in Figure 3. Meanwhile, we find *Medusa* to be more globally coherent, while still having well defined edges between classes and in depth discontinuities. This is due to the proposed MSA head, which can effectively combine the best features from each scale.

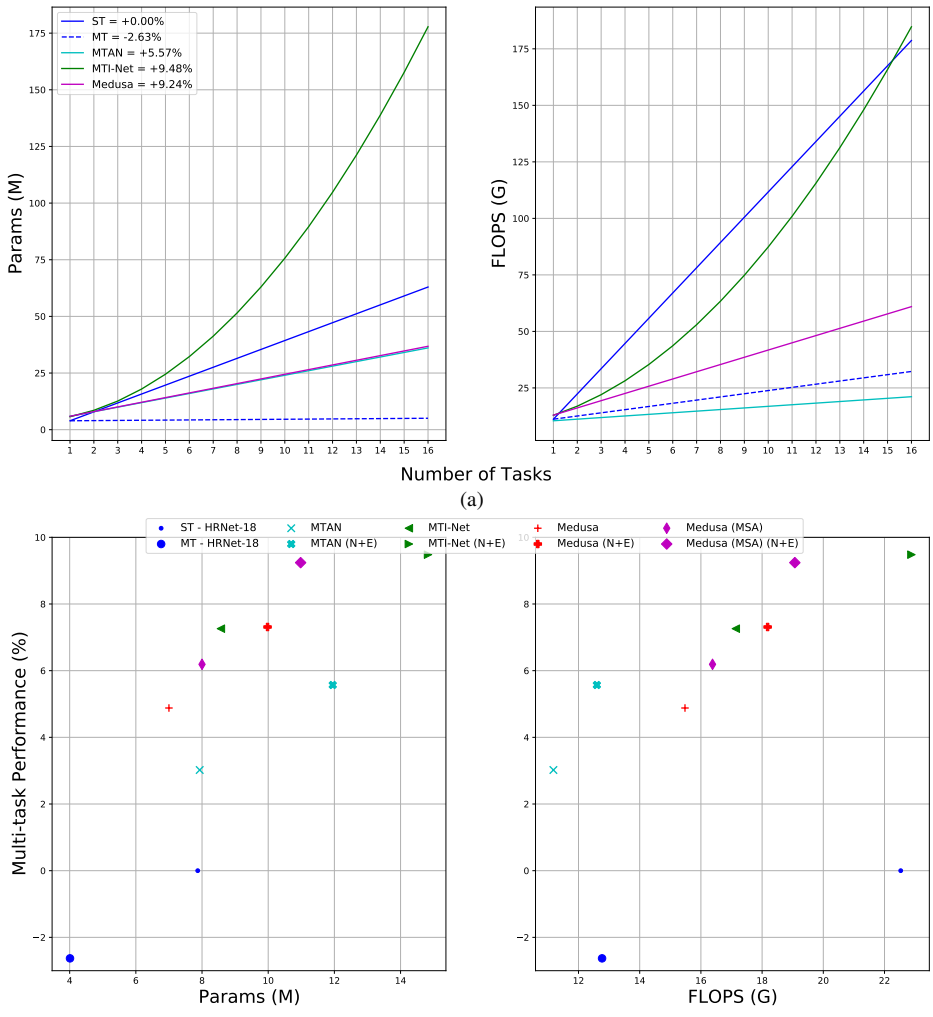


Figure 4. **Resource Usage.** Modelling the relationships between each pair of tasks and scales [42] results in a quadratic increase in paramters/GFLOPS w.r.t. the number of tasks. This does not scale well to an increasing number of tasks. *Medusa*'s independent task heads lead to a much more efficient scaling, while focusing on features that are more generic and reusable.

Resources. Figure 4 shows how different approaches scale w.r.t. the number of tasks. The most resource efficient approach is the MT baseline. However, since it only uses basic task heads without intermediate predictions or attention, its performance is lacklustre. Other approaches with independent task heads (ST, MTAN) are relatively efficient, since the increase in task head parameters is linear, but their performance (see Table 1) is not on par with *Medusa*. MTI-Net is the only approach with results comparable to *Medusa*, but it does not scale well to increasing numbers of tasks. After only three tasks, MTI-Net requires more parameters than the ST baseline, which trains a completely separate network for each task. This gap only increases, due to the quadratic parameter-complexity introduced by the connections between all possible pairs of tasks.

4.2. Universal Feature Learning

The following experiment evaluates *Medusa* in the highlighted UFL task. The objective is to learn generic shared features that can be adapted to new, unseen tasks on unseen datasets without additional finetuning at the backbone level. This is contrary to MTL, where the objective is to learn features that perform well in the specific set of training tasks without generalization to other tasks. It is also contrary to traditional transfer learning, where the objective is instead to solve the domain shift between different modalities of a single task.

We show the ability of *Medusa* features to transfer to new tasks on new datasets through the PASCAL-Context [6] dataset, containing semantic segmentation, human part seg-

Table 3. **Universal Feature Learning.** We use the pretrained backbone features from Table 1 to train task heads on new tasks on new datasets. The features learned by *Medusa* provide large improvements over the commonly used ImageNet pretrained features, despite the fact that we train with orders of magnitude less data.

