VA Plant Image Classification

A Preprint

Huilin Chang
School of Data Science
University of Virginia
Charlottesville, VA 22904
hc5hq@virginia.edu

Yihnew Eshetu
School of Data Science
University of Virginia
Charlottesville, VA 22904
yte9pc@virginia.edu

Celeste Lemrow
School of Data Science
University of Virginia
Charlottesville, VA 22904
ctl7t@virginia.edu

March 27, 2021

Proposal

Overview

Virginia has more than 3,000 square miles of water [16], including lakes, rivers, ponds, streams, and the Chesapeake Bay. The quality, health, and environmental ecosystems of those bodies of water are under threat from invasive aquatic plant species, which can choke out native species and prevent them from thriving, spur harmful changes in the water's chemistry, negatively impact the development of fish and aquatic organism populations, and make the water less hospitable for recreational use [4]. One of the most common invasive aquatic species in Virginia is hydrilla [14], a vine-like plant with small, whorled groups of leaves. Hydrilla was first discovered in the area in 1982, in the Potomac River [15], and it is frequently found today in the Chesapeake Bay and a number of rivers, lakes, and ponds throughout the state. The objective of our project is to create a deep learning image classification model that accurately identifies hydrilla plants, compared with other types of submergent aquatic plants that are not invasive species but have similar appearance, to help identify which plants should be removed from a body of water as part of a conservation effort. Tracking and monitoring these species is labor intensive and can involve considerable expense; a deep learning model can reduce the time and labor needed to accurately identify where hydrilla is overtaking a body of water, to better support more targeted treatment of the problem and more efficient allocation of limited water conservation resources. The goal is to scale up and scale out an algorithm to do what would normally require a labor force of considerable size. While the present focus of the study is Virginia, the algorithm could be used to benefit other areas of North America where hydrilla is also a problem.

Dataset

We will start building the dataset using images from forestryimages.org and invasive.org, which are part of the Bugwood Images archive, a project of the University of Georgia, Center for Invasive Species and Ecosystem Health. The archive contains more than 300,000 images of trees, plants, invasive species, insects, wildlife and other types of natural resources, available for free for educational purposes [2]. The Bugwood Images archive contains just over 200 images of hydrilla alone [3], and more than 2,000 images of aquatic plants, including watercress and coontail, two Virginia native species that we are considering as a comparison class, given their similar appearance. We are also investigating image data sources for waterweed, another Virginia native species that is considered similar in appearance to hydrilla. We will expand the dataset using other publicly-available images of hydrilla and other aquatic plants as needed, such as from Shutterstock and Google Images, once we conduct a thorough review of the images in the Bugwood Archive. Our initial investigation of image availability overall suggests that we will be able to construct a dataset of sufficient size, with an adequate number of images in each class (hydrilla/ invasive species, not hydrilla/ native species). The images in the archive are of variable dimensions, so we will need to resize as needed as part of data preparation. The images, however, do have different backgrounds, settings, angles, and other distinct qualities that should provide diversity and "noise" among the images to help with training the model. We may also need to conduct data augmentation as part of building the dataset.

Related Work

Given that our proposed topic is different from what we initially discussed in our literature review, we will need to continue to build that out with additional references as we move forward with the project. To start, we have reviewed a paper that uses CNNs to identify an invasive species, smooth cordgrass, in the Yellow River Delta in China [5]. The authors state that their work would help protect and manage the coastal wetland ecosystem by early detection of Spartina Alterniflora. This particular study used high-resolution satellite images of the Yellow River's wetland areas from 2012 to 2019. However, they preferred images collected from July to September, as there was less cloud cover and minimal influence of tides. Once the authors had their images processed, they examined two image super-resolution models: super-Resolution Convolutional Neural Networks (SRCNN) and Fast Super-Resolution Convolutional Neural Networks (FSRCNN) [5]. After thoroughly studying each model's benefits, the authors implemented both SRCNN and FSRCNN and discussed the limitations of their data regarding each model. In conclusion, the authors determined that FSRCNN was more effective and efficient at estimating a small size of Spartina Alterniflora.

