Multipronged Approach for Weed and Invasive Plant Species Identification

Elijah Reber and Eric Song

***Abstract*—In this paper, we describe our methods involved with the automatic identification of weed species in Australia. Our technique has been designed to provide interpret-ability by using explainable features based on plant photography. The first of these features is shape based detection of weed leaves which takes into account the formation and number of leaves in an image. The second of these is texture distribution which used an existing method, Local Binary Patterns, to find portions in the images which corresponded to edges, flat areas, and corners. In order to give ourselves a measurement to base our goals on, an unsupervised autoencoder network was trained and tested on the data set to compare accuracy. Our algorithm is designed to work with weed pictures taken in nature and introduces the possibility of high accuracy and highly explainable identification systems.**

I. Introduction

Worldwide, the growth of weeds and invasive species in agricultural crop fields results in damages from crop loss and weed control well into billions of dollars. In Australia alone, it is estimated that agriculturalists are hit by $2.5 billion of lost crop production from the growth of weeds in their fields and spend another $1.5 billion to combat the growth. Apart from economic damages, environmental and human health impacts can be seen from the growth of weeds in agricultural areas. [1].

As such, there is a great deal of incentive to develop systems capable of automatically classifying and eradicating destructive weed species. Once plants are properly identified, other technologies such as drones and robotics can perform high precision spraying of pesticides and weed killers to maintain fields. Mobile applications can also be developed that can be used by farmers to identify the species of a weed that is growing in their fields. Other computer vision work on plant species so far have gained high accuracy from using black box neural networks or restricted image inputs. [4]

In order to combat this growing issue, we propose a solution that will allow agriculturalists to use their smart phones to identify weeds and other invasive plant species in their fields and provide suggestions to combat the growth. The algorithm that we will build will be available to smaller-scale farmers, who may not have access to such technology otherwise. Previous technologies have been able to achieve high accuracies on the specified dataset, however, we are looking to improve upon these results using methods that have shown to provide success in computer vision tasks. The application itself would ideally be able to identify species of weeds given amateur, phone photography and then provide the reasoning as to why the model made that decision, as well as provide the decision that has been put forth by a deep learning network.

Our system will mainly use shape and texture features derived from a number of computer vision techniques. We learned through research that a very important part in recognizing plant species is looking at the leaves of the specimen. With that in mind, we analyze the plant leaves mainly through the shape and texture characteristics of the leaves.

In addition, our initial idea on how to approach this problem was also aimed at using deep learning neural networks. For our model, we wished to use an autoencoder network instead of the convolutional and residual networks done in other studies. For reasons that will be discussed later in the paper, we decided this was not the avenue to go down and instead began working on the interpretable algorithm.

The rest of the paper is organized as follows. The data we used is discussed in section 1A. Related works are discussed in section 2. We discuss our methods with the autoencoder network the computer vision model in section 3. In section 4, we discuss the results and findings from our algorithms. The work breakdown is discussed in section 5. Finally, we conclude and discuss future work in Section 6.

Eljiah Reber was with Department of Computer Science in the School of Electrical and Computer Engineering, The Pennsylvania State University, State College PA, 16801.
E-mail: see edr3@psu.edu

Eric Song was with the Department of Computer Science in the School of Electrical and Computer Engineering, The Pennsylvania State University, State College PA, 16801.
E-mail: see els5523@psu.edu

A. DATA

Our work will focus mainly on an image set "DeepWeeds" that includes 17,509 labeled images of eight specific weed species that are common to the area:

- Chinee apple
- Lantana
- Parkinsonia
- Parthenium
- Prickly acacia
- Rubber vine
- Siam weed
- Snake weed
- Negatives

The majority of the images (about 9,000) were examples of typical agricultural landscapes that did not contain any weeds. All of the data comes from northern regions of Australia, specifically across the state of Queensland. Overall, the size of the image set is 892 MB, with each image being labelled with the date, time, and id number of when that picture was taken. Along with the images is a .csv file of all the labels correctly identifying the species of weed in the picture.

