Minimum commitment Loss for Semantic Segmentation with partial labels

Fabien Delattre Vishnu Balakrishnan

University of Massachusetts, Amherst
College of Information and Computer Sciences

Abstract

Producing semantic segmentation annotations requires a large amount of human effort. It is often challenging to use and combine publicly available datasets due to disparities in their taxonomies. In this paper, we express the problem of combining multiple heterogeneously labeled datasets as a partial labels problem: We associate every pixel with a set of candidate labels, only one of which is the correct label. We propose to use the Minimum Commitment loss, a simple, yet effective loss that does not set any preferences among the compatible labels. We found that our loss outperforms traditional methods such as Cross-Entropy with uniform targets, or Multi-task learning.

1. Introduction

The semantic segmentation task, also known as per-pixel classification, has seen significant progress in the past years due to deep learning techniques [21]. However, training high-capacity learning models such as deep Convolutional Neural Networks (CNNs) requires a large amount of labeled data. Producing such annotations for semantic segmentation is especially expensive in terms of human labor.

Therefore one may find it easier to combine multiple existing datasets to gather more training data. Due to the hierarchical nature of the semantic segmentation task, datasets often don't share the same taxonomies and this often leads to ambiguities that cannot be solved without a substantial amount of annotation work. In the case where there is no publicly available dataset with all the needed classes, it can also be required to combine datasets with different taxonomies. An example of that is shown in figure 1. Each original dataset taken separately wouldn't have been sufficient to be able to differentiate between the three classes. Therefore, combining the two original datasets is needed. We refer to the *vegetation* label as an ambiguous label since it is the union of two labels that belong to the expected taxonomy.

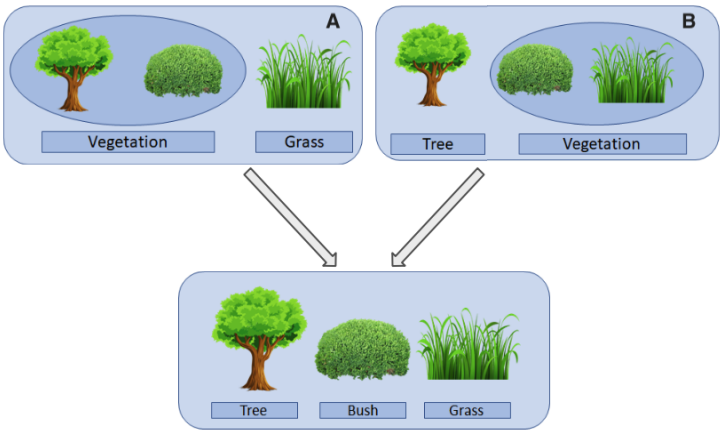

Figure 1: This is an example of ambiguities. Instead of manually annotating the *bush* class, we express the *vegetation* classes as sets of labels, i.e. {tree, bush} for the dataset A, and {bush, grass} for the dataset B.

Our goal in this paper is to avoid any additional annotation work when combining datasets that have different taxonomies. For that purpose, we express ambiguous annotations as sets of candidate labels. Instead of associating every pixel with a single label, we associate every pixel with a set of candidate labels, only one of which is correct. This classification problem is usually referred to as partial labels or multiple labels problem [10].

We propose to train a Convolutional Neural Network using a Minimum Commitment Loss [7]. To our knowledge, this is the first time this loss is used in the context of deep learning. We first compare the proposed loss against well studied losses such as Cross-Entropy or Binary Cross-Entropy. We then demonstrate the effectiveness of our method on Cityscapes [3] and BDD100k [20] datasets. We also compare it to other paradigms such as Multi-task learning and flat classification.

2. Related Work

The Partial labels problems are parts of the larger class of weakly-supervised learning problems. Weakly-supervised learning can be viewed as half-way between supervised learning (the exact labels are available) and unsupervised learning (no labels are available). The formulation and differences between the different learning settings are given below.

- In **partial labels** learning (also called **multiple labels** learning or **ambiguous labels** learning), each example is supplied with a set of candidate labels. Only one label among the candidate set is the correct label. Even if initially it was called **multiple labels** learning [10], it is now preferred to call it **partial labels** learning to avoid confusion with **multi-label** learning.