	NYUD-v2			PASCAL-Context		
	Seg ↑	Depth ↓	$\Delta_m\%$ ↑	Parts ↑	Sal ↑	$\Delta_m\%$ ↑
ST Baseline	34.57	0.606	+0.00	48.73	56.44	+0.00
MT Baseline	33.21	0.614	-2.63	36.13	51.96	-12.93
MTAN [23]	36.19	0.567	+5.57	47.37	57.84	+4.26
MTI-Net [42]	37.40	**0.540**	**+9.48**	51.50	60.19	+10.76
Medusa	**37.48**	0.545	+9.24	**52.24**	**61.91**	**+13.18**

mentation and edge detection. There are also pseudo-ground truth labels for surface normals and saliency [26] obtained from SotA models [1, 5]. Since three of the tasks are common to NYUD-v2 we evaluate on the two unique ones: human part segmentation and saliency estimation. To carry out this evaluation we use the previous models trained on NYUD-v2 in Section 4.1 with the auxiliary (N+E) tasks and check their transfer capability to the new target tasks in the PASCAL-Context dataset. This is done by freezing the shared feature backbone network and adding a new task head corresponding to either saliency estimation or human part segmentation. This can be seen as a form of continual learning. Since the shared backbone and previous task heads are frozen, we ensure that the network does not forget existing information. Instead, we expand its knowledge by learning a new task.

Table 3 shows the results from this experiment, including the previous MTL results on NYUD-v2 for comparison. This highlights the main difference between UFL and MTL, where MTL only performs well in the original training tasks. This is only exacerbated by the naïve multi-task implementation, resulting in a large amount of negative transfer between tasks. Meanwhile, *Medusa* provides the best transfer capabilities. It is worth noting the large improvement over ImageNet pretrained features from the single task baseline (ST), which are trained on orders of magnitude more data than the remaining MTL methods. However, since they are trained exclusively for global image classification, the learnt representations do not transfer well to complex dense tasks. Meanwhile, even through MTL performance is almost equal to MTI-Net (9.24% *vs*. 9.48%), the features learnt by *Medusa* generalize to a broader range of tasks (13.18% *vs*. 10.76%). This is due to *Medusa*'s design, which places a larger focus on the shared feature representation, which is therefore able to learn a more effective feature representation.

5. Conclusions & Future Work

In this paper we have highlighted the importance of *universal feature learning vs*. multi-task learning, requiring a feature learning system to perform well over a large variety of tasks without additional finetuning. This is in contrast to most current MTL approaches, which focus learning features specific to a given set of training tasks.

To this end we proposed *Medusa*, capable of training on multiple tasks simultaneously, while allowing new task heads to be attached and trained jointly or separately. Furthermore, thanks to the novel MSA head, we are capable of doing this in a very efficient manner. This helps to provide comparable results whilst using less resources than previous approaches. We additionally demonstrated the generality of the features leant by *Medusa* in the UFL task on unseen tasks and datasets, and showed its ability to outperform SotA features from both ImageNet and other MTL networks.

Whilst *Medusa* has shown its effectiveness in both MTL and UFL, it is not without limitations and challenges to address in future work. For instance, the data used during training is currently required to have labels for all target tasks. In practice, these labels can be challenging to obtain, especially as the number of tasks and images grows. *Medusa*'s performance is also dependent on the tasks used during training. If we wish to transfer to a task that is completely unrelated to the training tasks, it is likely that the features will not overlap. Both of these issues could potentially be addressed by making the training process more flexible, without requiring each item to have labels for all tasks or by training with multiple unrelated datasets.

References

[1] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. PixelNet: Representation of the pixels, by the pixels, and for the pixels, 2017. 8

[2] David Bruggemann, Menelaos Kanakis, and Stamatios Georgoulis. Automated Search for Resource-Efficient Branched

Multi-Task Networks. In *British Machine Vision Conference*. BMVA Press, 2020. 2

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

[4] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, apr 2018. 4

[5] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, volume 11211 LNCS, pages 833–851. Springer International Publishing, sep 2018. 5, 8

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Conference on Computer Vision and Pattern Recognition*, pages 1979–1986. IEEE Computer Society, sep 2014. 7

[7] Wenyuan Dai, Gui Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 210–219, New York, New York, USA, 2007. ACM Press. 2

[8] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught Clustering. In *Proceedings of the 25 th International Conference on Machine Learning*, 2008. 2

[9] Hal Daumé. Frustratingly easy domain adaptation. In *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, 2007. 2

[10] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language Modeling with Gated Convolutional Networks. In *International Conference on Machine Learning*, pages 933–941. PMLR, jul 2017. 3

[11] Jesse Davis and Pedro Domingos. Deep transfer via second-order Markov logic. In *ACM International Conference Proceeding Series*, volume 382, pages 1–8, New York, New York, USA, 2009. ACM Press. 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE Computer Society, mar 2010. 4

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 337–349. IEEE Computer Society, dec 2018. 1

