Chesapeake Bay Program Non-Tidal Wetland Mapping

Final Report 12/13/2024

Prepared by:
Dr. Michael Evans, Deputy Director
Conservation Innovation Center
Chesapeake Conservancy

SUMMARY

The goal of this project was to improve the ability of the Chesapeake Bay Program to inventory and monitor non-tidal wetland ecosystems within the Chesapeake Bay watershed (CBW) and create a wetland mapping system that can contribute to National Wetland Inventory data updates. To meet these goals, we developed methods that delineate and classify wetland areas using free and publicly available remote sensing and GIS data. These methods are automated, repeatable, and scalable. We built and trained two deep-learning models (i.e., AI) that ingest a combination of National Agriculture Imagery Program (NAIP) multispectral imagery, digital elevation model (DEM) data, SSURGO soil characteristics, and Sentinel-1 synthetic aperture radar returns, and produce output arrays containing the per-pixel probability of membership in each of five exclusive categories: emergent wetlands, forested wetlands, farmed wetlands, open water wetlands, and non-wetland. The use of free, nationally available datasets allows these models to be used throughout the Chesapeake Bay watershed to update and improve existing wetland data. Both deep-learning models exhibited a strong ability to delineate wetlands during model training, exhibiting > 0.9 precision and recall on evaluation data. Models exhibited acceptable precision > 0.4 and recall > 0.7 relative to existing NWI data in test areas within the Chesapeake Bay watershed. Qualitatively, these outputs capture areas of open water and forested wetlands that can be used to update and improve existing wetland maps. In conjunction with trained models, we developed a cloud-computing architecture and workflow that used free and publicly available tools to produce wetland data using trained deep-learning models quickly across large spatial extents. The production process produces new wetland data in county-sized areas in 3 – 6 hours. The process requires minimal user input and does not require download or local storage of input data making it portable and scalable. We validated the utility of this wetland mapping system by mapping and classifying wetlands within 14 counties that span the full range of states and physiographic provinces present within the CBW. This capability will allow wetland maps to be updated relatively quickly when either new data (e.g., 2024/25 NAIP) or improved models become available.

METHODS

Deep Learning Architectures

We used two well-studied deep-learning architectures to delineate and classify wetlands by vegetation type from freely available remote sensing and GIS data. First, we adapted U-Net – a well-studied fully convolutional encoder-decoder architecture used for image segmentation – to ingest three-dimensional array data and produce three-dimensional arrays in which pixels store the probabilities of class membership. Our U-Net model consisted of five consecutive convolutional encoder blocks, which increase the feature space of the data while reducing spatial resolution, and five decoder blocks that restore spatial detail. Encoder blocks have two sequences of a convolutional layer, a batch normalization layer and a rectified linear unit activation, followed by a max pooling step to reduce spatial resolution. Decoder blocks have a deconvolution layer that increases spatial resolution, the output of which is concatenated with output from the reciprocal encoder layer, followed by two sequences of convolution, batch normalization, and rectified linear unit activation. We hereafter refer to this sequence of layers as the U-Net model. Python code to build this U-Net model with an arbitrary number of input variables and output classes is available through a public GitHub repository (see Outputs).

The U-Net model delineates wetlands using a single stack of three-dimensional array data representing conditions at a single point in time. Because wetlands are dynamic and variable in space and time, we developed a second deep-learning architecture designed to explicitly take advantage of time-series data; the Long Short-Term Memory (LSTM) network. LSTM networks are a recurrent neural network designed to efficiently use temporal data. They consist of a sequence of cells equal to the number of timesteps present in input data. Each cell combines the input from a given time step, output from the previous cell, and a hidden state that is propagated forward through the entire sequence. The calculation of the output of a given cell is controlled by three gates: input, output, and forget. The forget gate determines to what extent the previous cell state is remembered. The input gate determines how strongly the new input contributes to the new cell state. Finally, the output gate is used to calculate the new hidden state. Our approach used convolutional LSTM cells that will use 3×3 kernels which are convolved over three-dimensional input within each gate. A convolutional LSTM model uses both the spatial context and temporal sequencing information inherent to time series of remotely sensed data. Our second model architecture combined the penultimate outputs of a U-Net model with the penultimate outputs from an LSTM model ingesting a time-series of Sentinel-1 radar returns. The output cell state from the final LSTM cell was concatenated with the final layer of the U-Net model and convolved by a 1 x 1 kernel with a softmax activation function producing a vector of values corresponding to the probability of a pixels' membership in each of the five wetland classes. Additionally, the penultimate outputs from the U-Net layers within this model were also subjected to softmax activation to produce a vector of probabilities corresponding to membership in a course three category wetland classification system, within which the full five categories are nested (hence the 'hierarchical' description of this model). We refer to this model as the 'hierarchical' model. The hierarchical model takes two sets of inputs – a three-dimensional array equivalent to the U-Net model, and a time-series of three-dimensional arrays (i.e., four-dimensional array).

Training Data

Deep-learning models were trained with paired sets of covariate input data (i.e., remote sensing data) and wetland label data (i.e., wetland class labels) sampled at random points throughout the CBW. We generated 10,000 random points inside National Wetland Inventory (NWI) polygon and line features in and around the CBW (Figure 1). Following guidance from CBP and NWI partners, we restricted NWI data to those produced by projects that were completed in 2014 or more recently (Table 1). This subset of NWI projects was chosen to ensure that the polygons used as wetland 'truth' were produced using updated quality control standards. We note that linear features representing streams may be older than polygons for a given NWI project as they are often ingested directly from the National Hydrography Dataset. To create training data representing non-wetland areas, we generated an additional 10,000 points falling outside of NWI wetland polygons and lines, but within the same NWI project area to restrict non-wetland point sampling to areas in which wetlands were completely delineated. Following our initial sampling, we tabulated the number of samples falling within each reclassified wetland class (Table 1). Based on these numbers and the project focus on identifying non-tidal, forested wetlands we generated an additional 5,000 points within polygons identified as Palustrine, forested wetlands (PFO1A).