In regards to the images themselves, the weed images are taken in their environment, meaning they are with neighboring plants and flora. The pictures are mostly taken from an angled top-down viewpoint. This is effective for training as these weeds generally grow low and will be looked at from these angles.

The quality of these photos does vary quite a bit. While many photos have a single plant clearly as the subject, others are surrounded by other foliage. Other factors include the health of the weed or shadows covering large parts of the plant that is being photographed.

Since the color values were being analyzed, the angle at which the photograph was taken, as well as the time of day, would be able to impact the decision. Generally, the contrast between different sections in the images would be the most clear when the sun is emitting the most amount of light. The angle at which the photograph was taken may accidentally result in a shadow being casted over a distinguishing feature, thus making classification a non-trivial problem. Examples of the dataset can be seen in Figure 1.

II. Related Work

In regards to plant identification, more work has been done in general on croplands rather than rangelands. As a result, there have not been many datasets and studies formed with rangelands in mind. The work that has been done involved deep learning and has used similar approaches with Inception-V3 and ResNet-50 which are a convolutional neural network and a residual neural network respectively [2]. These techniques and studies were meant to set a baseline benchmark with average accuracies of 95.1% and 95.7% respectively. The strong results obtained from these models bode well for the usage of deep learning in rangeland weed identification.

While these high accuracy networks already exist, we have not found much work done on the explainable and interpretable based approach side. There have been some applications such as LeafSnap that focus on taxonomic identification of plants based on the shape of their leaves. These, however, require photographs taken on a plain, uniform colored background and have been trained to identify other foliage and flora than rangeland species. Our goal is to utilize techniques to in order to identify weeds in pictures that were taken in nature.

From our research, we also found that there have not been many practical tools for farmers to use for weed identification. However, there has been a number of studies that have looked into the the possibility of using techniques such as precision spraying of weeds with machines or even drones. [3]

Our work will attempt to compare the effect of using a more easily explainable model for extracting information in identifying weeds to these deep learning benchmarks. If the accuracy is acceptable, this method may be a better fit as it can give reasons for why a plant is identified to be a certain species.

III. Method

A. AUTOENCODER

For our initial deep learning approach for weed identification, we used an artificial neural network. Specifically, we researched 3 different types of autoencoder networks to train and test on the Australian weed dataset. The different implementations we attempted are Vanilla, Sparse, and Convolutional autoencoders. One of our reasons for choosing this type of neural network was to try out a different form of deep learning that has not already been used in plant identification. Presumably, the autoencoders would be able to encode information through unsupervised learning which would then be used in a classification algorithm. In order to narrow our work, we created, trained, and tested each different implementation on a sample of 1000 images taken from the data. We would use the results of this analysis to then decide on which of the three autoencoders to train on the entire dataset.

1) In our **vanilla autoencoder**, we created a structure with 1 hidden layer that had 20 neurons. This took the 256x256 images of weeds and encoded them

Fig. 1. Various qualities of pictures within the dataset of weeds. The picture on the left and center are considered good quality. The weed on the right is covered in shadow.

