Zero-Shot Task Transfer and Self-Supervised Learning

Jinde Sagar
IIT Hyderabad
ai19mtech11006@iith.ac.in

Vinay Prakash
IIT Hyderabad
ai19mtech11003@iith.ac.in

1. Introduction

Modern computer vision tasks and other deep neural network models require a lot of data. Efficacy of deep models is largely dependent upon the labeled data due to their large parameter spaces. However, availability of large-scale hand-annotated data-sets for every vision task is not practical. It is very expensive to collect data and requires a huge amount of time. Due to this shortcoming, many of the vision tasks are considered expensive. We try to address this problem in this work. We try to build an alternative approach that can obtain model parameters for tasks without any labeled data. We call this task *Zero-Shot Task Transfer*.

We present our meta-learning algorithm that computes Encoder-Decoder parameters for novel tasks for which no ground truth is available (called zero-shot tasks). In order to adapt to a zero-shot task, our meta-learner learns from the Encoder-Decoder parameters of known tasks (with ground truth) and their task correlation to the novel task. Formally, given the knowledge of m known tasks $\{ \tau^1, ..., \tau^m \}$, a meta-learner $\mathcal{F}(.)$ can be used to find out the parameters of $\tau^{(m+1)}$, a novel task. We will drop Encoder-Decoder subscript and will be using τ^i instead of τ^i_E and τ^i_D for the simplicity.

The "known" task here are self-supervised task. Tasks whose model parameters can be learnt using self-supervised learning are called self-supervised task. Self-supervised learning is a form of unsupervised learning where the data itself provides the supervision. Self-supervised learning is still supervised learning, but the datasets do not need to be manually labelled by a human. Hence, in this paper, we are computing Encoder-Decoder parameters for a novel task without any labeled data at all.

We need to know the relationship between tasks, otherwise it may not be plausible to learn a meta-learner, as its output could map to any point on the meta-manifold. Therefore, we consider the task correlation between known tasks and a novel task as an additional input to our framework.

Idea of task correlation can be found similar to the recently proposed idea of Taskonomy but our method and objectives are different in many ways. However, main difference is that taskonomy used task correlation for transfer of one task model to another, while our model tries to learn the parameter for an novel task for which no labeled data is available.

2. Related Work

Transfer Learning: In transfer learning, we reuse the components of a trained model and fine-tune it for a new similar task. From the early experimentation of CNN features [41], it was clear that initial layers of deep networks learn similar kind of filters. Methods is [3], [23] augment generation of samples by transferring knowledge from one category to another. Zamir *et al*. [42] used this idea for Taskonomy. However, unlike our work, it cannot be generalized to a novel task without accessing the labeled data.

Domain Adaptation: The main focus of domain adaptation is to transfer domain knowledge from a data-rich domain to a domain with limited data [27] [9]. Learning domain-invariant features requires domain alignment. Such matching is done either by mid-level features of a CNN [13], using an autoencoder [13], by clustering [36], or more recently, by using generative adversarial networks [24]. In some recent efforts [35] [6], source and target domain discrepancy is learned in an unsupervised manner. However, a considerable amount of labeled data from both domains is still unavoidable. In our methodology, we propose a generalizable framework that can learn models for a novel task from the knowledge of available tasks and their correlation with novel tasks.

Meta-Learning: Earlier efforts on meta-learning (with other objectives) assume that task parameters lie on a low-dimensional subspace [2], share a common probabilistic prior [22], etc. Unfortunately, these efforts are targeted only to achieve knowledge transfer among known tasks and tasks with limited data. Recent meta-learning approaches consider all task parameters as input signals to learn a meta manifold that helps few-shot learning [28], [37], transfer learning [33] and domain adaptation [13]. A recent approach introduces learning a meta model in a modelagnostic manner [12][17] such that it can be applied to a variety of learning problems. Unfortunately, all these methods depend on the availability of a certain amount of labeled data

in target domain to learn the transfer function, and cannot be scaled to novel tasks with no labeled data. Besides, the meta manifold learned by these methods are not explicit enough to extrapolate parameters of zero-shot tasks. Our method relaxes the need for ground truth for zero-shot tasks, by leveraging task correlation among known tasks and novel zero-shot tasks. To the best of our knowledge, this is the first such work that involves regressing model parameters of novel tasks without using any ground truth information for the task.

