Final report: Land Cover Change Detection using Neural Network for Satellite Images

Ashkan Bozorgzad (ab5243), Hari Prasad Renganathan (hr2514), Karveandhan Palanisamy (kp2941),
Masataka Koga (mk4528), Yewen Zhou (yz4175), Yuki Ikeda (yi2220)

12/17/2022

Problem Definition and Progress Overview

This project is sponsored by JPMorgan Chase, an investment banking company. The goal of the project is to create high-resolution (1m / pixel) land cover change maps of a study area, the state of Maryland, USA, given multi-resolution imagery and label data. This project aims to provide an example of situations commonly found worldwide. In the field of earth observation, new images produce faster than high-quality, high-resolution labels. However, old and low-resolution labels are available, for example, 30m National Land Cover Database (NLCD) in the United States or 500 m MODIS land cover available worldwide. Therefore, it is significant to investigate how machine learning can be used to build a model that predicts high-resolution change without having a lot of higher-resolution change data.

Since the last progressive report, the following tasks have been conducted:

- Labeling high-resolution test images
- Study the bottlenecks of the base model and try to find ways to improve it
- Study previous studies and investigate the potential appropriate models for our dataset
- Add innovations to the previous models and try to improve the results
- Using new data (Dynamic World label) to compare its performance with NLCD labels

In the following sections the above task will be explaned in more details.

Labeling high-resolution test images

As mentioned in the first progressive report, we initially considered 50 NAIP aligned tiles (1m / pixel, high resolution) selected from the 2250 training tiles for the test data. However, their high-resolution land cover label is not available. We classified all pixels of test images into the four target classes to create high-resolution labeling for the images to evaluate models using manual and semi-automatic tools in GroundWork. This task is so time-consuming that we could not label all test data. So, we decided to use DFC 2021 contest web page instead for a main evaluation source as written in the next section. It should be mentioned since our model is supposed to compare the land cover changes between two different years (2013 and 2017) for each tile, a pair of images (2013 and 2017) should be labeled. We kept labeling the test images; as of today, we have four pairs of high-resolution labeled images. We developed a code to convert a geojson file of a manual label on GroundWork into numpy array. This code is mathematically advanced since it converts a number of manually drawn multi-polygons into an image.

Evaluating loss/gain IoUs on DFC 2021 contest web page on Codalab

Since we are not professionals for manual labeling, our manual labels may not be reliable. Hence, we decided to use a copy environment of DFC 2021 contest web page on Codalab as a main source of evaluation which enables us to evaluate our model performance in terms of loss/gain IoUs in exactly the same way as the contest, The inputs to Codalab are model predictions for selected 57 couples of NAIP images for 2013 and 2017, and the outputs are loss/gain IoUs from 2013 to 2017 for some unknown (to contest participants) parts of the 57 couples of images. We appreciate Professor Yokoya who is one of the contest organizer and kindly created the copy encironment for us.

Bottlenecks of the base model

For the base model, we used two different models, NLCD diff and a single-layer convolution neural network on the whole dataset. Table 1 shows the result of the models:

Table 1 IoU scores for our baseline models. −C and +C denote the loss and gain of class C, respectively.

Algorithm	-W	-TC	-LV	-I	+W	+TC	+LV	+I	Avg.
NLCD diff	0.000	0.106	0.002	0.000	0.000	0.000	0.092	0.003	0.025
1-layer FCN	0.022	0.063	0.080	0.032	0.010	0.085	0.064	0.034	0.049

The higher IoU score is better and as can be seen, the average IoU score is 0.025 and 0.049 for NLCD diff and 1-layer DCN respectively. As can be seen in the table the lowest IoU belongs to the water category for both models. In other words, the model did a poor job of identifying water changes. Therefore, we need to find a way to improve water classification and changes. The second lowest IoU score is I changes.

Another intuitive restriction of the FCN is that we only use one layer consisting of one 3x3 filter size. This single filter can not capture all the features and paters that exist in the depth of data. In the proposed model these problems should be addressed.

Literature review of previous models

Table 2 displays the models, inputs, and outputs that have been used for land cover change labeling. 4 studies were used to create this table. These 4 studies we used since they generated better results in compared to the other studies.

Table 2 Previous studies summary models for land cover change labeling

Paper Name	**Models**							
[2]	Phase 1			Phase 2			Phase 3	
	Model	Input	Output	Model	Input	Output	Model	Input
	Siamese Skip_FCN (LR) (13&17)	Landsat-8	Landsat-LRL	fusion models (separate years) (HRNet, Deeplabv3+ and Skip_FCN)	NAIP	The arithmetical average assignment is used to integrate their outputs and intersection operation is implemented to maintain the common high confidence parts in bitemporal predictions	Shadow-removal, NDVI restriction and morphological process are implemented as post-processing steps.	intensity channel of Hue-Saturation-Intensity (HSI) color model
		NLCD with both low-resolution						
	Siamese Skip_FCNs (HR) (13&17)	NAIP	resolution-improved labels, called "NAIP-HRL"		NAIP-HRL (transfer 15 to 4 labels)			near infrared (NIR) channel of images
		the output pseudo labels of the first phase (Landsat-LRL)						
[1]	Step1			Step2			Step 3	
	Model	Iput	Output	Model	Iput	Output	Comparing the probability of Steps 1 and 2 assign the class	
	FCN (5 feature extraction layers and 1 classification layer)	NAIP	Pseudo-labels (4 probs for each category)	five FCN models	Pseudo-labels	improve the separability of water		
		NLCD			NAIP, MNDWI from 8-Landsat (New feature)			
[5]	Step1			Step1				
	Model	Iput	Output	Model	Iput	Output		
	FCN (5 feature extraction layers and 1 classification layer)	NAIP	Pseudo-labels (4 probs for each category)	FCN, U-Net, Deeplabv3, LinkNet, PSPNet, PAN, FPN (pixel vote)	NAIP, Pseudo-labels (4 probs for each category)	HR label resulted from both model		
		NLCD		2 class clasifier just to detect water	NLCD + NLCD aug			
[3]	Models	Input	Output					
	FCN tile	NAIP, NLCD	HR Label					
	FCN all							
	-net all							

