CS 5330: Final Project Report

Segmentation of invasive plant communities in Cape Elizabeth, Maine from high-resolution aerial imagery using U-Net

Philip Englund Mathieu*, GunGyeom James Kim†, and Hao Sheng Ning‡
Khoury College of Computer Sciences
The Roux Institute at Northeastern University
Portland, ME, USA
Email: *mathieu.p@northeastern.edu, †kim.gu@northeastern.edu, ‡ning.ha@northeastern.edu

***Abstract*—This project demonstrates a method for segmenting aerial imagery using the U-Net CNN to detect the presence of *Fallopia japonica*, a plant species that is categorized as invasive in Maine. The project is based on [1], which achieved a similar goal using unmanned aerial vehicle (UAV, i.e. drone) imagery in a different part of the US. This project utilizes ortho-rectified aerial imagery obtained from the Maine GeoLibrary. The proposed solution will enable early detection, mapping, and monitoring of invasive plants in Maine, thereby improving the state's ecological, social, and economic conditions. All code and images used in this project are available in a public repository.**

***Index Terms*—CNN, convolutional neural network, remote sensing, satellite imagery, segmentation, computer vision, cv, U-Net, biodiversity, community ecology, forestry, invasive species**

I. Introduction

Invasive plant species are a significant problem in Maine, and their negative impact on the state's ecosystems, agriculture, and outdoor recreation demands urgent attention. Invasive plants pose a challenge in monitoring and mapping due to their rapid growth rate, ease of spread, and their ability to blend with native plant species, making it hard to identify and eradicate them [2]. As of 2019, the Maine Natural Areas Program has identified 125 species of plants "found to pose a threat to habitats and natural resources in Maine," [3]. One of these species, *Fallopia japonica*, is known to be an issue in the town of Cape Elizabeth, Maine based on the lead author's previous experience working for a local conservation organization.

Though *Fallopia japonica* does have culinary uses [4], it is primarily considered an environmental hazard. Per the Maine Cooperative Extension:

> "Japanese knotweed is a robust perennial herb that emerges early in the spring and forms dense thickets up to nine feet in height. Thickets may be so dense that virtually all other plant species are shaded out. Large colonies frequently exist as monocultures, reducing the diversity of plant species and significantly altering natural habitat," [5].

During spring (the time at which the imagery used in this project was collected), *Fallopia japonica* appears as large patches of dry, brown, "bamboo-like" stalks from the preceding season, with new stalks emerging from the base of the batch. See Figure 1 for an example taken in late April near the lead author's home.

Fig. 1: *Fallopia japonica* growing near the lead author's home. Photo taken in late April, roughly the same time of year that the aerial images used in this project were collected. The authors ate some of the green shoots shown in this photo and can attest that it tastes vaguely of rhubarb.

TABLE I: Datasets Generated with ArcGIS Pro's "Export Training Data for Deep Learning" Toolbox

Folder/Dataset Name	# of Images	Images with Target Features	Bands	Spatial Resolution	Data Type
Image_Chips_128_nostride_balanced_dem	155	100%	Red	0.075 m	uint8
			Green	0.075 m	uint8
			Blue	0.075 m	uint8
			DEM	0.075 m (interpolated)	float16*
Image_Chips_128_nostride_unbalanced_dem	5329	3%	"	"	"
Image_Chips_128_overlap_balanced_dem	640	100%	"	"	"
Image_Chips_128_overlap_unbalanced_dem	21,526	3%	"	"	"

*Normalized and converted to uint8 on a per-image basis during loading to maximize resolution

Based on the lead author's experience, aerial imagery is increasingly useful as a tool for conservation planning. The State of Maine collects and distributes a database of orthorectified aerial imagery for use by municipalities, NGOs, and private parties [6]. This project seeks to utilize this imagery resource to create an accurate segmentation of *Fallopia japonica* in Cape Elizabeth. The proposed solution will enable earlier detection, better mapping, and more effective control of invasive plants in Cape Elizabeth.

II. Related Work

This research is based on the U-Net architecture, which was originally developed as a method for segmentation on medical imaging [7]. Recent research has suggested that this architecture can effectively segment ortho-rectified aerial imagery to identify the areal coverage of plant communities [1].

However, for this technique to be broadly valuable to the environmental community, it needs to be replicable on existing bodies of image data. Similar research has been done using proprietary tools (see [8]). This project seems to achieve similar results using a more flexible, PyTorch-based network.

The code for this project is based on a fork of the "U-Net: Semantic segmentation with PyTorch" GitHub repository [9], which was originally developed for the Carvana image segmentation competition. The final repository is publicly available at [10].