Learning with Weak Supervision: Task correlation is used as a form of weak supervision in our methodology. Recent methods such as [32][38] proposed generative models that use a fixed number of user-defined weak supervision to pragmatically generate synthetic labels for data in near-constant time. Alfonseca et al. [1] use heuristics for weak supervision to accomplish hierarchical topic modeling. Broadly, such weak supervision is harvested from knowledge bases, domain heuristics, anthologies, rules-of- thumb, decisions of weak classifiers or obtained using crowd-sourcing. Structure learning [4] also exploits the use of distant supervision signals for generating labels. Such methods use factor graph to learn a high fidelity aggregation of crowd votes. Similar to this, [30] uses weak supervision signals inside the framework of a generative adversarial network. However, none of them operate in a zero-shot setting. We also found related work zero-shot task generalization in the context of reinforcement learning (RL) [29], or in lifelong learning [16]. An agent is validated based on its performance on unseen instructions or a longer instructions. We find that the interpretation of task, and primary objectives, are very different from our present study.

3. Motivation

The major driving force behind modern computer vision, machine learning, and deep neural network models is the availability of large amounts of curated labeled data. Deep models have shown state-of-the-art performances on different vision tasks. Effective models that work in practice entail a requirement of very large labeled data due to their large parameter spaces. Some tasks require extensive domain expertise, long hours of human labor, expensive data collection sensors - which collectively make the overall process very expensive. Even when data annotation is carried out using crowd-sourcing (e.g. Amazon Mechanical Turk), additional effort is required to measure the correctness (or goodness) of the obtained labels. We seek to address this problem in this work, viz., to build an alternative approach that can obtain model parameters for a novel tasks without any labeled data.

4. Methodology

The primary objective of our methodology is to learn a meta-learning algorithm that regresses nearly optimum parameters of a novel task for which no ground truth (data or labels) is available. To this end, our meta-learner seeks to learn from the model parameters of known self-supervised tasks (with ground truth in the input itself) to adapt to a novel zero-shot task. Consider there are m known self-supervised tasks.

Lets denote it by $\tau = \{ \tau_1, .., \tau_m \}$ and we know their corresponding model parameters $\{ \theta_{\tau_i} : i = 1,..,m \}$.

Complementarily, we have no knowledge of the ground truth for the zero-shot tasks $\{ \tau_{(m+1)}, .., \tau_K \}$.

Our aim is to build a meta-learning function $F(.)$ that can regress the unknown zero-shot model parameters $\{ \theta_{\tau_i} : i = (m+1),..,K \}$ from the knowledge of known model parameters. i.e.,

$$F(\theta_{\tau_1}, ..., \theta_{\tau_m}) = \theta_{\tau_j}, \qquad j = m+1, .., K$$

However, with no knowledge of relationships between the tasks, it may not be plausible to learn $\mathcal{F}(.)$. We hence introduce a task correlation matrix, Γ where each entry $\gamma_{i,j} \in \Gamma$ captures the task correlation between two tasks τ_i, $\tau_j \in \Gamma$. We obtain the task correlation matrix, Γ using crowdsourcing.

$$F(\theta_{\tau_1}, ..., \theta_{\tau_m}, \Gamma) = \theta_{\tau_j}, \qquad j = m+1, .., K$$

The function $\mathcal{F}(.)$ is itself parameterized by W. We design our objective function to compute an optimum value for W as follows:

$$\min_W \sum_{i=1}^{m} ||F((\theta_{\tau_1}, \gamma_{1,i}), .., (\theta_{\tau_m}, \gamma_{m,i}); W) - \theta^*_{\tau_i}||^2$$

According to the observation, only encoder parameters $\theta_{E\tau_i}$ is sufficient to regress zero-shot encoders and decoders for tasks $\{ \tau_{(m+1)}, .., \tau_K \}$

$$\min_W \sum_{i=1}^{m} ||F((\theta_{E\tau_1}, \gamma_{1,i}), .., (\theta_{E\tau_m}, \gamma_{m,i}); W) - (\theta^*_{E_{\tau_i}}, \theta^*_{D_{\tau_i}})||^2$$

