# Knowledge Assembly: Semi-Supervised Multi-Task Learning from Multiple Datasets with Disjoint Labels

Federica Spinola, Philipp Benz, Minhyeong Yu, Tae-hoon Kim
Deeping Source Inc., Seoul, Republic of Korea
{federica.spinola, philipp.benz, minhyeong.yu, pete.kim}@deepingsource.io

## Abstract

*In real-world scenarios we often need to perform multiple tasks simultaneously. Multi-Task Learning (MTL) is an adequate method to do so, but usually requires datasets labeled for all tasks. We propose a method that can leverage datasets labeled for only some of the tasks in the MTL framework. Our work, Knowledge Assembly (KA), learns multiple tasks from disjoint datasets by leveraging the unlabeled data in a semi-supervised manner, using model augmentation for pseudo-supervision. Whilst KA can be implemented on any existing MTL networks, we test our method on jointly learning person re-identification (reID) and pedestrian attribute recognition (PAR). We surpass the single task fully-supervised performance by* $4.2\%$ *points for reID and* $0.9\%$ *points for PAR.*

## 1. Introduction

With deep learning making its way into real products, it is crucial to deploy models that can efficiently and effectively understand the world around us. In scenarios like autonomous driving or surveillance, multiple tasks need to be solved simultaneously. Whilst networks can be trained for each individual task, this can lead to large inference times. In contrast, multi-task learning (MTL) aims to share information between tasks by learning them jointly, usually within a single network. Thus it can improve the individual tasks' performance [26] and overall inference time. MTL has been successfully applied to many computer vision tasks [13]. Nevertheless, it usually considers a single dataset labeled for all tasks [2, 16], whereas in real-world applications we are often faced with a scarcity of labeled data for the tasks at hand: each dataset is only labeled for one of the tasks. Moreover, the human effort required to label all the datasets with the missing annotations would be time-consuming and impractical. Therefore, there is a need to tackle this issue and leverage partially labeled datasets in a MTL framework.

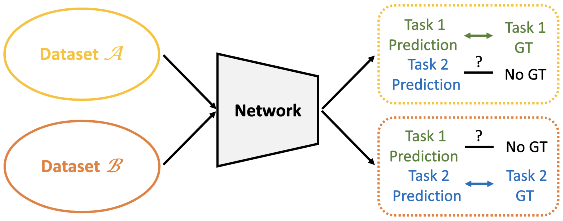

Figure 1. **Problem Definition.** The aim is to jointly learn two tasks, 1 and 2, given two datasets, $\mathcal{A}$ and $\mathcal{B}$ that only have labels for one of the tasks: dataset $\mathcal{A}$ for task 1 and dataset $\mathcal{B}$ for task 2.

In this work we consider the problem of learning multiple tasks from disjoint datasets, such that, for any given input image, the supervisory information is available for at least one but not all tasks at hand. An illustration of our problem setting is shown in Figure 1. In the considered setting the training data can come from different domains, the domains can be represented in an unbalanced way, and supervision signals can be sparse. We could devise several methods to compute only supervised losses when ground-truth labels are available. However, we would discard a lot of information from the images that could otherwise be leveraged to guide the learning of the other tasks. Semi-supervised learning (SSL) demonstrated that using additional unlabeled data can improve the network's performance and enhance its generalization capabilities [29]. Therefore, we propose Knowledge Assembly (KA), which aims to use all the data available within a MTL framework by leveraging SSL techniques.
Our contributions are the following:

- We propose a novel framework to train a single model for MTL by leveraging disjoint datasets.
- We suggest to utilize the unlabeled data for each task by introducing a semi-supervised consistency loss resulting from model augmentation.
- We evaluate our framework on the person re-identification (reID) and pedestrian attribute recognition (PAR) tasks, and show that our method outperforms all baselines.

