Learn transferable features for semantic image segmentation in the presence of label noise

First Author
Institution1
Institution1 address
firstauthor@i1.org

Second Author
Institution2
First line of institution2 address
secondauthor@i2.org

July 26, 2017

Abstract

The ABSTRACT HERE

1 Introduction

Why noisy labels?:This paragraph should discuss the difficulties for collecting perfect segmentation annotations on a large scale

The recent success of deep neural networks benefits from the availability of large-scale supervised datasets such as ILSVRC[25]. However, collecting a dataset for semantic image segmentation on a large scale can be expensive and time-consuming, especially when error-free labels are required. Enormous efforts have been made to create "gold standard" annotations for the current benchmark segmentation datasets[6, 20, 16]. [1] These datasets allow deep neural network models for semantic segmentation [17, 32] to develop assuming the perfect segmentation ground truth exists, where perfect means that all the instances were annotated, no non-target object was misannotated, and no instance was misclassified. However, it is natural for human beings to make mistakes due to the lack of expertise, the inherent ambiguity of tasks or unconscious bias. Strong supervision is required to correct the mistakes, including double-checking the annotations over and over again and ensembling opinions from multiple annotators. In some cases, for example in the medical imaging, the "gold standard" itself can be ambiguous and cause disagreement among different annotators. [2] Also, there are some freely available labels, which may or may not be accurate, for specific problems. For example, one can use digital maps, like OpenStreetMap, to annotate aerial images, but such segmentation annotations constructed from maps suffer from the incomplete annotation as well as registration problems.[19] This motivates us to explore how to learn from these noisy annotations.

Why representation?: This paragraph should discuss the idea of noise-robust representation

Deep convolutional neural networks (DCNN) based models often contain two principal components: a stack of convolutional layers to extract hierarchical features and a few task-specific layers to fit the training objectives. Both parts are trained jointly in an end-to-end manner via back propagation and are both important to achieve good performance for a particular task. [3] The convolutional features were proved "transferable" to a new dataset[30] or even to a different task[8]. The state-of-art DCNN based semantic image segmentation models relies on transferring the pre-trained convolutional filters as well.[17] A typical method to pre-train the convolutional filters is to

[1] M: Do we actually have a reference for that? Or other proof?
J: I think the preparation methods described in these papers can be supportive. Microsoft COCO even mentioned the total working hours in the paper.

[2] M: But that's OK or not? This is what probabilities solve...

[3] M: As a non-expert it is rather unclear to me how to identify or design generic layers and task-specific ones, or what it actually means or why it is important to make this distinction. In addition, I wonder to what extent end-to-end training actually turns generic layers into task-specific ones.

train a classification model with the large-scale ILSVRC dataset [25] However, this method constrains the semantic image segmentation models to have the same CNN architecture as the image classification models. The CNN design for semantic image segmentation does not necessarily follow the design of image classification architectures. The segmentation models need both global and local information to predict the category and give a fine segmentation, whereas the classification models care less about local information for object localization. For instance, the presence of the max-pooling layers enable the following convolutional filters to have larger receptive fields but, at the same time, reduce the resolution of the features. [4] Additional upsampling layers can recover the shape of the output segmentation but cannot fully recover the information dropped by subsampling. This first-pooling-and-then-upsampling pipeline can result in coarse segmentation output [2] with non-shape boundaries and blob-like shapes. [5] Alternatively, one can also extract convolutional features with semantic segmentation tasks. But it is more difficult to collect well-annotated dataset for semantic image segmentation than for object recognition. A large number of annotated images are of great value to train sufficiently generalized representation and avoid overfitting, given the "data-hungry" nature of DCNNs. [6] Allowing noisy annotations to exist could help significantly increase the number of annotated images and the number of training samples could compensate the impact of annotation errors. We will further discuss this in the related works.

This paragraph should summarize the main ideas.