The model parameters thus obtained not only should minimize the above loss function on the meta-manifold but should also have low loss on the original data manifold (ground truth of known tasks). Adding data model consistency loss, we get

$$\min_W \sum_{i=1}^{m} ||F((\theta_{E\tau_1}, \gamma_{1,i}), .., (\theta_{E\tau_m}, \gamma_{m,i}); W) - (\theta^*_{E_{\tau_i}}, \theta^*_{D_{\tau_i}})||^2$$

$$+\lambda \sum_{x,y} L(D_{\theta_{D\tau_i}}(E_{\theta_{E\tau_i}}(x)), y)$$

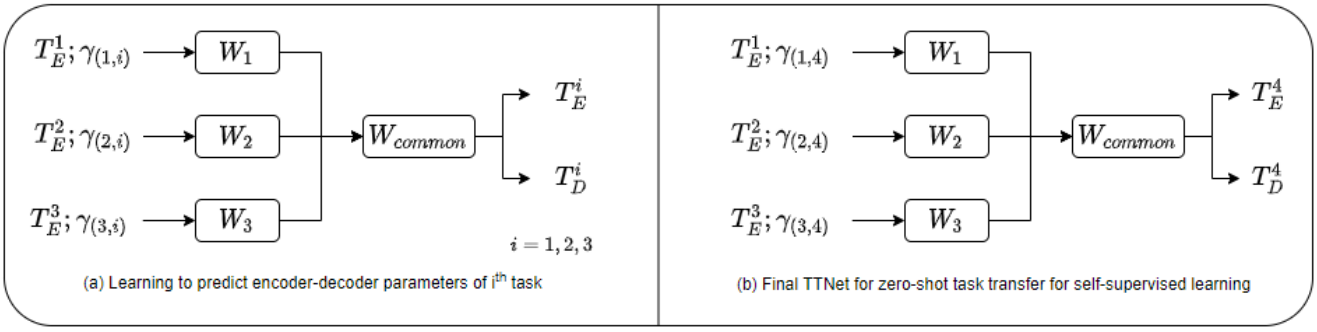

Figure 1. (a) represents the training model. Once TTNet is trained using (a), we use (b) to regress the parameters for zero-shot self-supervised learning task.

Network: To accomplish the aforementioned objective in above equation, we design $\mathcal{F}(.)$ as a network of m branches, each with parameters $\mathcal{W}_1, ..., \mathcal{W}_m$ respectively. These are not coupled in the initial layers but are later combined in a $\mathcal{W}_{common}$ block that regresses encoder and decoder parameters. For simplicity, we refer W to mean $\mathcal{W}_1$, ... ,$\mathcal{W}_m$ and $\mathcal{W}_{common}$.

Input: To train our meta network $\mathcal{F}(.)$, we need a batch of model parameters for each known task $\tau_1, ..., \tau_m$. This process is similar to the way a batch of data samples are used to train a standard data network. To obtain a batch of p model parameters for each task, we closely follow the procedure described in [40]. This process is as follows. In order to obtain one model parameter set $\Theta^*_{\tau_i}$, for a known self-supervised task τ_i, we train a base learner (autoencoder), defined by $\mathcal{D}(\mathcal{E}(x;\theta_{E_{\tau_i}});\theta_{D_{\tau_i}})$. Hence, we learn one $\Theta^1_{\tau_i} = \{\theta^{*1}_{E_{\tau_i}}, \theta^{*1}_{D_{\tau_i}}\}$. Similarly, p subsets of labeled data are obtained, i.e, $\Theta^*_{\tau_j} = \Theta^{*1}_{\tau_j}, ..., \Theta^{*p}_{\tau_j}$ for task τ_j. A similar process is followed to obtain p "optimal" model parameters for each known task $\{\Theta^*_{\tau_1}, ..., \Theta^*_{\tau_m}\}$

We pass model parameter of each known self-supervised task $\{\{\Theta^*_{\tau_1}, ..., \Theta^*_{\tau_m}\}$ which server as an input to our meta network.

Training: The meta network $\mathcal{F}(.)$ is trained on the objective function in equation mentioned above in two modes: a self mode and a transfer mode for each task.

Self mode is similar to training a standard autoencoder, where $\mathcal{F}(.)$ learns to project the model parameters θ_{τ_j} near the given model parameter (learned from ground truth) $\theta^*_{\tau_j}$. In transfer mode, a set of model parameters of tasks (other than j) attempt to map the position of learned θ_{τ_j} , near the given model parameter $\theta^*_{\tau_j}$ on the meta manifold.

Regressing Zero-Shot Task Parameters: Once we learn the optimal parameters W^* for $\mathcal{F}(.)$ using Algorithm 1, we use this to regress zero-shot task parameters, i.e.