We used the same set of NWI polygon and line features to create wetland label data. Because the focus of this project was to develop methods that can identify non-tidal, forested wetlands we did not use the full classification taxonomy available in NWI wetland data to define wetland categories in our label data. However, initial testing showed that a simple binary non-wetland/wetland classification resulted in models underrepresenting forested wetlands and confounding agricultural fields. Following advice and guidance from NWI partners and Ducks Unlimited Staff, we reclassified NWI wetland data into five general and distinct categories based on the WETLAND_TYPE attribute (Table 2). We trained both models to identify non-wetland areas, emergent wetlands, forested wetlands, farmed wetlands, and open water.

Covariate input data for the U-Net model included the red, green, blue, and near infrared reflectance values in multispectral NAIP imagery, lidar-derived digital elevation data, and SSURGO soil characteristics including annual minimum depth to water, drainage class, flooding frequency, and hydric class percentage. These variables were selected following discussions with NWI staff as well as a literature review that identified the most important and consistently used soil covariates in other wetland mapping research. Flooding frequency and drainage class are ordinal, categorical variables. We converted these to scaled, continuous values by first re-ordering the levels from driest to wettest conditions, setting these conditions to 0 and 1, respectively, and interpolating the intermediate levels. (e.g., 'Excessively drained' = 0, 'Subaqueous' = 1). We clipped the maximum value of annual depth to water at 200 cm, and the maximum value for hydric class percentage at 100. For all soil characteristic variables, we replaced missing values with the value corresponding to the driest condition based on the understanding that missing values represent places with no soil (e.g., bare rock).

Covariate inputs to the LSTM architecture within the hierarchical model were time series of the radiometrically terrain corrected VV and VH polarized backscatter returns from Sentinel-1 C-band synthetic aperture radar. Sentinel-1 data have been collected historically by a pair of sun-synchronous, near-polar orbiting satellites, producing global radar backscatter data every six days. The C-band synthetic aperture radar instruments are right-looking, meaning that the same location will have different backscatter signatures based on the north-south direction of the orbit at that location during a given pass. We used only images collected during the ascending phase of the orbit.

All remotely sensed and GIS data were accessed through the Microsoft Planetary Computer - a free cloud-based platform for programmatically accessing and working with publicly available geospatial data. The Planetary Computer maintains up-to-date catalogs of NAIP, USGS 3DEP digital elevation, SSURGO, Sentinel-1, and NWI data. At each sampling point, we extracted a 360 x 360-meter array of all U-Net covariates. These data are provided in different native resolutions, therefore these 'chips' had different pixel dimensions. NAIP data prior to 2017 have 1 m resolution, and 0.6 m resolution thereafter (i.e., 600 x 600 x 4-pixel arrays). We used the vintage of NAIP imagery that matched the vintage of imagery used to delineate the NWI polygons at a given point, as recorded by the NWI project documentation. This ensured that the multispectral imagery used to train the model reflected the same conditions represented in the wetland label data. DEM data were 1 m resolution (360 x 360 x 1-pixel arrays). SSURGO data are available as a 10 m raster that stores map unit identifiers per pixel, and tables containing the soil characteristic attributes for each map unit. At each point we extracted 36 x 36 x 4-pixel arrays from the map unit raster and joined the relevant tabular soil characteristics to these pixels based on the map unit values. We converted NWI vector data at each point to raster using the affine transform matrix, bounds, and shape of the corresponding NAIP chip, and converted categorical 1 – 5 values to one-hot vectors (i.e., factors) per pixel. This label data therefore indicated a pixels' exclusive membership in one of each wetland class. We resampled all U-Net covariate chips and NWI rasters to a common 0.6 m resolution (i.e., 600 x 600-pixel arrays) to facilitate easy integration with future NAIP data. Like NAIP data, we accessed a year of Sentinel-1 images collected during the same year as the images from which NWI data were delineated at a given point. For projects using imagery prior to 2016 we use the earliest available year of Sentinel-1 data. At each point we aggregated Sentinel-1 data into median composite images per 60-day window and extracted a 360-meter x 360-meter chip. This produced a 6 x 36 x 36 x 2-pixel chip. All covariate and NWI arrays were then exported to separate directories per variable in an Azure Blob Storage container (see Outputs). File names provide a unique id based on the coordinates of the sampling point and were used to match variable arrays originating from the same location. This segregated storage facilitated easy experimenting with different combinations of covariates during model training. During export from the Planetary Computer to Azure Blob storage, the data were randomly assigned to training or evaluation directories with 0.7 and 0.3 probability, respectively. Code used to access, process, and sample remote sensing and GIS datasets are available through a GitHub repository (see Outputs).

