Data exploitation: multi-task learning of object detection and semantic segmentation on partially annotated data

Hoàng-Ân Lê
hoang-an.le@irisa.fr
Minh-Tan Pham
minh-tan.pham@irisa.fr

IRISA, Université Bretagne Sud,
UMR 6074, 56000 Vannes, France

Abstract

Multi-task partially annotated data where each data point is annotated for only a single task are potentially helpful for data scarcity if a network can leverage the inter-task relationship. In this paper, we study the joint learning of object detection and semantic segmentation, the two most popular vision problems, from multi-task data with partial annotations. Extensive experiments are performed to evaluate each task performance and explore their complementarity when a multi-task network cannot optimize both tasks simultaneously. We propose employing knowledge distillation to leverage joint-task optimization. The experimental results show favorable results for multi-task learning and knowledge distillation over single-task learning and even full supervision scenario. All code and data splits are available at https://github.com/lhoangan/multas

1 Introduction

Although both object detection and semantic segmentation aim to understand the image content, the two problems differ in spatial structure and information granularity. Object detection performs at the object level outputting unordered list of bounding boxes with corner coordinates and object types while semantic segmentation provides per-pixel predictions; object detection distinguishes object instances while semantic segmentation recognizes each category as a whole and also amorphous regions such as ground, sky, sea, *etc*.

Attempts have been made to jointly learn both tasks in a single model. Methods such as Mask R-CNN [13] overcomes the spatial structure difference by generating an object mask for each predicted bounding box, effectively predicting instance segmentation. On the other hand, the introduction of panoptic segmentation [17] can be seen as resolving the information granularity difference in which instance-level objects and amorphous categories are tackled together as a dense prediction problem. Combining both tasks under the common form of instance segmentation, however, leaves the original tasks unfinished: Mask R-CNN does not provide segmentation masks for stuff categories nor does panoptic segmentation directly provide bounding box coordinates.

Multi-task learning is a research area that allows training different problems under the same model. The general assumption is that several tasks are inherently related to one

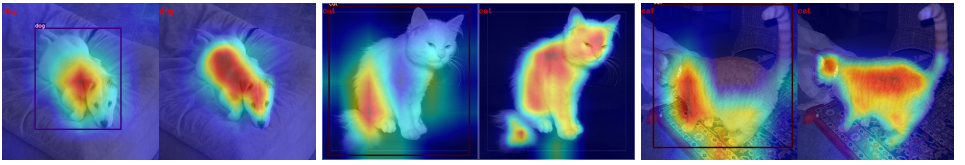

Figure 1: Class activation maps at the same feature layers of an object detection network (left) and semantic segmentation (right) showing incompatible feature attentions: detection only activates a (few) feature at the scale producing fit boxes while segmentation activates all those belong to the objects.

another and by optimizing them together for each input image, the network could extract common features and pick up the salient interrelationships. Although training multiple tasks could potentially increase tasks coherency and, for particular setups, also allow self-supervision [4, 35], it is challenging as each task would require specific architecture and optimization criteria, and maintaining a training dataset with consistent annotations for all tasks proves to be expensive.

In this paper, the joint learning of object detection and semantic segmentation is considered, which despite their popularity as single tasks, seems to receive limited attentions in the literature. Due to different targets, although the two tasks are closely related, the features learned for each task are not readily compatible. Figure 1 shows the activation map using Grad-CAM [30] at the same layers of two networks with the same encoder architectures, trained for object detection and semantic segmentation. Semantic segmentation activates (nearly) all the features covering the object of interest while object detection activates only those at the feature scale that produces fitted bounding boxes, no matter if they belong to the objects. Table 1 and Sec. 4.2 show that an encoder trained for one task cannot immediately be used for the other tasks when only the task-specific head is finetuned.