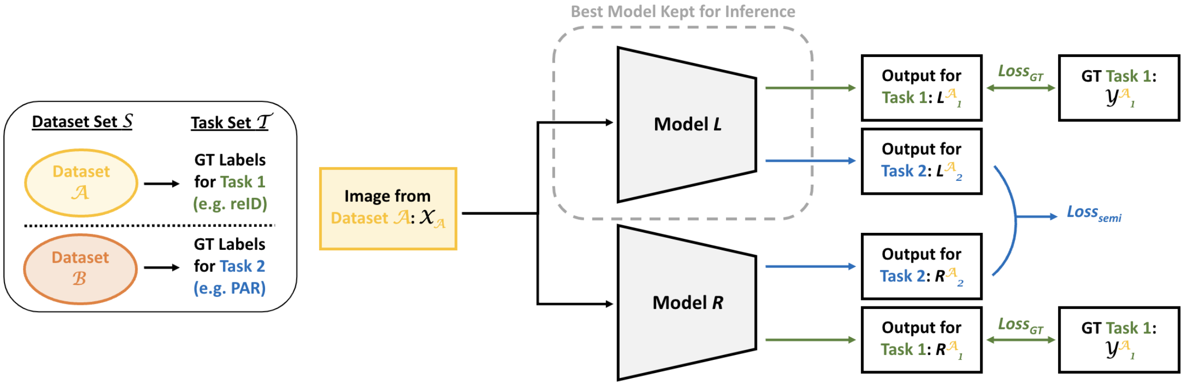
Best Model Kept for Inference
Dataset Set $\mathcal{S}$
Task Set $\mathcal{T}$
Dataset $\mathcal{A}$
GT Labels for Task 1 (e.g. reID)
Dataset $\mathcal{B}$
GT Labels for Task 2 (e.g. PAR)
Image from Dataset $\mathcal{A}$: $\mathcal{X}_{\mathcal{A}}$
Model L
Model R
Output for Task 1: $L^{\mathcal{A}}_1$
Output for Task 2: $L^{\mathcal{A}}_2$
Output for Task 2: $R^{\mathcal{A}}_2$
Output for Task 1: $R^{\mathcal{A}}_1$
$Loss_{GT}$
$Loss_{semi}$
$Loss_{GT}$
GT Task 1: $\mathcal{Y}^{\mathcal{A}}_1$
GT Task 1: $\mathcal{Y}^{\mathcal{A}}_1$

Figure 2. **KA Method Overview.** For training we initialize two instances $L$ and $R$ of the same model. A mini-batch containing both datasets $\mathcal{A}$ and $\mathcal{B}$ is fed into the networks, and outputs for both tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ are predicted. If the corresponding input has a ground-truth (GT) label then a supervised loss $\mathcal{L}^{GT}$ is computed, else a semi-supervised loss $\mathcal{L}^{semi}$ is calculated. The latter loss ensures the outputs' consistency between the two network. In this Figure we show the forward path for one instance of dataset $\mathcal{A}$ (in yellow) that only has GT labels for task $\mathcal{T}_1$ (in green). During inference, the best of the two models is kept. Best viewed in color.

## 2. Related Work

**Multi-Task Learning.** The MTL framework is being successfully applied to multiple computer vision domains, especially in the realm of scene understanding [10, 23, 27]. MTL is also used in the domain of face analysis [9] or person identification [14, 20]. Most MTL techniques assume a single dataset labeled for all the relevant tasks, or use knowledge transfer across tasks (to learn more about MTL please refer to [19]). However, they do not tackle the simultaneous learning of multiple tasks from different datasets with partial labels.

**Semi-Supervised Learning.** In deep learning, the need for quality labeled data is crucial. However, more often than not, we encounter tasks for which labeled data is hard to get and cumbersome to annotate. SSL is needed to train models with partially unlabeled data [18]. Successful methods for semi-supervised object classification [21, 25] consist of using a self-consistency loss to ensure that a network outputs similar predictions for different input image augmentations. Conversely, [1] develops a semi-supervised framework based on model augmentation for semi-supervised semantic segmentation. Instead of using weak augmented images as pseudo-labels for strong augmented images, [1] initializes two instances of the same network and uses the output from one network to supervise the other network (and vice-versa). However, semi-supervised learning is usually restricted to scenarios where the labeled and unlabeled data come from the same distribution, and few works successfully manage to leverage semi-supervised learning in cross-domain settings [5]. Moreover, the above-mentioned works do not tackle the problem of semi-supervised learning in a MTL framework.

**Multiple Datasets with Disjoint Labels.** A few works tackle the problem of learning multiple tasks from disjoint datasets [4, 7, 8, 11, 24]. The general approach is to use a MTL framework and deal with the encountered forgetting effect by leveraging knowledge distillation [4, 7, 11]. However, these methods do not leverage the unlabeled data from one of the tasks to improve the other task's performance. Thus, other works [8, 24] use adversarial learning jointly with SSL to learn multiple tasks from disjoint datasets, whilst utilizing all the data at hand for each task. In our work, we also focus on the problem of MTL from disjoint datasets, and leverage unlabeled data using semi-supervised learning. Moreover, our training strategy can directly be applied to existing networks and does not require additional adversarial network designs.

