Large Scale High-Resolution Land Cover Mapping with Multi-Resolution Data

Caleb Robinson
Georgia Institute of Technology

Le Hou
Stony Brook University

Kolya Malkin
Yale University

Rachel Soobitsky
Chesapeake Conservancy

Jacob Czawlytko
Chesapeake Conservancy

Bistra Dilkina
University of Southern California

Nebojsa Jojic*
Microsoft Research

Abstract

In this paper we propose multi-resolution data fusion methods for deep learning-based high-resolution land cover mapping from aerial imagery. The land cover mapping problem, at country-level scales, is challenging for common deep learning methods due to the scarcity of high-resolution labels, as well as variation in geography and quality of input images. On the other hand, multiple satellite imagery and low-resolution ground truth label sources are widely available, and can be used to improve model training efforts. Our methods include: introducing low-resolution satellite data to smooth quality differences in high-resolution input, exploiting low-resolution labels with a dual loss function, and pairing scarce high-resolution labels with inputs from several points in time. We train models that are able to generalize from a portion of the Northeast United States, where we have high-resolution land cover labels, to the rest of the US. With these models, we produce the first high-resolution (1-meter) land cover map of the contiguous US, consisting of over 8 trillion pixels. We demonstrate the robustness and potential applications of this data in a case study with domain experts and develop a web application to share our results. This work is practically useful, and can be applied to other locations over the earth as high-resolution imagery becomes more widely available even as high-resolution labeled land cover data remains sparse.

1. Introduction

Land cover mapping is a semantic segmentation problem: each pixel in an aerial or satellite image must be classified into one of several land cover classes. These classes describe the surface of the earth and are typically broad categories such as "forest" or "field". High-resolution land cover data ($\leq$ 1m / pixel) is essential in many sustainability-related applications. Its uses include informing agricultural best management practices, monitoring forest change over time [10] and measuring urban sprawl [31]. However, land cover maps quickly fall out of date and must be updated as construction, erosion, and other processes act on the landscape.

In this work we identify the challenges in automatic large-scale high-resolution land cover mapping and develop methods to overcome them. As an application of our methods, we produce the first high-resolution (1m) land cover map of the contiguous United States. We have released code used for training and testing our models at `https://github.com/calebrob6/land-cover`.

Scale and cost of existing data: Manual and semi-manual land cover mapping of aerial imagery is currently expensive and scales poorly over large areas. For example, the Chesapeake Conservancy spent 10 months and $1.3 million to produce a high-resolution (1m) land cover map of the Chesapeake Bay watershed in the Northeast US. This project, the largest of its kind, labeled only $\sim$160,000 km^2, or 2% of the US [5]. Existing benchmark land cover segmentation datasets and studies are limited to even smaller scales. The DeepGlobe challenge dataset [6, 24] covers a total area of 1,717 km^2, the Dstl satellite imagery dataset [2] covers $\sim$400 km^2, the UC Merced land use dataset [30, 4] covers just 7 km^2, and the ISPRS Vaihingen and Potsdam dataset [1] contains fewer than 36 km^2 of labeled data. In comparison, a single layer of aerial imagery of the contiguous US covers 8 million km^2 (8 trillion pixels at 1m resolution), occupying 55 TB on disk – two orders of magnitude larger than ImageNet, a standard corpus for training computer vision models. Deep learning-based approaches for land cover mapping have shown to be effective, however, in limited-size studies: [19] compare common CNN image classification architectures at a 6.5m spatial resolution in a small part of Newfoundland, Canada, while [7] use a multi-resolution approach for handling panchromatic and multispectral bands separately at a 1.5m spatial resolution in a $\sim$4000 km^2 area.

*jojic@microsoft.com

Figure 1: Example 1 km^2 image patches. **Top row:** NAIP imagery from 2012, NAIP imagery from 2015, ground truth land cover. **Bottom row**: Landsat leaf-on imagery, Landsat leaf-off imagery, NLCD land cover.

Model generalization: High-resolution land cover labels at 1m resolution only exist at concentrated locations. Such localized label sets have not been successfully used to classify land cover on a larger scale. Indeed, we show that neither standard random forest approaches [11], nor common semantic segmentation networks generalize well to new geographic locations: models trained on a single Northeast US state see their performance degrade in the entire Northeast region, and further in the rest of the country. Existing GIS methodology such as Object Based Image Analysis (OBIA) [31, 15] suffers from the same generalization issues, yet costs more in terms of data and effort to deploy. For example, OBIA methods have been used to create high-resolution (1m) land cover maps in part of a county in Indiana [17] and the city of Phoenix, Ariz. [16], but rely on human-engineered features and hand-derived rule-based classification schemes.

In view of this, we develop methods for generalizing models to new regions, achieving **high-quality results in the entire US**. Specifically, we augment high-resolution imagery with low-resolution (30m) satellite images, extend labels with low-resolution land cover data that we use as weak supervision, and augment data with inputs from multiple points in time (see Fig. 1). We evaluate models trained with these methods in the US: a) with ground-truth labels from the Chesapeake Bay area in the Northeast US; b) through visualizing their outputs in other US regions; and c) by comparing their predictions with low-resolution land cover labels over the entire US. As low-resolution satellite and land cover data sources are widely available, such as Landsat satellite imagery, or Global Land Cover [23], our methods are applicable wherever high-resolution imagery exists.