- In **supervised** learning, each example is associated with a single label. This can be viewed as a special case of **partial labels** learning where the candidate set contains only one label.

- In **semi-supervised** learning [22], some examples are associated with a single label (as in **supervised** learning) and some example are associated with no labels (as in **unsupervised** learning) This also can be formulated as a **partial labels** learning problem where the candidate set for the labeled examples contains one label and the candidate set for unlabeled examples contains all the labels.

- In **multi-label** learning, every example can be associated with multiple labels, all of them being correct.

Most of the methods for learning from ambiguously labeled data involve two types of approaches. The first is called the *identification-based* methods where the label confidence and model parameters are iteratively and alternatively updated. Several papers, [10, 13, 15, 12] have used this approach to estimate the model parameters and the true label, for ambiguously labeled data, mostly with the help of the expectation-maximization (EM) algorithm [5]. For example, [10] applies the EM algorithm to find which label among the given set is more appropriate. Starting with the assumption that every class label within the set is equally likely, they train a conditional model $p(y|x, \theta)$. Then, with the help of this conditional model, they estimate the label distribution $\hat{p}(y|x_i)$ for each data point. With these label distributions, they refit the conditional model $p(y|x, \theta)$ and so on.

On the other hand, the second approach called the *average-based* methods treat all the candidate labels equally, assuming they contribute equally to the trained classifier. These methods require the use of different loss functions to estimate the class probabilities given the partially labeled data [4, 9]. [2] explains the general conditions under which the probability of the true class given the observation can be estimated from training data with ambiguous class labels. To do so, they conceptualize losses as functions of ambiguous labels, and show that the capability to estimate class probabilities using a given loss depends on the relation of the ambiguous labels with the true class of the data.

Most of the work mentioned previously have dealt with ambiguously labeled within a single dataset. Some more recent research [6, 11, 16, 19] has been directed toward training a single model using multiple datasets for better learning and generalization. [6] uses a generalized version of the binary cross-entropy loss where the loss is normalized by the proportion of known labels rather than the total number of classes. [16] trains a Convolutional Neural Network called SMILE on multiple datasets, each of them corresponding to one class. During training, binary cross entropy loss is used to compute the loss for each class and the final loss is the summation of these losses. [19] proposes a novel loss function based on the Dice similarity coefficient to adaptively learn multi-organ information from heterogeneously labeled computed tomography (CT) abdominal scans. [11] uses a new loss function where during training, the model does not apply softmax on labels that are not available in a particular dataset but applies a sigmoid on the output of CNN for those labels.

A third approach is to use multiple classifiers. [14] trains multiple classifiers on the specific task of street scene semantic segmentation. They define the semantic hierarchy between the different datasets and train one classifier per dataset. Therefore, the network is able to handle different semantic level-of-detail and annotation types.

3. Formal description

Formally we define the problem as follows:
Let $x \in \mathcal{X}$ be an input where $\mathcal{X}$ is the set of all inputs. Let $\mathcal{Y} = \{y_1, y_2, ..., y_k\}$ be the set of all possible labels an input x can take. In the partially labeled dataset, every datapoint x_i is associated with a set of candidate labels $S_i = \{y_i \mid y_i \in \mathcal{Y}\}$ and only one of these labels is the true label for the datapoint x_i. Thus the dataset consists of the pairs (x_i, S_i) where $x_i \in \mathcal{X}$ is the input and $S_i \subseteq \mathcal{Y}$ is the set of labels for the x_i one of which is the true label and the algorithm does not know which label within S_i is the true label. Given such a dataset, the task of learning from ambiguous label is to learn a function, $f : \mathcal{X} \to \mathcal{Y}$, such that $f(x_i) \to y_i$ with a high probability, where y_i is the true label for x_i and $y_i \in S_i$.