Intended Experiments

The experiment's goal is to identify invasive species in Virginia wetlands using deep learning image classification techniques. We hope to use deep learning to accurately determine if invasive species exist in an image, thus helping biologists protect Virginia's ecosystem.

One of our experiment's first steps will be downloading images of hydrilla and other aquatic non-invasive species plants from the forestry site. We will also expand our dataset by downloading additional images from publicly available images. Once we have downloaded all of the images, we plan on saving them on a shared google drive folder. Using a shared drive will allow each individual in our group to have access to the images in Google Colab. We plan on ensuring the dataset is balanced between the two classes. We also plan on saving images into two main directories: invasive and non-invasive species. We will randomly place images into the training, validation, and test folder within these two directories. We plan to structure our dataset like this because we plan to use Keras image data preprocessing flow function, flow_from_directory. This function takes in a path to a directory and generates batches of augmented data. The function allows the resizing of images to a default size of our choosing and converts the images to grayscale, RGB, and RGBA. We also plan on conducting different types of position data augmentation, such as cropping and rotating. We believe these techniques will allow us to increase the size of our dataset. We also plan on normalizing the data.

Convolutional neural networks (CNN) are deep learning neural networks that have had great success solving image recognition problems. A CNN consists of convolutional layers, pooling layers, and dense layers [8]. Convolutional layers apply a convolution operation on an input matrix using a nxn convolutional filter matrix. In our experiment, we will need to determine the optimal kernel size for our convolutional filter. Pooling is a technique that reduces the matrix created by the convolutional later. Often Max Pooling and Average Pooling are used for pooling. Again, we need to explore these two options in our project to determine if we should use one technique or a combination of both in our network. We will also explore and study various CNN architectures like LeNet and GoogLeNet and learn commonly-used practices in developing CNN models.

Our input dimension will consist of a $nxnx3$ input size. Input size of 256x256 is most commonly preferred, but we will explore other input sizes. We also understand we will need to conduct multiple trials to determine the number of hidden layers and nodes within a hidden layer. Since we will be using a CNN model, it is recommended to use ReLU as the activation function. However, we will use Softmax at the last layer of the network to interpret the two classes' probability because we are solving a binary classification problem. Furthermore, we will also focus on tuning and optimizing our deep learning model. This includes but is not limited to using different optimizers (SGD, Adam, etc.), using a predetermined piecewise learning rate, and comparing loss versus iteration epoch. We can also explore batch normalization and using regularization techniques such as early stopping and dropout.

Model Design and Implementation

We downloaded data from multiple sources to create our project dataset: invasive.org, Google, gbif.org, and Shutterstock; invasive.org and gbif.org provide open source image data of plants. Our data set consists of 450 images with five types of plants: Duckweed (98), Watercress (100), Arrowhead (76), Grassy Mud Plantain (75), and Hydrilla (101). As discussed previously, these are aquatic plants native to Virginia; Hydrilla is an invasive plant, and the others are indigenous and do not cause harm to the water ecosystem. The purpose of our project is to analyze and identify non-invasive and invasive plant species using this dataset. Although this was originally a binary problem, we altered our research question towards a multi-classification approach, using one invasive species class (Hydrilla) and four non-invasive species classes (Duckweed, Watercress, Arrowhead, and Grassy Mud Plantain). Once we created our data set, we uploaded the dataset to Google Drive and mounted our drive to Google Colab. Next, we created a function that takes in image height, width, and batch size as parameters and returns a training, validation, and test set using the image_dataset_from_directory function. We split our initial dataset into 80 % training, 10 % validation, and 10 % test. We then visualized our images to ensure that our images were loaded correctly. Due to the size of our dataset, we implemented an image augmentation step in our process. Using Keras experimental preprocessing, we added steps to flip, rotate, contrast, and zoom images randomly. Then we tested our image augmentation layer to ensure that original images are getting augmented. For our initial model implementation, we focused on transfer learning, in order to maximize classification accuracy, and designed one initial model with our own architecture. We experimented with four types of transfer learning models – Xception, EfficientNet, DenseNet, and VGG – with demonstrated varying levels of performance. In all cases, we used the Softmax function as the activation function in the output layer, given that we are focusing on a classification problem. Before we downloaded each of the model architectures, we created an input layer with set image width and height. Then we passed the input layer into a Keras data augmentation sequential layer. Using the Keras Application, we instantiated each architecture using the ImageNet weights. We ensured the design did not include the top layers and passed the data augmentation sequential result into the architecture. Afterward, we turned on all layers of the architecture for training. We added a GlobalAveragePooling2D layer to output the feature maps spatial average. Within some of the models, we experimented with a Batch Normalization layer following the Global Average Pooling layer to standardize the inputs and aim to reduce the number of training epochs. We also experimented with a dropout layer in some of the models. For each model, we compiled using a sparse categorical cross-entropy loss function, accuracy as the metric, and tried different optimizers and learning rates.