III. Method

A. Data and Data Preprocessing

All source data was collected and preprocessed in ArcGIS Pro 3.1 using the Image Analyst extension. The orthoimagery layer was obtained from the Maine GeoLibrary and consists of 4-band (Red, Green, Blue, and Near Infrared), 8-bit unsigned integer pixels at 0.075m spatial resolution [6]. The digital elevation model (DEM) was obtained from Maine GeoLibrary and consists of 32-bit float pixels at 1.0m spatial resolution. The DEM dataset was upsampled to match the spatial resolution of the imagery using cubic interpolation. Finally, a vector polygon overlay of known *Fallopia japonica* patches was provided by the lead author based on previous experience.

The above imagery and labels were exported as GeoTIFF files using the Export Training Data for Deep Learning toolbox [11]. A summary of the datasets is shown in (see Table I). Each resulting image chip contains 128x128 pixels. Four datasets were created covering the choices of 50% or 0 overlap and "balanced" or "unbalanced" (with the former omitting any image chips that contain 0 pixels with the target class present).

B. Data Augmentation

Data augmentation is a common technique used in machine learning to increase the amount of data available, improve the diversity of the dataset, and expose the model to a wider range of image variations, which in turn helps the model to generalize better to new and unseen data. Moreover, data augmentation helps prevent overfitting by introducing variations in the training data, making it less likely for the model to memorize the training set. It also enhances the utilization of available data by creating new and unique data samples from the existing data. Finally, data augmentation improves the robustness of computer vision models by training on a wide range of augmented images, which helps the model to be more resilient to variations in lighting, orientation, and other factors that can affect image appearance in real-world scenarios. In our project, we multiplied the data using 4 different transformer functions:

1) Horizontal Flip
2) Vertical Flip
3) Random Crop
4) Everything above combined + random scale + shear

Due to the square nature of the dataset images, only horizontal and vertical flips are used to augment the data's rotations. For example, applying a 45-degree rotation would push some parts of the image out of the box and leave some parts of the image empty, thereby increasing the complexity of pre-processing steps.

The original images are 128x128, random crop takes a 64X64 snapshot of the image as the output. The location of the crop is random.

(a) Original Image

(b) Horizontal Flip

(c) Vertical Flip

(d) Random Crop

(e) all+random scale+ shearing

Fig. 2: Examples of Data Augmentations

Random scale zooms in on the image by a random factor between 1 and 1.5. Zooming out is not included, because it would have the same problem as arbitrary rotation where some parts of the image box would be empty.

Shearing distorts the dataset images by shifting their pixels along a certain direction, either horizontally or vertically, based on a certain angle or magnitude. The angle used in the project is 10-degrees.

The data augmentation step takes the input data in as a PyTorch Dataset, and applies custom transform function. The output of the data will be populated inside a separate directory.

C. Network architecture and loss functions

1) Network architecture: The U-Net architecture is semantic segmentation network and it starts with a double convolution layer using batch normalization and ReLU activation followed by a series of downsampling blocks. Each downsampling block consists of a convolution followed by a 2x2 maxpooling and doubles the number of channels while halving the width and height of the image. This process proceeds until a 1024-channel image is obtained.

The upsampling part of the network consists of a series of interpolation steps followed by convolution. Each block doubles the image resolution while dividing the number of channels by a factor of two. Additionally, the input to each upsampling block consists of the previous block concatenated with the output of the downsampling layer with the same resolution, allowing the upsampling blocks to utilize information from the original layers of the network. The symmetrical design of the network, as illustrated in Figure 3, is the shape of a "U", hence the name of the network.

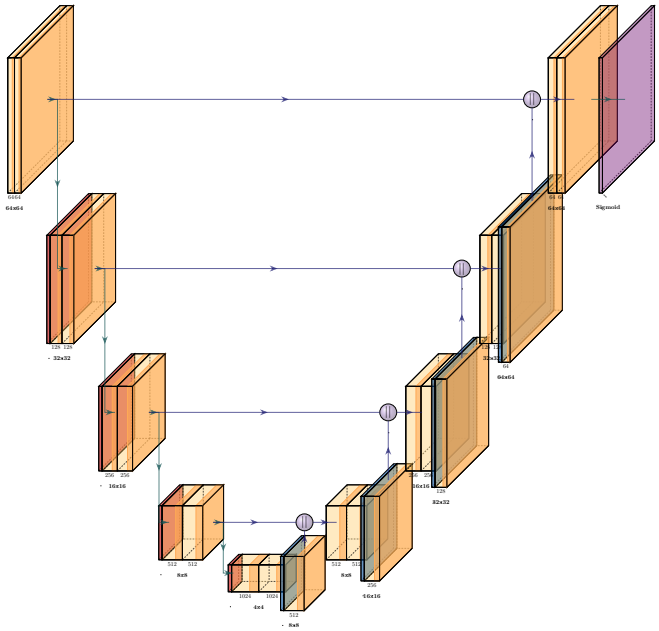

Fig. 3: U-Net Architecture

2) Loss functions: Two loss functions are back-propagated in the network. These loss functions are commonly used in both the deep learning and remote sensing communities, though they are sometimes known by different names[1].