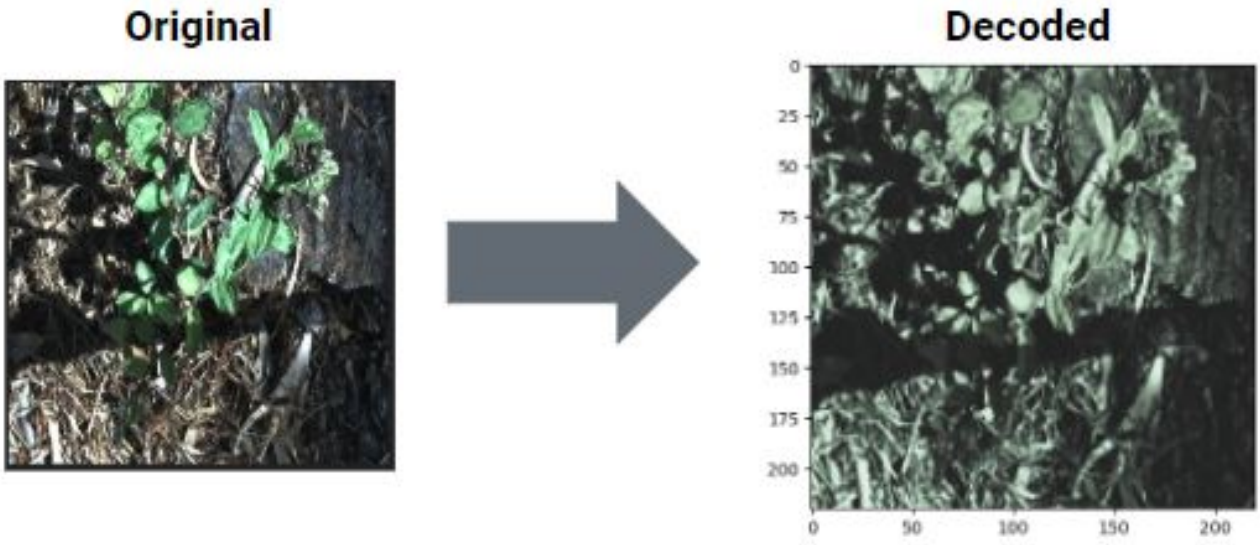

Fig. 2. Results taken from the Convolutional Autoencoder. Image on right is the result of being encoded and then decoded. The image is reformed relatively well, but with some color information lost.

down into a compressed representation before decoding them back to their original dimensions. The network is trained to reform the image back into the same image from the encoded layer. Thus, the hidden layer should be able to retain enough information while also applying dimensionality reduction and de-noising certain features. In general, we did not believe the vanilla autoencoder would perform as well as other methods but thought it was a good model to start with. This is also part of the reason we decided to use 1 hidden layer. With this, we have fewer weights to train and can thus cut down on training time. Realistically, we considered the idea that our machines were not capable of training on the entire dataset in a reasonable amount of time. However, we wanted to wait until we obtained our results to make a decision.

2) The **Sparse autoencoder** we created had a similar structure with 1 hidden layer that had 20 neurons. In this case though, we added a sparsity constraint to the model. We use L1 regularization in order to penalize the activation of neurons in the hidden layer. That is, we take the number of neurons that fired and multiply it by a hyperparmeter lambda in order to determine how much weight to give this constraint. As a result, fewer units should fire at a given time and we should end up with a

more sparse representation of the data. A reason as to why this model is being tested is because sparse encodings are often found to perform better in classification tasks. The reasoning for 1 hidden layer here is similar to that for the vanilla encoder in that it cut down on training time significantly compared to deeper models.

3) The third structure of artificial neural network we looked at is the **Convolutional autonencoder** and seems to fit our problem the best out of the three since it is naturally suitable to computer vision problems. With this sort of network, we apply convolutions and maxpooling much like the usual Convolutional Neural Networks. For our autoencoder, we applied 3 layers of convolutions with 3x3 windows. After each of these convolutions, we applied a maxpooling to the result. Initially, the images are 256x256, however, they had to be resized to 220x220 in order to better fit the model. To do this, the middle pixels were taken first and the edges were cropped out. Thus, we had layers that used dimensions 220x220, then 110x110, then 55x55, and finally 23x23. These dealt with 3 channels, then 16, then 8, and then 4. In the decoding layers, we once again used 3 convolutions, each of which were followed by an up sampling layer to reform a 220x220 image.

Once all of the encoding and decoding portions were trained for the autoencoders, we began to train these neural networks to classify the images correctly. To do so, we took the encoding portions of each model and fit a softmax layer to the end. The data itself came with labels. Thus, we found the 1000 corresponding images and labels we had sampled and began training. Once the models were done, we tested on another sample of 500 images and used these accuracies.