The following results can be concluded from Table 2 and can be incorporated in the proposed model:

- It seems that multi-phase models resulted in a better result than a single-phase model
- Alsomst all previous studies used 5 layers consisting of 64 three-by-three with relu activation function
- To improve the result of one specific category, some studies used a specific neural network to label that category and then add and combine the result of this model to the main model.
- To improve the result of the main model, extra information from different sources such as landsat data can be used
- For the main model some of the famous pre-trained models such as HRNet, Deeplabv3+ were used

The potential proposal models

According to the bottlenecks of our base models, previous studies, and the timeline of this project we decided to follow these steps to generate a final model with acceptable results:

Step1: in this step, we decided to generate an FCN model and train on the entire dataset. The 5 layers consisting of 64 three-by-three with relu activation function can result in a more accurate model. If we have time this model can be used as phase 1 of the entire 3-phase model we are going to propose. In the 3-phase model, this model will be used as preprocess model to generate higher resolution labels for phase 2 (main model). The model will be trained on the entire dataset (2013 and 2017). If we do not have enough time we can propose this model as a final model[3].

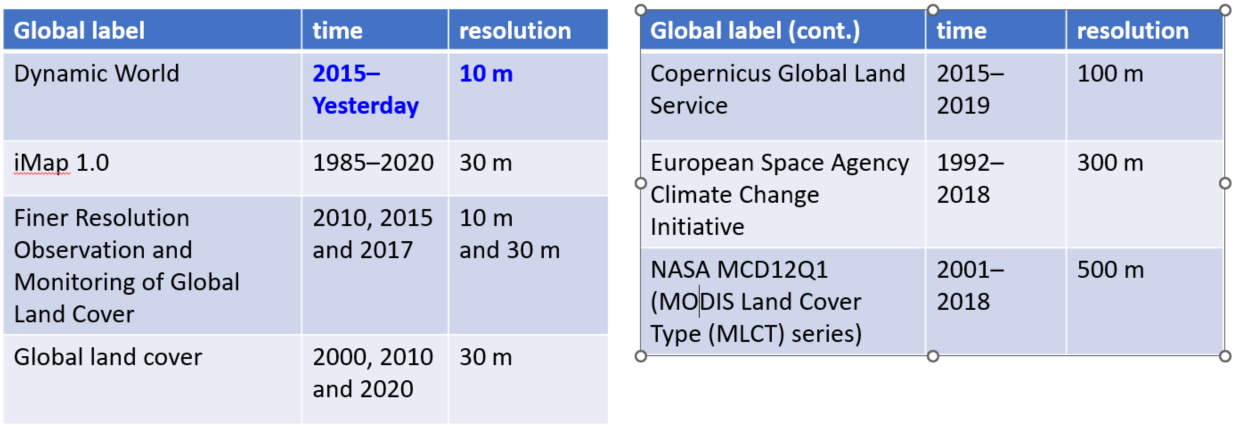
Step2: in this step, we use some pre-trained models and do the hyper-tuning for our dataset. We consider to used MobileNetV2, HRNet, and Deeplabv3+, Inception V3, and compare the result to pick the three of them as the main models. If we have time we can consider this step as the second phase of the 3 phases model and move to the last phase. In this step, two different models will be trained in two different years (2013 and 2017) and two final models will be produced. Otherwise, we can compare the result of this phase with the previous phase and propose the combination of the 3 of them as the final model.

Step3: in this phase, two models will be used for two years to produce two different labels (for each year) and take the average and compare

Two new ideas in the proposed model

Using Dynamic World label

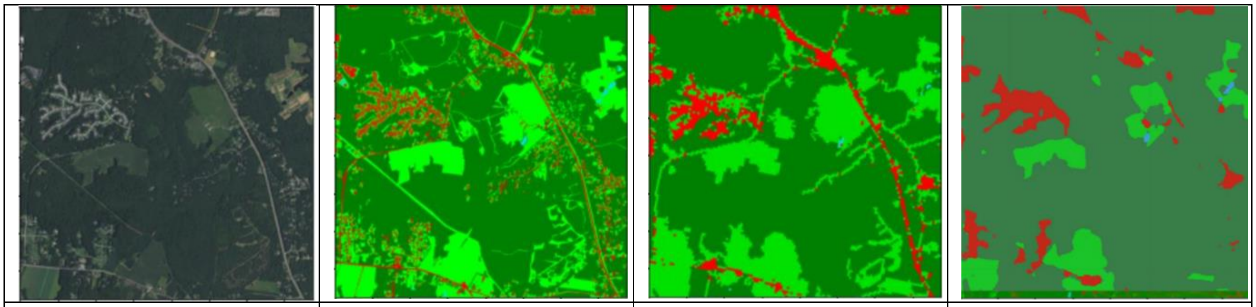
In addition to NLCD labels, we also tried using Dynamic World label (henceforth DW) for a label to train our models. DW is a 10m-resolution land-cover label which is predicted altomatically by deep learning model trained on Sentinel-2 satellite images [8]. For each pixel DW has nine classes: water, trees, grass, flooded vegetation, crops, shrub & scrub, built, bare, snow & ice, and DW has two kinds of bands: label band and probability bands. Label band shows the class with the highest probability over the nine classes, and probability bands show the probability of each class, predicted by the model. Though there are already a number of global land-cover labels, as shown in Table 3, they are limited in time or of coarcer resolution. On the other hand, DW is available globally and near real-time—meaning that being available until today (at least conceptually)—since June 2015 on the Google Earth Engine (GEE) and with finest resolution (10m) among these global labels. Hence, it is of interest to see the predictive performance when using DW as a label for training. As we see later, our model trained on DW has almost the same performance to the one trained on NLCD. Moreover, since there are also local land-cover labels such as NLCD in US and CORINE Land Cover [10] for Europe, we also investigated the performance of the model average of two models trained on NAIP+NLCD and NAIP+DW respectively. It turns out that the model average worked well as we see later.