The different levels of the hierarchical features in DCNNs are believed to play different roles in extracting information from the images. The low-level features process the local information within small neighborhood and the high-level features ensemble information from lower-level features to extract abstract information. The high-level features were found to significantly dependent on the exact categories compared to the low-level features which show extraordinary category independence.[30] Previous studies[27, 22] have shown that training with noisy labels can lead to significantly higher classification errors than training with clean labels if the total number of training samples are fixed. It is nevertheless unclear how the annotation errors would influence the learned multiple level features. We made a hypothesis that annotation errors do not necessarily lead to a "bad representation" because the "generality" of low-level features may contribute to a robustness to the errors when we transfer the learned features to a new dataset with new categories. [7]

Table of contents

In the next section, we summarized the related works for deep learning with noisy labels. In Section 3 we formulate the annotation errors into three categroies: misannotation, misclassification and incomplete annotation. We tested our hypothesis in Section 3.1, studying whether the misannotation and misclassification had an impact on learning "transferable" features. In Section 4 we connected training with inexaustive annotations to Positive and Unlabeled Learning.

2 Related work

Transfer Learning *transfer learning* We sometimes have a learning task in one domain of interest, but we only have sufficient training data in another domain of interest, where the two domain may or may not share the same feature space and have the same data distribution. Transfer learning arises in this scenario to transfer knowl-

[4] M: Is it the presence of max-pooling or the presence of subsampling that enables larger receptive fields? And what do you mean by resolution? Should low resolution always mean reduced / "thrown away" information? J: Max-pooling is a particular type of subsumpling. Actually both the conv layers and the pooling layers lead to gradually larger receptive field.

[5] M: Also for this claim, I think we need a reference or something like that. Or other proof indeed... In addition, I wonder whether we can explain why this may be the case? J: Yes, I can refer to a paragraph in the intro of CRFasRNN and enrich the discussion a bit.

[6] M: For the rest a bit vague... what do we really mean by suffer? Maybe this becomes clear later in the intro...? J: Refrased. M: This is still quite unclear to me. I would interpret "data-hungry" as an NN that overtrains even with large amount of data, but one would typically try to fix this by introducing some form of regularization or just using smaller networks. All in all [as a maybe-too-stubborn non-deep learner ;-)] I could still wonder what is really the problem...

[7] M: We hypothesize? For the rest I find the actual hypothesis pretty vague. Firstly, you use quotation marks quite often, which doesn't make "things" clearer. Either don't use that and make sure that the words you use are indeed the words you like to say or expand the part between ""s and use some more sentences to really explain what you have in mind. Secondly, for me the part "contribute to a robustness to the errors" needs further explaining. You might not have to get mathematically precise here, but I don't understand what you want to say here...

edge from one domain to anthoer and to improve the performance of learning by avoiding much expensive data-labeling efforts.[21] Recently, a form of knowledge that shows outstanding transferability is the weights of convolutional neural networks. For example, weights trained on ImageNet images to perform image classification were shown successfully transfered to new categories and new learning problems[8, 17, 26]. Convolutional neural networks on images are believed to extract hierarchical features, among which the low-level features look for specific patterns and the high-level features ensemble the information from low-level features. Low-level features were found more *general*, i.e., less dependent on a particular category, than the high-level features. By training a CNN on a random half of the ImageNet categories and transfer features, varying from the bottom layers to the top layers, to the other half, Yosinski et al. found transferability of features drop due to representation specificity increase as well as optimization difficulties due to co-adaption.[30] They also found that transferability of features decreases as the distance between the base task and target task increases, the first two layers have features still showing significant transferability to disimilar categories. Transferring features even from distant tasks can be better than using random features. In Section 3.1, we wanted to explore whether transferability of features is robust to annotation errors in the pre-training dataset.

Unsupervised pre-training Apart from supervised pre-training, one can also obtain pre-trained features in an unsupervised or a semi-supervised way. The most common method is to train a generative model with either *auto-encoder* variants or *deep beilief networks*. Vincent et al.[28] trained multiple levels of representation robust to the corrupted inputs with stacked denoising auto-encoders. Masci et al.[18] presented a stacked convolutional auto-encoder unsupervised pre-training for hierarchical feature extraction. Hinton et al.[11] proposed a greedy learning algorithm to train *deep belief nets* one layer at a time to train hierarchical features. Lee et al.[13] presented a *convolutional deep belief network*, to learn hierachical convolutional representations. A few studies[5, 4, 1] highlighted the advantage of unsupervised pre-training compared to the random initialization, connecting unsupervised pre-training to a norm of regularization and a method that help disentangle the sample variations. However, better random initialization strategies, for example, xavier initialization[9] and its variants, have shortened the gap between unsupervised pre-training and random initialization. Using unsupervised pre-training or not now becomes a tradeoff between the time and resources invested and the performance gain. Unsupervised deep representation learning is in general not comparable to supervised representation learning especially when large scale dataset is available. A proper method to learn features in the presence of label noise should at least outperform unsupervised pre-training because noisy information is still better than no information.