## 3. Method

In this Section, we will present our method, which is illustrated in Figure 2.

### 3.1. Problem Definition

Following a similar setup to that of [24], we consider a set of tasks $\mathcal{T}$, an input space consisting of a set of partially labeled datasets $\{\mathcal{X}_s\}_{s\in\mathcal{S}}$ and an output space consisting of a set of task-specific outputs $\{\mathcal{Y}_t\}_{t\in\mathcal{T}}$. $\mathcal{S}$ is the set of all datasets, where each dataset has labels for at least one of the tasks but can have missing labels for all the other tasks. We wish to train a single model $\mathcal{F}(x;\Theta,\theta_t) : \mathcal{X}_s \rightarrow \mathcal{Y}_t \; \forall s \in \mathcal{S}, t \in \mathcal{T}$, where $\Theta$ are the shared network parameters and $\theta_t$ are the task specific network parameters. In this work we consider two input datasets ($\mathcal{S}_{\mathcal{A}}$ and $\mathcal{S}_{\mathcal{B}}$) and two tasks ($\mathcal{T}_1$ and $\mathcal{T}_2$), where dataset $\mathcal{S}_{\mathcal{A}}$ has only ground-truth (GT) labels $\mathcal{Y}_1^{\mathcal{A}}$ for task $\mathcal{T}_1$ and $\mathcal{S}_{\mathcal{B}}$ has only GT labels $\mathcal{Y}_2^{\mathcal{B}}$ for $\mathcal{T}_2$.

### 3.2. Model Augmentation

Our method is designed within a SSL framework similar to the one proposed by [1]: consistency regularization through network perturbation. Indeed, consistency regularization has been proven to be an effective learning signal for SSL as it forces outputs to be consistent under different perturbations [1,21,25]. We initialize two instances of the same network, $\mathcal{F}_L(x;\Theta^L)$ and $\mathcal{F}_R(x;\Theta^R)$, and feed any input image through both networks. The semi-supervised consistency loss is designed as follows: the output $L$ of model $\mathcal{F}_L$ is used as a pseudo-label to supervise the output $R$ of model $\mathcal{F}_R$, and vice-versa.

More specifically, we include this semi-supervised training strategy within a MTL framework. Thus for an input $\mathcal{X}^{\mathcal{A}}$ ($\mathcal{X}^{\mathcal{B}}$) there will be two outputs for each task (four in total): $L_1^{\mathcal{A}}$ ($L_1^{\mathcal{B}}$) and $R_1^{\mathcal{A}}$ ($R_1^{\mathcal{B}}$) for task $\mathcal{T}_1$ (green path in Figure 2) from models $\mathcal{F}_L$ and $\mathcal{F}_R$, and $L_2^{\mathcal{A}}$ ($L_2^{\mathcal{B}}$) and $R_2^{\mathcal{A}}$ ($R_2^{\mathcal{B}}$) for task $\mathcal{T}_2$ (blue path in Figure 2) from models $\mathcal{F}_L$ and $\mathcal{F}_R$. Input $\mathcal{X}^{\mathcal{A}}$ ($\mathcal{X}^{\mathcal{B}}$) has an associated GT label $\mathcal{Y}_1^{\mathcal{A}}$ ($\mathcal{Y}_2^{\mathcal{B}}$) for task $\mathcal{T}_1$ ($\mathcal{T}_2$), with which a supervised loss $\mathcal{L}_1^{GT}$ ($\mathcal{L}_2^{GT}$) can be computed for both models. However, $\mathcal{X}^{\mathcal{A}}$ ($\mathcal{X}^{\mathcal{B}}$) does not have any GT label for task $\mathcal{T}_2$ ($\mathcal{T}_1$). Therefore, a semi-supervised consistency loss $\mathcal{L}_2^{semi}$ ($\mathcal{L}_1^{semi}$) is computed by using the output $L_2^{\mathcal{A}}$ ($L_1^{\mathcal{B}}$) as pseudo-supervision for $R_2^{\mathcal{A}}$ ($R_1^{\mathcal{B}}$), and by using the output $R_2^{\mathcal{A}}$ ($R_1^{\mathcal{B}}$) as pseudo-supervision for $L_2^{\mathcal{A}}$ ($L_1^{\mathcal{B}}$).