Diverging from the usual multi-task learning assumption that annotations are available in all tasks for each training example, we limit the scope of the paper to multi-task partially annotated data, where each image is annotated for a single task and there are no images containing both task annotations. This is interesting because (1) the network cannot optimize both tasks for the same input and is hindered in attempt to learn joint features and salient interrelationships; (2) therefore, this setting would illustrate the complementarity of the two tasks of interest; and (3) it is data efficient and would be an alternative method to ameliorate the data scarcity problem as more data with single-task annotations could be used for training, allowing for expanding training ability.

To that end, we employ a simple multi-task learning framework to study the combination of object detection and semantic segmentation. We experiment with various setups and observe each task's performance under different input conditions. By varying the datasets, the interaction between the two tasks can be observed which can be useful for further study. The simple feature-imitation knowledge distillation model is employed for cross-task optimization which is seemingly not possible for partially annotated data.

The paper contributions are as follows. We explore the combination of object detection and semantic segmentation in a multi-task learning framework for partially annotated data. Extensive experiments are performed to evaluate both quantitatively and qualitatively the benefit of one task to the other. A knowledge distillation method is employed and evaluated for joint-task optimization.

2 Related work

2.1 Multi-task learning

Multi-task learning (MTL) trains a single model that can infer different task targets from a given input. One of the main assumptions is the compatibility of the features learned for each task and by optimizing them for each input, the network could learn the common knowledge that benefits and complements one another [20, 26].

Several methods have been proposed to accommodate various tasks and network architectures to improve shared information among the tasks [3] using attention mechanisms [23] and gating strategies [1], and to study cross-task relationships [26, 33]. The fully supervised learning strategy requires annotations available for all tasks per training example for optimization, which is costly and hinders scalability. Therefore, attempts have been made for semi-supervised learning [6, 15] that allows learning from unlabelled data and relaxes the number of annotations, yet all-task annotations per training sample are still required.

Closely related to the problem in our paper is the work of Li *et al*. [19], in which each training data point is only required to contain an annotation for a single task, or the multi-task partial annotation scenario. The cross-task consistency constraint is proposed and the task-specific annotations are projected to the joint pairwise task-space from which supervised signals are provided to the training process. The method requires the dense spatial structures of the annotations making it inapplicable for object detection in this paper.

2.2 Knowledge distillation

Hinton *et al*.[14] has shown that a network could benefit from a larger or an ensemble of models, called teachers, by mimicking the predicted logits or imitate the deep features [11] learned by them. Depending on the purpose, knowledge distillation (KD) could be seen as model compression which aims to reduce model complexity with less performance sacrificing, or a self-training technique [32, 37] where a network is trained using the combination of available annotations and pseudo-labels provided by the teacher's predictions. Self-training with uncertain teachers for object detection has recently been studied [27], where the teachers are trained with a small number of supervised data disjoint with the students' training set, or for a different task (segmentation). Different from their paper which also involves detection-segmentation multi-task training but focuses only on the detection benefit, our work interests in both tasks' performances and shows the multi-task advantage with TIDE [1] error analysis. Multi-task learning has seen other applications with self-training such as the extension of Born Again Network [10] for learning context in NLP problem [7] using a weight annealing strategy to update the distillation and multi-task losses. Li *et al*. [18] apply knowledge distillation to solve the unbalanced loss optimization problem in multi-task learning and show favorable results for fully annotated semantic segmentation and depth prediction training.

3 Method

To study the relationship between object detection and semantic segmentation, we apply a simple multi-task learning framework following the encoder-decoder principle. Common to many approaches is a shared encoder, comprising a backbone (*e.g*. the ResNet family) and a neck (*e.g*. the FPN family), which extracts and aggregates features from input images while multiple decoders, or heads, provide task-specific predictions. The overview framework