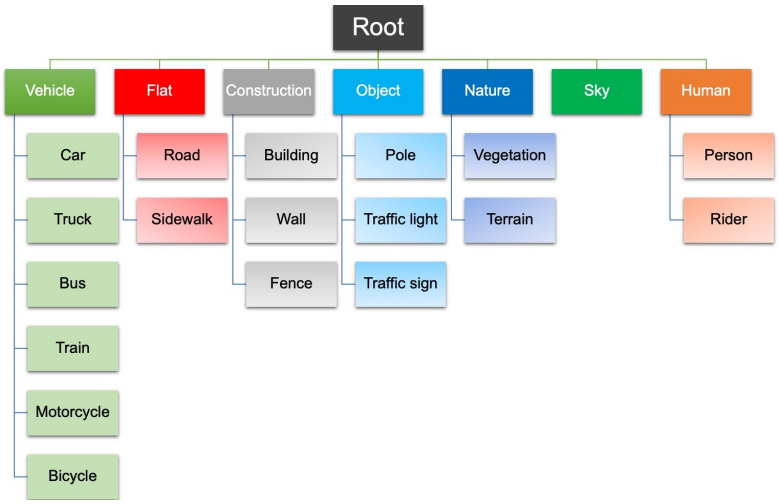

Figure 2: Semantic label hierarchy for the 19 classes and 7 categories in the dataset

4. Loss functions

4.1. Cross-entropy

A common loss used for semantic segmentation, and more generally for classification problems is the Cross-Entropy loss (CE) given by

$$\mathcal{L}_{CE} = -\sum_{i=0}^{N} y_i \log(p_i)$$

where N is the number of classes, y_i is the target probability and p_i is the probability output by the model. p_i is given by the softmax function defined as

$$p_i = \frac{e^{o_i}}{\sum_{j=0}^{N} e^{o_j}}$$

where o_i is the ith ouput of the model.

The simplest strategy to use the Cross-Entropy loss in the context of partial labels is to mask out the pixels for which the candidate set contains more than one label. One of the pioneering work [10] proposed to use the Cross-Entropy Loss and assume a uniform probability distribution among the targets belonging to the set of candidates S. Throughout this paper, we will refer to it as the Uniform Cross-Entropy loss (UCE).

$$\mathcal{L}_{UCE} = -\frac{1}{|S|}\sum_{i=0}^{N} \log(p_i)\mathbb{1}_{y_i \in S}$$

Based on this work, [17] reused the same loss in the context of deep Neural Networks. Even if this seems a natural approach from a probabilistic viewpoint, assuming a uniform distribution of the unknown variable can lead to major drawbacks. Starting from a perfect classifier, further training the model using the UCE loss on partial labels would affect the performance of the classifier since it would increase the entropy of the predictions inside the candidate sets.

4.2. Binary Cross Entropy

The Binary Cross Entropy (BCE) loss is widely used in the multilabel classification literature [6].

$$\mathcal{L}_{BCE} = -\sum_{i=0}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

It can also be used for partial labels. One advantage over the Cross Entropy loss is that Binary Cross Entropy can be used to explicitly lower the probabilities of incompatible targets.

4.3. Minimum commitment

We propose to use the Minimum Commitment (MC) loss defined as

$$\mathcal{L}_{MC} = -\log(\sum_{i=0}^{N} p_i \mathbb{1}_{y_i \in S})$$

Contrary to the Uniform Cross-Entropy loss, the minimum commitment loss does not set any preferences among the compatible labels. It only considers the sum of the output probabilities belonging to the set of candidates. It can be viewed as a Cross-Entropy loss where the set of candidate labels is consider as a label itself.

5. Model and training procedure

We used a model based on the FCN [18] architecture. We used a ResNet-50 [8] as our backbone. We replaced the stride 2 of the 3x3 convolutions in the block 5 with a dilation 2. Our classifier is composed of 2 convolutional layers with kernel 3x3 and dilation 6 following the Field-of-View enlargement method used in DeepLab V2 [1]. Each of them is followed by a batch normalization layer and a ReLU activation function. Our classifier has a final convolutional layer with kernel 1 for classification.

We randomly resized the input images with a scaling factor between 0.5 and 2. We took random crops of 769×769, we randomly flipped the images horizontally and finally we normalized the images with the mean and the standard deviation of the Cityscapes dataset. When training on multiple datasets, we concatenated the datasets and randomly sampled training images from the result of the concatenation.

We trained the models with a batch size of 4, using Adam optimizer with a learning rate of 0.0001 and a weight decay of 0.0001.

6. Experiments

We performed two types of experiments. We first created two toy experiments to validate the proposed loss against other well-known losses. We then conducted a larger scale experiment to compare our method to other methods for partial labels.