Evaluation: An important consideration for large-scale land cover mapping tasks is the cost associated with executing a trained model over massive scales. We run our best model, a U-Net variant, over the entire contiguous US to produce a **country-wide high-resolution land cover map**. This computation took one week on a cluster of 40 K80 GPUs, at a cost of about $5000, representing massive time and cost savings over the existing methods used to produce land cover maps. We provide a web tool through which users may interact with the pre-computed results – see `http://aka.ms/cvprlandcover` – exposing over 25TB of land cover data to collaborators.

In practice, land cover models must be verified and updated with human input. Our proposed models can be adapted to new regions with relatively little human labor. In a **study with domain experts**, we evaluated our best model (trained in the Chesapeake Bay area from the Northeast US) on a region in Iowa, then obtained manual corrections on $\sim 1\%$ of this territory. Using these corrections, we fine-tuned our model output, which reduced both the overall error and the manual labor required to perform corrections over the entire area.

2. Multi-Resolution Data Fusion

We assume that we are given a training set of pairs of high-resolution satellite or aerial imagery and high-resolution land cover labels, $\{(X^{(t)}, Y^{(t)})\}_{t=1}^{T}$ where $X^{(t)} = \{X_{ijk}\}_{i,j,k}^{(t)} \in \mathbb{R}^{h \times w \times c}$ is a multispectral image with height h, width w, and channel depth c, and $Y^{(t)} = \{Y_{ij}\}_{i,j}^{(t)} \in \{1, \ldots, L\}^{h \times w}$ are the associated land cover labels. A straightforward approach for training a deep neural network, $f(X;\theta) = \hat{Y}$, on this fully supervised semantic segmentation problem involves minimizing a standard loss function with respect to the network's parameters,

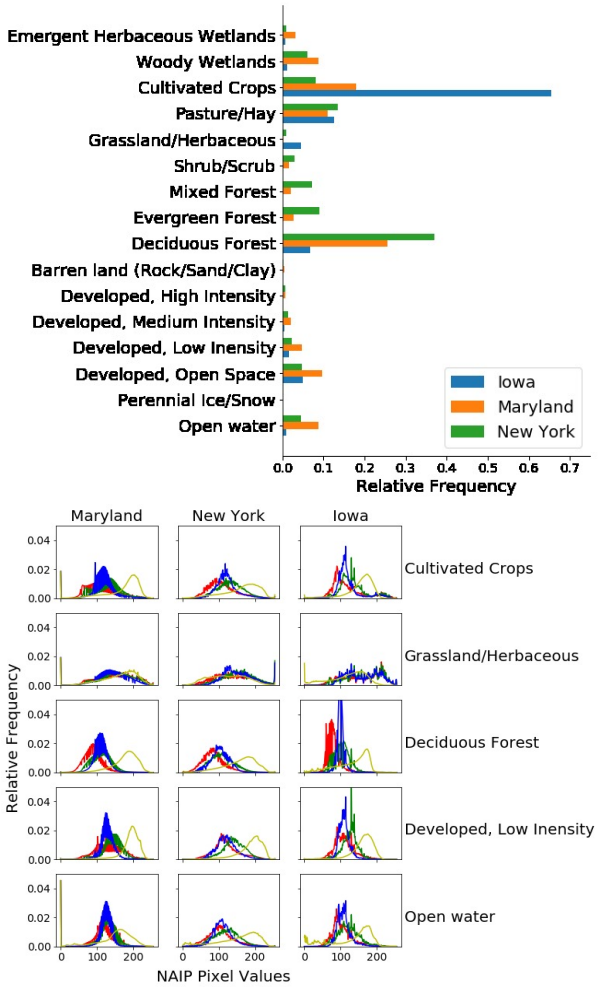

Figure 2: **Top** Inter-state differences in NLCD class composition, **bottom** Inter-state NAIP color histograms per NLCD class. Different states cover different geographies and have different *land use* purposes (e.g., over 60% of Iowa is covered with cultivated crops, while Maryland has a more uniform distribution of land cover) and different color profiles for each class.

i.e., $\min_\theta J(Y, \hat{Y})$. This approach generally works well in problems where potential *test* images are sampled from the same generating distribution as the training images.

This assumption does *not* hold for the land cover mapping problem as high-resolution input images will vary due to differences in: geography of the earth, atmospheric conditions (e.g., cloud cover), ground conditions (e.g., flood conditions), quality and type of sensor used in capturing the image, time of day or season that the image was captured, etc. Indeed, these differences are obvious in the input data we use later in this study, see Figure 2. As a result, models trained with standard deep learning methods fail to generalize over wide areas, see Section 5. We propose the following methods to improve model performance:

Low-Resolution Input Augmentation (LR): Publicly available low-resolution satellite data has been collected globally since 1972, starting with the Landsat 1 satellite. We find that augmenting high-resolution imagery with low-resolution imagery that has been averaged over large time horizons improves model performance. This averaged low-resolution imagery is less susceptible to sources of local noise that impact high-resolution imagery and can therefore be used by models to smooth such noise. Formally, for every high-resolution image, $X^{(t)}$, we assume that we can access a low-resolution image $Z^{(t)} \in \mathbb{R}^{h' \times w' \times c'}$. We resample the low-resolution imagery to the same spatial dimensions ($h \times w$) as the high-resolution imagery, then concatenate the two image sources, giving new input imagery $X'^{(t)} \in \mathbb{R}^{h \times w \times (c+c')}$.