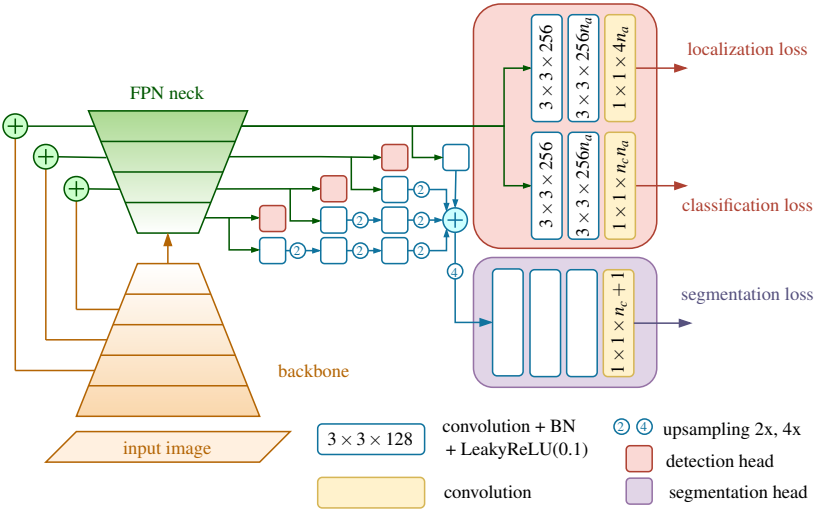

Figure 2: A general view of the network architecture including an encoder (backbone and neck) and 2 heads for object detection and semantic segmentation. Object detection performs at each pyramid scale while semantic segmentation aggregates all scales and upsamples them to the image-size.

is shown in Figure 2. In this work, the one-stage anchor-based object detection architectures [22, 25] are studied. The detection head (red boxes) at each pyramid scale comprises 2 output branches with the same architecture for localization and classification losses. For semantic segmentation, the multi-scale pyramid features are aggregated using the architecture by Kirillov *et al*. [17]. The aggregation is performed by alternating between convolution and double up-sampling the features at each scale until one-fourth of the input size before element-wise adding together and finally quadruple up-sampling to the input size. The aggregated and segmentation features' dimensions are set to 128 following the original work while detection heads features are 256 as output from the encoder. The detection head uses the Focal Loss [22] and the Balanced L1 Loss [28] for localization and classification while the semantic segmentation head uses the regular cross-entropy with softmax loss.

Multi-task training. As each data point is annotated for only a single task, not all the losses can be optimized together. Two optimization approaches are considered, alternating the tasks (1) every epoch or (2) every iteration. For the former, the network is trained with one task for one epoch with the gradients computed from the respective task-specific head and leaving the other task head untouched before being trained with the other task in the following epoch. For the latter, a mini-batch of images with annotations for one task is passed through the network immediately after one with the other task. The gradients from each mini-batch is computed for the corresponding head and accumulated for the encoder. Only after mini-batches from both tasks have been fed in and gradients accumulated are the network parameters updated. As a result, both tasks start and end an epoch together. Thus, the task with fewer annotations will randomly have some images repeated in waiting for a new epoch. We show in Sec. 4.2 and Table 1 the performance difference of the two strategies.

Knowledge distillation. We concatenate the features of all scale levels along the flattened spatial dimensions. The features of the student network are projected by a 1×1 convolution before being compared to the corresponding teachers'. The simple Mean Square Error (MSE) [34] is applied for feature imitation distillation. The illustration is shown in Figure 3.

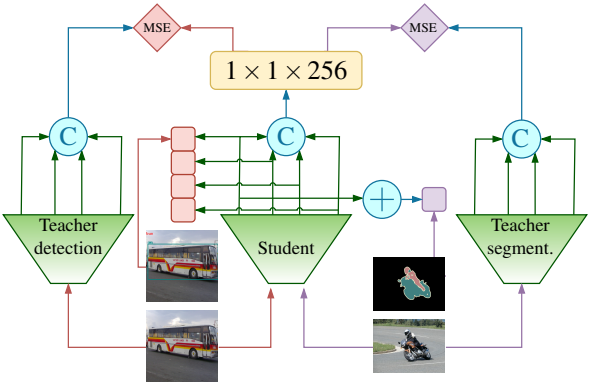

Figure 3: Visualization the knowledge distillation process in the multi-task learning for partially annotated data. Images and task-specific annotations are fed to the student and the respective teachers, with KD losses computed on teacher-student flattened and concatenated neck features.

