Training a single multi-class convolutional segmentation network using multiple datasets with heterogeneous labels: preliminary results

Fanjie Kong[1], Cheng Chen[1], Bohao Huang[1], Leslie M. Collins[1], Kyle Bradbury[2], Jordan M. Malof[1]

[1]Department of Electrical & Computer Engineering, Duke University, Durham, NC 27708
[2]Energy Initiative, Duke University, Durham, NC 27708

Abstract— **Segmentation convolutional neural networks (CNNs) are now popular for the semantic segmentation (i.e., dense pixel-wise labeling) of remote sensing imagery, such as color or hyperspectral satellite imagery. In recent years a large number of hand-labeled datasets of overhead imagery have emerged, leading to breakthrough performance for CNNs. However, these datasets are typically used in isolation of one another because they are either (i) annotated with heterogeneous object type labels, or (ii) they are collected over different geographic areas. This imposes a major bottleneck on the value of these datasets. In this work we present what we call a class-asymmetric loss function that makes it possible to train a single multi-class network using multiple datasets that are heterogeneously-labeled. We show, for example, that it is possible to train a segmentation algorithm for Buildings, roads, and background using two datasets: one annotated with buildings and one annotated with buildings. We propose a class asymmetric loss that under certain common conditions, allows for one to train models on datasets in which the target class is unlabeled.**

Keywords—semantic segmentation, convolutional neural networks, deep learning, aerial imagery, building detection

1. INTRODUCTION

Convolutional neural networks have garnered substantial attention in recent years for the segmentation of overhead imagery (e.g., satellite or aerial imagery). Such models have achieved top performance on several satellite image computer vision benchmarks such as a recent Urban Challenge [1], the INRIA building labeling competition [2], and the recent DeepGlobe satellite challenge [3].

High-capacity learning models such as convolutional neural networks (CNNs) require substantial quantities of labeled training data to perform well without overfitting to training data. Therefore, in addition to the development of effective CNNs, a major cause for recent performance breakthroughs has been the development of large annotated remote-sensing datasets. Table I summarizes several large recent benchmark datasets.

Thus far researchers and benchmark competitors train separate models: one model for each label-homogenous subset of data. This approach is illustrated in Fig. 1a. However, to increase model generalizability we would ideally represent the greatest diversity of geographic domains possible within a training set, and so we consider the problem of training a single CNN using all of these datasets. In aggregate, the datasets in Table 1, for example, provide an unprecedented level geographic and temporal diversity (time of day, season of the year, etc.) Furthermore, such datasets are expensive and time-consuming to create, providing additional motivation to combine existing training data resources.

One challenge with this goal however is that different datasets contain annotations of different object categories, leading to heterogeneity of labels across the datasets. For example, one dataset may contain road (R) annotations, and another dataset may contain building (B) annotations even when both datasets have both roads and buildings visible in the imagery. This is the case with the datasets in Table I. Even within the same benchmark dataset, e.g., DeepGlobe, the building and road labels may be made on geographically disjoint subsets of the data.

Table 1: Recent labeled overhead imagery datasets. The datasets contain annotations of either Roads (R) or Buildings (B). A bar over the letter indicates the set complement. Some datasets, such as DeepGlobe have two spatially disjoint datasets: one labeled with R, and one labeled with B. The classes are labeled as $\{\boldsymbol{R}, \bar{\boldsymbol{R}}\}$, $\{\boldsymbol{B}, \bar{\boldsymbol{B}}\}$ to indicate this. When labels spatially co-occur we use $\{\boldsymbol{R}, \boldsymbol{B}, \boldsymbol{Bg}\}$ where $\boldsymbol{Bg} = \bar{\boldsymbol{B}} \cup \bar{\boldsymbol{R}}$

Name	Year	Classes	No. of Cities	Area (km^2)
DeepGlobe [3]	2018	$\{R, \bar{R}\}, \{B, \bar{B}\}$	7	~11000
INRIA Building [2]	2017	$\{B, \bar{B}\}$	10	360
Duke road & building dataset [5]	2017	$\{R, B, Bg\}$	9	60
Urban challenge [1]	2017	$\{B, \bar{B}\}$	3	361

1.1. Multi-task versus multi-class problems

One straightforward solution to this problem – which has become popular in the deep learning community - is multi-task learning [4]. A multi-task approach to training with all datasets in Table I, for example, might involve a CNN with different parallel output nodes as illustrated in **Error! Reference source not found.**b. This is potentially an effective solution, however the classes in this case (R and B) are mutually exclusive (i.e., they cannot co-occur on the same pixel). In other words, a single pixel cannot (usually) be labeled as both B R. However, a multi-task solution assumes they *can* co-occur. A multi-class formulation of the problem,

shown in Fig. 1c, however does not permit labeling the same pixel with two labels. Such a model returns a probability distribution over the *three* possible classes: R, B, $Bg = \overline{B \cup R}$. Bg here is termed the "background" class in many contexts. As a result of this label mutual exclusivity, we will focus on multi-class CNN formulations in this work. We will propose a simple adjustment to the standard training procedure for SCNNs in order to facilitate training a single multi-class SCNN on multiple partially-labeled datasets. This will facilitate our subsequent goal, explained next.