Label Overloading (LO): The available hand-labeled land cover maps are created from a single time point of aerial imagery, but high-resolution imagery is collected periodically. Given that the true land cover of a location is not likely to change over short time scales, we augment our training dataset by pairing high-resolution training labels with high-resolution image inputs from different points in time. Specifically, given an image and labels (X,Y) at some point in time, we augment our training set with all pairs (X',Y), where X' ranges over imagery from all time points when it is available. Although this method has the potential to introduce confusion in cases where the high-resolution labels do not match the content of the other images (due to land cover change by construction, flooding, etc.), we demonstrate that it allows models to learn invariance to spurious high-resolution image differences.

Input Color Augmentation (Color) We found that within small geographical regions, individual pixel color is a very predictive feature for land cover classification, whereas across geographical locations, color is very inconsistent for each class. As a result, models trained on limited geographical locations overfit on color. Thus, we choose to add random color augmentation to input images. Given a training image, we randomly adjust the brightness and contrast per channel by up to 5%. Specifically, given a single channel of an image, $X_c \in \mathbb{R}^{h \times w}$, and the mean pixel intensity for that channel, $\overline{X}_c$, we sample $t,b \in \mathbb{U}(0.95,1.05)$, as the contrast and brightness adjustments, then compute the transformed image as $X'_{ijc} = t(X_{ijc} - \overline{X}_c) + b\overline{X}_c$.

Super-Resolution Loss (SR): We augment the training set with additional *low-resolution* labels from outside of the spatial extent in which we have *high-resolution* training data to better inform the model. We incorporate pairs of high-resolution imagery and low-resolution *accessory labels* corresponding to the same spatial extent as the imagery, but where low-resolution labels are assigned to larger (e.g., 30×30m) *blocks* of the image. We assume each accessory label class c determines a (known) distribution over frequencies of each high-resolution label, ℓ. We then use a variant of the super-resolution loss function of [20], which encourages the model to match its high-resolution predictions to the fixed distributions given by the low-resolution

labels while favoring high certainty of predictions.

Specifically, we assume each low-resolution label c determines a distribution $p_{mean}(\ell|c)$ over the frequency of labels of high-resolution class ℓ in a block labeled c, with mean $\mu_{c,l}$ and variance $\sigma^2_{c,l}$. These parameters are computed on a small subset of labeled data where both kinds of labels are available. Alternatively, they could be manually set. We view the probabilistic output of the core segmentation model, p_{net}, as generating labels independently at each high-resolution pixel, inducing a corresponding distribution $p_{\text{out}}(\ell|c)$ over label counts in each block. We then minimize the super-resolution loss, $\text{KL}(p_{net}\|p_{mean})$, over all blocks in the input image.

We incorporate this metric into the overall loss function by minimizing a weighted sum of the standard high-resolution loss (categorical cross-entropy) and the super-resolution loss:

$$J(Y_i,\hat{Y}_i) = \gamma(\text{HR loss}) + \eta(\text{SR loss}). \quad (1)$$

In offline experiments we have found that a ratio of $\gamma:\eta = 200:1$ balances the two losses effectively. We use this setting in all experiments in this work.

3. Data

Imagery data sources: High-resolution (1m) aerial imagery from the USDA National Agriculture Imagery Program (NAIP), and low-resolution (30m) multispectral satellite imagery from the USGS's Landsat 8 satellite.

Label data sources: High-resolution (1m) land cover labels from the Chesapeake Conservancy [5], based on imagery from the years 2013-2014, and low-resolution (30m) land cover labels from the 2011 National Land Cover Database (NLCD) [12].

Figure 1 shows aligned example images from each of these data sources. Combined, these datasets are $\sim$165TB on disk.

We use NAIP data from 2011 to 2016, which provides 2 to 3 layers of high-resolution imagery for each location in US. This allows us to implement the **Label Overloading** method by pairing our high-resolution labels with multiple years of NAIP imagery. We implement the **Low Resolution Input Augmentation** method by creating two sets of Landsat 8 Tier 1 surface reflectance products: a median of non-cloudy pixels from 2013 to 2017 over the April-September months (leaf-on) and a similar product over the October-March months (leaf-off). These layers are both resampled to the 1m-resolution grid used in the NAIP data.

The high-resolution land cover labels from the Chesapeake Conservancy consist of 4 land cover classes – water, forest, field, and impervious surfaces – for the Chesapeake Bay watershed, outlined in black in Figure 4. The low-resolution NLCD labels are from the 2011 data product and consist of 16 land cover classes covering the contiguous US. We use these low-resolution labels as additional training supervision with the **Super-Resolution** data fusion method. Each label at the 30m resolution suggests a distribution of high-resolution labels: e.g., an NLCD label "Developed, Medium Indensity" suggests on average 14% of the block is forest and 63% of the block is impervious surface. See Section 2 in the SI for more details about these correlations.