As each training image can optimize a single task, there are 3 cases for distilling the student features per iteration: (1) from the teacher whose task is annotated (1mse) so the student's features are forced to follow the teacher's while learning from the provided ground truths at the same time, (2) from the task teacher *without* annotations (0mse) so that the head is trained with one task (using ground truth) while the encoder is forced to follow the other's teacher, and (3) is the combination of both (2mse).

4 Experiments

4.1 Setup

Datasets. All the experiments are conducted on the Pascal VOC [9] containing 20 object categories with 8,218 bounding-box annotated images for training, 8,333 for validation, and 4,952 for testing. Due to limited semantic segmentation annotations originally provided (1,464 and 1,449 for training and validation, respectively), the common practices use extra annotations provided by [12], resulting in 10,582 training images.

To simulate the partial supervision scenario, images are randomly sampled into 2 subsets, one for detection ◖ whose semantic annotations are held back and the other for semantic segmentation ◗ whose bounding-box annotations are kept out, resulting in 7,558 and 7,656 respectively (images without semantic segmentation ground truth are prioritized to the detection subset). For validation, the originally provided validation set for semantic segmentation with both task annotations are used with 1,443 images (6 images with only semantic segmentation are withheld). Unless stated otherwise, the image lists are kept the same in all experiments.

For out-of-domain experiments, the Cityscapes [8] dataset with 2,975 training, 500 validation images, and 7 semantic classes is employed for semantic segmentation. We resize the images to 128×256 to speed up the training process following [19, 23].

Network architectures. For comparison purposes, two backbone models from the ResNet family are employed, including ResNet50 backbone with PAFPN [24] neck (RN50+PAFPN), and ResNet18 backbone with FPN [21] neck (RN18+FPN). A few modifications are made

Training	Detection		Segmentation	
	RN18+FPN	RN50+PAFPN	RN18+FPN	RN50+PAFPN
Single task	42.81	50.22	64.55	72.32
Finetuning head	31.80	36.08	61.08	65.84
Finetuning full	43.30	50.61	65.21	72.01
ReCAM [5]			63.30	70.11
Multi-task (epoch)	44.51	51.62	66.99	72.93
Multi-task (iteration)	**44.78**	**52.10**	**67.57**	**73.66**
Multi-task (full)	46.83	53.68	67.47	73.08

Table 1: Comparing single tasks and multi-task learning on partially annotated data. Multi-task are trained by alternating the tasks every epoch or iteration. Training both tasks (full) results are included for reference. Multi-task learning outperforms all other settings.

Training	Detection		Segmentation	
	RN18+FPN	RN50+PAFPN	RN18+FPN	RN50+PAFPN
Single task	42.81	50.22	64.55	72.32
	38.10	43.73	63.02	68.96
Multi-task	44.78	**52.10**	**67.57**	**73.66**
	40.89	47.27	65.39	73.17
	44.99	51.43	66.16	73.03

Table 2: Single-task and multi-task performance when trained with half annotated detection and segmentation. Single-task is impacted more from the reduced training sizes for the respective task.

following the implementation of [34], including removing the first max-pooling layer of ResNet as in ScratchDet [36], and adding the context enhancement module as in ThunderNet [29]. The number of convolutional blocks in the detection head subsets is reduced from 4 to 2 to speed up the training time. The two networks are also used for knowledge distillation as teacher and student, respectively, with parameter ratio of 1.61. The networks are trained for 30 epochs, with learning rate of 5×10^{-3}.

Evaluation and analysis. We employ, for semantic segmentation the conventional IOU score [16] and, for object detection, the mAP metric implemented by the Detectron2 library [31], which follows the original VOC code but averages APs at multiple IOU thresholds in the range $[.5, .95, .05]$. The detection results at IOU of 50%, *i.e.* AP50, are used as inputs to the TIDE [1] framework for analyzing the error sources. TIDE breaks down detection errors into 6 types and estimates the isolated contribution of each to the overall performance, as follows: (1) **Cls** errors localize correctly but classify incorrectly; (2) **Loc** errors classify correctly but localize incorrectly; (3) **Both** errors classify and localize incorrectly; (4) **Dupe** errors would be correct if not for a higher scoring detection; (5) **Bkg** errors detect background as foreground; (6) **Miss** errors are all undetected ground truths not already covered by Cls or Loc error.