Model Training

During model training, we implemented data augmentation and standardization to ensure a consistent, memory-efficient range of values for all incoming input data, and to increase the variability of data seen by the model. Together, these techniques maximize deep-learning models' ability to accurately identify wetlands in diverse geographies and times. For each input variable, we rescaled native values to a range of [0, 1] based on empirical minima and maxima. For all NAIP bands we used a minimum and maximum value of 0 and 255, for digital elevation data, we used a maximum of 0 and 2000, and for Sentinel-1 VV and VH returns we used a minimum and maximum of -50 and 0. We additionally augment NAIP bands spectrally and all covariate data morphologically. Immediately prior to model ingestion during training, the brightness and contrast of each multispectral band were randomly adjusted by ±5%. Finally, the entire stack of input variables and label data were randomly rotated by 0, 90, 180, or 270 degrees and randomly flipped vertically and horizontally.

We trained all models for 200 epochs using batches of 8 training data examples per iteration. We optimized model weights by minimizing a weighted generalized dice coefficient loss function calculated between one-hot label vectors and softmax class probability vectors using the Adam optimizer with initial learning rate of 1e-4 and a decay rate of $\beta 1 = 0.9$, $\beta 2 = 0.999$. At the end of each epoch, we evaluated model performance by calculating mean intersection over union (IoU), precision, and recall between predicted wetland class probabilities and wetland labels using among validation data at the end of each epoch.

*IoU: (wetland probabilities * wetland labels) / (wetland probabilities + wetland labels)*

Precision: True Positive/(True Positive + False Positive)

Recall: True Positive/(True Positive + False Negative)

Model weights were saved if they improved performance in terms of IoU on evaluation data. The evaluation data contains 5,000 examples of paired predictor variables and reclassified NWI labels. These evaluation examples are never used during training to update model weights. Therefore, they provide an evaluation of the model's performance on out-of-bag examples. The metrics are mean values across classes weighted by class prevalence. Model training was performed using Keras with Tensorflow backend with batches of 16 chips per training step. We created NVIDIA Tesla K80 virtual machines with 2 GPU cores accessed through Microsoft Azure and used these machines for model training.

Data Production

We developed a portable workflow to generate wetland probability data using trained model weights. A georeferenced polygon and date range defining the location and time for which outputs are desired are used as inputs to query the Planetary Computer catalog and acquire the required covariate data (i.e., NAIP, DEM, and SSURGO). Covariate data are then pre-processed as described during model training (i.e., rescaled, resampled, etc.), except for the random data augmentation steps used to increase data variability during model training. Covariate data are then stacked into a single image and subdivided into overlapping chips which are provided to the relevant deep-learning model, and for which output inference data are obtained. Areas of overlap are removed from each output chip to create a seamless, contiguous output raster with the same spatial extent and shape as input polygons. To speed inferencing, we parallelized the access and processing of input covariate data chips across multiple available CPU cores, in which each core takes a different coordinate and retrieves the relevant input covariate data. We used this system and the best performing weights for each model to generate maps of wetland probability within 14 test counties in and around the CBW (Figure 1). Outputs for counties in Pennsylvania were shared with Ducks Unlimited staff who are performing an NWI wetland delineation project in those counties. Ducks Unlimited staff provided qualitative evaluation of the accuracy of model outputs and their utility in expediting NWI data production. We evaluated model performance quantitatively using outputs generated within ten small $10km^2$ test areas. In these areas, we converted the multiband probability outputs into a single-band classification raster and calculating precision, recall, and IoU using reclassified, rasterized NWI wetland polygons as true reference data. We used a receiver operating curve analysis to identify the probability threshold that best converts probability raster to wetland polygons. We used a series of probability thresholds [0.1, 0.2, 0.2, 0.4, 0.5] to assign pixels to a wetland or non-wetland class. For each threshold, we assigned the pixel to the highest probability wetland class given that probability was above the given threshold. Pixels in which no wetland class exceeded the given probability threshold were assigned to non-wetland. For example, a pixel with the following probabilities:

Non-Wetland: 0.4; Emergent Wetland: 0.1, Forested Wetland: 0.3, Farmed Wetland: 0.1, Open Water: 0.1

would be assigned to Forested Wetland for thresholds, 0.1, 0.2, and 0.3 and non-wetland for thresholds 0.4, and 0.5. At each threshold we calculated precision, recall, and IoU. The threshold at which IoU was maximized was selected as the optimum threshold for polygon creation. We constructed receiver operating characteristic curves from this series of metrics and compare the performance of model architectures in terms of area under the curve.

RESULTS

We trained both the U-Net and hierarchical models for 200 epochs. The best performing model weights files (.hdf5) are available in an Azure Blob Storage container at URLs (see Outputs). Both models exhibited ~ 0.90 precision and recall, and between 0.7 and 0.8 IoU using model evaluation data (Table 3). Precision and recall calculated against NWI data within 10km^2 county subsets ranged from 0.70 & 0.45 to 0.24 & 0.82 for U-Net, and from 0.88 & 0.22 to 0.34 & 0.73 for the Hierarchical model (Figure 2). Receiver operating characteristic analysis indicated that the U-Net model (AUC = 0.65) was superior to the Hierarchical model (AUC = 0.57). For both models, the maximum IoU was achieved using a probability threshold of 0.3 (Figure 2).

We used the best-performing U-Net weights to generate wetland probability rasters within 14 test counties, spanning all five states and six physiographic provinces within the Chesapeake Bay Watershed. Parallelization across four cores decreased the time required to produce data for a county by approximately 1/3 (18 hours to 6). These are multiband rasters, in which each band represents the model-estimated softmax probabilities of membership in a different class. Band 1 stores non-wetland probability, band 2 stores emergent wetland probability, band 3 stores forested wetland probability, band 4 stores farmed wetland probability, and band 5 stores open water probability (Figures 3 & 4). Band values at a given pixel sum to one. These outputs are available as cloud optimized geotiffs located in a public Azure Blob Storage container (see Outputs).