### 3.3. Objective function

The networks are trained by computing losses over mini-batches. Each mini-batch is created by sampling data from both datasets: samples $\mathcal{S}_{\mathcal{A}}^b$ from dataset $\mathcal{S}_{\mathcal{A}}$ and samples $\mathcal{S}_{\mathcal{B}}^b$ from dataset $\mathcal{S}_{\mathcal{B}}$. We compute all the relevant task-specific supervised and semi-supervised losses. The total supervised loss $\mathcal{L}^{GT}$ is calculated as:

$$\mathcal{L}^{GT} = \frac{1}{|\mathcal{S}_{\mathcal{A}}^b|}\sum_{s\in\mathcal{S}_{\mathcal{A}}^b}(\mathcal{L}_1^{GT}(L_1^{\mathcal{A}},\mathcal{Y}_1^{\mathcal{A}}) + \mathcal{L}_1^{GT}(R_1^{\mathcal{A}},\mathcal{Y}_1^{\mathcal{A}})) + \frac{1}{|\mathcal{S}_{\mathcal{B}}^b|}\sum_{s\in\mathcal{S}_{\mathcal{B}}^b}(\mathcal{L}_2^{GT}(L_2^{\mathcal{B}},\mathcal{Y}_2^{\mathcal{B}}) + \mathcal{L}_2^{GT}(R_2^{\mathcal{B}},\mathcal{Y}_2^{\mathcal{B}})) \quad (1)$$

The consistency loss on unlabeled data $\mathcal{L}_u^{semi}$ is computed as:

$$\mathcal{L}_u^{semi} = \frac{1}{|\mathcal{S}_{\mathcal{A}}^b|}\sum_{s\in\mathcal{S}_{\mathcal{A}}^b}(\mathcal{L}_2^{semi}(L_2^{\mathcal{A}},R_2^{\mathcal{A}}) + \mathcal{L}_2^{semi}(R_2^{\mathcal{A}},L_2^{\mathcal{A}})) + \frac{1}{|\mathcal{S}_{\mathcal{B}}^b|}\sum_{s\in\mathcal{S}_{\mathcal{B}}^b}(\mathcal{L}_1^{semi}(L_1^{\mathcal{B}},R_1^{\mathcal{B}}) + \mathcal{L}_1^{semi}(R_1^{\mathcal{B}},L_1^{\mathcal{B}})) \quad (2)$$

We also compute the consistency loss for labeled data $\mathcal{L}_l^{semi}$. Therefore the total semi-supervised loss becomes: $\mathcal{L}^{semi} = \mathcal{L}_u^{semi} + \mathcal{L}_l^{semi}$.

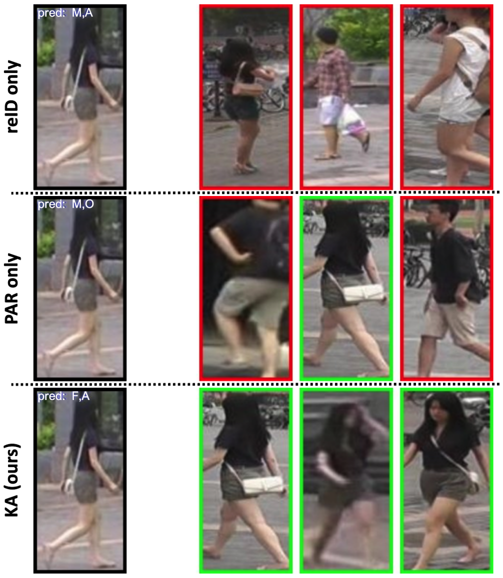

Figure 3. **Qualitative Results on Market1501.** We show the top three reID results (colored on the right) for a query image (black on the left). Correct reID is in green and wrong reID is in red. The query's attributes are also predicted. We display gender (M: male, F: female) and age (Y: young, A: adult, O: old). Best viewed in color.

Finally, the total training objective for each batch is the following:

$$\mathcal{L} = \mathcal{L}^{GT} + \lambda\mathcal{L}^{semi} \quad (3)$$

Where $\lambda$ is a hyperparameter to balance the supervised and consistency losses.

## 4. Experiments

Whilst our method can be applied to any two or more related tasks, we implement it to the tasks of person reID and PAR ($\mathcal{T}_1$ corresponds to reID and $\mathcal{T}_2$ to PAR). The experimental details and results are outlined in Section 4 below.