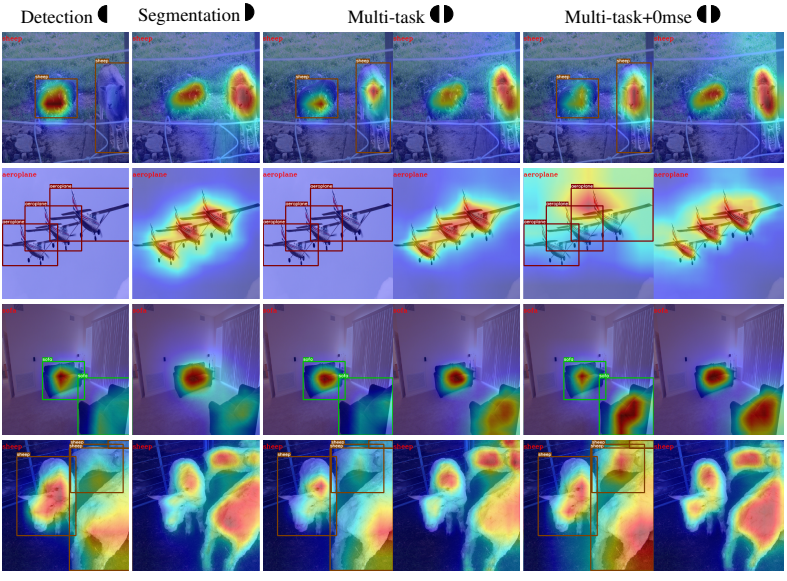

Figure 4: Class activation maps on the same layers for single task object detection and semantic segmentation networks (first 2 columns) and multi-task network (last 2 columns). Failed detected objects are recovered by multi-task networks and further with cross-task enforcing by KD (+0mse).

	AP50↑	Cls↓	Loc↓	Both↓	Dupe↓	Bkg↓	Miss↓	FP↓	FN ↓
◑	40.89	4.23	7.16	0.70	**0.37**	**1.45**	8.38	**13.12**	16.58
◐	**44.99**	**3.75**	**6.08**	**0.58**	0.43	1.46	**6.54**	14.02	**13.09**
Δ	4.10	-0.48	-1.08	-0.12	0.06	0.01	-1.84	0.90	-3.49
◑	47.27	2.69	6.81	0.56	0.37	1.07	8.57	9.55	15.12
◐	**51.43**	**2.46**	**6.06**	**0.51**	**0.31**	1.07	**6.39**	**9.47**	**12.30**
Δ	4.16	-0.23	-0.75	-0.05	-0.06	0	-2.18	-0.08	-2.82

Table 3: TIDE analysis of detection from RN18+FPN (top) and RN50+PAFPN (bottom).

4.2 Single-task learning and multi-task learning

In this experiment, we confirm the benefit of multi-task training for partially annotated data where each training example is only annotated for a single task. Possible data exploitation includes (1) training a single-task network with the provided annotated data; (2) training a single-task network by finetuning one pretrained for the other task; (3) training a single-task network with provided annotated data and pseudo labels generated by a semi-supervised learning method for the other task's data; and (4) training a multi-task network with 2 decoders. Except for (1), all other strategies involve the data from the other task in the training process: (2) is a standard transfer learning approach and (3) formulates as a weak-supervised problem. The ReCAM method [5] is applied to generate semantic mask for images from the detection subset ◖. For the transfer learning approach, we also include experiments where the pretrained backbone and neck are kept frozen during finetuning to show the compatibility of the features extracted for one task to the other. The results are shown in Table 1.