We use an additional set of high-resolution land cover labeled data in the case study in Iowa (Sec. 5.2), derived from multiple dates of aerial imagery and LiDAR elevation data, as a held out test set [13]. We map the 15 land cover classes in this Iowa dataset to the same 4 Chesapeake land cover classes that our model is trained on according to the Iowa class specifications.

As expected, the distribution of NLCD low-resolution classes and their appearance varies between states (see Figure 2 for class distributions and color histograms). In addition, there is not a standardized national method for collecting NAIP imagery: it is collected on a 3-year cycle by different contractor companies, with collection years differing between states (see Figure 1). These sources of variability in the NAIP imagery must be accounted for in order to build models that will generalize over the entire US using only *high-resolution* training data from the Chesapeake Bay region, motivating our study.

4. Experiments

4.1. Neural Network Models

We consider three network architectures: **FC-DenseNet**, **U-Net**, and **U-Net Large**. Each of these architectures contains the basic structure of four down-sampling and four up-sampling layers. For down-sampling, we use a simple 2×2 max-pooling. For up-sampling, we use deconvolution (transposed convolution) with fixed interpolation, which is useful for reducing checkerboard artifacts [21]. The U-Net models [25] contain three convolutional layers between successive down/up-sampling modules, with batch normalization after each convolution operation and before a ReLU activation function. The FC-DenseNet model [14] instead contains "dense blocks" made up of three convolutional-batchnorm-ReLU layers. The FC-DenseNet model uses 32 filters in a convolution layer immediately after the input and 16 filters in all other convolutional layers. The U-Net model contains 64 3×3 filters in the first three convolutional layers and 32 3×3 filters in all other convolutional layers. The U-Net Large model contains 32 3×3 filters in the first three layers and double the number of filters after each pooling layer, except in the representational bottleneck layer that uses 128 filters.

For training, we use the largest minibatch that will fit in GPU memory and the RMSProp optimizer with a learning rate schedule starting at 0.001 with a factor of 10 reduction every 6000 mini-batches. We use the Python CNTK library for implementation [26].

		North Chesapeake Test Set		Iowa Test Set	
Training Set	**Models**	**Accuracy**	**Jaccard**	**Accuracy**	**Jaccard**
Maryland	RF	37.11%	15.60%	74.95%	31.47%
	FC-DenseNet	71.05%	44.92%	77.87%	41.01%
	U-Net Large	78.06%	50.50%	82.31%	47.06%
	U-Net	61.19%	39.62%	79.07%	47.28%
	U-Net + Adapt	63.33%	42.55%	79.69%	44.10%
South Chesapeake	RF	41.16%	17.96%	72.33%	30.52%
	FC-DenseNet	72.46%	47.83%	74.07%	38.34%
	U-Net Large	72.38%	46.51%	61.56%	37.44%
	U-Net	59.42%	40.47%	71.00%	40.93%
	U-Net + Adapt	62.88%	41.60%	62.95%	39.28%

Table 1: Models that are trained solely on high-resolution labels generalize poorly, regardless of the choice of architecture, training, and testing sets. Compared to the results in Tab. 2, we see that almost all models without multi-resolution data fusion perform worse than any of the models with multi-resolution data fusion.

			North Chesapeake Test Set		Iowa Test Set	
Training Set	**Data Fusion Methods**	**Models**	**Accuracy**	**Jaccard**	**Accuracy**	**Jaccard**
Maryland	LR + Color	RF	64.37%	47.27%	83.03%	49.86%
	LR + Color + LO	RF	75.06%	54.57%	81.94%	49.90%
	SR	U-Net	84.72%	57.72%	80.91%	40.45%
	SR + Color	U-Net	85.11%	59.16%	86.50%	45.03%
	SR + LR + Color	U-Net	88.45%	70.90%	90.95%	62.17%
	SR + LR + Color + LO	U-Net	89.52%	74.11%	92.36%	68.91%
	SR + LR + Color + LO	FC-DenseNet	89.74%	74.30%	91.81%	68.81%
	SR + LR + Color + LO	U-Net Large	**90.31**%	75.41%	**92.93**%	70.66%
South Chesapeake	LR + Color	RF	67.15%	49.08%	88.90%	54.60%
	LR + Color + LO	RF	77.57%	53.86%	83.86%	52.89%
	SR	U-Net	86.85%	62.49%	77.83%	42.03%
	SR + Color	U-Net	87.11%	63.34%	79.71%	42.68%
	SR + LR + Color	U-Net	89.13%	72.83%	93.07%	67.66%
	SR + LR + Color + LO	U-Net	90.61%	76.29%	93.06%	71.12%
	SR + LR + Color + LO	FC-DenseNet	90.52%	76.16%	93.28%	71.17%
	SR + LR + Color + LO	U-Net Large	**90.68**%	76.60%	**93.35**%	71.32%

Table 2: We show the effect of our data fusion methods. (1). Regardless of the choice of models (RF, U-net), the training set (Maryland, South Chesapeake), and the testing set (North Chesapeake, Iowa), adding data fusion methods significantly improved the results. (2). Increasing model capacity, only provides diminishing accuracy and Jaccard returns. The U-Net Large model only performs slightly better than the U-Net model. (3). Our best performing models are able to generalize excellently to Iowa, with an accuracy of 93.35% and Jaccard score of 71.32%.