$F_{W^*}((\theta_{E\tau_1}, \gamma_{1,j}), .., (\theta_{E\tau_m}, \gamma_{m,j}))$, j = (m+1,...,K)

Algorithm 1: Training our meta network, TTNet

Result: Trained TTNet model, $\mathcal{F}(.)$
for *Num_Epoch* **do**
  for i=1,...,m **do**
    for p *steps* **do**
      /* self mode */
      update weights $\mathcal{W}_i$, $\mathcal{W}_{common}$ of $\mathcal{F}(.)$

$$||\mathcal{F}((\theta_{E\tau_1}, \gamma_{1,i}), .., (\theta_{E\tau_m}, \gamma_{m,i}); W) - (\theta^*_{E_{\tau_i}}, \theta^*_{D_{\tau_i}})||^2 + \lambda \sum_{x,y} L(D_{\theta_{D\tau_i}}(E_{\theta_{E\tau_i}}(x)), y)$$

    end
    for p *steps* **do**
      /* transfer mode */
      update weights $\mathcal{W}_{\neg i}$, $\mathcal{W}_{common}$ of $\mathcal{F}(.)$

$$||\mathcal{F}((\theta_{E\tau_1}, \gamma_{1,i}), .., (\theta_{E\tau_m}, \gamma_{m,i}); W) - (\theta^*_{E_{\tau_i}}, \theta^*_{D_{\tau_i}})||^2 + \lambda \sum_{x,y} L(D_{\theta_{D\tau_i}}(E_{\theta_{E\tau_i}}(x)), y)$$

    end
  end
end

5. Results

To evaluate our proposed framework, we consider the self-supervised learning vision tasks defined in [42]. We consider autoencoding, jigsaw and denoising as known tasks and surface-normal as unknown or zero-shot task.

5.1. Dataset

We evaluated TTNet on the Taskonomy dataset [42]. We have total of 9,464 images, out of which we used 1000 ran-

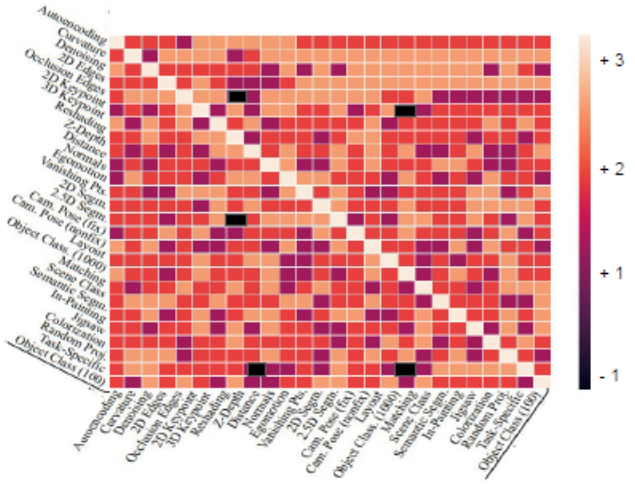

Figure 2. **Task correlation matrix.** We get the task correlation matrix Γ after receiving votes from 30 annotators. We use this Γ to build our meta-learner TTNet.

Component	Layer	Channel(i/p, o/p)
Encoder	Conv	(3,5)
	ReLU	-
	Conv	(5,8)
	ReLU	-
Decoder	Transpose-conv	(8,5)
	ReLU	-
	Transpose-conv	(5,3)
	ReLU	-

Table 1. Source network configuration. Kernel size is 3 for all the convolution layers.

domly picked images (with replacement) for training and 189 images for validation.

5.2. Implementation Details

Network Architecture: Each data network is considered an Encoder-Decoder network. We have used 2 convolution layers with kernel size of 3 followed by ReLU activation in the encoder and 2 transpose-convolution layers with kernel size of 3 followed by ReLU activation in the decoder. See Table 1 for detailed information. To make input samples for TTNet, we created 48 samples of the model parameters for each task, each of which is obtained by training the model for 100 iterations on 1000 data points sampled (with replacement) from the taskonomy dataset and the ground truth weights were obtained by training the model for 2700 iterations. These data networks were trained with mini-batch SGD optimizer using a batch size of 32 and learning rate of 0.001.

TTNet: TTNet is shown in Figure 1. The TTNet initially has 3 branches; one branch corresponding to each source task. Each of these branches is comprised of two linear layers; each layer is followed by a dropout layer and a ReLU activation function. See Table 2 for more details. These 3 branches are then concatenated to form a common layer which is then followed by 3 linear layers, out of which, each of the first two linear layers is followed by a dropout layer and a ReLU activation function. We used Adam optimizer for training our model with batch size of 16, learning rate = 0.001 and momentum of 0.9.