Deep Learning with Noisy Labels A few studies[27, 22] investigated the impact of label noise on classification performance with convolutional neural networks assuming the labels were randomly transited from one to another given the probabilities fall in a transition matrix. They found a significant decrease in classification performance along with the increase of false label proportion when the total number of examples is fixed. They then proposed methods to handle this label noise at random (NAR)[7] situation by either introducing a linear noise layer on top of the output layer[27] or correcting the loss functions with an estimation of the noise transition matrix[22]. Xiao et al.[29] integrated a probabilistic graphic model to an end-to-end deep learning system to train predicting class labels, either correct or wrong, as well as to correct the wrong labels. Reed & Lee[23] proposed an empirical way of taking into account the *perceptual consistency* for large-scale object recognition and detection when incomplete and noisy labels exist by introducing a bootstrapping modification to the negative log-likelihood, in either a “Hard” or a “soft” favor.

Noise robustness In contrast to the works above, Rolnick et al.[24] argued that deep neural networks can learn robustly from the noisy dataset as long as an appropriate hyper parameters choice was made. They studied instead of replacing the correct labels with noisy labels but diluting correct labels with noisy labels to support their argument. They then concluded sufficiently large training set is of more importance than lower the level of noise. This work is closely related to our work in Section 3.1, except that we focus on the label noise robustness regarding the feature transferability instead of the classification performance. Additionally, most of these studies focus on the classification problems, whereas our work inclined more to the semantic segmentation problem.

Positive and Unlabeled Learning If we consider the in-exhaustive annotation issue only, i.e., only a proportion of the target instances were annotated, the problem becomes similar to a so-called *positive and unlabelled learning* (PU learning) setup[15]. In the positive and unlabeled learning setup, the training dataset has two sets of examples: the *positive (P) set*, contained only positive examples, and the *unlabeled (U) set*, contained a mix of positive or negative examples. If we categorize the pixels into either *foreground pixels* or *background pixels*, the correctly annotated instances form the positive set, and the unannotated instances are mixed with the background pixels, forming an unlabeled set. The previous studies about PU learning mainly focus on the binary classification for linear-separable problems[3, 14], whereas we showed in Section 4 that it is possible to train deep neural networks for multiple classes with only "positive" and unlabeled examples.

3 Annotation noise-robustness of representations

From feature generality to feature robustness The first-layer features of convolutional neural networks for images are often observed converged to either Gabor filters or color blobs even training with different datasets and different objectives[31, 13, 12, 26]. Because these standard features on the first layer often occur independent of the exact cost function and natural image dataset, we call these first-layer features *general*. By contrast, the last-layer features must significantly depend on the given labels otherwise the training errors would be high which is against learning objective. These features are then denoted as *specific*. As we mentioned in Section 2, Yosinski et al. [30] studied the features in the intermediate layers and found the weights transferability decreases from the first layer to the last layer, alongside the specificity increases from the first layer to the last layer. Given the evidence that low-level features can be independent of a particular category, we wonder if the learned features are robustness to label noises regarding their transferability to a new task. For instance, if some dogs were incorrectly annotated as cats in the base dataset for pretraining, would these annotation noises influence transferability of the learned features to a new task recognize or detect sheep?

We experimented, in Section 5, how transferable the learned features are in the presence of three types of annotation noises: *misannotation*, *misclassification* and *inexaustive annotation*.