Outputs within counties in Pennsylvania produced by the U-Net model were provided to Ducks Unlimited to evaluate their utility in complimenting or enhancing photo interpretation wetland delineation. Feedback provided by DU staff indicated that open waterbodies identified by the model were of sufficient accuracy and quality to be automatically included in final production data without manual correction. Staff communicated that forested wetland polygons were generally accurate in terms of their location and extent and offered an improvement over existing NWI data in Pennsylvania (Figure 5). However, the boundaries of model outputs did not exactly match those drawn by expert photo-interpreters, requiring manual correction to be of sufficient quality for inclusion in final data production (i.e., to meet Federal Geographic Data Committee Wetland Mapping Standard requirements). Emergent wetlands were identified as problematic, specifically in terms of commission errors in agricultural fields.

SUMMARY & NEXT STEPS

This project has produced two deep-learning models that can automatically map non-tidal wetlands using free, public, and nationally available remotely sensed and GIS data, along with a cloud-computing infrastructure and workflow to quickly produce wetland maps at the county scale. Evaluation of the model performance on both evaluation data ‘chips’ as well as over larger areas of interest indicated that a U-Net architecture ingesting NAIP, elevation, and soil characteristic data was superior to a more complicated architecture that additionally used C-band synthetic aperture radar time series data. This may indicate that when using a deep-learning approach there is sufficient information available in the multispectral reflectance, elevation, and soil characteristics to accurately identify wetland areas, and that SAR returns added little additional informational value. However, the hierarchical model may still have value for future wetland mapping analyses due to its inherent ability to simultaneously produce wetland membership probabilities for multiple classification schemes. Specifically, the model returns a simpler wetland vs. open-water vs. non-wetland probability output that may outperform the five-class U-Net model in identifying wetland areas more generally. If this capability is of interest to the Chesapeake Bay Program or its partners, we would recommend a follow up analysis comparing the performance of these models at this coarser aggregation of wetland classes.

Our analysis also indicates that the U-Net model largely accurately mapping true wetlands, based on both quantitative and qualitative assessment. Recall rates relative to NWI data in county subsets were relatively high, while precision rates were lower. This is what we would expect if a model were identifying wetlands accurately in cases where existing data are accurate but incomplete (i.e. Figure 5). In this scenario, lower precision may correspond to the identification of wetlands missing from existing data. Evaluation by Ducks Unlimited indicated that this may be the case, at least in Pennsylvania counties, especially for newly created (i.e., post NWI data production) open water and forested wetland areas. The Conservancy will be acquiring field delineated wetland boundaries from Ducks upon completion of their Pennsylvania NWI update project and can re-run the accuracy analysis on these high-precision, up-to-date data. Similarly, Exelon utility companies including Baltimore Gas and Electric, Delmarva Power, and PEPCO have offered to provide field-delineated wetland boundaries from past projects that can similarly serve as an additional set of ground truth data. The Conservancy will perform these analyses to strengthen the quantitative assessment of model performance in situ, and the results can be used to guide any potential fine-tuning of the model for improved performance.

The assessment of model outputs by Ducks Unlimited indicated additional strengths, limitations, and potential next directions for the development of deep-learning wetland mapping. First, Ducks Unlimited staff expressed high confidence in the open-water class produced by the U-Net model, such that they would use these outputs to create open-water NWI wetland polygons instead of the existing 1 m resolution land cover data. The ability to automate the mapping of open-water wetlands (e.g., ponds, lakes, etc.) could expedite the creation of new NWI data relative to manual photointerpretation. Second, Ducks Unlimited staff reported that U-Net model outputs successfully identified and delineated forested wetland areas. Differences in the exact boundaries around these wetlands, and the five-meter horizontal accuracy requirement, precluded staff from automatically ingesting model outputs to create NWI forested wetlands. An outstanding question that the Conservancy identified is the degree to which discrepancies between model outputs and the polygons drawn by photointerpreters represent true differences in the accuracy. Remote wetland delineation involves some subjectivity, and it is impossible to ascertain ‘truth’ absent a site survey. Comparing both the U-Net model outputs and photointepreted polygons with field-delineated would help answer this question and potentially provide a path for incorporating forested wetland model outputs in NWI data production. Even absent the level of precision and confidence needed to directly incorporate model outputs into NWI data, the Ducks Unlimited team recognized the utility of the automated wetland mapping methods developed in this project to rapidly assess and update wetland presence information in new areas. These capabilities can be used in a number of other contexts such as performing rapid inventories of wetland loss, identifying areas requiring new manual delineation, and providing advanced guidance for in situ site surveys.

In addition to the trained deep-learning models, this project delivered computational architecture and workflow that can be used to repeatably produce wetland data at large spatial scales (i.e. counties). We solved challenges related to accessing cloud-hosted data, data pre-processing, spatial reprojection and alignment, and parallel computing that can be leveraged in a more general way. This capability means that future efforts can focus on refining the deep-learning, statistical, or mechanistic models used to estimate wetland probability from input data, as these methods can be inserted into the current workflow.