6.1. Toy experiments

The two toy experiments are conducted on two different sets of datasets created from Cityscapes. The goal of conducting two experiments is to see the performance of the loss function in two different scenarios. In the first scenario, one label is always part of an ambiguous candidate set, i.e. all the candidate sets where this label appears have a cardinality of at least 2. In the second scenario, all the labels appears in a non-ambiguous candidate set in at least one dataset.

For the first toy experiment, we created 2 datasets containing 50 images each. Three labels "Construction", "Nature" and "Flat" were created from merging original Cityscapes labels. In the first dataset, the classes "Construction" and "Nature" were combined into a single label. In the second dataset, the classes "Flat" and "Construction" were combined to a single label. Therefore the class "Construction" is always part of an ambiguous set of labels. The validation set is composed of 100 images from Cityscapes with the labels "Construction", "Nature" and "Flat". The labels present in each dataset are summarized in figure 3.

- ❑ Training set 1: Flat, {Construction, Nature}
- ❑ Training set 2: {Flat, Construction}, Nature
- ❑ Validation set: Flat, Construction, Nature

Figure 3: Training and validation dataset and classes for toy experiment 1

For the second toy experiment, we created 2 new datasets containing 50 images each. Four labels "Construction", "Nature", "Flat" and "objects" were created from merging original Cityscapes labels. In the first dataset, the classes "Nature" and "Construction" are combined into a single label. In the second dataset, the classes "Flat" and "Object" are combined into a single label. The validation set is composed of 100 images from Cityscapes with the labels "Construction", "Nature", "Flat" and "Objects". The labels present in each dataset are summarized in figure 4.

- ❑ Training set 1: Flat, Object, {Nature, Construction}
- ❑ Training set 2: {Flat, Object}, Nature, Construction
- ❑ Validation set: Flat, Object, Nature, Construction

Figure 4: Training and validation dataset and classes for toy experiment 2

6.2. Larger scale experiment

For the larger scale experiment we use the BDD100K and Cityscapes datasets. Both datasets originally share the same classes. To introduce ambiguity, we merged the 19 cityscapes labels into 7 categories as show in Figure 2. We train the model using 300 images of BDD dataset with 19 classes and 1000 images of Cityscapes with 7 categories. Training a model this way would account for the ambiguity that may arise from training a single model on multiple heterogeneously labeled datasets. The validation set is composed of 500 images Cityscapes. The model is evaluated on the original 19 Cityscapes' labels.

We compare our model with a multitask model, a flat classifier and the CE baseline where ambiguities are ignored. For the multitask model, we used the same model described in section 5. We ran experiments while including 1, 2 and 3 layers in the classification head. We kept the model that gave the best performance. We created two classification heads, one for each dataset. For the flat classifier, we concatenated the labels and the categories. By doing so, we obtained a classification problem with 26 classes. While

(a) GT (b) CE (c) BCE (d) UCE (e) MC

Figure 5: Example of segmentation masks produced by the different losses for the toy experiment 1

(a) GT (b) CE (c) BCE (d) UCE (e) MC

Figure 6: Example of segmentation masks produced by the different losses for the toy experiment 2

running the validaiton, we ignored the categories classes and perform the argmax over the labels only.

7. Results

7.1. Toy experiments

Figure 8 shows the results for the first toy experiment. The proposed Minimum Commitment loss performs significantly better than the three other losses. It achieves a validation mean Intersection over Union (mIoU) 0.17 greater than the Uniform Cross Entropy loss and 0.29 greater than the CE and BCE baselines. The IoU and accuracy are reported for every individual classes in Table 1 and Table 2. The minimum commitment loss achieves better mIoU and better mAcc than the other losses. We also notice that the Uniform Cross Entropy loss leads to a mIoU curve with great volatility. This is caused by the incentive for the network to have uniform output inside the sets of candidates. The Cross-Entropy and Binary Cross-Entropy losses achieve low mIoU and mAcc because of the *Nature* class that always appears in sets of candidates that contain at least 2 classes. Therefore, CE and BCE are never trained on the *Nature* class. An example of the segmentation masks produce by each loss can be seen in Figure 5