Global label	time	resolution
Dynamic World	**2015–Yesterday**	**10 m**
iMap 1.0	1985–2020	30 m
Finer Resolution Observation and Monitoring of Global Land Cover	2010, 2015 and 2017	10 m and 30 m
Global land cover	2000, 2010 and 2020	30 m
Global label (cont.)	**time**	**resolution**
Copernicus Global Land Service	2015–2019	100 m
European Space Agency Climate Change Initiative	1992–2018	300 m
NASA MCD12Q1 (MODIS Land Cover Type (MLCT) series)	2001–2018	500 m

Table 3 IoU List of global labels (resolution finer than 1 km) [8], [9]

For this study, we wrote a code to download DW for the state of Maryland. This code extracts the coordinates of the NLCD labels we have and downloads the corresponding DW. As already menthoned, DW is available from 2015, so we used only DW for 2017. Figure 1 shows one sample NAIP image (id: 3716), its manual label by authors, NLCD label, and DW label. Despite of higher resolution (10m) than NLCD (30m), DW looks more rough than NLCD. This tendency is also applicable to other NAIP images, but as we see later, the accuracy of DW is actually no worse than NLCD.



(a) NAIP image (b) Manual label (c) NLCD image (d) DW label

Figure 1 NAIP, manual label, NLCD, and DW label (Note that Tree Canopy is drawn with slightly different dark green color in DW for a technical reason)

Thoughout this study, we integrated the nine classes of DW (mentioed above) into four classes of interest (Water, Tree Canopy, Low Vegetation, and Impervious) as follows: "water" to Water, "trees" and "flooded vegetation" to Tree Canopy, "grass", "crops", "shrub & scrub", and "bare" to Low Vegetation, and built to Impervious. There was no “snow & ice” (in the state of Maryland for the period we specified). We downloaded DW label for each NLCD label (i.e. the same coordinates). In detail, we specified the period as June 23rd - September 29th (i.e. during the summer) in 2017 and downloaded the average probability over the period of each class for each pixel using python API of GEE. We used the probabilities as soft labels (henceforth DW 2017 soft labels). We also prepared hard labels (henceforth DW 2017 hard labels), but we did not use the label band originally prepared on GEE which is the class with the highest probability over all nine classes. Rather we took the argmax over the integrated four classes and made hard labels. Considering that DW is 10m-resolution, to save the data size, we down-sampled DW by 1/10, e.g. size 388 by 388 for a label of size 3880 by 3880 originally. We also rounded the probabilities by multiples of 0.5% and saved as uint8 upon multiplied by 200. When using in training, we transformed them back into normalized probabilities again.

Handle the uncertainty of the model

All previous studies used hard labeling for training models. This causes some uncertainty in classifying some categories and some inaccuracy. To overcome this problem, we plan to train the model and used soft labeling with the probability of each category instead of hard labeling. For the implementation of using soft labeling. To decrease the uncertainty (Decrease the variance and bias) in phase 2 of the main model we used the Ensemble technique. We are going to train multiple models in parallel and take the average of the result as the final result. Another technique we are going to implement is DUQ. In this technique (please explain this technique)

It should be mentioned that our dataset is large and our models have so many parameters. The models are ready for training and hyperparameter tuning. However, we did not have access to the computational resources until November 9th. Therefore, we only train U-Net (all dataset) and 5 layers FCN (all dataset) as follow and present the result.

Results

5 layers FCN

As mentioned in the previous sections, we are going to use FCN for the first phase of the model. We trained 5 layers of FCN on a partial of the dataset due to the total dataset size for this project, around 254 GB, and our computational resource limitations. The architecture of the model is the number of input channels = 4 for (Red, Green, Blue, and Near-Infrared), the number of output classes = 5 (Water, Tree Canopy, Low Vegetation, Impervious, and None), filter size = 3, stride = 1, and padding = 1. The None output class is for the case there are pixels we cannot classify into the four target classes, and we do not evaluate its IoU scores. We added padding to the images to make sure the model trained on the same size images before training because the size of images is different in some images. Since we used cross-entropy as the loss function, the model architecture is the same as the multiclass logistic regression with 36 features (= 4 pixel colors * 9 surrounding pixels) to estimate the label of each pixel. We predicted labels for one pair of our test images (1950, 3313, and 3716, each in 2013 and 2017) and compared them with the high-resolution labels produced by GroudWork. The IoU scores of this model are written in Table 4.