TABLES

Table 1. *NWI projects from which wetland vector data were used.*

NWI Project ID	NWI Project Name	Source Imagery Year	# of Samples
R05Y11P02	Chesapeake Bay Update	2013	5,916
R09Y16P04	Virginia Fix	2013	1,729
R05Y13P07	James River Updates	2012	3,845
R06Y18P03	Mapping for Scalable Data Areas of New York	2014 - 2017	963
R05Y19P02	NWI Update for Eight West Virginia Counties	2019	2,547

Table 2. *Crosswalk between original NWI wetland types and reclassified wetland categories used to train deep-learning models. Model outputs are the probability of membership in reclassified categories.*

NWI Wetland Type	Reclassification	Output Band	Samples
Non-Wetland	Non-Wetland	1	10,000
Estuarine and Marine Wetland	Emergent Wetlands	2	1,628
Freshwater Emergent Wetland	Emergent Wetlands	2	903
		Total	*2,531*
Freshwater Forested/Shrub Wetland	Forested Wetlands	3	15,311
Freshwater Pond	Open Water	5	564
Lake	Open Water	5	288
Estuarine and Marine Deepwater	Open Water	5	215
Riverine	Open Water	5	936
		Total	*2,003*
Other	Farmed Wetlands	4	155

Table 3*: Accuracy metrics calculated on model evaluation data using the best-performing weights for each model.*

Model	**Precision**	**Recall**	**IoU**
Hierarchical	0.887	0.892	0.745
U-Net	0.901	0.911	0.792

OUTPUTS

1. ***Model Training Data***

Provided as numpy (.npy) files containing 600 x 600 x N-pixel arrays of covariate input data. The filename provides metadata indicating the vintage, data type, array dimensions, and unique id of the array, delimited by '_'. All files are stored in an Azure Blob Storage container at:

https://aiprojects.blob.core.windows.net/wetlands?si=CBP_NonTidal_deliverables&spr=https&sv=2022-11-02&sr=c&sig=wTXD4yHoAPIEEyVmeQr6ezN553vQyQJji9QXtmpzlJ4%3D

The directory structure within the Azure Blob Container is

test

```
|- train
    |- dem
        |- 2012_dem_360x360_1395362_1750747.npy
        |- ...
    |- naip
        |- 2012_naip_360x360_1395362_1750747.npy
        |- ...
    |- nwi
        |- 2012_nwi_360x360_1395362_1750747.npy
        |- ...
    |- s1
        |- 2012_s1_360x360_1395362_1750747.npy
        |- ...
    |- ssurgo
        |- 2012_s1_360x360_1395362_1750747.npy
        |- ...
|- eval
    |- dem
    |- naip
    |- ...
```

2. ***Trained Model Weights***

Best-performing model weights from the U-Net and Hierarchical models are provided as single keras weights (.hdf5) files in Azure Blob Storage Containers. The U-Net model can be accessed at:

https://aiprojects.blob.core.windows.net/wetlands/models/hierarchical/hierarchical_best_weights_30Sep24_157.hdf5?si=CBP_NonTidal_deliverables&spr=https&sv=2022-11-02&sr=c&sig=wTXD4yHoAPIEEyVmeQr6ezN553vQyQJji9QXtmpzlJ4%3D

The Hierarchical model can be accessed at:
https://aiprojects.blob.core.windows.net/wetlands/models/unet/unet_best_weights_12Sep24_172.hdf5?si=CBP_NonTidal_deliverables&spr=https&sv=2022-11-02&sr=c&sig=wTXD4yHoAPIEEyVmeQr6ezN553vQyQJji9QXtmpzlJ4%3D

3. ***County Wetland Data***

Rasters containing the probability of membership in either non-wetland, emergent wetland, forested wetland, farmed wetland, or open water in every pixel as estimated by the best-performing U-Net model are available as multiband Cloud-optimized geotiffs. These data are publicly available for each of 14 counties within the CBW at the following URLs:

County Name	County FIPS	Cloud-Optimized Geotiff URL
Amherst, VA	51009	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/amhe_51009/unet172/cogs/multiband.tif
Allegheny, PA	42003	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/alle_42003/unet172/cogs/multiband.tif
Cambria, PA	42021	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/cambria_42021/unet172/cogs/multiband.tif
Dauphin, PA	42043	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/daup_42043/unet172/cogs/multiband.tif
Fauquier, VA	51061	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/fauq_51061/unet172/cogs/multiband.tif
Floyd, VA	51063	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/floy_51063/unet172/cogs/multiband.tif
Howard, MD	24027	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/howa_24027/unet172/cogs/multiband.tif
Jefferson, WV	54037	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/jeff_54037/unet172/cogs/multiband.tif
Lancaster, PA	42071	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/lanc_42071/unet172/cogs/multiband.tif
McKean, PA	42083	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/mcke_42083/unet172/cogs/multiband.tif
Morgan, WV	54065	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/morg_54065/unet172/cogs/multiband.tif
Montgomery, NY	36057	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/mont_36057/unet172/cogs/multiband.tif
Powhatan, VA	51145	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/powh_51145/unet172/cogs/multiband.tif
Susquehanna, PA	42115	https://cicwebresources.blob.core.windows.net/wetlands-ai/data/inferencing/susq_42115/unet172/cogs/multiband.tif

4. Data Processing, Model Training, and Inferencing Code

Python scripts used to access data from the Planetary Computer, process data, train deep-learning models, and perform inferencing are available in a private GitHub repository:

https://github.com/conservation-innovation-center/NonTidal_Wetlands

Contact Michael Evans (mevans@chesapeakeconservancy.org) to obtain access.