B. COMPUTER VISION MODEL

It was first decided that the more interpretable model would use a more classical computer vision approach, the Bag of Features (BoF) model with a random forest classifier. The BoF model is essentialy collecting both global and local features that are predetermined to be indicative of a label and then feeding the resulting feature vectors into a machine learning algorithm. This model, along with the accompanying random forest, was chosen due to it's explainable nature, as the random forest is easily able to say just how much each of the features accounted for towards the classification.

In order to identify features that would aid in the classification of the weeds, botany journals were first consulted. With this domain knowledge, it was discerned that features of leaves are the main way to identify any type of leafy plant. The main ways that leaves can be distinguished from one another is by looking at their shape and texture features.

In order to extract shape and texture features that would be indicative of the different species, the images first had to go through some pre-processing. First, the images were put through a filter to extract only the green parts of the images. Originally, just the RBG (Red, Blue and Green) values were being analyzed and the areas with a low green value were set to black. This resulted in a bad filter that included many areas that were mainly dirt, but had a green value that was high enough to be classified as green. Using a filter in the HSV (Hue, Saturation and Value) color space resulted in a filter that was able to more accurately extract the green parts of the image.

After the green parts of the image were extracted, the images were put through a median smoothing and an average smoothing. These smoothings helped to both get rid of jagged edges in the pictures and random, small areas of green that were not actually leaves.

After the images were fully pre-processed, they were put through a Hough transform, in order to find the leaves. The Hough transform was relaxed, so the identified circles did not have to be perfect circles, as to account for the fact that the leaves were not perfect circles. From the output of the Hough transform, the density, average area, total area and total amount of leaves were recorded as features. The average HSV values that were in the circles found in the transform were also recorded as features.

For the texture features, the Local Binary Pattern (LBP) of the images were calculated. The LBP is a measure of how similar the pixels are to its nearest neighbors. Using this similarity measure, the texture patterns are extracted by finding the edges, corners and flat areas of the image. The flat areas are the areas where the similarity measure is large between the pixels, indicating that they are all in the body of one object. The corners and edges are areas where the pattern in the pixels changes.

The LBP result is vectorized by putting the data into a histogram with 25 bins. The resulting distribution vector is then used as a feature, with each of the histogram proportions being a separate feature, in order to visualize which texture pattern gives the most information towards classification.

Fig. 3. The Convolutional Autoencoder would often mix up the Prickly Acacia, Parkinsonia, and Chinee Apple species. It is difficult to understand exactly why this is because it is a deep learning model.

IV. RESULTS & DISCUSSION

A. AUTOENCODER

The results of these deep learning networks were around what were expected them to be. The vanilla autoencoder only classified around 56.8% percent of the test data correctly. The sparse autoencoder classified 55.3% correctly. The convolutional autoencoder ended up classifying 61.6% of the images correctly. As a result, we decided that the Convolutional Autoencoder would be used in our next steps. Another good reasons as to why we chose the Convolutional model is because of training time. With convolutional autoencoders, there are far fewer weight that need to be trained on the model than with the fully connected vanilla or sparse implementations which results in quicker training.

From here, we recreated the neural network and trained the autoencoding/decoding portion of the network. Then once again, we took the encoding layers and added a dense softmax layer to the end to train on the dataset. From the 17,509 images, we took 1000 to be used as the test set and obtained a final accuracy of 59.6%.

Some results we noticed is that the autoencoder classification network would often confuse specific species. For instance, in Figure 3, it shows 3 pictures that were mislabeled as one of the others. This pattern of Prickly Acacia and Parkinsonia being mislabeled was found throughout the testing. On the other hand, the Chinee Apple was usually not classified incorrectly along with these other two, however there are still some instances like this one. In general, it seemed that the leafy plants were often misclassified as each other while the thinner, narrower leaf plants were also confused with each other. As for reasons to why this occurs, it is difficult to pinpoint because it utilizes deep learning. It may be the plant features themselves that are causing these results. However, it could also be that the photographs of these plant groups could have been taken in the same way. This sort of analysis might explain why the model would sometimes classify very dissimilar looking plants like the Chinee Apple as something like the Parkinsonia.