4.2. Baseline Methods

Random forests (RF) have been used extensively in previous literature for low-resolution land cover classification [9, 3], usually with Landsat imagery, and recently for high-resolution land cover classification [11, 15]. The RF results in the high-resolution setting are promising in areas for which there are high-resolution labels, however show problems generalizing to new geographies [11]. We therefore train a baseline Random Forest model (**RF**) to predict the land cover class of a single pixel from raw pixel values of that pixel and the surrounding pixels within a given radius (in the L_∞ metric). Our offline experiments show that increasing this *feature radius* hyperparameter improves model performance slightly when no augmentation techniques are used, but does not increase performance with Low Resolution data augmentation is used. The RF model we use has a

feature radius of 1, is created with 100 trees, and uses the default parameters from the Python scikit-learn library [22] otherwise.

To improve the generalization ability of supervised models in an unsupervised fashion, domain adaptation methods [8, 18, 29, 32, 27] learn to map inputs from different domains into a unified space, such that the classification/segmentation network is able to generalize better across domains. We use an existing domain-adversarial training method [8] for the land cover mapping task (**Adapt**). In particular, we attach a 3-layer domain classification sub-network to our proposed U-Net architecture. This subnetwork takes the output of the final up-sampling layer in our U-Net model and classifies the source state (New York, Maryland, etc.) of the input image as its "domain". In addition to minimizing segmentation errors on limited image domains, we also train the segmentation network to *maximize* the error of the classification sub-network. In this way, the segmentation network learns to generate more domain-invariant features.

4.3. Model Training and Evaluation

We train all models on two sets: the state of **Maryland** and its superset, the lower half the Chesapeake Bay region (**South Chesapeake**). We test on a set consisting of the upper half of the Chesapeake Bay region (**North Chesapeake**) as well as held out land cover data from Iowa (**Iowa**). In training, we uniformly sample $\sim$100,000 240$\times$240 pixel patches with high-resolution land cover labels from the training set. If Adapt or Super Resolution is used, we sample an additional $\sim$150,000 240$\times$240 patches from across the US. In the Adapt case, these additional samples are without labels, while in the Super Resolution case, we include their low-resolution NLCD labels. For a given set of tile predictions, we compute the accuracy and average Jaccard index (i.e. intersection-over-union) over the four high-resolution classes.

The relationship between training on data from a single state and testing on the held out North Chesapeake set mimics the relationship between the entire Chesapeake Bay region and the rest of the US. Maryland data is restricted both geographically (i.e., models trained there will not be able to observe features found in other parts of the Chesapeake Bay) and in observed NAIP sensor variance (i.e., all the imagery in Maryland from a given year will be collected in the same manner). A similar relationship will hold between the Chesapeake Bay region and the remainder of the US, e.g., it is impossible to observe deserts in the Chesapeake Bay, and there will be NAIP imagery conditions that are unobserved in the Chesapeake Bay region, but present in other parts of the country.

Training on South Chesapeake exposes models to more of the variation that is likely to be present in North Chesapeake, thus making the generalization easier than that of Chesapeake to the whole US. Indeed, the NLCD class composition of **South Chesapeake** is similar to that of **North Chesapeake**, but not similar to the remainder of the US.

5. Results

5.1. Model Generalization

The results in Table 1 show the performance of our models when trained solely on high-resolution labeled data, i.e., without our multi-resolution data fusion methods. These results show that the models are not generalizing well: adding more training data (South Chesapeake vs. Maryland training sets) results in poorer performance on the Iowa test set. The models that are trained in Maryland have Jaccard scores of less than 50% on the Iowa test set, but relatively high accuracies, which suggests that they are biased towards predicting the majority class (overwhelmingly "field" in Iowa). The benefits of using higher-capacity models, like the U-Net Large, or more complex models, like the FC-DenseNet, are not expressed in this land cover mapping problem. Lastly, of note, the domain-adaptation method we use does not give a significant increase in model performance.

Table 2, however, shows how the progressive addition of our data fusion methods improves model performance. More specifically, for models trained in **Maryland**, each data fusion method increases the performance in both the North Chesapeake set *and* in the held out Iowa set in terms of accuracy and Jaccard scores. For models trained on South Chesapeake, the benefits of **LO** are not as prevalent as with the restricted Maryland training subset. In this case, the South Chesapeake set must contain additional features that are not present in the Maryland set before Label Overloading is used. Of note, increasing model capacity provides diminishing accuracy and Jaccard returns. The U-Net Large model only performs slightly better than the U-Net model. Our best-performing models are able to generalize excellently to Iowa, with an accuracy of 93.35% and a Jaccard score of 71.32%.