The workflow used to sample training data is provided in a *Jupyter* notebook in ./notebooks/WetlandSampling.ipnb. Training and inferencing scripts are provided in the ./local directory (e.g. ./local/train_unet.py, ./predict_unet_pc.py, etc.) This project repository relies on an external repo containing utility functions for working with Planetary Computer data, building deep-learning data ingestion tools, and building deep-learning models. When cloning the wetland repository, you will need to include the –recurse-submodules flag:

git clone https://github.com/conservation-innovatin-center/NonTidal_Wetlands.git --recurse-submodules

FIGURES

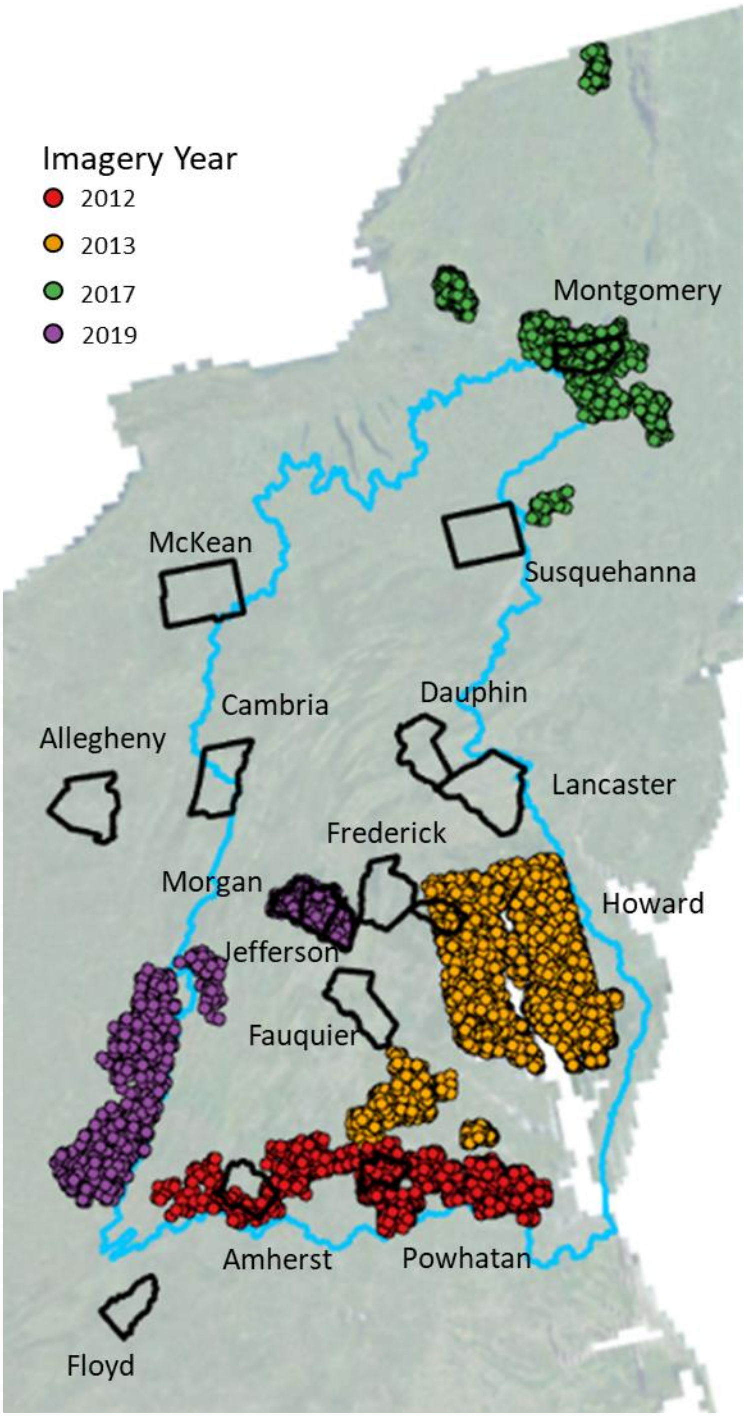

Figure 1. *Chesapeake Bay watershed (blue) study area showing the distribution of data sampling locations (points) and counties for which wetland data were produced (black*

polygons). Points indicate locations at which data used to train deep-learning models was sampled from within recently updated NWI projects. Point colors indicate the NAIP vintage year for a given NWI project.