Figure 9 shows the results of the second toy experiments. The proposed Minimum Commitment loss performs slightly better than the three other lossses. The IoU and accuracy are reported for every individual classes in Table 3 and Table 4. Since all the labels appear at least once in a non-ambiguous set, the Cross-Entropy and Binary Cross-Entropy losses perform significantly better than in the first toy experiment. Moreover, both datasets are sample from Cityscapes, i.e. the same distribution, which explains why there is less benefit in taking advantage of ambiguities. An example of the segmentation masks produce by each loss can be seen in Figure 6

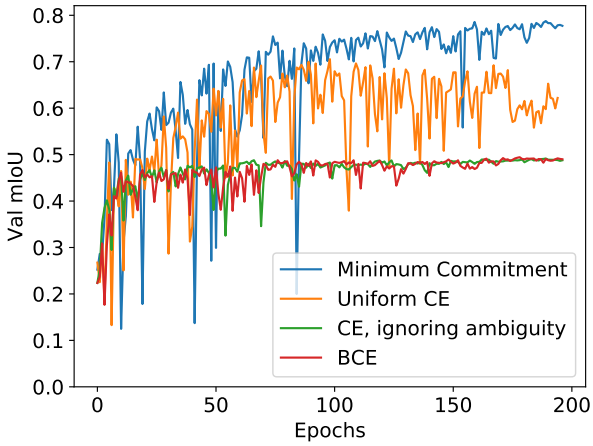

Figure 8: Evolution of the val mIoUs during the training for the toy experiment 1

Loss	Flat	Construction	Nature	mIoU
MC (ours)	**0.9149**	**0.7307**	**0.6880**	**0.7779**
UCE	0.7369	0.6280	0.4496	0.6048
CE	0.8866	0.5668	0.0000	0.4845
BCE	0.9014	0.5726	0.0000	0.4913

Table 1: Validation IoUs for the toy experiment 1

(a) GT (b) CE (c) Flat classifier (d) Multi-task (e) MC (ours)

Figure 7: Example of segmentation masks produced by the different methods for the larger scale experiment

Loss	Flat	Construction	Nature	mAcc
MC (ours)	**0.9613**	0.8322	**0.8244**	**0.8726**
UCE	0.8106	0.7245	0.7299	0.7550
CE	0.9264	**0.9693**	0.0000	0.6319
BCE	0.9560	0.9560	0.0000	0.6373

Table 2: Validation accuracy for the toy experiment 1

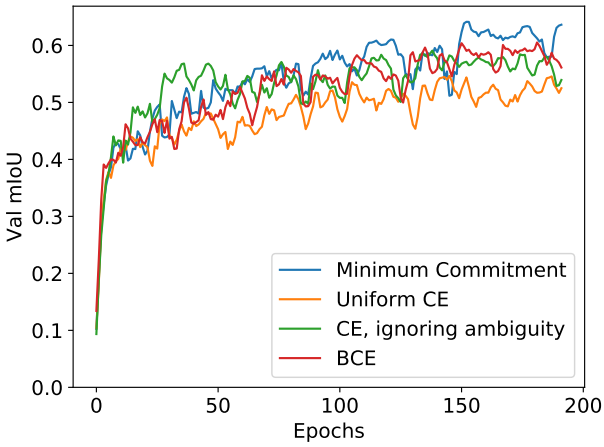

Figure 9: Evolution of the val mIoUs during the training for the toy experiment 2

Loss	Flat	Constr.	Nature	Object	mIoU
MC (ours)	0.745	**0.603**	**0.724**	0.4048	**0.619**
UCE	0.653	0.516	0.500	0.331	0.500
CE	**0.883**	0.414	0.6150	0.329	0.560
BCE	0.693	0.527	0.682	**0.424**	0.582

Table 3: Validation IoUs for the toy experiment 2

Loss	Flat	Constr.	Nature	Object	mAcc
MC (ours)	0.932	**0.831**	**0.847**	0.627	**0.809**
UCE	0.816	0.751	0.569	0.777	0.743
CE	**0.943**	0.462	0.692	**0.780**	0.719
BCE	0.942	0.737	0.801	0.753	0.808