In addition to the quantitative model results, we visualize the land cover output from several of our models over a set of hand-picked scenes from locations outside of the Chesapeake Bay in Figure 3. We choose these locations to capture potential failure cases and to display model behaviour in interesting settings. In most locations, our best model (last row) correctly identifies features mislabeled by the **RF** baseline and other versions of the model trained without data fusion methods. In the last column, an image from Tucson, Arizona, we observe that the two baseline models, without data augmentation, are not able to identify a collection of houses in an ambient desert. Our best-performing model in the last row is able to correctly identify the houses, but does not identify the road.

5.2. Middle Cedar Watershed Case Study

Our partners at the Chesapeake Conservancy are working with the Iowa Agricultural Water Alliance (IAWA) to pilot new techniques to facilitate watershed management planning throughout the state of Iowa. High-resolution land cover data is important in this setting to rapidly identify specific recommendations for how to improve land management and

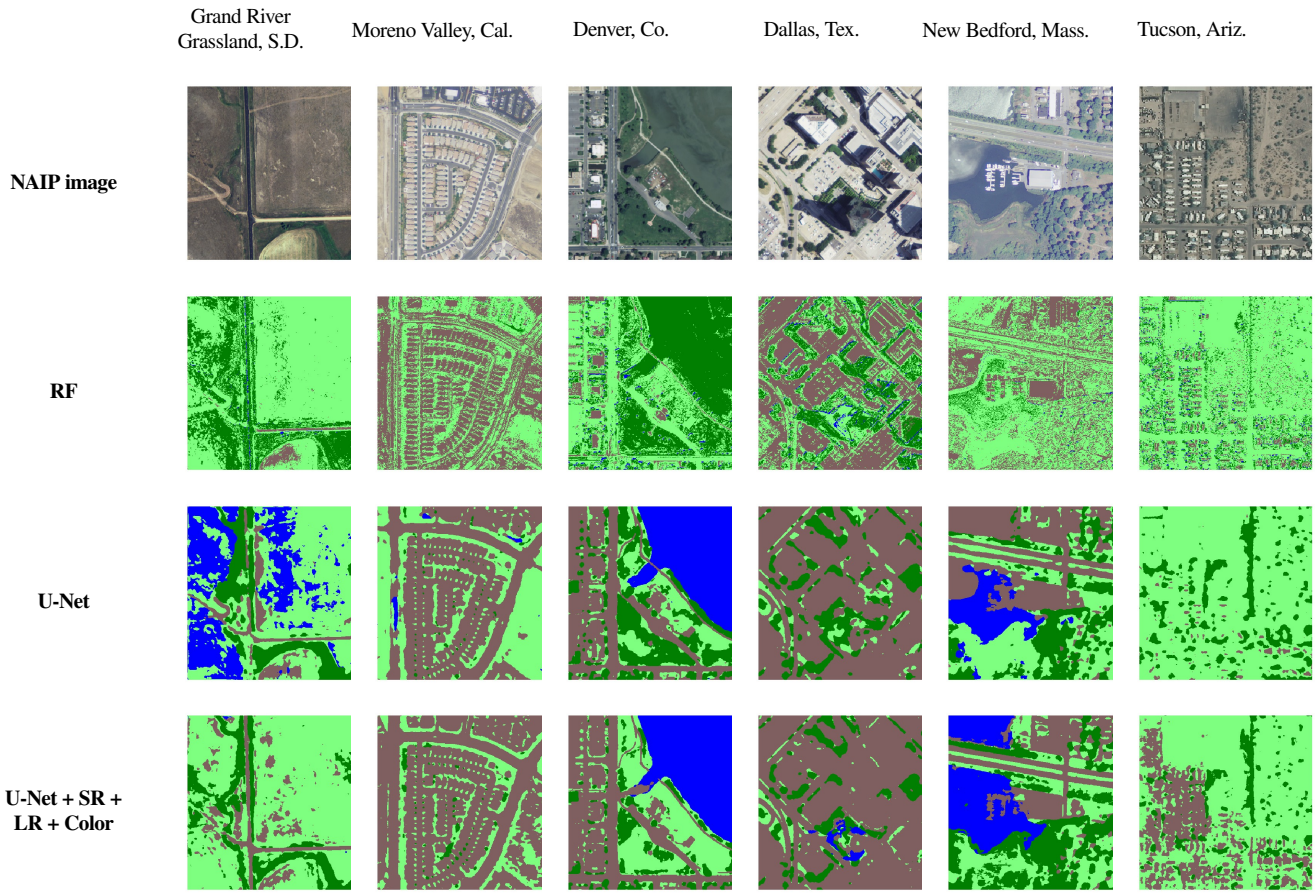

Figure 3: High-resolution land cover predictions, and accompanying NAIP imagery, for different models in choice locations where ground truth labels are not available. Here, the color map is the same as in the high-resolution ground truth image from Figure 1.

water quality while minimizing the impact to farm operations.