Once we created our deep learning model, we decided to change our work in the direction of a more explainable algorithm. There are numerous reasons that we used the autoencoder network for our first step. Our only reference for accuracies in the area of rangeland weeds classification came from the state of the art, deep neural networks like Inception-V3. We did not believe beating these benchmarks were possible with our resources and knowledge. However, these results were in the range of 93%-95%. While we knew we were not able to achieve this near 100% accuracy, we wanted a metric that was more useful. Autoencoders are unsupervised and would not be able to achieve accuracies of such a high level, but were also deep learning. Thus, we believed that they would be able to provide a much more reasonable result for us to compare our next approach to.

B. COMPUTER VISION MODEL

1) Classification: Overall, the BoF model with a decision tree was able to achieve an accuracy of 61%.

As can be seen from the confusion matrix, while the accuracy was 61%, the precision values for the species were actually quite low. Since the negative images made up a majority of the dataset and the precision of the negatives was relatively high at 86%, it made the model

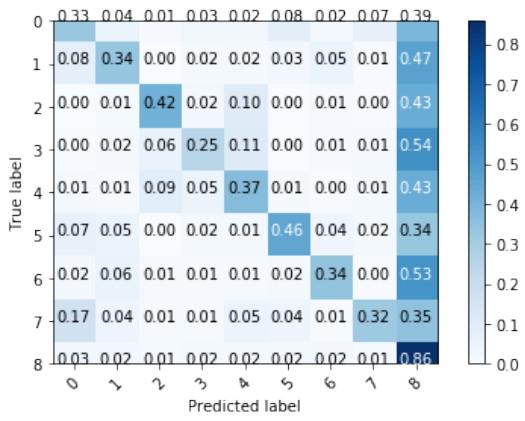

Fig. 4. Confusion matrix for CV algorithm, including negatives. The labels follow the same order as the list of species in the DATA section. Accuracy = 0.61

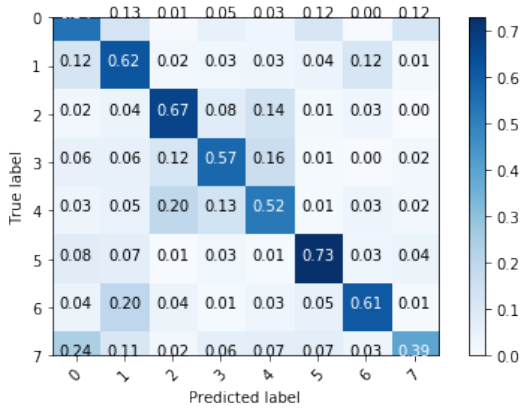

Fig. 5. Confusion matrix for CV algorithm, excluding negatives. The labels follow the same order as the list of species in the DATA section. The top left value is 0.54. Accuracy = 0.57

seem to be more accurate than it actually was. It can be inferred that the existence of the negative images confused the model. The algorithm actually classified five of the species as more likely to be a negative image than the specie itself.

Since this work would be used in an application for farmers, it can be inferred that the farmer might be able to know themselves whether or not the object they are looking at is actually a weed. If they were to have such knowledge, then they would not use the application on an object that is not a weed. Given this logic, the algorithm was tested again, without the negative images.

Without the negative values, the accuracy did drop down to 57%. This is probably due to taking out the 9,000 negative images that were classified at a relatively high precision rate. Without the negatives, however, the individual precision rates were increased by an average of 22.75 percentage points.

One interesting note is that the same failures can be seen in the new confusion matrix, however more magnified. The confusion around labels 2, 3 and 4 is present in the confusion matrix with the negative images, however it is more noticeable in the confusion matrix without the negatives. These labels happen to correspond to two of the labels that the deep learning algorithm got confused with (Prickly Acacia and Parksonia), but substituted the Parthenium species into the confusion, instead of the Chinese Apple. This may point to some similarities between the plants that makes it hard for even sophisticated algorithms to deal with.