Transferability of features can be evaluated by how much they can improve the performance of training a new dataset compared to training with random weights initialization. [30]

Problem Formulation Semantic segmentation and other dense prediction tasks can be considered as pixel-wise classification problem. Given an image x, a segmenting model $f : R^{h \times w \times c} \rightarrow R^{h \times w}$ predicts a label for each pixel and output a label map y that has the same size as x. h, w are image height and weight respectively and c is the number of channels for images. Supposing there are K predefined categories, each pixel is assigned a label $y_{ij} = k$, where ij denotes a pixel from a full set of pixel in the image, P, and $i \in [1, h], j \in [1, w]$. The assigned label $k \in [1, K]$ if the pixel corresponds to an object from one of the K categories; $k = 0$ if the pixel is correspondent to the background. We can also say pixels with $k = 0$ are unannotated as they were not annotated to one of the predefined categories.. The probability of obsesrving a pixel label $\tilde{y_{ij}}$ depends on inputs x, the true label of that pixel y_{ij}, and labels of the rest pixels in that image:

$$p(\tilde{y_{ij}}|x, y_{ij}, \{y_{mn} : mn \in P \setminus \{ij\}\})$$

where P is the full set of pixels in one image and $i, m \in [1, h], j, n \in [1, w]$.

It is nevertheless difficult to model the probability conditioning on the distribution of x. [8]

The annotation errors considered in this work all occured to the whole object instead of to individual pixels. That means if one pixel for an object has a wrong label, the rest

[8] J: A bit more explaination needed.

pixels that belongs to this object will also have the same wrong label. By doing so, we exlude errors such as inprecise boundaries, oversegmenting or undersegmenting the objects from discussion. These types of errors are not the focus of our works and may lead to future studies.

Misannotation Misannotation denotes the errors wrongly segmenting a semantically meaningful but not from the predefined categories object as one of the pre-defined categories. These misannotated objects have pixel labels transited from 0 to k with probability $p(\tilde{y_{ij}} = k|x, y, y_{ij} = 0)$, where $\tilde{y_{ij}}$ means the observed label and y_{ij} is the true label. For this type of error we assume it depends on both inputs x and true labels y, meaning that the labels are *noisy not at random (NNAR)*[7]. That is natural because it is less likely misannotation would happens to partitions that have no semantically meaning or are not distinguishable against the background. The dependence of x results in the difficulty of modeling p by fitting to

Misclassification Misclassification means objects misclassified from one pre-defined category to another. Labels of the corresponding pixels flipped from k to l with probability $p(\tilde{y_{ij}} = l|y(y_{ij} = k))$, where $k, l \in [1, K]$.

We assume noise at random

Inexaustive annotation Inexaustive annotation denotes that there exists objects from the pre-defined categories unsegmentated. Pixels for the unannotated objects have labels flipped from k to 0. $p(\tilde{y_{ij}} = 0|y(y_{ij} = k))$, where $k \in [1, K]$

We assume labels are missing at random

The exhaustive annotations can introduce bias to both the decoding layer and the encoding layers because they negatively contribute to the activations in all the layers. [9] The inexhaustive annotations need to be properly handled given the prior knowledge modeling the missing pattern of the annotations. Given that we believe any annotated instance provide information, all the foreground pixels that correspond to the annotated instances become reliable and the background pixels may contain both the true background pixels and object pixels unannotated. That satisfies a Positive and Unlabeled learning setup where the training dataset contains only the positive examples and unlabeled examples that are the mixed of the positive samples and negative samples.

3.1 Pre-train features by learning “objectness”

4 Positive and Unlabeled Learning

One sentence summary of Positive and Unlabeled Learning

Formulation *This part should explain the Positive and Unlabeled Learning setup with mathematical representation when necessary.*

Weighted Logistic Regression *This part should discuss the linear model for observing positive conditioning on true positive and its relationship to changing the class weight.*

Exponential Loss for unlabeled examples *This part should explain why the exponential loss could perform better than the cross-entropy loss, potentially with a figure of 2D Gaussians.* The “soft” bootstrapping loss in [23] is actually equivalent to a softmax regression with *minimum entropy regularization*[10] which was originally proposed for semi-supervised learning. Minimum entropy regularization encourages the model to have a high confidence in predicting labels.

This paragraph should explain why fade-in was introduced to avoid all-positive inital prediction

This part should explain the influence of the imbalanced problem and how to overcome.

5 Results

What Table 3 tell us.