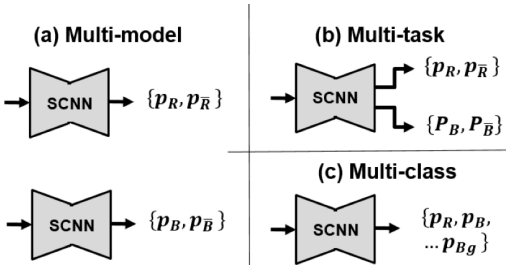

Fig. 1. Illustration of three possible formulations of the building and road segmentation problems. Let p_R and p_B refer to the probability of Road (R) and Building (B) respectively. The bar over a letter indicates a set complement. (a) A multi-model formulation, in which a separate model is trained for each problem. (b) A multi-task formulation, in which a single model is trained with two separate binary outputs. A multi-class formulation (b) results in a 3-way output, where we define $Bg = \overline{B \cup R}$.

1.2. A class-asymmetric loss for differently-labeled data

An important consequence of mutual exclusivity in the classes is that it provides an opportunity for additional supervision during training. Consider a multi-class (R,B,Bg) network that we wish to train on a dataset annotated only with $\{R, \bar{R}\}$. The network in this case can be penalized for predicting B, given that the true label is R.

		Predicted label		
		R	***B***	***Bg***
True label	***R***	√	X	X
	$\bar{R}$ = (Bg ∪ B)	X	O	√

Fig. 2. A truth table for 3-class SCNN making predictions on a dataset with only 2-classes labeled. The SCNN predicts one of three classes: Road (*R*), Building (*B*), and Background (*Bg*). The ground truth is assumed to be labeled with roads, and therefore the labels are *R* and $\bar{R}$ (equivalent to $Bg \cup B$). Red boxes with 'X' indicate an incorrect prediction, green boxes with checkmarks indicate a correct label, and yellow boxes with 'O' indicates ambiguity. Because the classes *B* and *R* are mutually exclusive, it is possible to say the network is making an error when the true label is *R*, even though the dataset was not annotated with building labels.

In this work we also propose a simple loss function, which we call class-asymmetric (CA) loss, that incorporates the extra supervision available when the data is labeled with mutually exclusive class labels, as illustrated in Fig. 2. In this preliminary work, we conduct experiments on the Duke Road/Building dataset in Table I and compare it to two conventional baseline approaches. The first baseline approach is simply to train one model on each of the available training datasets. The second baseline is to train a 3-class SCNN on all of the available datasets, but without utilizing the proposed class-asymmetric loss. For this we will use our simple modification to the standard SCNN training procedure, allowing the network to be trained on multiple datasets, but without leveraging the proposed CA loss. We then show that the proposed loss, when added to multi-dataset training, yields substantial performance improvements over the two baseline approaches.

The remainder of this paper is organized as follows: Section 2 describes the Duke dataset; Section 3 discusses how we train a single multi-class network on partially-labeled datasets, as well as the proposed class-asymmetric loss; Section 4 and 5 discuss the experimental design and results, respectively; Section 6 discusses the conclusions.

2. OVERHEAD IMAGERY DATASET

In this work we use the Duke Building/Road dataset [5], which is summarized in Table 1. This dataset is comprised of 30cm resolution color (RGB) overhead imagery from the US Geological Survey. Although the dataset is comprised of 10 cities, we use the city Arlington, MA because it has building and road labels over the same geographic area, providing greater control for our experiments (see Section 4.1). Our final dataset is therefore comprised of three large image tiles covering a total of 7.5 km^2 of area. In our experiments, we crop the original images into chips with 572 ×572 pixels. Ultimately, we have 507 patches of the images in total.

3. LOSS FUNCTIONS TO LEVERAGE HETEROGENEOUSLY-LABELED DATASETS

In this section we explain the basic cross-entropy loss function that is standard for SCNNs, followed by a simple adjustment we make in order to train a multi-class network on a partially-labeled dataset. This will facilitate training a single network on multiple partially-labeled datasets. The last subsection discusses our proposed class-asymmetric loss. For simplicity, in all of these discussion we will assume a we have a set of only two possible class labels, $\ell = \{1,2\}$, although the methods can plausibly be extended to greater numbers of classes.