Xception

Xception merges the ideas of GoogLeNet and ResNet, claiming the usage of fewer parameters, less memory and fewer computations than a regular convolution layer, yet with better performance. We experimented with two configurations to solve our image classification question. The first approach adopted an Adam optimizer with a learning rate of 0.01. BatchNormalization was adopted and serves as an inference mode. The model accuracy from the training data is 99% compared to 85% for the validation dataset. The second configuration included a stochastic gradient descent (SGD) optimizer, a momentum of 0.9, a learning rate of 0.01, and a decay rate of 0.001. BatchNormalization was also adopted and serves as inference mode. The model accuracy from the training data is 99%, while the validation dataset accuracy is 94%. The model accuracy improved with the SGD optimizer compared to Adam.

EfficientNetB6

Most common CNN models are developed on fixed resources and can be scaled up for better accuracy if there are more resources. However, there are cases where there are limitations in resources, and that is where EfficientNet comes into play. Using EfficientNet, we can uniformly scale a network's depth, width, and input resolution using a compound coefficient. We picked EfficientNet because it has shown to be more accurate and efficient than past CNNs under resource constraints.For our model experimentation, we decided to try EfficientNetB6 because it had a higher accuracy on the Imagenet dataset than the other EfficientNet models and fewer parameters than EfficientNetB7. We added a BatchNormalization layer to standardize the inputs and reduce the number of training epochs, following the GlobalAveragePooling2D layer. We included a dropout layer and a dense softmax layer with five classes. Once the model was designed, we compiled our model using a sparse categorical cross-entropy loss function, accuracy as the metric, and looked at three different optimizers, all with a learning rate of 0.01: SGD, NAdam, and RMSProp. We fit all models using an epochs value of ten. The model with SGD as the optimizer had a validation accuracy of 78.8%, while the training accuracy was 95%. This result clearly shows that the model is overfitting, and we can address this issue in our further model development by adding more images, modifying our image augmentation layers, and adding more regularization. The model with NAdam as the optimizer had a validation accuracy of 16.6%, while the training accuracy was 33.3%. The model with RMSProp as the optimizer had a validation accuracy of 20%, while the training accuracy was 33.8%. One significant difference between SGD and the other optimizers was the training time. The SGD took 25 seconds per epoch, while RMSProp and NAdam took 120 seconds per epoch. We also noticed that the RMSProp and NAdam models' validation accuracy fluctuates significantly between epochs, and the accuracy of the model is extremely low. We plan on examining these results further to assess what may be causing a dramatic difference between SGD and the other optimizers. We plan to tune the learning rate and other arguments to see if that impacts the accuracy. We also plan to incorporate EarlyStopping to stop training when the accuracy stops improving in the future. For the EfficientNet models we did not evaluate the models against the testing set, we plan on including that in our future model design.