[9] J: This argument need evidence too, or experiment/discussions in details in Section 4

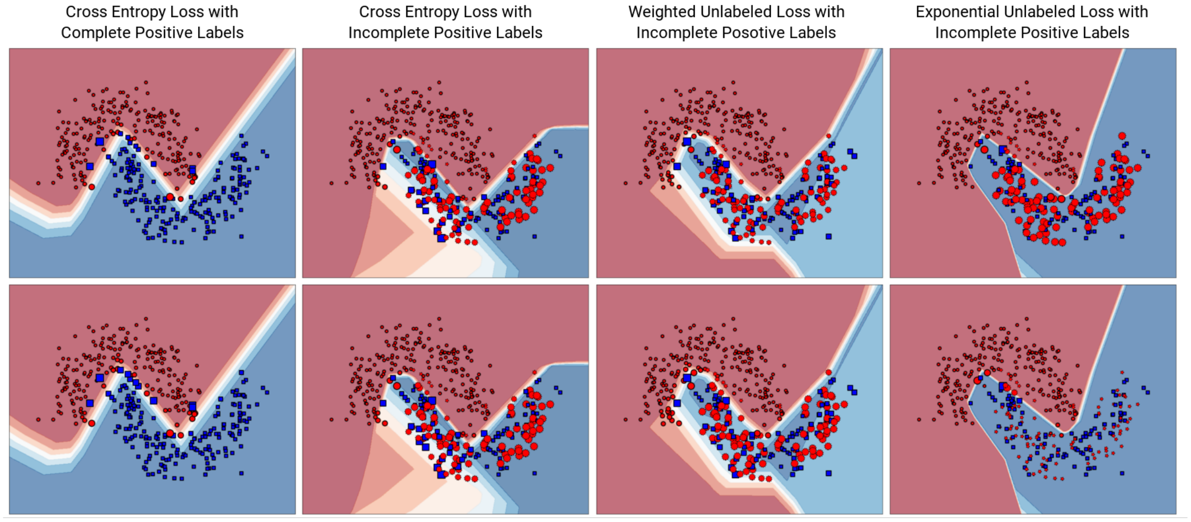

Figure 1: 2D moons dataset with non-linear separable decision boudary. Four hundreds samples per class were drawn randomly from two interleaving half circles with noises added with a minor standard deviation. A **red circle** indicates an example labelled as positive whilst a **blue square** indicates the example has a negative label. The **leftmost** figures have complete positive labels, meaning the positive and negative labels are all correct, whereas, in **the other figures** only half of the positives were correctly labelled and the rest were mixed with the negative samples. The **background colors** represent the probability for the area to be positive given by the classifier trained with the given samples and labels: **red** for high probability areas, **blue** for low probability areas and **white** for the class transition areas, i.e.decision boundaries. The **size of the markers** in the top row denotes the per-class normalized training losses and the **size of the markers** in the bottom row the per-class normalized derivatives w.r.t the output of the last layer for the trained Multilayer Perceptron (MLP) with the different losses.

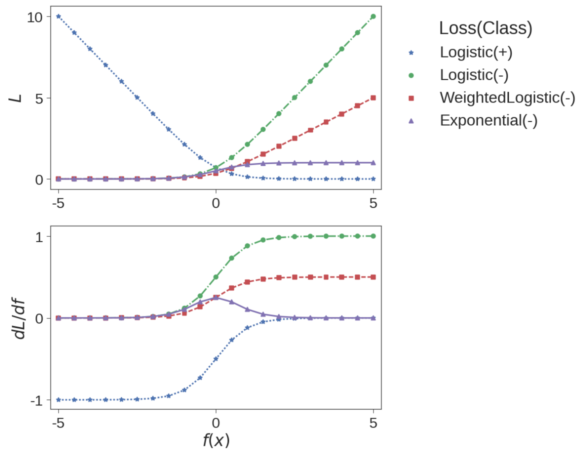

Figure 2: The Logistic Loss, Weighted Logistic Loss, Exponential Loss and their dirivatives with respect to the model output.

