Sensing Brown Tail Caterpillar Infestations Using RGB to NIR Image-to-Image Translation

Computer Science: AI Henry Landay Advisor: Tahiya Chowdhury

Research Goal

- Develop a computer vision pipeline to detect Brown Tail Moth (BTM) infestations using drone footage by performing image processing, data analysis, training an AI model, and mapping the detected infestations.

Background

- Brown Tail Moths (BTM) are an invasive species.
- Larval BTM excessively feed on the foliage of trees causing reduction of growth and occasional death of valued trees.
- BTM caterpillars create distinctive winter webs in which they overwinter; these are what we hope to identify using computer vision. [10]
- Cities spend hundreds of thousands of dollars on manual identification methods; using AI to automate this process using can save cities vast amounts of money and resources

Prior Work

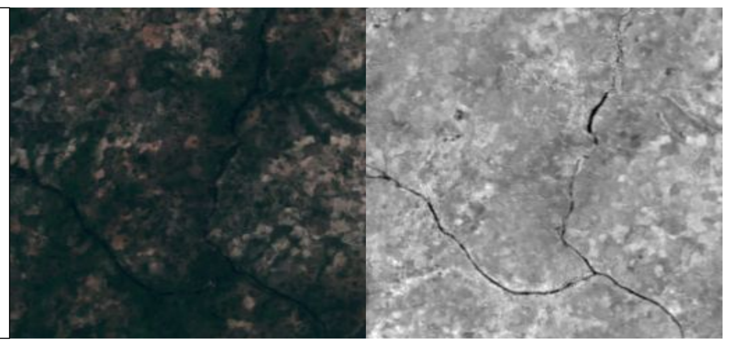
- Use of regular RGB images and manually labeled images to develop an object classification neural network proved infeasible because of small the overwintering pods are and how well they blend-in to the image background.

The Data

1) Drone Captured Images
   - Set of 5,000 drone captured images; metadata (latitude, longitude, etc.) from images extracted into a CSV.
2) SEN12MS Satellite Images
   - Open source library of over 180,000 RGB-NIR paired images.

Image coordinates mapped using Folium Python library.

RGB-NIR paired images



Methods

- Use "pix2pix" Generative Adversarial Network (GAN) to conduct RGB to Near-Infrared (NIR) image translation.
- By simulating the near NIR channel we hope to make the overwintering pods of the BTMs more visible to a neural network.
- Train on SEN12MS RGB-NIR paired images, conduct RGB-to-NIR image translation on drone captured images.
- Generator: UNet Architecture (Fig. 1)
- Discriminator: PatchGAN Architecture (Fig. 2)