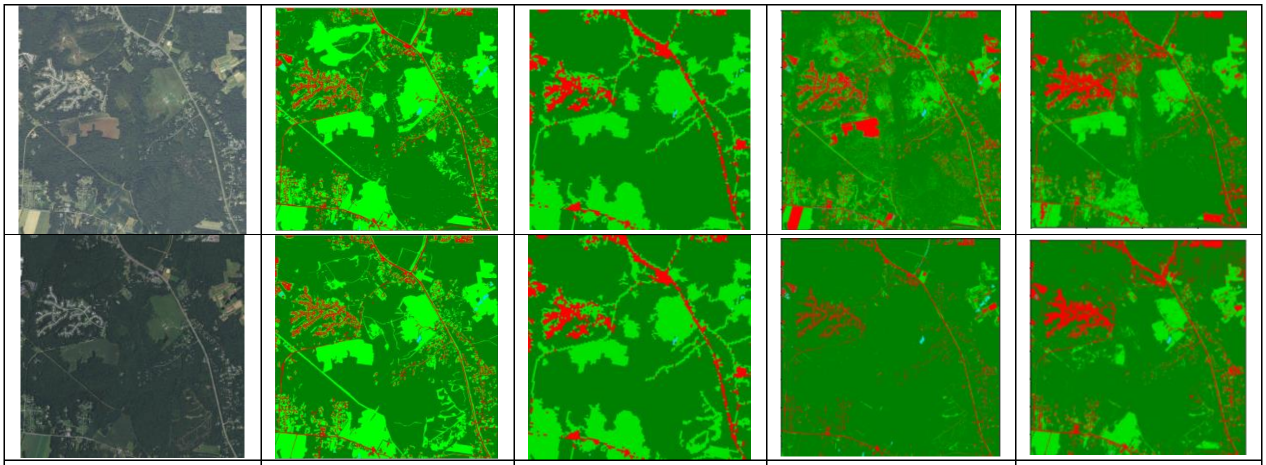
Table 4 IoU scores for our baseline models and 5 layers FCN. −C and +C denote the loss and gain of class C, respectively.

Algorithm	-W	-TC	-LV	-I	+W	+TC	+LV	+I	Avg.
NLCD diff	0.000	0.053	0.047	0.000	0.004	0.000	0.036	0.093	0.029
1-layer FCN	0.016	0.106	0.162	0.129	0.005	0.126	0.122	0.145	0.101
5- layer FCN	0.005	0.106	0.110	0.021	0.039	0.078	0.057	0.103	0.070
U-Net	0.038	0.121	0.145	0.044	0.038	0.100	0.080	0.164	0.091

Table 5 IoU scores evaluated on Test labels used in the MSD contest. −C and +C denote the loss and gain of class C, respectively.

Algorithm	-W	-TC	-LV	-I	+W	+TC	+LV	+I	Avg.
NLCD diff	0.148	0.167	0.282	0.014	0.031	0.001	0.106	0.362	0.139
1-layer FCN	0.393	0.468	0.383	0.037	0.116	0.167	0.200	0.274	0.255
5- layer FCN	0.459	0.582	0.455	0.026	0.070	0.212	0.321	0.408	0.317
U-Net	0.456	0.506	0.522	0.163	0.217	0.332	0.380	0.550	0.391

As can be seen in Table 4, the average IoU of the model improved for 5-layer FCN. F displays the prediction of 5 layers FCN and comparison with base model FCN.

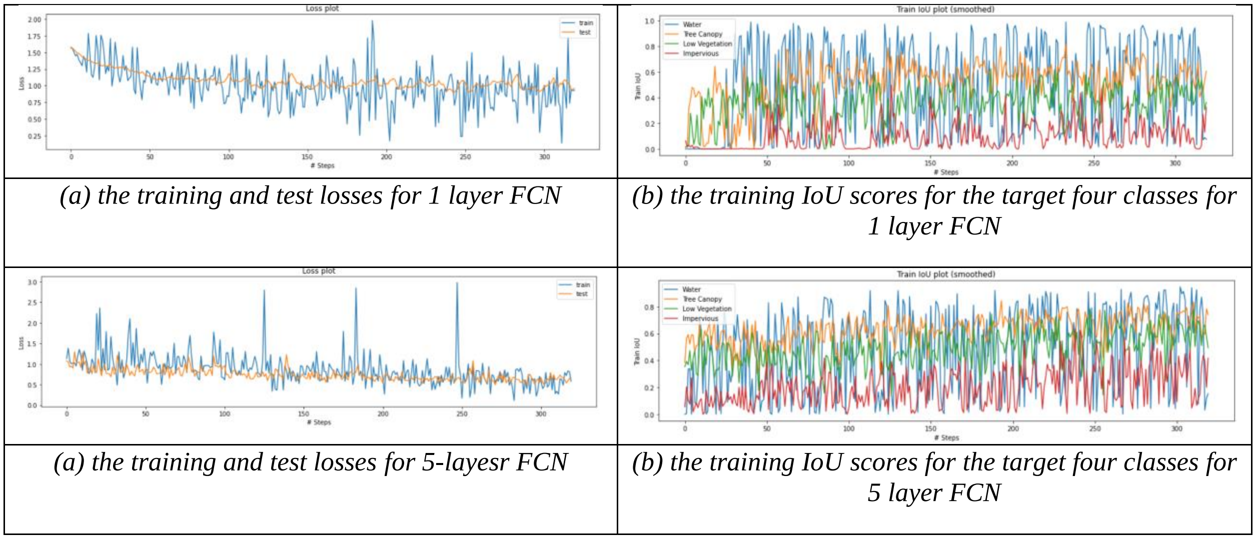


NIAP image | *GroundTruth image* | *(C) NLCD image* | *(d) 1 layer FCN* | *(e) 5 layer FCN*

Figure 2 Comparison between base model prediction(1 layer FCN and 5 layers FCN) and NAIP images, ground truth labels, and NLCD labels

As can be seen in Figure 2, the 5 layers FCN improved the prediction. However, we can improve the prediction by training this model on the entire dataset rather than just a partition of the training dataset.

Figure 3 shows the training and test losses and the training IoU scores for the target four classes on each step, not for the gains and losses of the four classes.