		AP50↑	Cls↓	Loc↓	Both↓	Dupe↓	Bkg↓	Miss↓	FP↓	FN ↓
STL	◖	**42.81**	4.71	**6.17**	0.73	0.40	1.61	**6.70**	16.40	**12.85**
MTL	◑	40.89	**4.23**	7.16	**0.70**	**0.37**	**1.45**	8.38	**13.12**	16.58
Δ		-1.92	-0.48	0.99	-0.03	-0.03	-0.16	1.68	-3.28	3.73
STL	◖	**50.22**	2.91	**6.45**	**0.54**	**0.34**	1.25	**6.48**	10.51	**13.21**
MTL	◑	47.27	**2.69**	6.81	0.56	0.37	**1.07**	8.57	**9.55**	15.12
Δ		-2.95	-0.22	0.36	0.02	0.03	-0.18	2.09	-0.96	1.91

Table 4: TIDE analysis of detection from RN18+FPN (top) and RN50+PAFPN (bottom).

It could be seen that the performance of multi-task learning even with partially annotated data are highest for all settings. The results when the two tasks are alternated every iteration seems to perform slightly better than every epoch. We also include the results when an image is annotated with both task, *i.e*. fully supervised setting with all annotations from [12] for reference (the results are not comparable as there are 10,476 images with both task annotations). Even with fewer effective images, optimizing both tasks shows superior performance. Training a semantic network with joint annotated data and pseudo-label generated by ReCAM method does not help even when compared to single task learning.

Regarding transfer learning scenario, finetuning the whole network pretrained with the other task improves over single-task learning while finetuning with frozen encoders plunges, showing the incompatible features learned by one task to the other. Class activation maps generated by Grad-CAM [30] are shown in Figure 4 for single and multi-task networks at the same layer output by the encoder. The multi-task activation seems spreading out for both detection and segmentation and could recover a miss-detected object.

Multi-task with fewer data We extend the study by adjusting the number of training data used for multi-task learning. Half number of images from one task subset are randomly removed while retaining those of the other. The results are shown in Table 2.

Although reducing training data size takes a great toll on the respective single task performance, the multi-task results seem to have less impact, especially for ResNet50+PAFPN with semantic segmentation task. From a multi-task point of view, reducing detection set also affects semantic segmentation performance, especially for RN18+FPN, while reducing segmentation data does not seem to affect the performance of the detection counterpart.

TIDE analysis on Table 3 shows that the most contribution to the difference between ◑ and ◔ is FN, especially the missing objects (Miss, Δ=-1.84, -2.18), then faulty localization with correct classification (Loc, Δ=-1.08, -0.75). The Classification error (Cls) seems to be affected at a lesser degree (Δ=-0.48, -0.23). As the semantic segmentation masks do not contain precise locations of object instances, even when VOC has limited instances per image, using semantic masks helps more with classification and less with localization. This is confirmed in Table 4 where STL trained with full detection data ◖ is compared to MTL with half ◑. Although inferior in general performance, MTL with half detection has lower classification error (Cls,Δ=-0.48,-0.22) background confusion (Bkg,Δ=-0.16,-0.18), and generally FP (Δ=-3.28,-0.96).

Multi-task with different category sets To understand the interrelationships between object detection and semantic segmentation, we gradually deviate one task from the other and

	Detection (20 classes)		Segmentation (4 classes)	
	RN18+FPN	RN50+PAFPN	RN18+FPN	RN50+PAFPN
Single task	42.81	50.22	78.47	81.82
Multi-task	**44.48**	**50.38**	**79.32**	**81.89**

Table 5: Single-task and multi-task performance when the tasks have different label sets. The performance gap decreases yet still in favor of multi-task learning.