DenseNet121

Developed to address the impact of the vanishing gradient problem on accuracy, DenseNet121 has 121 layers, most of which are packaged within four "dense blocks" of layers in the architecture. We used an SGD optimizer with a learning rate of 0.2 for the first pass with the base layer frozen, and a momentum parameter of 0.9 and a decay of 0.01. In the second pass with all layers made trainable, we used Adam as the optimizer with a learning rate of 0.001. The model demonstrated 84.4% accuracy on the test set. We used early stopping, which stopped the training process after 36 epochs.

VGG19

VGG19 has 19 layers and focused on the use of 3 x 3 fixed-size kernels within increasingly deep layers in the architecture. We used an SGD optimizer with a learning rate of 0.2 for the first pass with the base layer frozen, and a momentum parameter of 0.9 and a decay of 0.01. In the second pass with all layers made trainable, we used Adam as the optimizer with a learning rate of 0.001. The model demonstrated 75.5% accuracy on the validation set in the first pass, but early stopping engaged after only 12 epochs in the second pass, with a test set accuracy of 26.6%. The low performance on the test set was not a surprise, given the indications in the training epochs that did run. We did some hyperparameter tuning, which did not improve performance. We plan to assess the results further and conduct further hyperparameter tuning to see if performance using this model improves.

Custom Neural Network

We used a custom neural network in our image classification approach. The motivation for utilizing CNN is to compare the performance with pretrained models. In addition, creating a convolutional neural network could be treated as a baseline at the first step. A CNN convolves learned features with input data and uses 2D convolutional layers. The architecture is composed of two main blocks. The first block acts as a feature extractor, with which we filter the feature maps obtained with new kernels, which gives us new feature maps to normalize and resize, and we can filter again, repeat the process, and so on. Finally, the values of the last feature maps are concatenated into a vector. The second block at the end of all the neural networks is used for classifications. The Softmax function was adopted as an activation function which is suited for our multi-class classification question. We noticed that a dropout layer is necessary to prevent overfitting issues. One dropout layer was adopted. When the dropout layer was not adopted, the model accuracy for the training dataset was 99%, but the accuracy for the validation dataset was 85%, with a gap of 14 percentage points. The gap between the accuracy of training and validation datasets narrowed when a dropout layer was included, with the former being 95% and the latter being 82%, with a difference of 13 percentage points.

Literature Review

Per discussion with the professor and TA that we do not need to re-submit the literature review, given our topic change, we will plan to update this section for the final paper.

Overview

As both global population growth and effects of climate change increase, the field of agriculture research faces an increasing need for analytic approaches that support increased yields, improved planting and harvesting efficiency, and mitigation of pests and diseases that negatively impact food production across all regions of the world. In addition, with the increasing globalization of food production and food distribution, there is a need for greater technological efficiency in the monitoring and study of crop production to provide more direct and comprehensive support to farmers in the field at a faster pace and lower labor cost. Several of these identified research needs align with the comparative advantages of deep learning. Opportunities to leverage computer vision to study crop yields and diseases and pests, quickly synthesize larger datasets of satellite imagery to assess plant growth, and predict production volume and potential environmental condition impacts to production all can lend themselves to deep learning techniques. This literature review presents a brief synthesis of a corpus of papers we curated that focus on the use of deep learning for precision agriculture, pest and disease identification, yield prediction, and environmental prediction, in support of our intended research focus on deep learning for agricultural applications

Deep Learning in Agriculture

Overall, the set of references we surveyed demonstrated successful implementations of deep learning techniques, including: focusing on image classification related to positive or negative pest and disease identification [10, 13, 18] in support of maximizing crop production; prediction of environmental conditions to support decision-making about planting and cultivation technique adjustments [6], and computer vision to support the use of robotics to automate agricultural labor tasks, such as harvesting or weed removal [9]. Some studies relied exclusively on deep learning algorithms for classification [7]; others utilized a hybrid approach, turning to deep learning for advanced feature extraction and then using a supervised machine learning algorithm, such as support vector machines, for the classification phase [13]. The surveyed studies provided useful insights to support initial parameter selection and optimization approaches for models, specifying details such as activation function, number of layers, and use of pooling layers [1, 7], which can be leveraged in the design and adjustment of models in future studies. Many of the studies we examined mention the importance of resizing images to impact the model. From one study to another, we noticed unique techniques were used for image manipulation. Furthermore, certain studies built their own neural networks while others used existing architecture such as LeNet as a foundation for their network.