(a) the training and test losses for 1 layer FCN

(b) the training IoU scores for the target four classes for 1 layer FCN

(a) the training and test losses for 5-layesr FCN

(b) the training IoU scores for the target four classes for 5 layer FCN

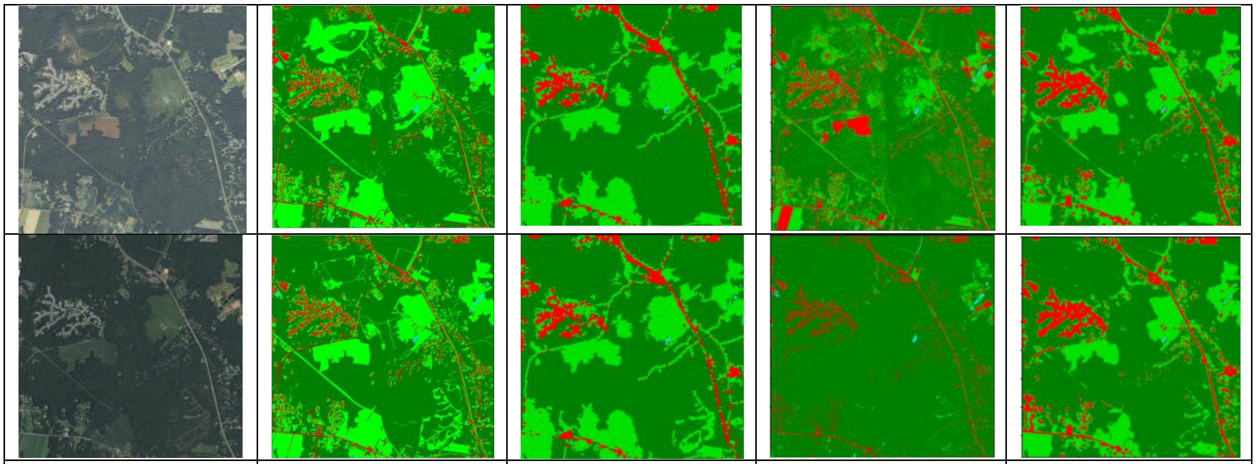
Figure 3 The training loss and training IoU scores of the 5-layer FCN and 1-layer FCN baseline model on each step

As can be seen in this figure, with this model, we decrease the loss and IoU and improve the results.

U-Net (all datasets)

This method is similar to FCN / all. However, we will use a U-Net family architecture. U-Net is an FCN for image semantic segmentation and consists of encoder and decoder parts connected with skip connections. The encoder part consists of convolution and maxpooling layers. It learns to classify the object by downsampling the input image into feature representations. The decoder part replaces convolutional layers with upsampling and concatenation layers and learns to semantically increases the resolution of the previous outputs. Developed originally for biomedical image sementation tasks, it is now widely adopted for being more precise and requiring less training data than vanilla FCN models. In our case…

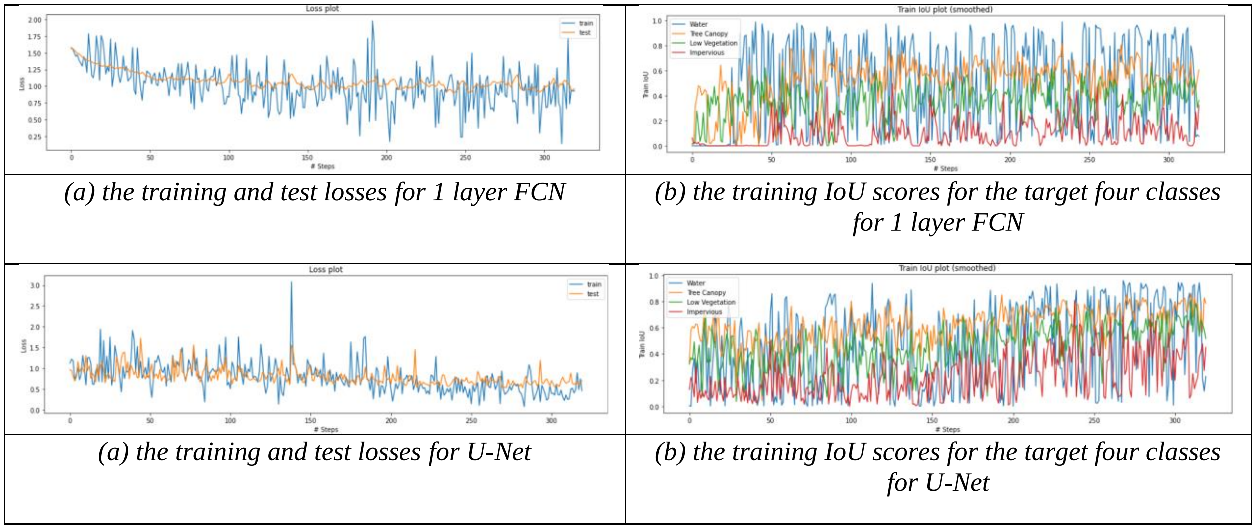
Figure 3 shows the average result of IoU for this model. As can be seen this model……



NIAP image	*GroundTruth image*	*(C) NLCD image*	*(d) 1 layer FCN*	*(e) U-net*

Figure 4 Comparison between base model prediction(1 layer FCN and U-net) and NAIP images, ground truth labels, and NLCD labels

Need to explain Figures 4 and 5 after having data according to the result.