Thus, we ran an early version of our model over the entire area of Middle Cedar watershed in Iowa, an area of 6260km^2, and gave the results to our partners. The partners used their quality assurance (QA) methodology to correct the model's systematic errors over a geography that was $\sim 1.1\%$ of the area of the total watershed. This methodology involves comparing the model output with NAIP imagery, a Normalized Difference Vegetation Index (NDVI) layer, and a Normalized Difference Surface Model (nDSM) layer to identify classification errors. The NDVI and nDSM layers help to identify misclassifications in vegetation and mistakes that can be captured with height differences (e.g. low vegetation misclassified as trees) respectively. The first round of this process resulted in corrections of three broad classes of errors: incorrect prediction of the "field" class bordering roads, rounded building corners, and water values predicted in shadows. The corrections represented $\sim 2\%$ of the pixels in the evaluated geography and cost 30 hours to perform. Using this feedback, we tuned our model's per-pixel class probabilities with a global transformation to best fit the corrections and generated a new map over the entire watershed using this transformation.

Formally, we are given n corrected samples from our set of model predictions. We sample another n pixels that were not corrected in the QA process in order to balance the dataset, then form a matrix $\mathbf{X} \in \mathbb{R}^{2n \times 4}$, of our model's probabilistic output, and vector, $\mathbf{y} \in \mathbb{R}^{2n \times 4}$, of the accompanying labels (one-hot encoded). We find a transformation $\mathbf{W} \in \mathbb{R}^{4 \times 4}, \mathbf{b} \in \mathbb{R}^4$, such that $\mathbf{XW} + \mathbf{b} = \hat{\mathbf{y}}$ minimizes the categorical cross-entropy with $\mathbf{y}$. The learned transformation, $\mathbf{W}$ and $\mathbf{b}$, can now easily be applied across any number of pixels. This method is able to correct 89.7% of the errors made by the original model, and, under the assumption that our model is making the same systematic errors across the whole testing region, is able to save the $\sim$2700 hours of manual labor that would be required to correct the entire area. In Figure 5 of the SI we display the progression of this feedback process for a small patch of land in a corrected area. This method is a cheap way to incorporate domain expert feedback to

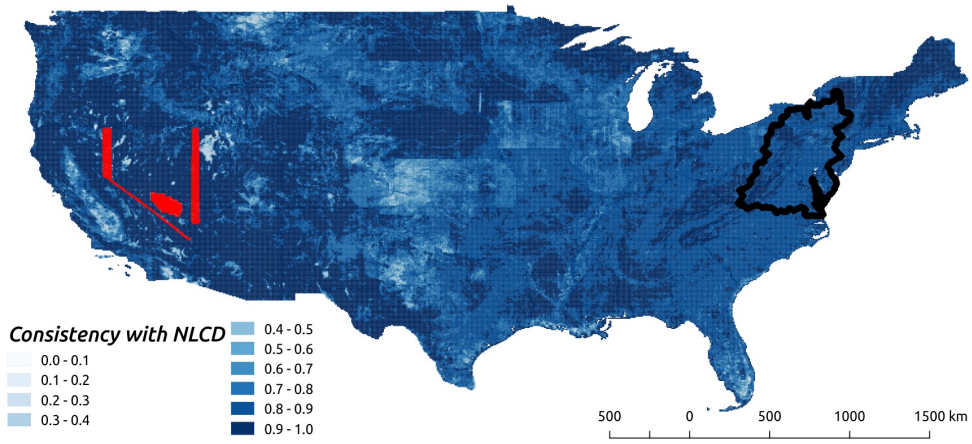

Figure 4: Maps showing the high-resolution *consistency with NLCD* over the entire US. Lower values represent an 'inconsistency' between our model estimates and the expected high resolution labels (using high-resolution label distributions per NLCD class from the Chesapeake Bay area). Areas for which there is no input data, or data errors prevented our model from running are shown in red.

a model's existing predictions, and can further be embedded in a model generation loop, where new versions of the original model are fine-tuned with samples from the broader corrected area, and updated predictions are looped back into the QA process.

5.3. US-Wide Land Cover Map

We used the approach of our best model – including all data fusion methods – to generate a full-US land cover map. For training data, we used high-resolution labels from the entire Chesapeake Bay region and low-resolution NLCD labels sampled over the entire US. The correlations between NLCD and high-resolution labels, $\mu_{n,c}$, for the Super Resolution loss were manually tuned for this model, rather than estimated from data in the state of Maryland (as in our experiments)[1].

Cost: The size and complexity of the network used to create the full-US land cover map will largely determine the cost of the operation. For example, the Dense Fusion Classmate network that won the DeepGlobe land cover mapping competition requires 8 GPUs to train and would be prohibitively costly for full-US inference [28]. The FC-DenseNet103 architecture [14], on which the Dense Fusion Classmate network is based, can fit on a single GPU but will incur an $\sim 270\%$ increase in cost over our U-Net Large model when run over the entire US. Our full-US map was generated with the U-Net Large architecture, which only has a 19% cost increase over the U-Net and FC-DenseNet models.

