Automatic Labeling of Data for Transfer Learning

Parijat Dube, Bishwaranjan Bhattacharjee, Siyu Huo, Patrick Watson, Brian Belgodere
IBM Research
Yorktown Heights, NY
pdube,bhatta,siyu.huo,pwatson,bmbelgod@us.ibm.com

John R. Kender
Columbia University, NY
jrk@cs.columbia.edu

Abstract

Transfer learning uses trained weights from a source model as the initial weights for the training of a target dataset. A well chosen source with a large number of labeled data leads to significant improvement in accuracy. We demonstrate a technique that automatically labels large unlabeled datasets so that they can train source models for transfer learning. We experimentally evaluate this method, using a baseline dataset of human-annotated ImageNet1K labels, against five variations of this technique. We show that the performance of these automatically trained models come within 6% of baseline.

1. Introduction

In many domains, the task performance of deep learning techniques is heavily dependent on the number of labeled examples. Labeling examples require annotation by subject matter experts, or crowds and are difficult and expensive to acquire. This demand for large labeled datasets has inspired alternative techniques, such as weak supervision or automated labeling, whose algorithms create plausible labels to be used to guide supervised training on other tasks.

Transfer Learning [16] is a well established technique to train a neural network. It uses trained weights from a source model as the initial weights for the training of a target dataset. A well chosen source with a large number of labeled data leads to significant improvement in accuracy.

In this work, we develop a content-aware model-selection technique for transfer learning. We take an unlabeled data point (here, an unlabeled image), and compute its distance to the average response of a number of specialized deep learning models, such as those trained for "animal", "person", or "sport". By applying this technique to an ensemble of specialized models, we create a "pseudolabel" for each piece of unlabeled data consisting of the ordered list of the domain-adapted model names (e.g., "animal-plant-building"), using this synthetic label we can then automatically label large numbers of unlabeled images, and use them to augment training data.

We describe five different methods of pseudo labeling using the above principle and evaluate them using the ImageNet1K [14] dataset. We compare the transfer learning accuracy obtained for a set of target workloads on models trained on the human annotated ImageNet1K labels against models trained on pseudo labels obtained through our methods. We show that using our methods, one can obtain transfer learning accuracies which come within 6% of the human annotated labels.

2. Related Work

There are several well-established approaches that attempt to automatically assign labels to unlabeled images. For example, some use clusters of features to predict labels [8], or augment image data with linguistic constraints from sources such as WordNet [1, 9]. These approaches augment tasks by pretraining models using larger unlabeled datasets. Pretraining approaches have also improved results when attempting a target task for which there is a limited amount of accurately labeled training data [11], by using *weakly* labeled data, such as social media hashtags, which are much more plentiful. However, effectiveness only appears to grow as the log of the image count. Further approaches use generative models such as GANs as in [13] to explore and refine category boundaries between clusters of data, which exploit the rich statistical structures of both real and generated examples, sometimes augmented with labels or linguistic constraints. All of these automatic approaches use the structures present in large unlabeled datasets, to extend the expressivity of known labels, and to augment the raw size of training sets. More broadly, a variety of approaches attempt to learn a representation of a class of data, and later use that representation in service of a target task. For example, [7] clustered images in an embedding space, and developed a meta-learner to find classifications which distinguished various clusters within this embedding. Later, demonstrating improved performance on classification tasks not originally used. Other approaches to mapping the feature space have used autoencoder [2].

Taken together these approaches suggests that there is often rich and meaningful structure present in the data from which useful features can be inferred.

From a practitioner's perspective: these approaches present a trade off: at one extreme, obtaining rich, appropriate, and novel labels from human annotators provides the most expressive power and accuracy for novel examples, but it is also the most expensive. At the other extreme, naively machine-labeling unlabeled data using pre-trained models is quick and inexpensive, but augmenting data sets with these labels is fraught. They are limited by the label set of existing training, and they may simply reinforce biases or unbalanced sets in previously collected data. The literature discussed here attempts to find hybrid approaches that find productive ways to leverage machine-learned distributions of examples to find new ways of characterizing unlabeled data. The current work presents a novel approach in this domain.