### 4.1. Datasets and Metrics

For all our experiments, we train our KA on the Market1501 [28] (contains only reID GT labels) and the PA100K [15] (contains only PAR GT labels from the UPAR dataset [22]) training datasets. We then evaluate reID on the test set of Market1501 and evaluate PAR on the test set of PA100K.

For person reID, we evaluate our performance based on the mean Average Precision (mAP) and the Cumulative Matching Characteristics (CMC). For CMC we report the rank-1, rank-5 and rank-10 accuracies. We evaluate the PAR's performance based on the label mean accuracy (ma), and the instance Precision, Recall and F1 scores.

Table 1. **Quantitative Experiment Results on ResNet-18.** We train our method on Market1501 and PA100K and evaluate reID and PAR on the test-sets of Market1501 and PA100K respectively. Best results are shown in **bold**.

| *ResNet-18* | Market1501 (reID) | | | | PA100K (PA) | | | |
|---|---|---|---|---|---|---|---|---|
| | mAP (%) | rank-1 (%) | rank-5 (%) | rank-10 (%) | ma (%) | F1 (%) | Prec. (%) | Rec. (%) |
| reID only | 52.6 | 72.2 | 88.2 | 92.2 | - | - | - | - |
| PAR only | - | - | - | - | 74.3 | 82.3 | 84.4 | 80.3 |
| reID ssl | 48.9 | 69.6 | 86.2 | 91.1 | - | - | - | - |
| PAR ssl | - | - | - | - | **75.1** | 83.1 | 85.0 | **81.3** |
| MTL baseline | 56.4 | 75.1 | **90.1** | **93.7** | 74.3 | 82.3 | 84.4 | 80.3 |
| KA (ours) | **56.8** | **75.9** | 89.8 | **93.7** | 74.4 | **83.2** | **85.2** | **81.3** |

## 4.2. Implementation Details

Our method is designed for any existing MTL architecture. Therefore, we experiment with a very basic network architecture: a ResNet-18 [6] backbone and two task-specific heads composed of a $1 \times 1$ convolution, two fully-connected layers and a final classifier. The network outputs a feature vector and a classification vector for each task. Pre-trained Imagenet [3] weights are used as a starting point to train the MTL network. We train our model $\mathcal{F}(.)$ for 100 epochs with batch-size 64, using the Adam [12] optimizer with the initial learning rate $lr = 0.0003$ and a cosine learning rate scheduler. $\lambda = 1.0$ from Equation 3.

We use an ID cross-entropy (CE) loss and an attribute binary cross-entropy (BCE) loss for the supervised reID ($\mathcal{L}_1^{GT}$) and PAR losses ($\mathcal{L}_2^{GT}$) respectively. For the reID consistency loss $\mathcal{L}_1^{semi}$, we compute a triplet loss on the reID output feature vectors, and a BCE loss on the dataset prediction output (part of the reID's classification output). For the PAR consistency loss $\mathcal{L}_2^{semi}$, we compute a DICE loss [17] on the PAR classification outputs and a triplet loss on the PAR output feature vectors.

## 4.3. Baselines

We define five baselines to compare our KA against.
***reID only*:** Model trained for reID with only reID data in a fully-supervised manner using $\mathcal{L}_1^{GT}$.
***PAR only*:** Model trained for PAR with only PAR data in a fully-supervised manner using $\mathcal{L}_2^{GT}$.
***reID ssl*:** Model trained for reID with reID and unlabeled data in a semi-supervised manner using $\mathcal{L}_1^{GT}$ and $\mathcal{L}_1^{semi}$.
***PAR ssl*:** Model trained for PAR with PAR and unlabeled data in a semi-supervised manner using $\mathcal{L}_2^{GT}$ and $\mathcal{L}_2^{semi}$.
***MTL baseline*:** Model trained for reID and PAR in a fully-supervised manner using only $\mathcal{L}_1^{GT}$ and $\mathcal{L}_2^{GT}$.