3.1. Brief review of convolutional segmentation networks

SCNNs vary in their architectures and training procedures, but, like other neural networks, SCNNs use a cross-entropy loss given by

$$\mathcal{L}_{CE} = -\sum_{i=0}^{|\ell|+1} y_i \log(p_i) \quad (1)$$

where M is the number of classes, p_i is the predicted probability of one class for each pixel and y_i is the true label of one class for each pixel. The output of the network is the one-hot coding of the predicted labels.

The major difference between SCNNs and other networks is that SCNNs provide pixel-wise labeling of their input imagery (as opposed to a single image-level label), and similarly, the cross-entropy loss is applied pixel-wise to the output of the SCNN.

3.2. Modifying the cross-entropy loss for training a multi-class network on partially-labeled datasets

The cross-entropy loss in equation (1) assumes that the output labels provided by the SCNN are identical to those annotated in the training dataset. In this section we describe a simple adjustment to $\mathcal{L}_{CE}$ in order to account for scenarios in which a subset of the total labels of the network, ℓ, are labeled in the dataset. This loss will be used in our experiments, and is a useful prerequisite to describing our class-asymmetric loss. For simplicity, let us assume a scenario again in which we have a set of only two possible class labels, $\ell = \{1,2\}$, and two dataset D_1 and D_2, where each dataset is annotated with just one of the two possible labels, indicated by their respective subscripts. For example, D_1 is only labeled with class 1. Then the proposed "partial-labeling" loss is given by

$$\mathcal{L}_{PL}(\boldsymbol{D_k}) = -\sum_{i \in \ell} \boldsymbol{y_i} \log\left(\boldsymbol{p_{i|D_k}}\right). \quad \textbf{(2)}$$

Here the loss function depends upon which of the available datasets is being employed for training, indexed by k. We still sum over all possible labels, but we compute the SCNN's softmax only over the subset of labels present in the training dataset; this alteration is denoted by $p_{i|D_k}$ and can be interpreted as the probability of class i, given that the probability of all labels not in dataset D_k are equal to zero (i.e., we apply a softmax only over the available labels). In this case, the prediction for the unlabeled class is always equal to zero, and of course the ground truth label of the missing class is also 0 (i.e., $y_{\ell/k} = 0$), causing the unlabeled class to have no impact on the loss function. Note that this loss does not assume, or exploit, mutual exclusivity of the class labels. This loss is a simple way to train a multi-class network even when labels are missing in the training dataset.

3.3. The class-asymmetric (CA) loss

Here we propose a "class-asymmetric" (CA) loss for exploiting mutual exclusivity among labels in order to better-leverage partially-labeled datasets. The motivation and intuition for exploring such a loss was described in Section I. The main idea of the CA loss is to implement the logic in Fig. 2. We will begin by presenting the CA loss, and then subsequently explain the terms in it. The CA loss is given by

$$\mathcal{L}_{CA}(\boldsymbol{D_k}) = -\boldsymbol{y_k} \sum_{j \in \ell / k} \boldsymbol{\sigma}\left(\boldsymbol{z_j}\right). \quad \textbf{(3)}$$

Once again this loss depends on the dataset currently being utilized for training, dentoed D_k. The index j refers to the output nodes from the SCNN for each class. As indicated in the summation, we only sum over those labels that are *not* provided in the training dataset, as indicated by the set difference ℓ/k; This is because we assumed only one label, k, is provided in the training dataset. The term y_k in front of the summation imposes that this loss is only applied when $y_k = 1$ for the pixel under consideration. The $\sigma(z_j)$ refers to a sigmoid function acting on the output of the SCNN for the label j, prior to the softmax operation. We do not sofmax with the missing label because it is not present in the data, but we use the sigmoid function to constrain the magnitude of the penalties on mistakse.

4. EXPERIMENTS

The major goal of our experiments is to explore how effectively we can leverage partially-labeled datasets with the proposed CA loss. For these experiments we will leverage the Duke dataset in Table I, and specifically the subset of data discussed in Section II. This dataset was used because it is small and experiments and proof-of-concept results can be obtained efficiently. The dataset includes two sets of labels – one for road and one for building - over *the same geographic region*, which will provide us with additional experimental control. Let Roads and buildings have the numerical labels 1 and 2, respectively. With these labels we can create three datasets: D_1,D_2, and $D_{\{1,2\}}$. Here $D_{\{1,2\}}$ is an ideal scenario, with all three class labels present in the dataset. We will use these datasets to develop several models in order to understand the benefits of the CA loss.