3. Approach

We present our technique using a specific case study involving images, and with source datasets created by vertically partitioning ImageNet22K [3] along its distinct subtrees: animal, plant, weapon, tools, music, fungus, sport, person, food, fruit, garment, building, nature, furniture, vehicle, and fabric. These 16 subtrees were used since they were easy to partition from Imagenet22K. However the method could be used with a different number also. We represent each such dataset by a single average feature vector. In this study, this vector is generated from the second last layer of a reference VGG16 [15] model trained on ImageNet1K, as shown in Figure 1, with the average taken over all the images in the dataset. These 16 datasets and their characteristics are shown in Table 1.

Each of these datasets was first split into four equal and disjoint partitions. One partition was used for feature-vector calculation while one was used for validation of target model during finetuning. One-tenth of the third partition was used to create a transfer learning target. For example, the fabric hierarchy has about 160K images. This was split into four equal partitions of about 40K each. The average feature vector for fabric was calculated using data of that size, whereas the target model was fine-tuned with one-tenth of data (∼4K) taken from one of the other partitions. The smaller target datasets is reflective of real transfer learning tasks.

Dataset	Images	Classes
sport	200692	170
tool	175095	317
fruit	185091	307
fabric	159110	232
building	193928	240
furniture	195052	187
garment	214172	252
music	137770	156
nature	192255	241
weapon	102946	138
person	1203512	2880
plant	2224817	4040
tree	562555	995
fungus	135919	299
food	999470	1500
animal	2783256	3796

Table 1: 16 datasets used in our evaluation created using Imagenet22K vertical partitions; 11 of these (shown in bold) are also used to create target datasets for finetuning. Details on how these datasets were partitioned for disjoint source and target tasks are in the text.

To label a new image, we first calculate its own feature vector, then compute the distance between it and each of the representatives of the datasets; in this study, we use the Kullback-Leibler (KL) divergence [10], after appropriate normalization. We have experimented with other measures like Jensen Shannon (JS) and Euclidean also and the results were similar. For the purpose of this paper we will be using the KL divergence measure for the distances. These 16 distances are then used to determine the synthetic label.

Observe that for discrete probability distributions, p and q, KL divergence is a measure of the difference between and is defined as

$$D_{KL}(p,q) = \sum_i p(i) log\left(\frac{p(i)}{q(i)}\right).$$

This is an asymmetric measure as $D_{KL}(p,q) \neq D_{KL}(q,p)$. We are using KL divergence like measure to quantify the difference between two feature vectors. Since KL divergence is defined over probability distributions, we need to appropriately normalize the feature vectors before calculating the KL divergence. For every unlabeled image that comes in, we calculate the KL divergence of the feature vector of the image from those of the datasets we possess, as shown in Figure 2. This distance measure is then used for labeling purposes as described in Section 4

An interesting analogy of our approach for (pseudo) labeling an image is with the "Blind men and an elephant"