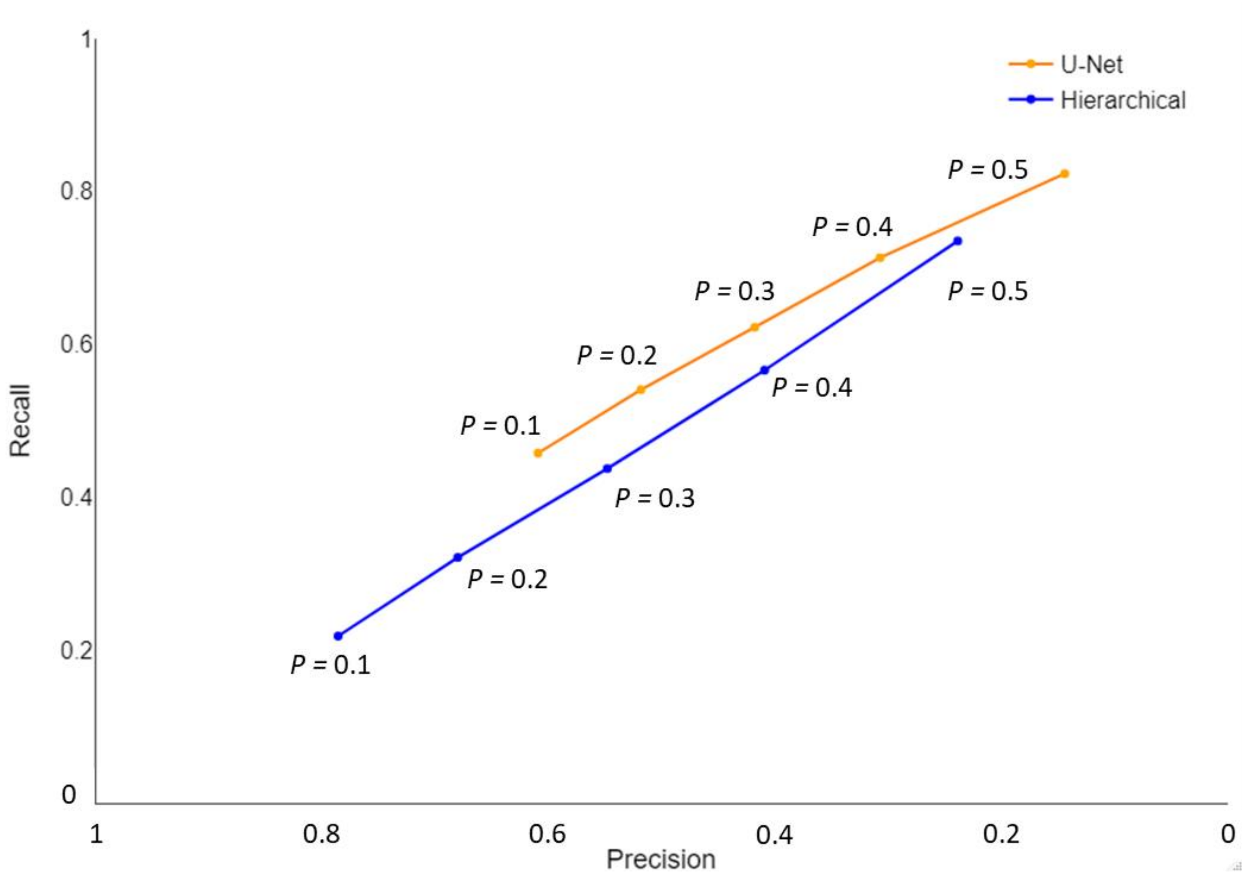

Figure 2. *Receiver operating characteristic curve comparing performance of the U-Net and Hierarchical models as evaluated against NWI data in 10km^2 test areas. Curves show the tradeoff in precision and recall for a series of probability thresholds used to categorize probabilistic model outputs into categorical wetland classes.*