Binary Cross Entropy with Logits Loss: This loss combines a *Sigmoid* layer and the *BCELoss* in one single class [13]. Binary cross entropy loss is the binary version of cross entropy loss that penalizes divergence of the probability distribution of predicted and that of true. Binary Cross Entropy with Logits Loss ℓ of probability distribution x and y is:

$$\ell(x, y) = mean(L) = mean\{l_1, l_2, \ldots, l_N\}^T \quad (1)$$

$$l_n = -w_n[y_n \log \delta(x_n) + (1 - y_n) \log(1 - \delta(x_n))] \quad (2)$$

where N is batch size and δ is *sigmoid*:

$$\delta(x) = \frac{1}{1 + exp(-x)}$$

Dice Coefficient Loss [1]: This loss penalize the dissimilarity of two samples. Calculated as:

$$DCL = 1 - \frac{2|X \cap Y|}{|X| + |Y|}$$

[1]For a detailed meta-analysis, see [12].

Thus, more similar the X and Y are, the less the loss will be. Vice versa, more different the X and Y are, the greater the loss will be.

Since *binary cross entropy with logits loss*(BCELL) can be very huge number and *dice coefficient loss*(DCL) is between 0 and 1, in the start of learning process BCELL will impact significantly and in the later phase, both will have about same impact.

IV. Experiments and Results

A. Hyperparameter Tuning

Because the U-Net model has a fairly fixed architecture, there are relatively few hyperparameters to tune. Specifically, for this project we considered values for learning rate (1e-1, 1e-2, 1e-3, 1e-4, and 1e-5) and batch size (16, 32, 64, 128). Each model was trained for 20 epochs with a 67/33 train-validation split. Each run took approximately 10 minutes on Google Colab with standard GPU (Nvidia Tesla T4). All runs were logged to WandB.

Fig. 4: Batch Size Tuning Results

Results from hyperparameter tuning are shown in Figures 4 and 5. Likely due to the fact that BCE loss is unbounded, the models with learning rates 1e-1 and 1e-2 failed to converge. Smaller learning rates resulted in similar validation set performance, though the rate of convergence varied slightly. Batch sizes 32, 64, and 128 outperformed batch size 16 on the validation set. Based on these results, the model with learning rate 1e-4 and batch size 32 was selected for application to the test set.

B. Test Set

Following hyperparameter tuning, we selected an optimal model and applied it to a test set consisting of 1530 image chips that were not included in the original data set. The test image, true mask, and predicted mask are shown in Figure 6.

The predicted mask captures nearly all of the pixels identified as *Fallopia japonica* in the true mask (low type II error) but also identifies many other regions erroneously (high type I error). The full performance on a pixel-by-pixel basis is summarized in a confusion matrix (Figure 7). Test statistics are summarized in Table II.