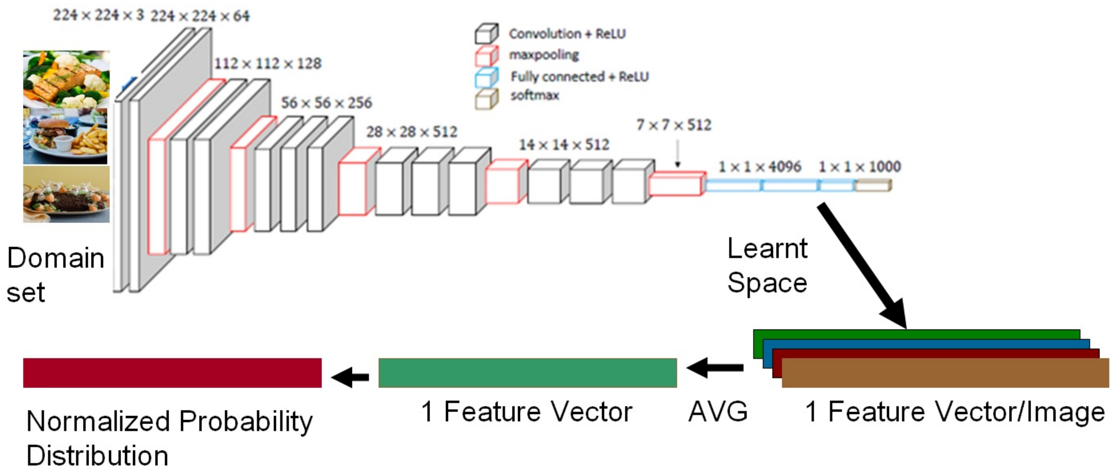


Figure 1: Feature extraction using VGG16 ImageNet1K trained model

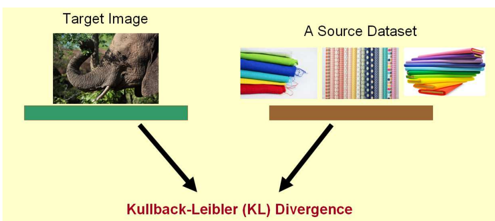


Figure 2: KL divergence calculation between target image and sources

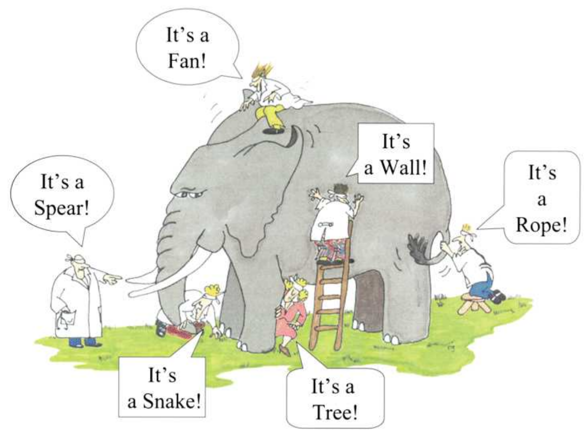


Figure 3: Blind men and an elephant parable cartoon (source unknown).

parable where a group of blind men (who have never learnt about an elephant) try to categorize an elephant just by touching it and then relating it to something that they already know. As illustrated in the cartoon in Figure 3 categorization of an elephant include tree, wall, snake, rope, fan, and spear. Basically by touching and feeling an elephant the blind men are measuring its closeness to things known by them. Our approach, also measures the closeness of an unknown image (in feature space) to existing *known* categories, and then generates label for it.

4. Labeling Methods

Generating rich pseudo labels from models trained on distributional similar data involves a trade off between an expressive, long label, and a generalizable, short label. Longer labels carry more information about similarity between previous models and the target image, and differences between the previously trained models could be critical for properly labeling new examples. For example, a novel set of data including pictures of household objects might be well described by combining the labels of tools, fabric, furniture.

To this end we propose two main methods for assigning psuedolabels: selecting the N "nearest" (Nearest-N), and "closest, farthest, max area" (CFA) as depicted in Figure 4. Using the former strategy should produce a label that has the most detailed information about positively correlated features, this corresponds to the smallest possible bounding hypertriangle, and is most useful in cases where previous models capture the domains features well. However, domains that possess substantial differences from previous data might be better defined by the magnitude and direction of such a difference. For example, the "flower" dataset shares some features with "plants," but is perhaps better defined by statements such as "flowers are very unlike furniture." In other, ambiguous cases, negative features may be necessary to distinguish between overlapping cases: a suit of armor might have similarities with the body shapes of people, but could be contrasted with these categories by its dissimilarity with "sports," a category otherwise close to