Table 4: Validation accuracy for the toy experiment 2

7.2. Larger scale experiment

The larger scale experiment results on fig 10 show that our method achieves better results than the multitask model, the flat classifier and the CE baseline. Our method converges to a val mIoU of 0.26 outperforming the multi-task setting by 0.06, the CE baseline by 0.10 and the flat classifier by 0.13. The flat classifier is under performing compared to the CE loss because of its incentive during training to discriminate similar classes across the datasets. It is also important to notice that our method requires less parameters than the multitask and the flat classifier settings. The mIoU being relatively low can be explained due to the fact that we only used a fraction from both datasets, and BDD100k and Cityscapes are quite different datasets; the transfer from one dataset to the other can be more tricky than what it seems. Firstly, images from BDD100k are twice smaller than images from Cityscapes and we did not apply any dataset specific pre-processing. Secondly, the distribution of the two datasets is different. The Cityscapes dataset only contain images from the cities and BDD contain a much greater variety of locations. An example of the segmentation masks produced by each methods can be seen in Figure 7

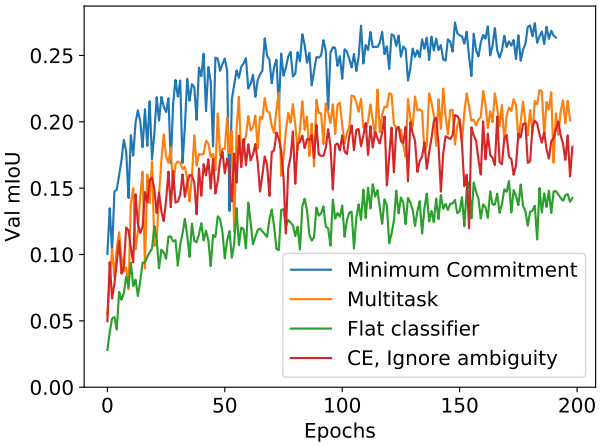

Figure 10: Evolution of the val mIoUs during the training for the larger scale experiment

8. Conclusion

In this paper, we considered the problem of combining multiple heterogeneously labeled datasets. We proposed to use the Minimum Commitment loss, and we showed that our loss performs significantly better than other losses or methods. In future work, one could study the performance of the Minimum Commitment Loss applied to tasks other than semantic segmentation.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:834–848, 2018. 4

[2] J. Cid-Sueiro. Proper losses for learning from partial labels. *Advances in Neural Information Processing Systems*, 2:1565–1573, 01 2012. 2

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1

[4] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 05 2011. 2

[5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1997. 2

[6] T. Durand, N. Mehrasa, and G. Mori. Learning a deep convnet for multi-label classification with partial labels. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 647–657, 2019. 2, 3

[7] Y. Grandvalet. Logistic regression for partial labels. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[9] E. Hullermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006. 2

[10] R. Jin and Z. Ghahramani. Learning with multiple labels. *Advances in Neural Information Processing Systems*, 07 2003. 1, 2, 3

[11] F. Kong, C. Chen, B. Huang, L. M. Collins, K. Bradbury, and J. M. Malof. Training a single multi-class convolutional segmentation network using multiple datasets with heterogeneous labels: preliminary results. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3903–3906, 2019. 2

[12] L. Liu and T. Dietterich. A conditional multinomial mixture model for superset label learning. *Advances in Neural Information Processing Systems*, pages 557–565, 2012. 2

[13] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. Progressive identification of true labels for partial-label learning. *ArXiv*, abs/2002.08053, 2020. 2

[14] P. Meletis and G. Dubbelman. Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1045–1050, 2018. 2

[15] N. Nguyen and R. Caruana. Classification with partial labels. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV*, pages 551–559, 2008. 2

[16] O. Petit, N. Thome, A. Charnoz, A. Hostettler, and L. Soler. Handling missing annotations for semantic segmentation with deep convnets. In D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 20–28, Cham, 2018. Springer International Publishing. 2

[17] J. Seo and J. S. Huh. On the power of deep but naive partial label learning. *ArXiv*, abs/2010.11600, 2020. 3

[18] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 2017. 4

[19] Y. Tang, Y. Huo, Y. Xiong, H. Moon, A. Assad, T. K. Moyo, M. R. Savona, R. Abramson, and B. A. Landman. Improving splenomegaly segmentation by learning from heterogeneous multi-source labels. *Proc SPIE Int Soc Opt Eng.*, 2019. 2

[20] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2020. 1

[21] B. Zhaoa, J. Feng, X. Wu, and S. Yan. A survey on deep learning based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017. 1

[22] X. Zhu and A. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 1–130, 2009. 2