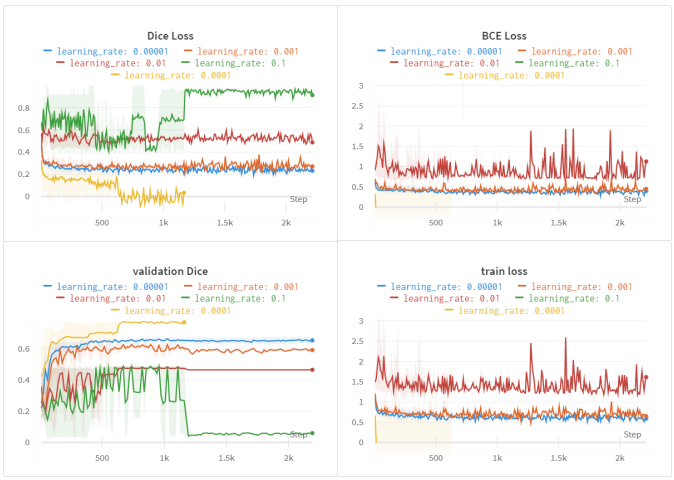

Fig. 5: Learning Rate Tuning Results

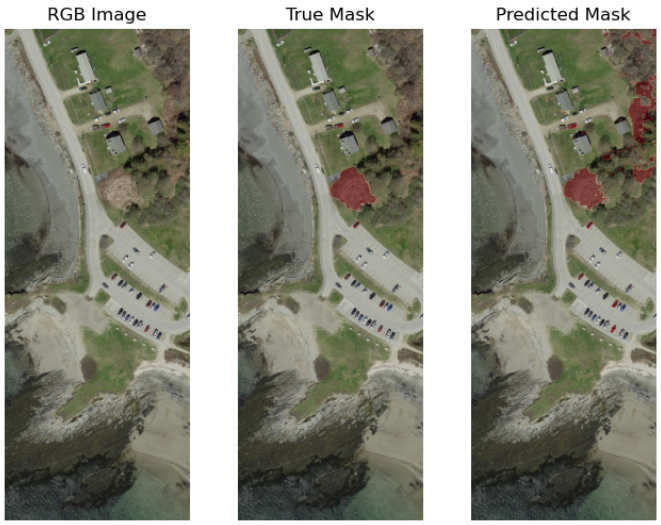

Fig. 6: Test Set Image (RGB bands shown) with True and Predicted Masks

TABLE II: Statistical Results from Test Set

	precision	recall	f1-score	support
Absent	0.998	0.982	0.990	6,209,478
Present	0.387	0.857	0.533	81,978
accuracy	0.980	0.980	0.980	0.980
macro avg	0.693	0.919	0.762	6,291,456
weighted avg	0.990	0.980	0.984	6,291,456

C. Alternative Band Combinations

Using the best hyperparameter combination, we also tested the following alternative band combinations:

1) Removing the DEM band (R/G/B)

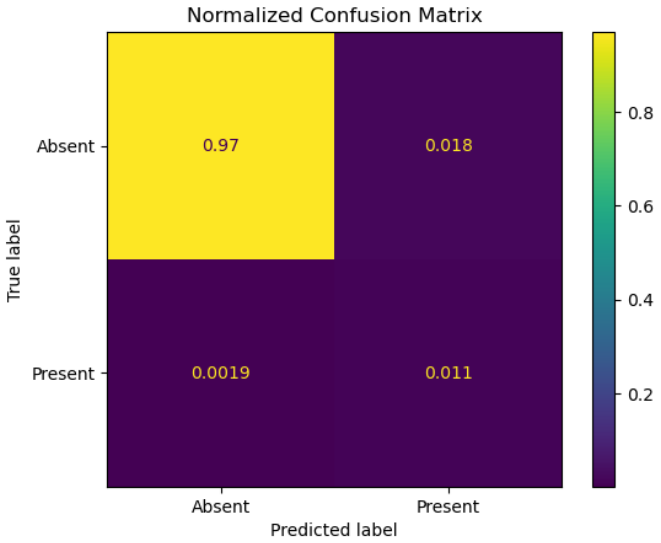

Fig. 7: Confusion Matrix for the Test Set

2) Adding the NIR band (R/G/B/NIR/DEM)

Both runs resulted in similar performance to the original run, suggesting that the RGB bands contain most of the information necessary to train the model. However, the nature of the error varied. In particular, the model without the DEM introduced an area of false positive pixels along a rocky outcropping (see Figure 8). It is possible that the DEM band helps to eliminate this error in the full model.

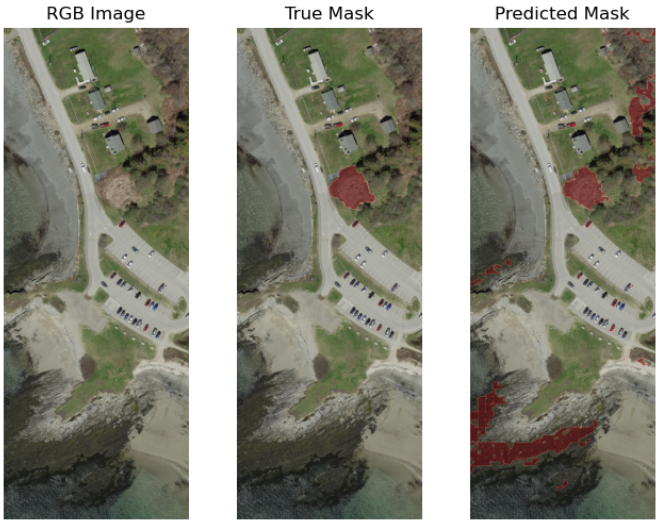

Fig. 8: Results from Model Trained with RGB Only (No DEM) Showing Region of False Positives

V. Discussion and Summary

The results of this project demonstrate the potential for applying CNNs to the problem of invasive species detection on existing orthoimagery datasets. The recall metric (85.7%) means that this network is capable of classifying 85.7% of pixels containing *Fallopia japonica* successfully. However, the precision metric (38.7%) indicates a significant number of false positives. In the context of invasive species detection, false positives are preferred over false negatives; thus, this tradeoff may be acceptable, but improved performance is still desired.