[14] Lixin Duan, Dong Xu, and Ivor W Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, volume 1, pages 711–718, 2012. 2

[15] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Conference on Computer Vision and Pattern Recognition*, pages 8084–8093. IEEE Computer Society, jun 2019. 1

[16] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 12379–12388. IEEE Computer Society, jun 2019. 2, 3

[17] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. NDDR-CNN: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Conference on Computer Vision and Pattern Recognition*, pages 3200–3209. IEEE Computer Society, jun 2019. 2, 5

[18] Pinghua Gong, Jieping Ye, and Changshui Zhang. Multi-Stage Multi-Task Feature Learning. Technical report, 2012. 2

[19] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to Branch for Multi-Task Learning. *International Conference on Machine Learning*, 2020. 2

[20] Hal Daumé Iii, Abhishek Kumar, and Avishek Saha. Co-regularization Based Semi-supervised Domain Adaptation. In *Advances in Neural Information Processing Systems*, volume 23, 2010. 2

[21] Iasonas Kokkinos. UberNet: Training a universal convolutional neural network for Low-, Mid-, and high-level vision using diverse datasets and limited memory. In *Conference on Computer Vision and Pattern Recognition*, pages 5454–5463. IEEE Computer Society, nov 2017. 2

[22] Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks. In *n Proceedings of the 24 th International Conference on Machine Learning*, 2007. 2

[23] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Conference on Computer Vision and Pattern Recognition*, pages 1871–1880. IEEE Computer Society, jun 2019. 1, 2, 3, 4, 5, 6, 8

[24] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2200–2207. Institute of Electrical and Electronics Engineers Inc., 2013. 2

[25] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. In *Conference on Computer Vision and Pattern Recognition*, pages 5334–5343. IEEE Computer Society, 2017. 2

[26] Kevis Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Conference on Computer Vision and Pattern Recognition*, pages 1851–1860. IEEE Computer Society, jun 2019. 1, 4, 5, 8

[27] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-Stitch Networks for Multi-task Learn-

ing. In *Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 3994–4003. IEEE Computer Society, dec 2016. 2, 5

[28] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1717–1724. IEEE Computer Society, sep 2014. 3

[29] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, pages 677–682, 2008. 2

[30] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, feb 2011. 2

[31] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning, oct 2010. 2

[32] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint*, jun 2017. 2

[33] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Conference on Artificial Intelligence*, pages 4822–4829. AAAI Press, 2019. 2

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, volume 7576 LNCS, pages 746–760. Springer International Publishing, 2012. 4

[35] Jaime Spencer, Richard Bowden, and Simon Hadfield. Scale-adaptive neural dense features: Learning via hierarchical context aggregation. In *Conference on Computer Vision and Pattern Recognition*, pages 6193–6202. IEEE Computer Society, jun 2019. 1

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 5686–5696. IEEE Computer Society, jun 2019. 4

[37] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11141 LNCS, pages 270–279. Springer Verlag, oct 2018. 2

[38] Jessica A F Thompson, Marc Schonwiesner, Yoshua Bengio, and Daniel Willett. How Transferable Are Features in Convolutional Neural Network Acoustic Models across Languages? In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 2019-May, pages 2827–2831. Institute of Electrical and Electronics Engineers Inc., may 2019. 3

[39] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. 2009. 2

[40] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched Multi-Task Networks: Deciding What Layers To Share. *British Machine Vision Conference*, apr 2020. 2

[41] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-Task Learning for Dense Prediction Tasks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, apr 2021. 2

[42] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. MTI-Net: Multi-scale Task Interaction Networks for Multi-task Learning. In *European Conference on Computer Vision*, pages 527–543. Springer International Publishing, aug 2020. 1, 2, 3, 4, 5, 6, 7, 8

[43] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. PAD-Net: Multi-tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing. In *Conference on Computer Vision and Pattern Recognition*, pages 675–684. IEEE Computer Society, dec 2018. 1, 2, 3, 4, 5

[44] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3712–3722. IEEE Computer Society, dec 2018. 3

[45] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 4101–4110. IEEE Computer Society, jun 2019. 2

[46] Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. A Modulation Module for Multi-task Learning with Applications in Image Retrieval. In *European Conference on Computer Vision*, volume 11205 LNCS, pages 415–432. Springer International Publishing, sep 2018. 2

[47] L Zhou, Z Cui, C Xu, Z Zhang, C Wang, T Zhang, and J Yang. Pattern-structure diffusion for multi-task learning. In *Conference on Computer Vision and Pattern Recognition*, pages 4514–4523. IEEE Computer Society, 2020. 2

[48] Fuzhen Zhuang, Ping Luo, Changying Du, Qing He, Zhongzhi Shi, and Hui Xiong. Triplex transfer learning: Exploiting both shared and distinct concepts for text classification. *IEEE Transactions on Cybernetics*, 44(7):1191–1203, 2014. 3

[49] Fuzhen Zhuang, Ping Luo, Hui Xiong, Qing He, Yuhong Xiong, and Zhongzhi Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization†. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):100–114, feb 2011. 3

[50] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning, jan 2021. 2