## 4.4. Results

Our main quantitative results are shown in Table 1. The reID's mAP improves by 1.8% points when trained alongside PAR (*MTL baseline*). Its accuracy further improves by 0.4% points when using our KA method of combining MTL and SSL to leverage unlabeled data. However, the performance drops for our *reID ssl* baseline. We conjecture that this is due to different domain and unbalanced data. The PAR's F1 score improves by 0.8% points when training with additional unlabeled data (*PAR ssl* baseline). Its score further improves by 0.1% points when using our KA method of combining SSL with MTL. In summary, our KA method outperforms all baselines for most metrics, meaning that training related tasks jointly and leveraging available but unlabeled data improves all tasks' performances. The qualitative results shown in Figure 3 further support this statement. Indeed, we notice that our KA method correctly predicts attributes (Female (F), Adult (A)) and re-identifies the person, unlike the single-task methods.

In Table 2 we outline our ablation study results. Firstly we show that using model augmentation instead of image augmentation [21] as a SSL framework yields higher performance for both tasks. Moreover, the network reaches its highest performance (mAP = 56.8% for reID and F1 = 83.2% for PAR) when computing an additional triplet loss on feature vectors for pseudo-supervision (*KA w/ NetAug+Tri*), instead of only calculating a standard classification loss (*KA w/ NetAug*).

Table 2. **Ablation Study Results on ResNet-18.** We test different augmentation strategies and losses for the semi-supervised loss computation. Best results are shown in **bold**.

| *ResNet-18* | Market1501 (reID) | | PA100K (PA) | |
|---|---|---|---|---|
| | mAP (%) | rank-1 (%) | ma (%) | F1 (%) |
| KA w/ ImgAug | 54.1 | 73.3 | 73.8 | 81.9 |
| KA w/ NetAug | 55.8 | 74.5 | 74.2 | 82.7 |
| KA w/ NetAug+Tri | **56.8** | **75.9** | **74.4** | **83.2** |

# 5. Conclusion

In this work we focused on training multiple tasks for real-world applications, where datasets are labeled for only some of the tasks. Our method, KA is capable of utilizing all the available data by leveraging semi-supervised consistency regularization from model perturbation. The results show that learning multiple tasks with labeled and unlabeled data using KA leads to better performances than training networks for single tasks in a fully supervised manner. Future work would explore training different and larger combinations of tasks and datasets.

## References

[1] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[2] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[3] 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 (CVPR)*. IEEE, 2009. 4

[4] Prithviraj Dhar, Amit Kumar, Kirsten Kaplan, Khushi Gupta, Rakesh Ranjan, and Rama Chellappa. Eyepad++: A distillation-based approach for joint eye authentication and presentation attack detection using periocular images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[5] Ran Gu, Jingyang Zhang, Guotai Wang, Wenhui Lei, Tao Song, Xiaofan Zhang, Kang Li, and Shaoting Zhang. Contrastive semi-supervised learning for domain adaptive segmentation across similar anatomical structures. *IEEE Transactions on Medical Imaging*, 42, 2022. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[7] Yan Hong, Li Niu, Jianfu Zhang, and Liqing Zhang. Beyond without forgetting: Multi-task learning for classification with disjoint datasets. In *International Conference on Multimedia and Expo (ICME)*. IEEE, 2020. 2

[8] Chao Huang, Hui Tang, Wei Fan, Yuan Xiao, Dingjun Hao, Zhen Qian, Demetri Terzopoulos, et al. Partly supervised multi-task learning. In *International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020. 2

[9] Zhizhong Huang, Junping Zhang, and Hongming Shan. When age-invariant face recognition meets face age synthesis: A multi-task learning framework. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[10] 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 (CVPR)*, 2018. 2

[11] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, Youngjin Yoon, and In So Kweon. Disjoint multi-task learning between heterogeneous human-centric tasks. In *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018. 2

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[13] 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 (CVPR)*, 2017. 1

[14] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern recognition*, 95, 2019. 2

[15] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[17] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Iinternational Conference on 3D Vision (3DV)*. IEEE, 2016. 4

[18] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 2

[19] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[20] Arne Schumann and Rainer Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 2

[21] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33, 2020. 2, 3, 4

[22] Andreas Specker, Mickael Cormier, and Jürgen Beyerer. Upar: Unified pedestrian attribute recognition and person retrieval. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 3

[23] 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*, 44, 2021. 2

[24] Yufeng Wang, Yi-Hsuan Tsai, Wei-Chih Hung, Wenrui Ding, Shuo Liu, and Ming-Hsuan Yang. Semi-supervised multi-task learning for semantics and depth. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. 2

[25] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33, 2020. 2, 3

[26] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34, 2021. 1

[27] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Jiaxiang Shang, Tian Fang, and Long Quan. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In *Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020. 2

[28] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[29] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005. 1