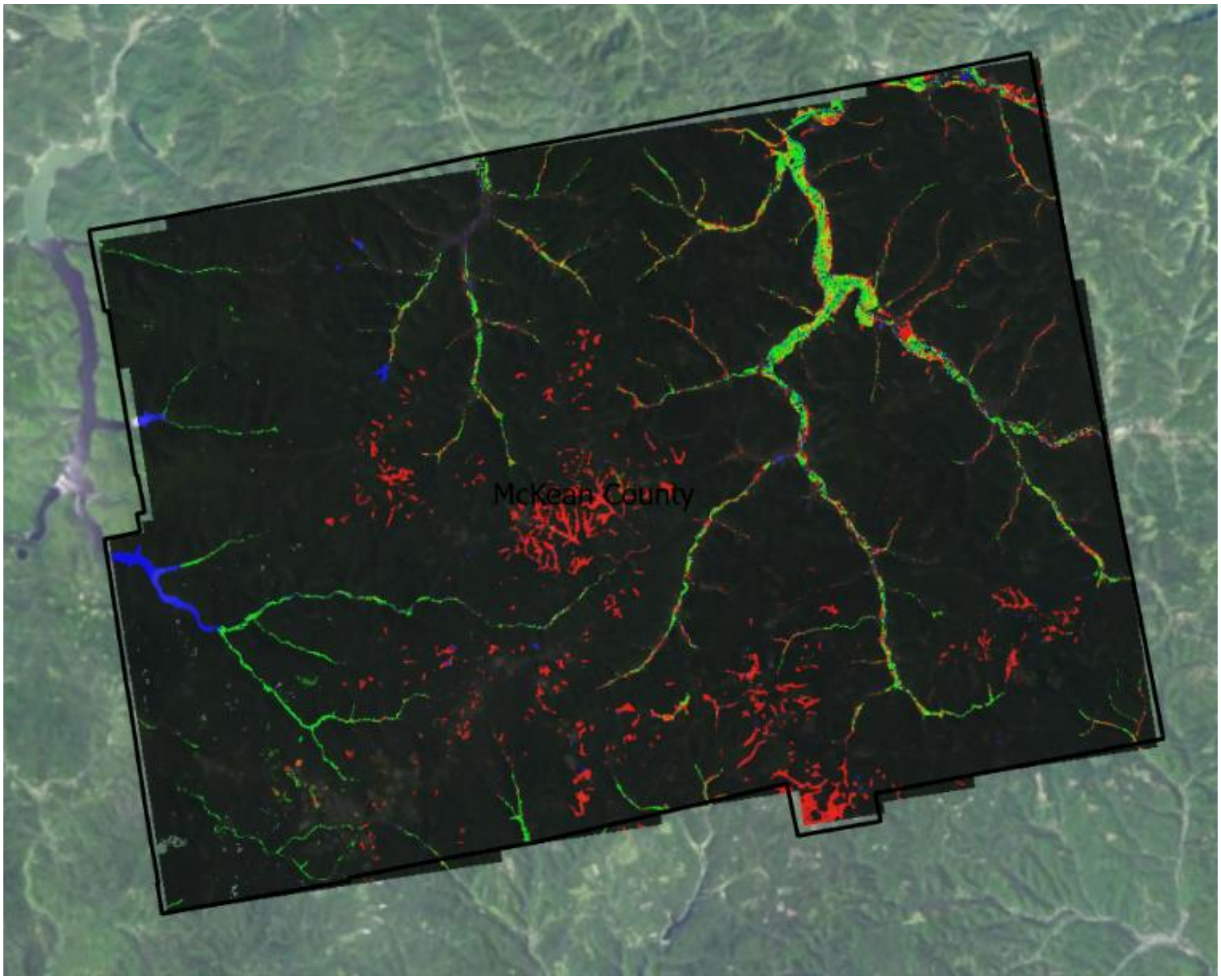

Figure 3. *Example of multiclass wetland probability outputs produced by the U-Net model in McKean County, Pennsylvania. Red areas indicate high probability of emergent wetlands, green areas indicate high probability of forested wetlands, and blue areas indicate high probability of open water. Black areas indicate a high probability of no wetlands.*

Figure 4. *Example of multiclass wetland probability outputs produced by the U-Net model in Morgan County, West Virginia. Red areas indicate high probability of emergent wetlands, green areas indicate high probability of forested wetlands, and blue areas indicate high probability of open water. Black areas indicate a high probability of no wetlands.*

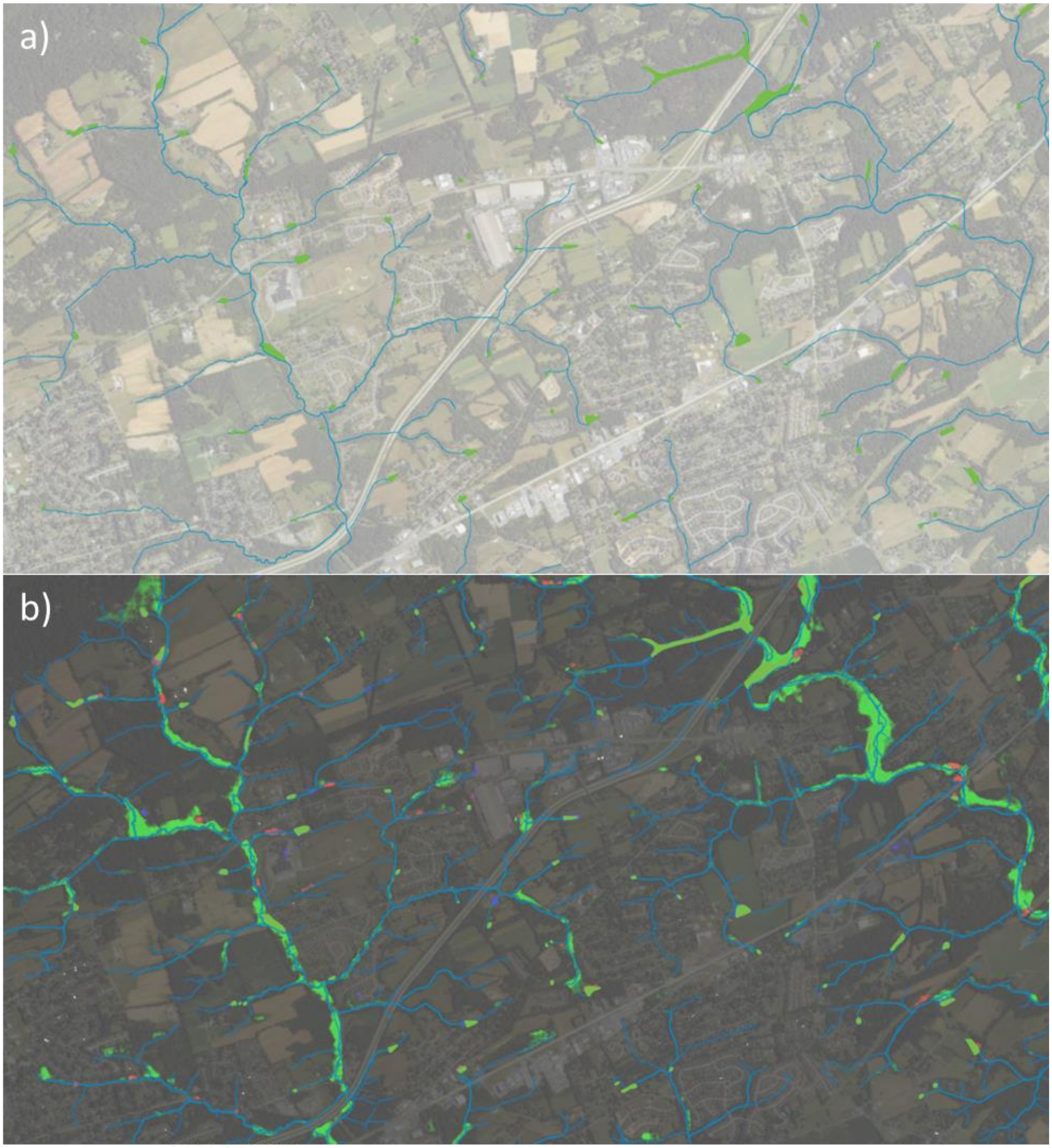

Figure 5. *Comparison of existing NWI wetland data created in 1983 (a) and AI-identified wetlands (b). Images are from Dauphin County, PA. In both panels, green areas represent wetland areas, and blue lines represent open water. In the lower panel, red areas represent emergent wetlands.*