Component	Layer	O/p Dimension
W_i	Linear	768
	Dropout	p=0.2
	ReLU	-
	Linear	256
	Dropout	p=0.2
	ReLU	-
W_{common}	Linear	1024
	Dropout	p=0.2
	ReLU	-
	Linear	512
	Dropout	p=0.2
	ReLU	-
Encoder	Linear	508
Decoder	Linear	503

Table 2. TTNet Configuration. Input dimension of the first linear layer is 508+1 (encoder dimension+correlation dimension). First two linear layers form $W_i's$ and are computed for all 3 tasks and then concatenated. It is followed by linear layers of W_{common} and finally fed into the encoder and decoder layers (in parallel).

Loss using TTNet	Loss using Supervised Model
0.125560	0.180921
0.123371	0.180921
0.122083	0.180921

Table 3. Comparing the losses of surface-normal task using parameters generated by TTNet and using parameters generated by supervised learnt model.

5.3. Quantitative Result

We can see the results in Table 3 in terms of mean-squared losses. As we can see, losses obtained using parameters from TTNet model is less compared to losses obtained by using parameters generated by supervised learnt model.

6. Conclusion

We obtained model parameters for a task that requires labeled data without using any explicit labeled data at all.

References

[1] E. Alfonseca, K. Filippova, J.-Y. Delort, and G. Garrido. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 54–59. Association for Computational Linguistics, 2012.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008

[3] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2252–2259. IEEE, 2011.

[4] S. H. Bach, B. He, A. Ratner, and C. Re. Learning the structure of generative models without labeled data. *arXiv preprint arXiv:1703.00854,* 2017.

[5] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5965–5974*, 2016.

[6] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty. Reweighted adversarial adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7976– 7985, 2018.

[7] H. Cohen and K. Crammer. Learning multiple tasks in parallel with a shared annotator. In *Advances in Neural Information Processing Systems*, pages 1170–1178, 2014.

[8] 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. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[9] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[10] K. G. Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3, 2010.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *In Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic metalearning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[13] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *xEuropean Conference on Computer Vision*, pages 597–613. Springer, 2016.

[14] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in neural information processing systems*, pages 1288–1296, 2010. 6

[15] R. Held and A. Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology,* 56(5):872, 1963.

[16] D. Isele, M. Rostami, and E. Eaton. Using task features for zero-shot knowledge transfer in lifelong learning. In *IJCAI*, pages 1620–1626, 2016.

[17] T. Kim, J. Yoon, O. Dia, S. Kim, Y. Bengio, and S. Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836,* 2018.

[18] S. Korman and R. Litman. Latent ransac. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6693–6702, 2018.

[19] A. Kumar and H. Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012.

[20] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. Roomnet: End-to-end room layout estimation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4875–4884. IEEE, 2017.

[21] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim. Single-image depth estimation based on fourier domain analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 330–339, 2018.

[22] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th international conference on Machine learning*, pages 489–496. ACM, 2007.

[23] J. J. Lim, R. R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Advances in neural information processing systems*, pages 118–126, 2011.

[24] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[25] S. Liu, S. J. Pan, and Q. Ho. Distributed multi-task relationship learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946. ACM, 2017.

[26] M. Long, Z. Cao, J. Wang, and S. Y. Philip. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems*, pages 1594–1603, 2017.

[27] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In Advances in Neural Information Processing Systems, pages 165–177, 2017.

[28] D. K. Naik and R. Mammone. Meta-neural networks that learn by learning. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pages 437–442. IEEE, 1992.

[29] J. Oh, S. Singh, H. Lee, and P. Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. *arXiv preprint arXiv:1706.05064,* 2017.

[30] A. Pal and V. N. Balasubramanian. Adversarial data programming: Using gans to relax the bottleneck of curated labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 1556–1565, 2018.

[31] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 283–291, 2018.

[32] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Re. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575, 2016.

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[35] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *arXiv preprint arXiv:1712.02560*, 3, 2017.

[36] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

[37] S. Thrun and L. Pratt. *Learning to learn*. Springer Science Business Media, 2012.

[38] P. Varma, B. He, D. Iter, P. Xu, R. Yu, C. De Sa, and C. Re. Socratic learning: Augmenting generative models to incorporate latent subsets in training data. *arXiv preprint arXiv:1610.08123,* 2016.

[39] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[40] [40] Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. In *NIPS*, 2017.

[41] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328, 2014.

[42] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 3712–3722, 2018.

[43] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. Generic 3d representation via pose estimation and matching. In *European Conference on Computer Vision*, pages 535–553. Springer, 2016.

[44] W. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *IEEE Transactions on Multimedia*, 19(5):935–943, 2017.

[45] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 2051–2059, 2018.