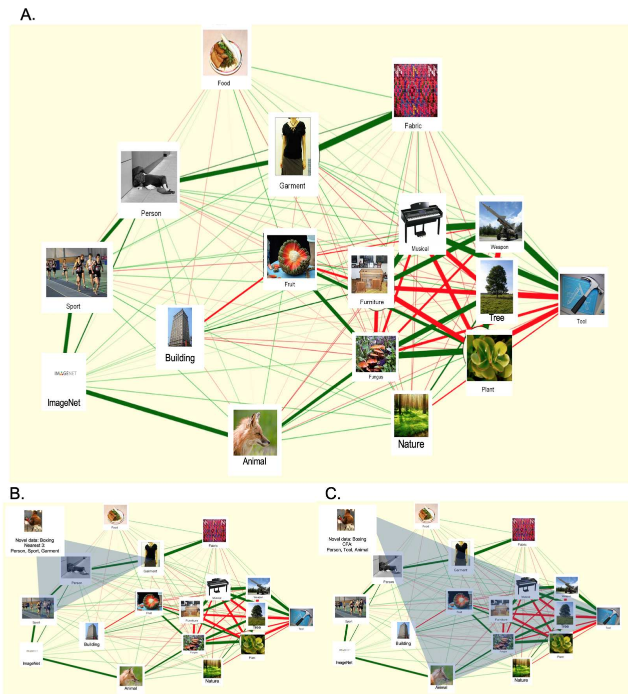


Figure 4: A) This figure shows the similarities between the distributions of 16 existing image models projected into 2d space. B) Projecting a novel data category (”boxing”) into this space, and psuedolabeling by nearest 3. C) Psuedolabeling this same novel category by CFA.

”person.” Thus, we contrast the Nearest-N metric with the CFA metric, which corresponds to the largest bounding hypertriangle.

To label an image, we encode its relative positions with respect to some subset of the known labeled datasets. To better explain the technique, we will use ImageNet1K as a source of images and ground truth labels. ImageNet1K has 1000 labels; the number of images/label is almost uniform; but the labels broadly fall under few broad categories like animals, vehicles, food, musical instruments, garment, furniture, buildings etc. About 446 out of 1000 labels ($\sim 45\%$) belong to animals while the other top categories are vehicles (5.2%), food (2.2%), musical instruments (2.1%) and garment (2.1%). The distribution of number of labels for different categories has a long tail as shown in Figure 5.

4.1. Nearest-N labels

Our first labeling methods choose labels to be the names of the N source datasets which were the closest to the image. We concatenate those names together, in order of closeness. Thus, the Nearest-3 method generates labels like ”tree-animal-fungus”. With 16 source datasets, the Nearest-3 approach yields 16x15x14 = 3360 possible pseudolabels;

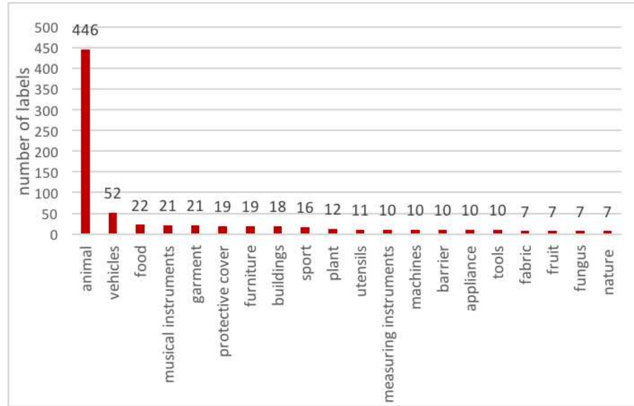


Figure 5: Distribution of labels in ImageNet1K for top 20 categories in decreasing number of labels

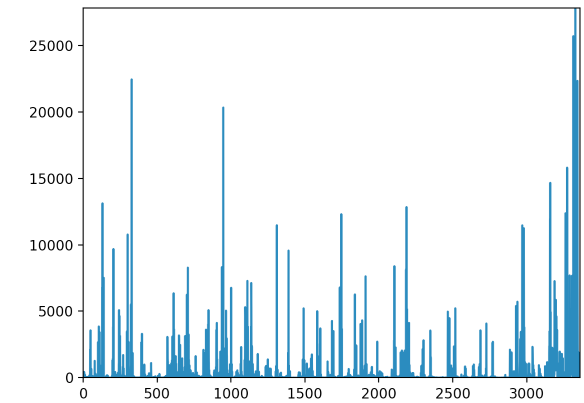
in our study with the 1.3M images from ImageNet1K, each label had a mean of 381 images and a standard deviation of 1503. In a similar fashion, the Nearest-2 and Nearest-1 pseudolabels were also computed, with 240 and 16 possible labels, respectively. (We found no benefits above Nearest-3.) Figure 6a shows the distribution of images for the Nearest-3 method. The high peaks on the right are unsurprisingly related to animals.