(a) the training and test losses for 1 layer FCN

(b) the training IoU scores for the target four classes for 1 layer FCN

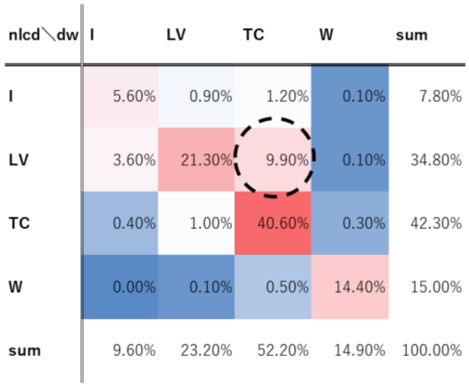
(a) the training and test losses for U-Net

(b) the training IoU scores for the target four classes for U-Net

Figure 5 The training loss and training IoU scores of the U-Net and 1-layer FCN baseline model on each step

Using Dynamic World labels

We first developed an explarotary data analysis on the nature of DW. Table 6 shows a comparison of NLCD 2016 and DW 2017 over the total 2250 NAIP 2017 images for the state of Maryland. One sees that they are identical in about 80% of all pixels. The major difference is 9.9% of all pixels where NLCD is Low Vegetation and DW is Tree Canopy and 3.6% of all pixels where NLCD is Low Vegetation and DW is Impervious. So, in general, NLCD tends to say Low Vegetation while DW tends to say Tree Canopy or Impervious.



nlcd╲dw	I	LV	TC	W	sum
I	5.60%	0.90%	1.20%	0.10%	7.80%
LV	3.60%	21.30%	9.90%	0.10%	34.80%
TC	0.40%	1.00%	40.60%	0.30%	42.30%
W	0.00%	0.10%	0.50%	14.40%	15.00%
sum	9.60%	23.20%	52.20%	14.90%	100.00%

Table 6 Comparison of NLCD (2016) and Dynamic World label (2017) over total 2250 NAIP (2017) images. Sum of the diagonal elements is 81.90%

Figure 6 shows a comparison of NLCD and DW for one sample NAIP image (id = 3716). Only in this figure, Tree Canopy is colored by blue, and pixels where NLCD and DW are identical are colored by black for improving the visibility. One can see from the figure that NLCD and DW are different mainly in pixels of edges of a land cover class (e.g. Low Vegetation) or along roads.

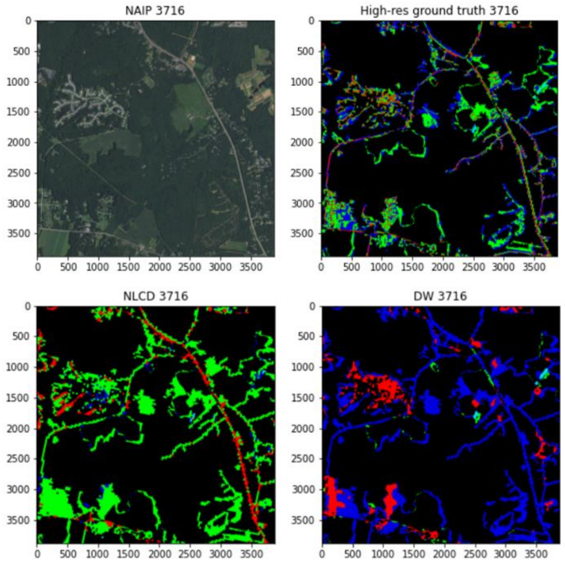


Figure 6 Comparison of NLCD (2016), Dynamic World label (2017) and our high-resolution manual label for NAIP 3716 (2017). To improve the visibility, *Tree Canopy is colored by blue, and pixels where NLCD and DW are identical are colored by black*

Next, we shows a naïve comparison of performance between NLCD and DW using our manual labels. Specifically, we compared NLCD 2016 or DW 2017 and our high-resolution manual labels for four NAIP 2017 images and computed IoUs which are shown in Table 7. Note that these IoUs are not "loss/gain" IoUs reported in this paper. One sees that DW is by no means more inferior than NLCD in terms of IoUs. This may be at least attributed to the disadvantage of NLCD that it is as of 2016, not 2017. This implies an advantage of DW that it is available for any years (from 2015). For ablation study, comparing with DW for 2016 can be our future task.

Class	NLCD	DW
W	71.7%	74.3%
TC	65.6%	74.1%
LV	51.5%	49.0%
I	38.0%	42.1%
simple ave	56.7%	59.9%

Table 7 IoU comparison of NLCD 2016 andDW 2017 for four manual ground-truth labels by authors

Finally, we investigated the performance of models trained on NAIP 2017 images and DW 2017 labels, compared with NAIP 2017 images and NLCD 2016 labels. Specifically, we compared the following four models:

1. Model trained on NLCD 2016

2. Model trained on DW 2017 hard labels
3. Model trained on DW 2017 soft labels
4. Model average of 1. and 3. (taking average of outputted probability of each model)

Since there are no DW labels for 2013, we trained our models from 1. to 3. using either NLCD 2016 or DW 2017, and apply those models not only for NAIP 2017 images but also NAIP 2013 images as well. For models, we only tried the same 5-layer FCN as already introduced but without batch normalization, i.e. the same structure as used in the baseline model github [1]. We used randomly selected 2000 pairs of NAIP 2017 images, NLCD 2016 and DW 2017 labels, and we trained our models for 0.5 epoch (i.e. 1000 images). Considering the difference in the brightness between NAIP 2013 (brighter) and 2017 (darker), we applied data augmentation with respect to the brightness of the NAIP image which is explained in more detail in the Appendix. So, the each pair of NAIP image and NLCD or DW label will be augmented to 2 pairs (original NAIP image and its brighter one with its label unchanged). Hence, the experiment flow is as Figure 7.