Data

Several of the studies discussed relevant gaps and considerations specific to applying deep learning to the agriculture sectors. One issue of note, for example, is the lack of datasets of sufficient size, labeling, and image quality; larger and more diverse image sets are needed to support the range of potential research studies [9]. In the case of lack of data many studies used data augmentation techniques such as scaling, cropping, and brightening to increase their dataset [17] Useful references to available datasets, however, were provided and discussed in one study, which we plan to use as part of our dataset review and selection [9].

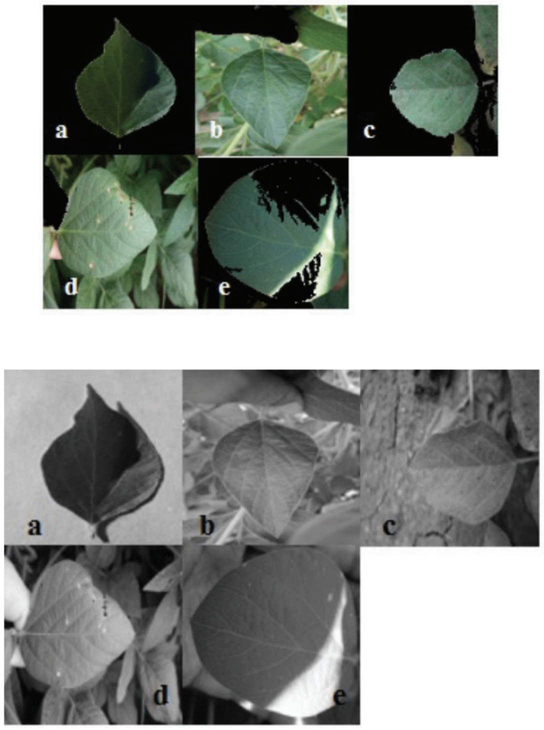

Figure 1: (**Top**): Resized Color Images (**Bottom**): Resized Grayscale Images [17]

One study explores the impact resizing has on images that are originally of different resolution. As seen in Figure 1, in order to prevent any bias, this particular study decided to resize all images to 128x128 pixels [17]. Furthermore, since many of the images were taken under different lighting conditions, they decided to create a separate dataset containing grayscale images to reduce noise, Figure 1. Using the two datasets, the test classification accuracy was compared.

Data Pipeline

In addition, we noted some useful discussion about data pipeline development to consider when dealing with satellite imagery, in particular, such as spectral normalization, solar correction, and atmospheric correction. We also found useful assessment of feature engineering approaches for different types of image information, such as spectral, spatial, and temporal information [11]. While these studies provided useful initial insights we can leverage in both data selection and data modeling and analysis, they also highlighted that there is room for growth in understanding about how these considerations can be studied further to enhance these types of deep learning models.

Figure 2, depicts a detailed view of the data pipeline architecture used in one study [1]. In this pipeline we see steps such as feature extraction, data augmentation, and image analysis.