4.2. Uniform clustering

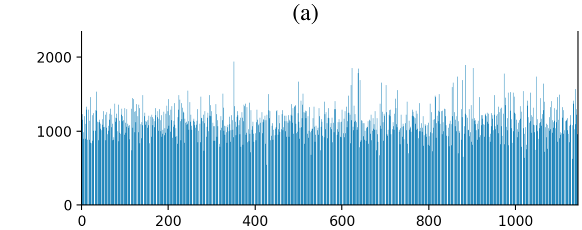
Our fourth method exploited all 16 distances. We first fixed the order of the 16 source datasets, forming a 16-dimensional coordinate system, so that each image could be viewed as a vector in the resulting 16-dimensional simplex of all possible positive distances to them. All of the unlabeled images were then k-means clustered within this space; we used 240 cluster centers to imitate the size of the Nearest-2 space. These resultant clusters could not be expected to be of uniform size, so a second round of splitting and merging, based on relative distances, balanced the cluster membership to be nearly uniform (see Figure 6b), with an average of 1000 images per cluster, and a standard deviation was 193. The ”names” of these final clusters were used as the pseudolabels.

4.3. Closest, Farthest, Max Area (CFA)

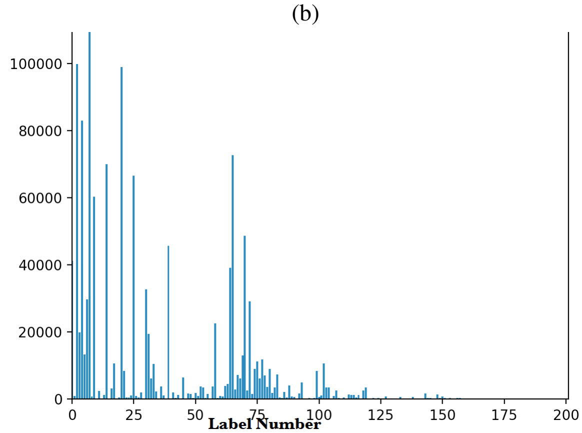
Our fifth method accommodates those incoming datasets that are characterized by a wide variety of low-level image features. These labels were again devised as a sequence of three source names, but chosen to span the 16-dimensional space as widely as possible. The first source dataset was the closest one, and the second was the farthest one. The third was chosen from the remaining 14 to maximize the area of the resulting triangle in 16-space, computed via Heron’s formula. This method not only captures what an image is about (positive description) but also what it is not like (neg-



(a)



(b)



(c)

Figure 6: Distribution of images per label in pseudo-labeled datasets (a) Nearest-3 (b) Uniform Clustering (c) CFA

ative description). In practice, this method only resulted in about 200 labels, with an average of about 6300 images per label, and with a very high standard deviation of about 17000 (see Figure 6c).

As an example, consider the two images, armor suit and elephant, in Figure 7, taken from ImageNet1K. Table 2 shows the KL divergence of these two images from the 16 clusters. The nearest cluster (Nearest-1) for the armor suit image is music. This is possibly due to the brass and metallic textures of the photo, similar to that seen in brass music instruments. The Nearest-2 and Nearest-3 labels are (music-weapon) and (music-weapon-person) respectively.

Figure 7: Pseudolabels assigned to example images from ImageNet1K

cluster	armor suit	elephant
sport	2.086 [A]	1.872
tool	1.947	2.004
fruit	2.379	1.818
fabric	2.016	1.805
building	2.007	1.745
furniture	1.947	2.085
garment	1.888	1.817
music	1.783 [1]	1.983
nature	2.480	1.590
weapon	1.787 [2]	1.898
person	1.842 [3]	1.840
plant	2.532	1.664
tree	2.517	1.473 [1]
fungus	2.475 [F]	1.507 [3, A]
food	2.062	2.126 [F]
animal	2.080	1.498 [2]

Table 2: KL divergence of body armour and elephant with source datasets; the datasets which are nearest first [1], second [2], and third [3] and the farthest [F] and area maximizing dataset [A] are marked.

In contrast, the label for CFA is music-fungus-sport. So for this image, fungus is the source which is most unlike it, and sport maximizes the area of the triangle defined by the third source.