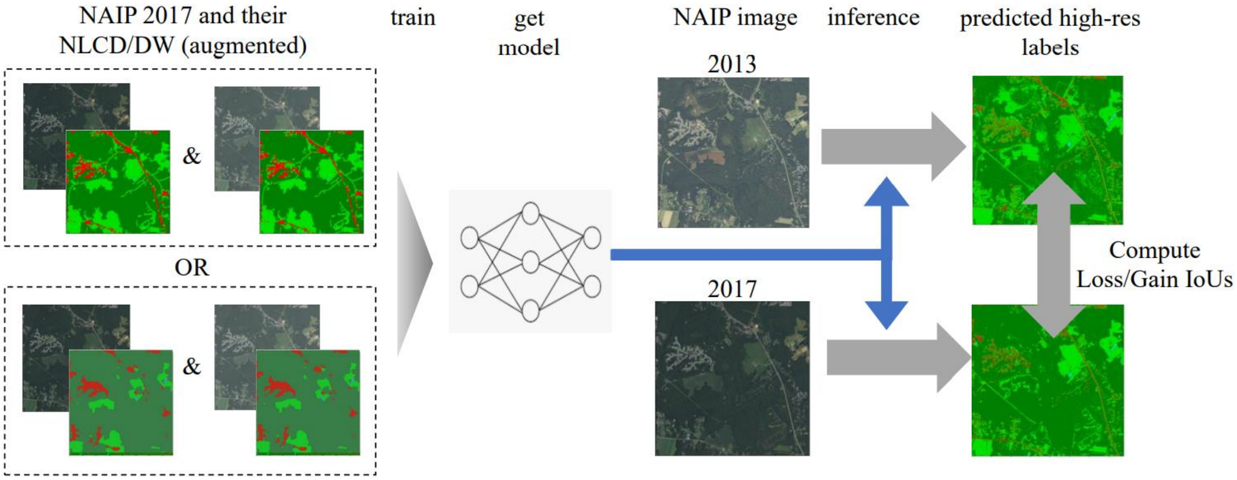


Figure 7 Experiment flow

The results are summarized in Table 8. First, DW soft label (solid orange) has similar performance to NLCD. This is important result since it implies the possibility that we can use DW instead of NLCD without degrading the predictive performance at least for the state of Maryland. Second, DW soft label is slightly better than DW hard label in average and for many categories, which encourage the use of soft labels rather than hard labels when they are available. Finally, the model average of NLCD and DW soft label performed better than both in average and for almost all categories, which suggests one way to combine global and local labels. Note that its average IoU was 0.3949, which is comparable to the 5-layer FCN models trained on both NAIP+NLCD 2013 and 2017.

	Ave	W-	TC-	LV-	I-	W+	TC+	LV+	I+
NLCD	0.3781	0.3480	0.4954	0.5397	0.3360	0.1253	**0.2604**	0.4992	0.4208
DW (hard)	0.3557	0.3190	0.4767	0.5446	0.2999	0.1106	0.1879	0.4688	0.4381
DW (soft)	0.3727	**0.4203**	0.4906	0.5226	0.2304	**0.1474**	0.2449	0.4476	0.4776
Model average: NLCD + DW(soft)	**0.3949**	0.3840	**0.4973**	**0.5683**	**0.3361**	0.1380	0.2496	**0.5071**	**0.4789**

Table 8 Result of IoU scores of 5-layer FCN baseline models

Goals and Next Steps

Our next goal is to use our computational resources to train the proposed model. As we mentioned we need to train 5-layer FCN for phase 1 and after hyper tuning and produce a higher-resolution label. Use the result for the second phase. In this phase, we are going to use Ensemble methods and pick the best three pre-trained models as the main models. The final result will be the average of the output of these models for each of the years. After finalizing the models in phases 1 and 2, we can do post-processing on the result and calculate land cover changes and other necessary parameters.

Contribution (accumulated):

- Ashkan Bozorgzad:
- Hari Prasad
- Karveandhan Palanisamy:
- Masataka Koga:
- Yewen Zhou:
- Yuki Ikeda: Developed the initial code of baseline model using PyTorch, developed analysis on Dynamic World label (downloading, EDA, training, writing this report and making slides), established the way of reliable evaluation using DFC2021 contest page on Codalab (including developing the initial code for submisstion), explored and shared an efficient way to use GroundWork for manual labling, and developed the code to convert a geojson file of a manual label on GroundWork into numpy array

Reference:

[1] N. Malkin, C. Robinson, and N. Jojic, “High-resolution land cover change from low-resolution labels: Simple baselines for the 2021 IEEE GRSS Data Fusion Contest,” Jan. 2021.

[2] Z. Li, F. Lu, H. Zhang, G. Yang, and L. Zhang, “CHANGE CROSS-DETECTION BASED ON LABEL IMPROVEMENTS AND MULTI- MODEL FUSION FOR MULTI-TEMPORAL REMOTE SENSING IMAGES State Key Laboratory of Information Engineering in Surveying , Mapping and Remote Sensing , School of Electronic Information , Wuhan Univer,” *Int. Geosci. Remote Sens. Symp.*, pp. 2054–2057, 2021.

[3] L. Tu, J. Li, and X. Huang, “High-Resolution Land Cover Change Detection Using Low-Resolution Labels via a Semi-Supervised Deep Learning Approach - 2021 IEEE Data Fusion Contest Track MSD,” in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 2058–2061.

[4] “2021 IEEE GRSS Data Fusion Contest: Track MSD” [Online]. Available: https://www.grss-ieee.org/community/technical-committees/2021-ieee-grss-data-fusion-contest-track-msd/.

[5] Q. Bao *et al.*, “MRTA : MULTI-RESOLUTION TRAINING ALGORITHM FOR MULTITEMPORAL SEMANTIC CHANGE DETECTION Qianyue Bao, Yang Liu, Zixiao Zhang, Dafan Chen, Yuting Yang, Licheng Jiao, Fang Liu Key Laboratory of Intelligent Perception and Image Understanding of Ministry,” *Int. Geosci. Remote Sens. Symp.*, pp. 2062–2065, 2021.