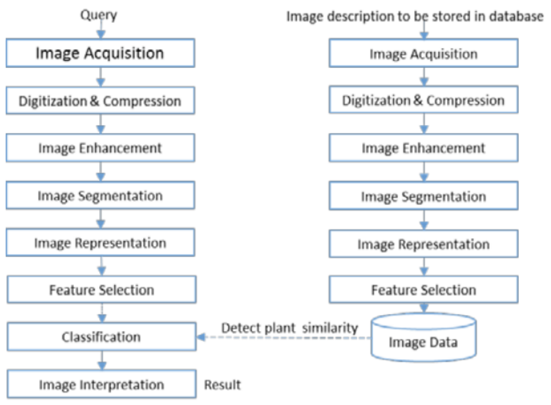

Figure 2: Detailed Illustration of an Image Pipeline [1]

Image Processing

Furthermore, there seem to be gaps in the literature regarding broader issues of image quality and image processing and engineering, and how those parameters impact both data processing decisions and performance and quality of models. Further investigation on the impact on analysis of different types of images, such as close-up photos compared with satellite imagery, may be warranted. While some papers discussed the merits and concerns of a particular type of imagery of interest, we did not find a lot of comparative discussion across image types, and how that may impact the development of a model and its results. This may be due to the particular set of references we have reviewed so far for this project; we will continue to look at this as part of additional background research.

In the specific area of deep learning for agricultural disease recognition, we found discussion of some particular limitations, including consideration of whether there is a large enough labeled dataset, which is important to prevent or limit overfitting. In addition, more complex models may rely such a large number of features and become so specialized to a particular disease classification or dataset, it's external validity may be very limited, which, in turn, raises the issue of high computational costs and effort to train new models from scratch for multiple contexts[18].

Conclusion

Overall, this literature review proved useful in helping us identify and formulate initial research questions for consideration as we move forward with dataset selection and full proposal development for our project. Based on the key themes and findings from the work we reviewed, some potential questions include:

Research Questions

- How can we use automation and artificial intelligence methods to solve traditional farming problems, given agriculture's significant role in the global economy?
- How can we leverage existing deep learning models and architecture to help solve issues in agriculture?
- Based on current research, are there areas of agricultural research where deep learning approaches have not been implemented, but have potential to be used given potential common characteristics and linkages with existing studies?
- Would the merging of datasets possibly help improve certain deep learning agriculture models? If so, what approaches would be best for doing that?
- How can we improve upon the current limits of highly-specialized models and limited external validity in studies focusing on deep learning for pest and disease identification to support more broadly applicable research?

As we move forward with data selection and research question development, and consider which sub-area to focus on, our assessment is that there is currently more completed research work regarding identification of plant disease and pests, although we noted that there seems to be more of a challenge in moving beyond just classification of a diseased plant, and determining the specific type or cause of the disease infestation. Work related to determination of yield, prediction of yield, and identification and prediction of environmental conditions appears to be more sparse. In addition, linked to the paucity of available image data, there is also space to generate classification models for additional species and types of crops. A key next step will be honing in further on a dataset that meets necessary criteria for deep learning in conjunction with determination of a more specific research focus that supports a Virginia-based need. Based on our review of datasets so far, we have identified several promising image collections with linkages to prominent Virginia crops, such as apples, that would provide utility for state-focused precision agriculture issues.

References

[1] Abdullahi, Halimatu Sadiyah, and Ray E. Sheriff. "Convolution Neural Network in Precision Agriculture for Plant Image Recognition and Classification." *International Conference on Innovative Computing Technology 2021. IEEE Xplore.* Web. 26 Feb. 2021.

[2] Center for Invasive Species and Ecosystem Health. "About Forestry Images" *https://www.forestryimages.org/about/*

[3] Center for Invasive Species and Ecosystem Health. "Hydrilla" *https://www.invasive.org/browse/subthumb.cfm?sub=3028*

[4] Center for Invasive Species and Ecosystem Health "I am a fisherman/boater. Why should I care about invasive species?" *https://www.invasive.org/101/FishermanBoater.cfm*

[5] Chen, Mengmeng, Yinghai Ke, Junhong Bai, Peng Li, Mingyuan Lyu, Zhaoning Gong, and Demin Zhou. "Monitoring Early Stage Invasion of Exotic Spartina Alterniflora Using Deep-learning Super-resolution Techniques Based On Multisource High-resolution Satellite Imagery: A Case Study in the Yellow River Delta, China." *International Journal of Applied Earth Observation and Geoinformation.* 20 June 2020. Web. 13 Mar. 2021.