Similarly, for the elephant image, the Nearest-3 label is tree-animal-fungus, and the CFA label is tree-food-fungus. The Nearest-2 and Nearest-1 labels are tree-animal and tree, respectively. Tree qualified as nearest since they have the same rough texture, shape of trunk and color. Apart from this, the image has trees in them. Food was determined to be the furthest from the image with most negative features. Those will include color, shape, texture. Fungus, which was the largest area selection, tends to have some features from both the trees and food. They are seen a lot among trees and greenery and are as colorful and textured like food items.

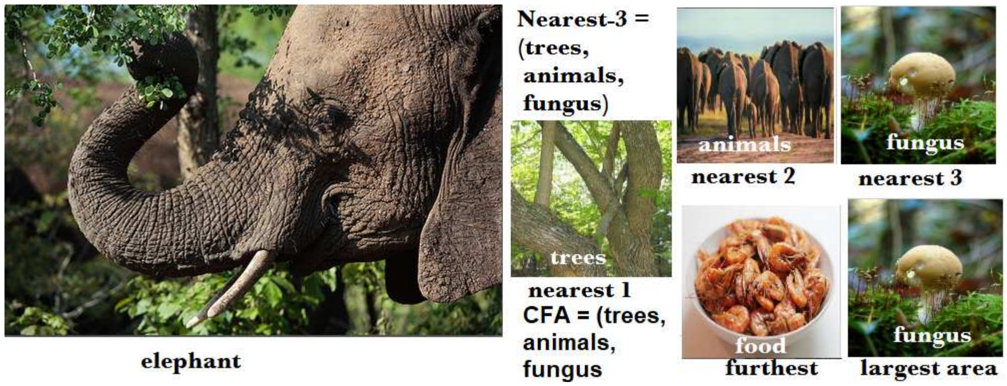


Figure 8: Pseudolabels assigned to elephant image of ImageNet1K

The pseudolabels for elephant image can be visually analyzed in Figure 8.

5. Experimental Evaluation

Experiment 1 Using techniques described in Section 4, we first created five pseudo-labeled datasets for the images in ImageNet1K, as shown in Table 3. We then trained ResNet27 using each of these pseudo-labeled datasets, creating base models for further transfer learning. For all our experiments we used ResNet27 as residual networks [6] are considered state of the art in image classification. The choice of ResNet27, in particular, was dictated by it being easy to train while being big enough for the sizes of datasets being used for the experiments.

We also created two baseline models, one using the vanilla ImageNet1K dataset of images and human-annotated labels, and a second by assigning random labels to the ImageNet1K images. For perspective, Table 3 shows the accuracy of these seven base models, but since we are interested in the *transferability* of representations from these base models to a target domain, their absolute accuracy is not the important measure. For target, we used 12 datasets; 11 of these created from Imagenet22K (refer to Section 3 for details) plus "flowers" dataset [12]. Then, for each of 12 candidate target datasets, we fine-tuned and calculated the transfer learning accuracy of each of the 7 base models; each of these 84 experiments were carried out with the same hyperparameters.

As shown in Table 3, the accuracy obtained with *Vanilla* as the base model can serve as an upper-bound on transfer learning accuracy for experiments using *Nearest-N, Uniform,* and *CFA* base models. Similarly, *Random* can provide a lower bound. Table 4 shows that *Nearest-3* gives the best accuracy for 8 out of 12 datasets. For two datasets, Nearest-2 performs slightly better than Nearest-3, while Uniform and CFA perform best for person and flowers datasets.

dataset	labels	images/label		accuracy
		mean	std-dev	
Nearest-1	16	80073	79325	78.74%
Nearest-2	240	5338	10146	55.75%
Nearest-3	3360	381	1503	37.07%
Uniform	1144	1119	193	33.49%
CFA	201	6373	17648	81.01%
Random	1000	1200	0	0.08 %
Vanilla	1000	1200	0	67.14%

Table 3: Characteristics of different base model datasets, and their accuracy

Experiment 2 To capture the performance of pseudo-labeled vs. human-labeled datasets, we define in the usual way the relative error of transfer learning accuracy between a pseudo-labeled dataset, i, and the *Vanilla* base model, v, as: $Err_i = (1 - accuracy_i/accuracy_v) \times 100\%$. For each target dataset, we also calculate their KL divergence with respect to ImageNet1K, as defined in Section 3.