Upper bound model: Because the Duke dataset contains both road and building labels, we can build $D_{\{1,2\}}$. This dataset provides an ideal scenario in which we have all labeling information, permitting us to estimate an upper bound on the performance achievable with partially labeled data over the same location. To obtain this upper bound, we will train a conventional three-class SCNN using $D_{\{1,2\}}$ using $\mathcal{L}_{CE}$ in equation (1). The remaining experiments will all assume that we have only D_1 and D_2. Although these datasets cover the same imagery, we will only be training with partial information about the true labels.

Lower-bound model: We estimate a lower bound on performance by simply training two separate two-class models using $\mathcal{L}_{CE}$ loss: one model for D_1 and another model for D_2. In this case the two models share no information from the two sets of available labels. Any approach that effectively utilizes D_1 and D_2 should outperform this lower bound.

Baseline model: Next we create a baseline approach using a naïve strategy for leveraging two partially-labeled datasets. In the baseline approach, we train a single three-class SCNN by alternating mini-batches from D_1 and D_2 using the partial-label loss, $\mathcal{L}_{PL}$, in equation (2). This is similar to a multi-task training regime, except we use $\mathcal{L}_{PL}$ to train a multi-class model. However, in this case we are not leveraging the mutual-exclusivity of the class labels.

Proposed approach: In this case, we perform the exact same procedure as we do for the Baseline model, except our loss function now includes the CA loss. The total loss function for the Proposed approach is given by

$$\mathcal{L}_{CA}(D_k) = \mathcal{L}_{PL}(D_k) + \lambda\,\mathcal{L}_{CA}(D_k). \quad (4)$$

The λ coefficient provides a way to balance the two loss functions. We set $\lambda = 1$ for simplicity, assuming equal weighting, but in principal it practice it could be adjusted.

4.1. The U-net segmentation network and training details

For our experiments we use a U-net network[6]. We use a (slightly) modified version and training procedure that recently achieved the highest accuracy on the Inria benchmark competition[2]. The final network architecture is illustrated in Fig. 3. We train our networks using Adam optimizer with learning rate 1e-4. The training is finished for 200 epochs. The input size of patches is 572×572 with symmetric padding. We use a batch size of 5 to train our networks. For every experiment, we split the dataset into training set, validation set and test set. Each set contains 169 patches from the original images. All experiments used exactly the same images in the training, validation, and testing datasets, respectively.

5. RESULTS

The final results of all the experiments are summarized in Table 2. At the inference time, we measure the performance of our models by IoU (Intersection over Union), also known as Jaccard index. The experimental results demonstrate that our proposed approach using the CA loss function is able to outperform (IoU 0.488) the more naïve approaches to this problem of using individual classifiers (IoU 0.372) and even a reasonable baseline (IoU 0.415). In this case, the multi-task learning approach (the Baseline) improved performance of the network compared to single task learning (the Lower Bound). All methods are outperformed by the single 3-class SCNN trained on homogeneously-labeled trained data (the best possible case). Of the cases in which the labels were heterogeneous across datasets our proposed method outperformed other approaches in this experiment. We summarize these results in Table 2.

Table 2: Experimental results

Method	Building IoU	Road IoU	Mean
Lower bound	0.540	0.203	0.372
Baseline	0.623	0.207	0.415
Proposed approach	0.656	0.319	0.488
Upper bound	0.728	0.511	0.620

6. CONCLUSIONS

In this work we explore techniques for leveraging the large number of annotated remote sensing datasets that have been developed recently for training segmentation convolutional neural networks (SCNNs). Combining these datasets is not straightforward because the datasets are often differently labeled: they are annotated with somewhat different objects. We present a (AC) loss function to help train multi-class SCNNs using datasets with differing labels. We compare the proposed approach using the AC loss to a baseline approach, indicating it provides performance improvements.

ACKNOWLEDGEMENTS

We thank the NVIDIA corporation, the NSF XSEDE computational environment, and Duke Research Computing for providing computing resources for this work. We also thank the Duke University Energy Initiative and the Sloan foundation for their support.

REFERENCES

[1] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition," vol. June, 2017.

[2] B. Huang, K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J. M. Malof, A. Boulch, B. Le Saux, L. Collins, K. Bradbury, S. Lefevre, and M. El-Saban, "Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark," *Int. Geosci. Remote Sens. Symp.*, 2018.

[3] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images," pp. 172–181, 2018.

[4] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," pp. 1–20, 2017.

[5] K. Bradbury, B. Brigman, L. M. Collins, T. Johnson, S. Lin, and R. Newell, "Arlington, Massachusetts - Aerial imagery object identification dataset for building and road detection, and building height estimation," 2016. [Online]. Available: https://figshare.com/articles/Arlington_Massachusetts_-_Aerial_imagery_object_identification_dataset_for_building_and_road_detection_and_building_height_estimation/3485204.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Miccai*, pp. 234–241, 2015.