Based on visual analysis of the test set results, we propose the following areas for further study:

1) The ”noisiness” of the predicted image seems to suggest an opportunity for a morphological filtering step to eliminate small patches of false negatives. This idea is backed up by domain knowledge, specifically the fact that *Fallopia japonica* grows in dense patches.
2) The predicted mask shows clear artifacts related to the boundaries of the image chips used to create the mask. This could be addressed by modifying the procedure for generating predicted masks, such as using an overlapping stride to generate the image chips followed by an AND operation.
3) This project separated the image chip generation, data augmentation, and training set preparation steps, primarily for convenience. Incorporating data augmentation directly into the PyTorch pipeline could simplify the project and reduce the amount of space needed for image storage. Additionally, developing a data loader that can generate chips on-the-fly from a single, high-resolution input raster would simplify both training and prediction stages.

In addition to the above performance-oriented goals, this project could be developed to eliminate the dependency on proprietary software. The open-source QGIS software package includes most if not all of the functionality utilized in ArcGIS. Additionally, python-based tools like rasterio could provide an alternative method of retrieving and preprocessing raster imagery, allowing for automation of the full pipeline.

References

[1] T. Kattenborn, J. Eichel, and F. E. Fassnacht, “Convolutional Neural Networks enable efficient, accurate and fine-grained segmentation of plant species and communities from high-resolution UAV imagery,” Sci Rep, vol. 9, no. 1, Art. no. 1, Nov. 2019, doi: 10.1038/s41598-019-53797-9.

[2] Maine Department of Agriculture, Conservation, and Forestry, “Invasive Plants,” Maine Natural Areas Program. https://www.maine.gov/dacf/mnap/features/invasive_plants/invasives.htm (accessed Apr. 22, 2023).

[3] Maine Department of Agriculture, Conservation, and Forestry, “Maine Advisory List of Invasive Plants - 2019 Revision,” Maine Natural Areas Program, 2019. Accessed: Apr. 22, 2023. [Online]. Available: https://www.maine.gov/dacf/mnap/features/invasive_plants/2019advisorylist_sciname.pdf

[4] C. Nast, “Meet the Massively Destructive Garden Weed That ‘Tastes Like Rain,’” Bon Appétit, May 31, 2016. https://www.bonappetit.com/test-kitchen/ingredients/article/japanese-knotweed-recipes (accessed Apr. 26, 2023).

[5] “Bulletin #2511, Maine Invasive Plants: Japanese Knotweed/Mexican Bamboo, Fallopia japonica, Synonym: Polygonum cuspidatum (Smartweed Family) - Cooperative Extension Publications - University of Maine Cooperative Extension,” Cooperative Extension Publications. https://extension.umaine.edu/publications/2511e/ (accessed Apr. 26, 2023).

[6] Maine Office of GIS, “Maine Orthoimagery Regional 2021.” Dec. 27, 2021. [Online]. Available: https://hub.arcgis.com/datasets/maine::maine-orthoimagery-regional-2021-imagery-layer/about

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[8] A. Abd-ELrahman, K. Britt, and T. Liu, "Deep Learning Classification of High-Resolution Drone Images Using the ArcGIS Pro Software," Oct. 14, 2021. https://edis.ifas.ufl.edu/publication/FR444 (accessed Apr. 22, 2023).

[9] A. Milesi, "U-Net: Semantic segmentation with PyTorch." Apr. 25, 2023. Accessed: Apr. 25, 2023. [Online]. Available: https://github.com/milesial/Pytorch-UNet

[10] P. E. Mathieu, G. J. Kim, and H. S. Ning, "unet-orthoimagery." Apr. 14, 2023. Accessed: Apr. 22, 2023. [Online]. Available: https://github.com/PhilipMathieu/unet-orthoimagery

[11] ESRI, "Export Training Data For Deep Learning," ArcGIS Pro — Documentation. https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/export-training-data-for-deep-learning.htm (accessed Apr. 22, 2023).

[12] A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review," Remote Sensing, vol. 13, no. 13, Art. no. 13, Jan. 2021, doi: 10.3390/rs13132450.

[13] "BCEWithLogitsLoss," PyTorch 2.0 Documentation. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html (accessed Apr. 22, 2023).