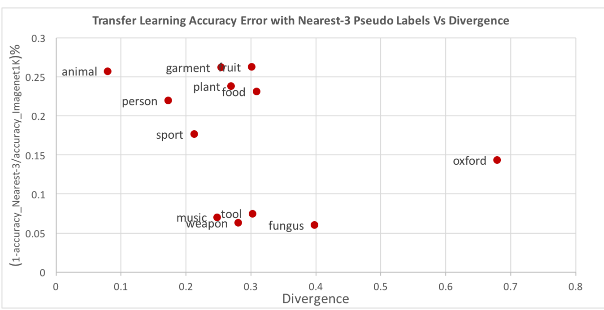
Figures 9a and 9b show plots of Err_i for 12 different target datasets. The average value of Err_i is 17.2%, with minimum and maximum being 6.1% and 26.3%. Thus, using base models trained with automatically generated labels, transfer learning accuracy is on average only 17.2% worse when compared to base models trained using human labeled images. Further, the error shrinks when the divergence increases. This implies that when the base dataset is far away in feature space, the transferability of representations is less sensitive to noise in the labels.

6. Observations

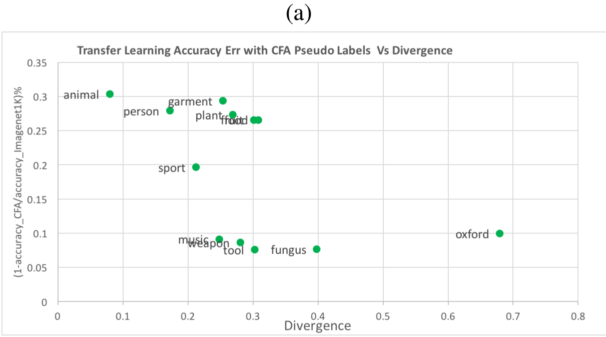
The experimental results in Table 4 indicates that the transferability from a given dataset is proportional to the quality of the labels. The human annotated ImageNet1k labels are of high quality and helps the model trained on it

Base →	Pseudo-labeled						Imagenet1K
Target ↓	Nearest-1	Nearest-2	Nearest-3	CFA	Uniform	Random	Vanilla
music	42.98%	43.60 %	**43.86%**	42.87 %	43.71%	1.57%	47.19%
tool	38.79%	39.12%	**39.44%**	39.40%	39.39%	1.24%	42.65%
weapon	29.51%	**30.24%**	30.21%	29.46%	29.92%	2.09%	32.25%
fungus	21.28%	21.96%	**22.16%**	21.78%	21.88%	1.60%	23.59%
flowers	75.94%	74.90%	72.88%	**76.64%**	72.36%	0.43%	85.13%
sport	28.68%	30.46%	**30.76%**	30.01%	30.74%	0.98%	37.37%
person	6.87%	7.25%	7.89%	7.29%	**8.05%**	0.12%	10.12%
food	8.58%	9.21%	**9.62%**	9.19%	9.36%	0.13%	12.52%
fruit	18.53%	**19.53%**	19.12%	19.05%	18.54%	0.82%	25.95%
garment	16.84%	17.30%	**18.05%**	17.29%	17.61%	0.68%	24.48%
animal	15.40%	17.54%	**18.46%**	17.31%	18.40%	0.10%	24.87%
plant	10.46%	11.27%	**11.69%**	11.14%	11.47%	0.08%	15.34%

Table 4: Transfer learning accuracy for different target datasets; best accuracy shown in bold.



(a)



(b)

Figure 9: Relative error between pseudo-labels and Vanilla, vs. divergence, for 12 target datasets (a) Nearest-3 (b) CFA

learning fine details including for basic features like colors, textures, shapes etc. This then gets transferred during transfer learning.