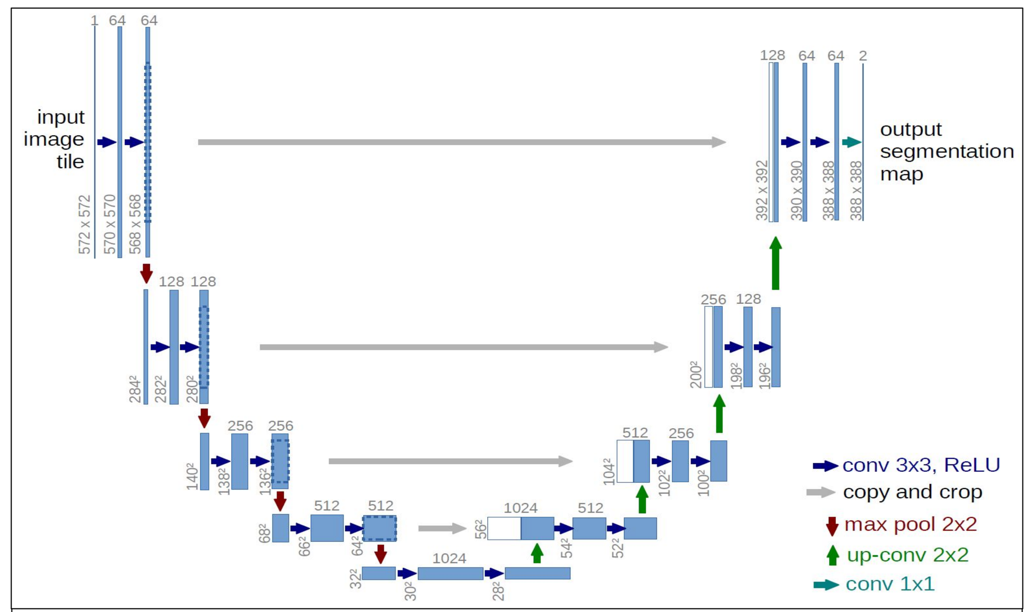


Figure 1: U-net generator architecture. Blue boxes correspond to a multi-channel feature map. Number of channels is denoted on top of the box. The x-y-size is at the lower left edge of the box. [15]

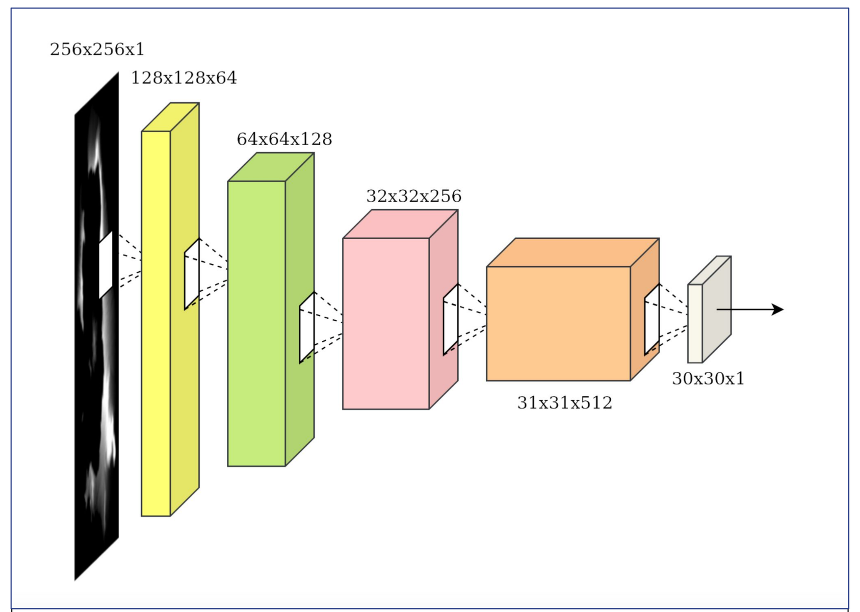


Figure 2: PatchGAN discriminator architecture. Classifies each N x N patch in an image as real or fake. Predicts overall output (real or fake) based on average of all responses. [7]

Results

- Generator, trained against a discriminator for 420 epochs, on drone-captured aerial images produced output images in *Figure 3*
- *Figure 3* shows the generator does not do a very good job at simulating the NIR channel using the drone captured images.
- Possible reasons that the model does not perform well are 1) the data it was trained is too dissimilar from the data it is being used for and 2) the model produces images in the RGB colorspace instead of in grayscale

Future Work

- Train for longer/more epochs (was restricted by Google Colab GPU free-usage limits).
- Alter the GAN model to predict images in grayscale instead of RGB.
- Quantify how well the NIR channel is simulated for the drone captured images (difficult because there is no NIR ground truth).
- Augment the drone images to look more like training input images (SEN12MS images).