Evaluation: In Section 5 we discuss a "benchmark" visualization set of patches that we use to inspect a model's performance on important terrain features, and in the SI we show a web application to interactively explore our models' predictions. However, these are not sufficient for discovering all cases where our model is performing poorly. It is prohibitively time-consuming to qualitatively evaluate the performance of our model by simply sampling patches of model input vs. predicted output. Considering this, we use the low-resolution labels to approximate the performance of our model across the entire US by computing a metric we call *consistency (of high-res labels) with NLCD*. First, we compute the high-resolution class distribution for each NLCD label, $p_{mean}(y|n) = \mu_{n,y}$, as described in the **SR** data fusion method. We let $\rho_{n,y} = \mu_{n,y} / \max_{y'} \mu_{n,y'}$, normalizing the high-resolution class means for each NLCD label by the maximum value in that distribution. Now, given a set of N high-resolution predictions, $\{y_1, \ldots, y_N\}$, and the associated NLCD labels, $\{c_1, \ldots, c_N\}$, we compute the *consistency with NLCD* value, $\lambda = \frac{1}{N} \sum_{i=1}^{N} \rho_{c_i,y_i}$. This definition can be thought of as a charitable "accuracy" score for a given set of predictions[2]. In general, the aim of this metric is to identify potential problem areas in our US-wide evaluation – areas in which the high-resolution labels *do not* have *consistency with NLCD*. Finally, we show the approximate accuracy map for this model run in Figure 4, with an average *consistency with NLCD* of 87.03%.

Acknowledgements

The authors thank Lucas Joppa and the Microsoft AI for Earth initiative for their support and the reviewers for their helpful comments on an earlier version of the paper. C.R. was partially supported by the NSF grant CCF-1522054 (COMPUSTNET: Expanding Horizons of Computational Sustainability).

[1] The input to the model we trained for this purpose has a small difference compared to the best model reported in Table 2: we used an median of all available Landsat 8 imagery, not separating leaf-on and leaf-off months.

[2] As an alternative, we could define a deterministic mapping from NLCD labels to high-resolution labels and directly compute an "accuracy" surrogate, but this will heavily penalize predictions in areas where multiple high-resolution classes may occur with high frequency, such as cities. We expand on and show results for this definition in Section 3 of the SI.

References

[1] ISPRS 2D semantic labeling dataset. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html.

[2] Dstl satellite imagery feature detection, 2017. [Online].

[3] M. Belgiu and L. Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[4] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva. Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*, 2015.

[5] Chesapeake Bay Conservancy. Land cover data project, January 2017. [Online].

[6] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[7] R. Gaetano, D. Ienco, K. Ose, and R. Cresson. A two-branch cnn architecture for land cover classification of pan and ms imagery. *Remote Sensing*, 10(11):1746, 2018.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[9] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.

[10] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. Goetz, T. Loveland, et al. High-resolution global maps of 21st-century forest cover change. *science*, 342(6160):850–853, 2013.

[11] M. M. Hayes, S. N. Miller, and M. A. Murphy. High-resolution landcover classification using random forest. *Remote sensing letters*, 5(2):112–121, 2014.

[12] C. Homer, J. Dewitz, L. Yang, S. Jin, P. Danielson, G. Xian, J. Coulston, N. Herold, J. Wickham, and K. Megown. Completion of the 2011 national land cover database for the conterminous united states–representing a decade of land cover change information. *Photogrammetric Engineering & Remote Sensing*, 81(5):345–354, 2015.

[13] Iowa DNR. Iowa 2009 land cover data, 2009. [Online].

[14] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

[15] T. Kavzoglu. Object-oriented random forest for high resolution land cover mapping using quickbird-2 imagery. In *Handbook of neural computation*, pages 607–619. Elsevier, 2017.

[16] X. Li, S. W. Myint, Y. Zhang, C. Galletti, X. Zhang, and B. L. Turner II. Object-based land-cover classification for metropolitan phoenix, arizona, using aerial photography. *International Journal of Applied Earth Observation and Geoinformation*, 33:321–330, 2014.

[17] X. Li and G. Shao. Object-based land-cover mapping with high resolution aerial photography at a county scale in midwestern usa. *Remote Sensing*, 6(11):11372–11390, 2014.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[19] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing*, 10(7):1119, 2018.

[20] K. Malkin, C. Robinson, L. Hou, R. Soobitsky, J. Czawlytko, D. Samaras, J. Saltz, L. Joppa, and N. Jojic. Label super-resolution networks. In *International Conference on Learning Representations*, 2019.

[21] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] B. Pengra, J. Long, D. Dahal, S. V. Stehman, and T. R. Loveland. A global reference database from very high resolution commercial satellite data and methodology for application to landsat derived 30m continuous field tree cover data. *Remote Sensing of Environment*, 165:234–248, 2015.

[24] A. Rakhlin, O. Neuromation, A. Davydow, and S. Nikolenko. Land cover classification from satellite imagery with u-net and lovász-softmax loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 262–266, 2018.

[25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] F. Seide and A. Agarwal. CNTK: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2135–2135. ACM, 2016.

[27] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016.

[28] C. Tian, C. Li, and J. Shi. Dense fusion classmate network for land cover classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 192–196, 2018.

[29] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[30] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279. ACM, 2010.

[31] H. Zhang, Z.-f. Qi, X.-y. Ye, Y.-b. Cai, W.-c. Ma, and M.-n. Chen. Analysis of land use/land cover change, population shift, and their effects on spatiotemporal patterns of urban heat islands in metropolitan shanghai, china. *Applied Geography*, 44:121–133, 2013.

[32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.