It is also seen that in general, having coarser labels (like Nearest-1) means the base model does not need to learn much to be able to differentiate between labels. This then becomes an hindrance for transferabilty. With more deeper labels (like Nearest-3), the base model learns to differentiate on specific features and all that knowledge helps in transferabilty. Thus although the top-1 accuracy as seen in Table 3 for Nearest-1 is higher than Nearest-2 or Nearest-3, it did not learn as much to achieve it, and that was an hindrance.

In these experiments we have used 16 different specialized models/datasets as anchor points. The diversity of these datasets does influence the quality of pseudo labels produced. A set of datasets which cover as much of the high dimensional feature space as possible would produce better pseudo labels. For example, from Figure 4, a diverse dataset set which has (plants, weapons, garments, building) would provide better label coverage than (plant, tree, fungus, fruit).

Our method, for every image in the target dataset, requires the KL divergence calculation for M feature vector pairs where M is the number of source datasets. In our experiments it is 16. To determine the nearest N labels, these M divergence numbers need to be sorted and the bottom N need to be picked. To determine the CFA labels, the top 1 and bottom 1 from the sorted list needs to be picked. Then Heron's formula is applied to determine the third label which maximizes area. This requires $M-2$ area calculations. Then to pick the the label maximizing the area, an additional $M-2$ comparisons are needed.

In our work we have only looked at feature vectors from the second last layer of reference convolutional neural network (CNN). A recent work [5] has shown that features from all the layers of a CNN can potentially be used for knowledge representation. This study implies devising pseudo labeling techniques exploiting representations not just from a single layer but multitude of convolutional layers in a CNN.

The results we reported did not involve any exploration of learning rate to improve transfer learning accuracy. It has been shown empirically in [4] that appropriate selection of learning rates for differ layers of a CNN during finetuning can significantly improve the accuracy. It will be interesting to investigate the possible improvements in transferability of pseudo labeled datasets by selecting the right learning rates during finetuning.

7. Conclusion

We have shown that generation of content-aware pseudolabels can provide transfer performance approaching that of human labels, and that models trained on psuedolabels can be used as source models for transfer learning. The automated approach presented here suggests that the internal representations of content models trained on specialized datasets contain some descriptive features of those datasets. By treating each of these specialized representations as a "word" in a longer "sentence" that describes a category of images, we can create labels such as a "music-weapon-person" to describe a suit of armor, or a "tree-animal-fungus" to describe an elephant. These rich labels capture features of these objects such as visual information about the materials they are made out of, that better describe the contents than reliance on a single label would produce. Using multiple, content-aware models to achieve greater descriptive power may be a valuable future avenue of research.

References

[1] Duygulu Pinar-Forsyth David Freitas Nando de Blei David Jordan Michael Barnard, Kobus. Matching words and pictures. *Journal of machine learning research*, 3(Feb):1107–1135, 2003. 1

[2] Gabriel B. Cavallari, Leonardo Sampaio Ferraz Ribeiro, and Moacir Antonelli Ponti. Unsupervised representation learning using convolutional and stacked auto-encoders: a domain and cross-domain feature space analysis. *CoRR*, abs/1811.00473, 2018. 2

[3] W. Deng, J.and Dong, L.-J.and Li K. Socher, R.and Li, and L FeiFei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on CVPR*, 2009. 2

[4] Parijat Dube, Bishwaranjan Bhattacharjee, Elisabeth Petit-Bois, and Matthew Hill. Improving transferability of deep neural networks. *CoRR*, abs/1807.11459, 2018. 7

[5] Dario Garcia-Gasulla, Ferran Parés, Armand Vilalta, Jonathan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. On the behavior of convolutional nets for feature extraction. *CoRR*, abs/1703.01127, 2017. 7

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on CVPR*, 2016. 6

[7] Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *CoRR*, abs/1810.02334, 2018. 2

[8] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 119–126, New York, NY, USA, 2003. ACM. 1

[9] Rong Jin, Joyce Y Chai, and Luo Si. Effective automatic image annotation via a coherent language model and active learning. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 892–899. ACM, 2004. 1

[10] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. 2

[11] Girshick Ross-Ramanathan Vignesh He Kaiming Mahajan, Dhruv. *Exploring the Limits of Weakly Supervised Pretraining: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 185–201. 09 2018. 1

[12] M.E Nilsback and A Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 6

[13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 1

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[15] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[16] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014. 1