Annotation	Loss	acc.	prec.	rec.	F_1
Complete	CrossEntropyU.	0.87 ± 0.01	0.88 ± 0.01	0.82 ± 0.01	0.85 ± 0.01
50%(P+N)	CrossEntropyU.	0.83 ± 0.01	0.84 ± 0.01	0.78 ± 0.01	0.80 ± 0.01
50%P+U	CrossEntropyU.	0.64 ± 0.04	0.93 ± 0.08	0.34 ± 0.02	0.44 ± 0.06
50%P+U	WeightedU.	0.78 ± 0.01	0.75 ± 0.01	0.75 ± 0.01	0.76 ± 0.01
50%P+U	ExponentialU.	0.82 ± 0.01	0.86 ± 0.01	0.73 ± 0.01	0.78 ± 0.01
50%P+U	BootstrapHard	0.74	0.81	0.60	0.67
50%P+U	DropoutReg.				

Table 1: Image classification with positive examples partially annotated. The complete dataset contains images from CIFAR10 as the **positive** (P) set and images from CIFAR110 as the **negative** (N) set. The unannotated positive examples from P set construct the **unlabeled** (U) set together with the N set.

6 Conclusion

References

[1] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

Figure 3: Varying percentage of annotated positives 10%, 20%, 50%, 80% and 100% with images from CIFAR10 as the positives and images from CIFAR110 as the negatives.

Annotation	Loss	pixel acc.	mean acc.	mean IU	f.w. IU
Complete	CrossEnt.U				
50%(P+N)	CrossEnt.U				
50%P+U	CrossEnt.U				
50%P+U	WeightedU				
50%P+U	ExponentialU				
50%P+U	BootstrapHard				
50%P+U	DropoutReg.				

Table 2: Image semantic segmentation with images contain single instance only from the PASCAL VOC2011 segmentation dataset. The complete **positive** (P) set denotes the foreground instances and the **negative** (N) set consists of the background. The unannotated instances from P set construct the **unlabeled** (U) set together with the N set.

[3] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM, 2008.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[5] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised

Initial Representation	mean IU (aerospace, bicycle, bird, boat, bottle)	mean IU (bus, car, cat, chair, cow)	mean IU (dining table, dog, horse, motorbike, person)	mean IU (potted plant, sheep, sofa, train, TV)
ImageNetModel	0.42 ± 0.01	0.51 ± 0.01	0.49 ± 0.01	0.47 ± 0.01
SingleCategory	0.26 ± 0.01	0.37 ± 0.03	0.29 ± 0.01	0.33 ± 0.04
BinaryLabels	0.30 ± 0.02	0.35 ± 0.01	0.29 ± 0.02	0.35 ± 0.03
TrueLabels	0.29 ± 0.01	0.36 ± 0.01	0.29 ± 0.01	0.37 ± 0.01
AllRandomLabels	0.29 ± 0.01	0.33 ± 0.03	0.26 ± 0.01	0.28 ± 0.01
HalfRandomLabels	0.29 ± 0.00	0.33 ± 0.00	0.26 ± 0.00	0.28 ± 0.00
IncompleteLabels	0.29 ± 0.00	0.29 ± 0.00	0.27 ± 0.00	0.30 ± 0.00
RandomWeights	0.29 ± 0.01	0.29 ± 0.03	0.27 ± 0.01	0.30 ± 0.02

Table 3: Performances of FCN with Alexnet trained to segment 5 categories from the PASCAL VOC2011 dataset with different representation initializations. *CompleteCategory* is the model pre-trained to segment the other 15 categories from the PASCAL VOC2011 dataset; The *PixelObjectness* model was pre-trained to distinguish the instance against the background; The *RandomCategory* model was pre-trained to segment instances with randomly assigned categories from 1 to 15.

Figure 4: Visualization of first-layer features from different pre-trained models.

pre-training. In *Artificial Intelligence and Statistics*, pages 153–160, 2009.

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[7] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[10] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[11] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.

[14] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pages 448–455, 2003.

[15] Xiao-Li Li and Bing Liu. Learning from positive and unlabeled examples with different data distributions. *Machine Learning: ECML 2005*, pages 218–229, 2005.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[18] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 52–59, 2011.

[19] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 567–574, 2012.

Figure 5: Varying the number of categories while pre-training the representation and the pre-trained weights were fine-tuned to segment 5 categories from the PASCAL VOC2011 dataset.

[20] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[21] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[22] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making neural networks robust to label noise: a loss correction approach. *arXiv preprint arXiv:1609.03683*, 2016.

[23] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[24] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[26] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[27] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

[28] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[29] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[30] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[31] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[32] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.