	Detection (VOC)		Segmentation (Cityscapes)	
	RN18+FPN	RN50+FPN	RN18+FPN	RN50+FPN
Single task	**38.688**	**44.683**	**71.389**	**72.398**
Multitask	37.531	39.910	69.481	70.247

Table 6: Single tasks and multi-task performance when the tasks are from different domains. Jointly learning 2 tasks from different domains does not help but hurt the performance.

observe the performance differences. In this experiment, the category set for semantic segmentation is modified. Various VOC semantic categories are merged into an "abstract" class representing the group of the original labels such as the *vehicle* group (from aeroplane, bicycle, boat, bus, car, motorbike, and train), *animal* (from bird, cat, cow, dog, horse, and sheep), *furniture* (bottle, chair, dining table, potted plant, sofa, and TV monitor), and *person* as a group in itself.

Arranging different classes into the same group arrives at a semantic segmentation task that aims to learn entirely different concepts from the object detection task. The results are shown in Table 5. Although the performance distance between multi-task and single-task is shortened as each task has to cope with its own concept targets, multi-task still has its superiority. There is a diminishing return with higher capacity architectures.

Multi-task with out-of-domain data In this experiment, the two tasks are further pushed to different data domains. To that end, the semantic segmentation images are taken from the Cityscapes dataset [8] with 7 classes. Some of the classes are shared between the two datasets, such as car, human, vegetation. The results in Table 6 show that jointly learning data from different domains worsens the multi-task performance. It is not surprising as the data belong to different distributions with different semantic concept targets, the jointly learned features from one task is not helping but impede the other's learned features.

4.3 Knowledge distillation

In this section, knowledge distillation (KD) is used to enact joint-task training for partially annotated data and multi-task learning. To that end, the ResNet50+PAFPN architecture is used as the teacher model and results of the student ResNet18+FPN are reported. The task-specific heads are kept the same for the two networks. The teacher-student parameter ratio is 1.61. Unless stated otherwise, the teachers are initialized with the corresponding weights for single tasks from the previous experiments and stay frozen during the training of the students. The results are shown in Table 7. By simply forcing the student encoders to imitate the output of the teachers, the corresponding results are improved, confirming the benefit of knowledge distillation. Distilling the encoder neck features using one task's teacher while training the

Training	Detection	Segmentation
Single task	42.907	65.291
+ KD	44.982	67.375
Multi-task	45.678	67.310
+ 1mse	45.989	69.126
+ 0mse	47.337	**70.056**
+ 2mse	**47.611**	69.911

Table 7: Adding feature imitation knowledge distillation. For multi-task learning, the distilled features can be on the task with (1mse), or without annotations (0mse), or both (2mse). The performances are in favor for (0mse) and (2mse).

	AP50↑	Cls↓	Loc↓	Both↓	Dupe↓	Bkg↓	Miss↓	FP↓	FN ↓
STK	42.81	**4.71**	6.17	0.73	**0.40**	**1.61**	6.70	**16.40**	12.85
+MSE	**44.98**	4.94	**5.70**	**0.67**	0.43	1.71	**5.39**	17.32	**11.67**
Δ	2.17	0.23	-0.47	-0.06	0.03	0.10	-1.31	0.92	-1.18
MTL	44.78	3.70	6.72	0.73	**0.40**	1.61	6.13	14.90	12.43
+MSE	**47.61**	**3.10**	**6.17**	**0.65**	0.50	1.61	**5.56**	**13.47**	**11.66**
Δ	2.83	-0.60	-0.55	-0.08	0.1	0	-0.57	-1.43	-0.77

Table 8: TIDE analysis of detection results with (+MSE) and without knowledge distillation.

other task's head using provided ground truths (0mse) shows favorable results over distilling the same task that has annotations (1mse). The results are even higher when both tasks are optimized simultaneously in the fully-supervised scenario in Table 1, showing the benefit of multi-task data exploitation and joint-task optimization using knowledge distillation.

Table 8 shows the errors reduced by KD for both single-task learning (STL) and multi-task learning (MTL). It could be seen that among the first 6 errors, KD helps the most with Miss detection and Cls. The effects, however, are not the same for STL and MTL: STL benefits substantially from Miss error (Δ=-1.31), reflecting also in FN (Δ=-1.18) while MTL benefits equally on Cls, Loc, and Miss, emphasizing on FP (Δ=-1.43). This suggests the more balancing performance of MTL over STL and the improvement of robustness by KD.