[6] Z. Li *et al.*, “The Outcome of the 2021 IEEE GRSS Data Fusion Contest—Track MSD: Multitemporal Semantic Change Detection,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 1643–1655, 2022.

[7] Z. Zheng, Y. Liu, S. Tian, J. Wang, A. Ma, and Y. Zhong, “WEAKLY SUPERVISED SEMANTIC CHANGE DETECTION VIA LABEL REFINEMENT FRAMEWORK State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing Wuhan University, Wuhan 430079, China,” *Int. Geosci. Remote Sens. Symp.*, pp. 2066–2069, 2021.

[8] Brown, C.F., Brumby, S.P., Guzder-Williams, B. *et al.* “Dynamic World, Near real-time global 10 m land use land cover mapping,” *Sci. Data*, vol. 9, 251, 2022

[9] Han Liu, Peng Gong, Jie Wang, Xi Wang, Grant Ning, Bing Xu, “Production of global daily seamless data cubes and quantification of global land cover change from 1985 to 2020 - iMap World 1.0,” *Remote Sensing of Environment*, vol. 258, 112364, 2021

[10] García-Álvarez, D., Lara Hinojosa, J., Jurado Pérez, F.J., Quintero Villaraso, J., “General Land Use Cover Datasets for Europe,” *Chapter*, pp. 313–345, 2022

Appendix.

A. Data augmentation with respect to brightness

When we trained models on NAIP 2017 and either DW 2017 or NLCD 2016, we applied data augmentation with respect to brightness of NAIP 2017 images. Specifically, we prepared one brighter transformed image for every NAIP 2017 image that we used for training with its label (either DW 2017 or NLCD 2016) unchanged. Since NAIP 2013 images are generally brighter than their corresponding NAIP 2017 ones, this is to generalize the model even to brighter NAIP 2013 images. In more detail, we applied γ-transformation to the numbers of the RGB channels and the infrared channel of each pixel of NAIP 2017 images as follows:

$$x_{transformed} = 255\left(\frac{x}{255}\right)^{1/\gamma},$$

where γ is set to 1.6 and x is the number of each channel. γ-transformation uses the exponential relationship between the input light volume and the output signal intensity by image sensors, and it is popular in image analysis to adjust brightness. In our case, this will increase the numbers of each channel and make the original images brighter. Note that we normalized all input images by dividing by 255, but we did not standardize them, i.e. we did not subtract mean or divide by standard deviation. The figure below shows the flow of the data augmentation in the case of NLCD 2016 label.

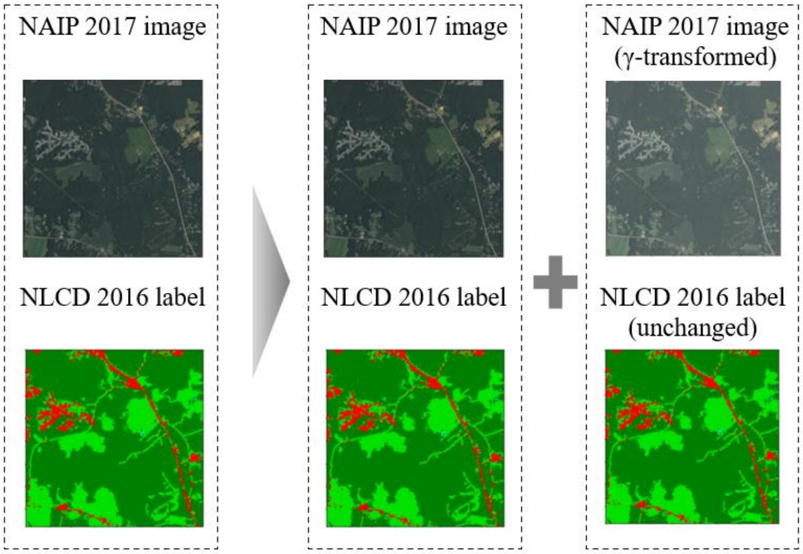


Figure A.1 Data augmentation with respect to brightness in case of NLCD label

The effect of this data augmentation is remarkable. Figure A.2 shows the predictions of a model trained with and without data augmentation (trained NAIP 2017 and NLCD 2016). Without data augmentation, the model trained on NAIP 2017 did not generalize well to NAIP 2013. However, training on augmented dataset led to a model generalized even to NAIP 2013.

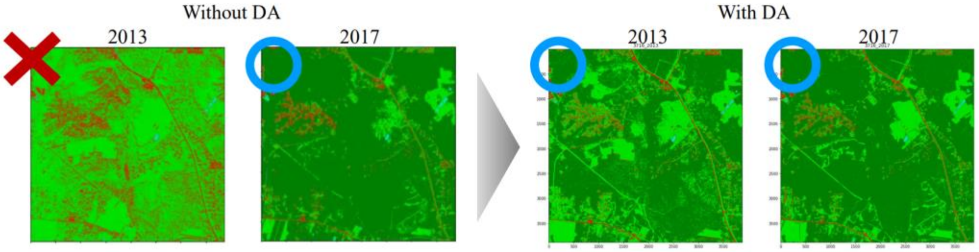


Figure A.2 Prediction of models with and without data augmentation

Note that another possible way is standardization of input images, however, if images for training and images for which predictions are made are quite different (e.g. taken from quite different regions), the standardized distributions of numbers of each channel would be quite different, which can still cause failures in model predictions. Also, it is not feasible for online prediction where we cannot know the distribution of the input images for which predictions are made.