[6] Chen, Shuchang, Bingchan Li, Jie Cao, and Bo Mao. "Research on Agricultural Environment Prediction Based on Deep Learning." *Procedia Computer Science.* Elsevier, 18 Oct. 2018. Web. 26 Feb. 2021.

[7] Dahikar, Snehal and Rode, Sandeep. "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach." *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering.* January 2014. Web. 26 Feb. 2021.

[8] Google Machine Learning. "Convolutional Neural Network definition, Google Machine Learning Glossary." *https://developers.google.com/machine-learning/glossary*

[9] Lu, Yuzhen, and Sierra Young. "A Survey of Public Datasets for Computer Vision Tasks in Precision Agriculture." *Computers and Electronics in Agriculture.* Elsevier, 08 Sept. 2020. Web. 26 Feb. 2021.

[10] Nasir, Inzamam Mashood, Bibi, Asima, Shah, Jamal Hussain, Khan, Muhammad Attique. "Deep Learning-Based Classification of Fruit Diseases: An Application for Precision Agriculture." *Computers, Materials & Continua.* Web. 26 Feb. 2021.

[11] Nguyen, Thanh Tam, Thanh Dat Hoang, Minh Tam Pham, Tuyet Trinh Vu, Thanh Hung Nguyen, Quyet-Thang Huynh, and Jun Jo. "Monitoring Agriculture Areas with Satellite Images and Deep Learning." *Applied Soft Computing.* Elsevier, 23 July 2020. Web. 26 Feb. 2021.

[12] Tahir, Muqadas Bin, Muhammad Attique Khan, Kashif Javed, Seifedine Kadry, Yu-Dong Zhang, Tallha Akram, and Muhammad Nazir. "Recognition of Apple Leaf Diseases Using Deep Learning and Variances-Controlled Features Reduction." *Microprocessors and Microsystems.* Elsevier, 15 Jan. 2021. Web. 26 Feb. 2021.

[13] Tombe, Ronald. "Computer Vision for Smart Farming and Sustainable Agriculture." *2020 IST-Africa Conference (IST-Africa) IEEE Xplore.* Web. 26 Feb. 2021.

[14] Virginia Department of Conservation and Recreation. "Virginia Invasive Plant Species List" *https://www.dcr.virginia.gov/natural-heritage/document/nh-invasive-plant-list-2014.pdf*

[15] Virginia Institute of Marine Science. "Hydrilla Overview" *https://www.vims.edu/research/units/programs/sav/species/hydrilla.php*

[16] Virginia Places. "How much of Virginia is water?" *http://www.virginiaplaces.org/watersheds/howmuchwater.html*

[17] Wallelign, Serawork, Mihai Polceanu, and Cédric Buche. "Soybean Plant Disease Identification Using Convolutional Neural Network." *Artificial Intelligence Research Society Conference.* 05 June 2018. Web. 26 Feb. 2021.

[18] Yuan, Yuan, Lei Chen, Huarui Wu, and Lin Li. "Advanced Agricultural Disease Image Recognition Technologies: A Review." *Information Processing in Agriculture.* Elsevier, 30 Jan. 2021. Web. 26 Feb. 2021.

[19] Zapotezny-Anderson, Paul, and Chris Lehnert. "Towards Active Robotic Vision in Agriculture: A Deep Learning Approach to Visual Servoing in Occluded and Unstructured Protected Cropping Environments." *IFAC-PapersOnLine. Elsevier, 31 Dec. 2019.* Web. 26 Feb. 2021.

[20] Zheng, Yang-Yang, Jian-Lei Kong, Xue-Bo Jin, Xiao-Yi Wang, Ting-Li Su, and Min Zuo. "CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture." *MDPI. Multidisciplinary Digital Publishing Institute*, 01 Mar. 2019. Web.