Input Image	Output Image

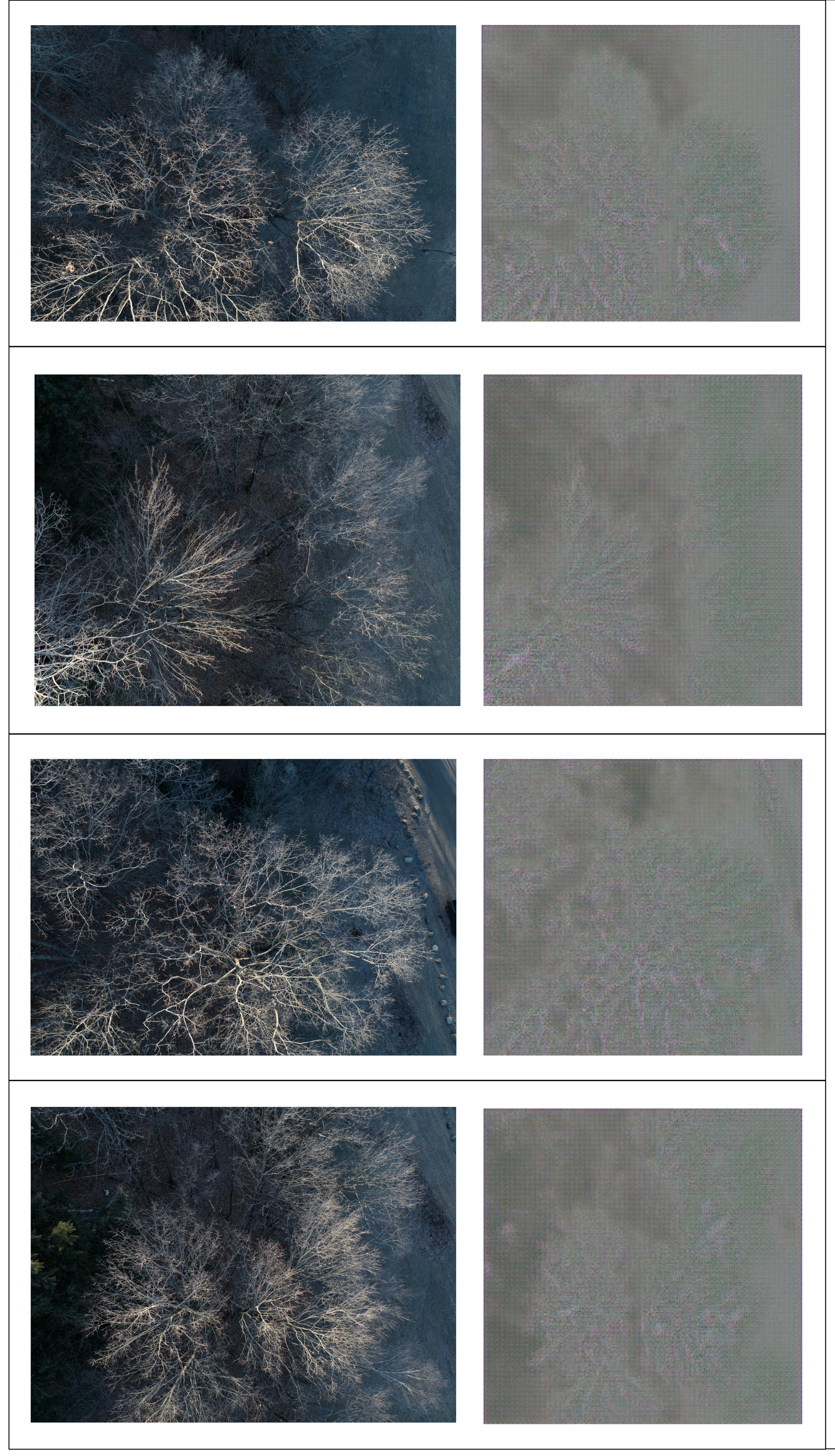


Figure 3: Series of images showing the output (right) of the generator after being trained on the SEN12MS dataset for 420 epochs with the input (left) being the drone captured images. One can clearly observe the outlines of the tree branches in these images. However, beyond the outlines, the simulated Near-Infrared images provide little additional information.

Input Images:
1. 2022-34
2. 2022-79
3. 2022-99
4. 2022-134

Acknowledgments

- Special thanks to Tahiya Chowdhury for her support and guidance as well as Tom Klepach for his work collecting data and advising the project. Thank you to everyone at DavisAI who has provided support and guidance on this project.

References