5 Conclusion

The paper studies the possibility for jointly learning object detection and semantic segmentation using partially annotated data. As there are no images with both task annotations, optimization is alternated between the tasks. The experiments show that by alternating every iteration, the networks could pick up useful information from the other task's data and improve over the single-task cases. Knowledge distillation could be an alternative method allowing to learn interrelationship between one task and the other's features.

Acknowledgments

This work was supported by the SAD 2021 ROMMEO project (ID 21007759) and the ANR AI chair OTTOPIA project (ANR-20-CHIA-0030).

References

[1] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *European Conference on Computer Vision (ECCV)*, 2020.

[2] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated Search for Resource-Efficient Branched Multi-Task Networks. In *British Machine Vision Conference (BMVC)*, 2020.

[3] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring Relational Context for Multi-Task Dense Prediction. In *IEEE/CVF Proceedings of International Conference of Computer Vision (ICCV)*, 2021.

[4] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth Prediction without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[5] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class Re-Activation Maps for Weakly-Supervised Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Zhihao Chen, Lei Zhu, Liang Wan, Song Wang, Wei Feng, and Pheng-Ann Heng. A Multi-Task Mean Teacher for Semi-Supervised Shadow Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. BAM! Born-Again Multi-Task Networks for Natural Language Understanding. In *ACL*, 2019.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset. volume 3, 2016.

[9] M Everingham, L Van∼Gool, C K I Williams, J Winn, and A Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 88(2), 2010.

[10] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born Again Neural Networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, 2018.

[11] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling Object Detectors via Decoupled Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE/CVF Proceedings of International Conference of Computer Vision (ICCV)*, 2011.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *IEEE/CVF Proceedings of International Conference of Computer Vision (ICCV)*, 2017.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Abdullah-Al-Zubaer Imran, Chao Huang, Hui Tang, Wei Fan, Yuan Xiao, Dingjun Hao, Zhen Qian, and Demetri Terzopoulos. Partly Supervised Multi-Task Learning. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[16] Paul Jaccard. The distribution of the Flora in the Alpine Zone. 1. *New Phytologist*, 1912.

[17] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic Feature Pyramid Networks. In *IEEE/CVF Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Wei-Hong Li and Hakan Bilen. Knowledge Distillation for Multi-task Learning. In *European Conference on Computer Vision workshop (ECCVw)*, 2020.

[19] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Learning Multiple Dense Prediction Tasks from Partially Annotated Data. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[20] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal Representations: A Unified Look at Multiple Task and Domain Learning. *arXiv preprint arXiv:2204.02744*, 2022.

[21] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] Shikun Liu, Edward Johns, and Andrew J Davison. End-To-End Multi-Task Learning With Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD : Single Shot MultiBox Detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Yao Lu, Soren Pirk, Jan Dlabal, Anthony Brohan, Ankita Pasad, Zhao Chen, Vincent Casser, Anelia Angelova, and Ariel Gordon. Taskology: Utilizing Task Relations at Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[27] Hoàng-Ân Lê and Minh-Tan Pham. Self-Training and Multi-Task Learning for Limited Data: Evaluation Study on Object Detection. In *IEEE/CVF Proceedings of International Conference of Computer Vision workshop (ICCVw)*, 2023.

[28] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: Towards Balanced Learning for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. ThunderNet: Towards Real-time Generic Object Detection on Mobile Devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[31] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[32] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-Training With Noisy Student Improves ImageNet Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust Learning Through Cross-Task Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] Heng Zhang, Elisa Fromont, Sebastien Lefevre, and Bruno Avignon. PDF-Distil: including Prediction Disagreements in Feature-based Distillation for object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

[35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *IEEE/CVF Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] Rui Zhu, Shifeng Zhang, Xiaobo Wang, Longyin Wen, Hailin Shi, Liefeng Bo, and Tao Mei. ScratchDet: Training Single-Shot Object Detectors From Scratch. In *IEEE/CVF Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking Pre-Training and Self-Training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.