Another interesting note is the improvement of the performance with classifying label 3, or the Parthenium. After the exclusion of the negative images, the precision rate increased by 32 percentage points. The algorithm originally classified the Parthenium as more likely to be a negative image originally, but did not think that it was likely to be any other specie. It can be inferred then that exclusion of the negative images narrowed down the options, so the algorithm had to pick it's second favorite label.

2) Important Features: Using the decision tree, the most imporant features were found to be the beginning and end of the LBP histogram distribution (corresponding to the areas found to be flat in the images) and the average HSV values in the extracted green areas.

Unsurprisingly, the features extracted that were aimed at describing the leaf size and density had very little impact - each accounting only about 2% or 3% of the classification. While the parameters in the transform were relaxed in order to account for more elliptical shapes of leaves, the features failed to accurately describe plants that did not have circular leaves, like the Parkinsonia.

In the cases where the leaves of the plant were not circular, the identified circles would include a lot of area in the circles that did not actually belong to the plant. Constraining the problem by setting a limit to the size of the circles leads to a less accurate representation of how big the leaves actually are, thus limiting the feature's utility.

3) Qualitative Analysis: By looking at the confusion matrix and the actual botanical features of the species, we can make claims as to how the model performed based on characteristics of the individual species. For instance, in the cases where the predicted labels were either 5, 6 or 7 (Rubber Vine, Siam Weed and Snake Weed), the true label was almost never 2, 3 or 4 (Parkinsonia, Parthenium and Prickly Acacia). When looking at images of Rubber Vine, Siam Weed and Snake Weed, it can be seen that they all have large, round leaves. Images of Parkinsonia, Parthenium and Prickly Acacia show more

Fig. 6. Image of a Parkinsonia plant

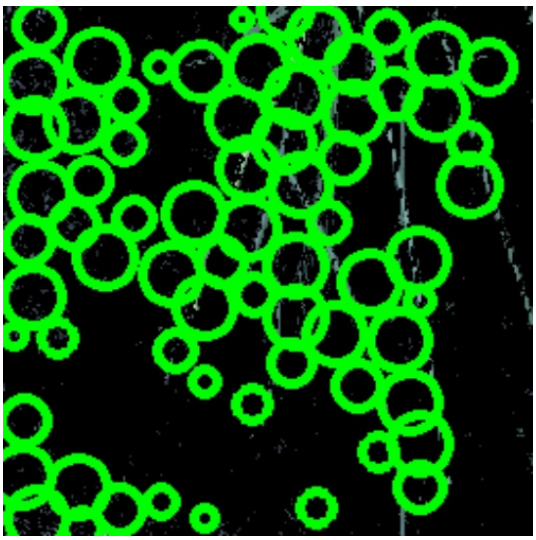

Fig. 7. Output of the Hough transform after pre-processing

twig-like structures with less well-defined leaves.

While the algorithm did not confuse these more leafy plants as the more twig-like plants, the opposite is not necessarily true. The classifier is seen to have struggled with classifying the Snake Weed, as it classified it as one of the more leafy plants about 13% of the time. It also believed that the Snake Weed was the Chinese Apple (another plant with round-ish and well-defined leaves) about 24% of the time.

This may not be the only conclusion that can be drawn from this observation. In Fig. 3, it can be seen that the Chinese Apple has a glossiness feature of it's leaves. Given that the average HSV values were among the most indicative features for the classifier, it could actually mean that the glossiness vs the dullness of the leaf could have been the deciding factor when classifying these species.

V. Work Breakdown

While both members did work in both approaches and all parts of the project in general, Eric Song was tasked mainly with the deep learning method while Elijah Reber applied himself mostly to the interpretable approach.

Both the code and writing for each portion are split similarly to this.

VI. Conclusions and Future Work

We will continue to try to improve the deep learning algorithm, however, we are skeptical as to its capability of beating the well-known ResNet and Inception networks, as there is a vast amount of work behind their architectures. With time permitting, we may try other versions of the autoencoder architecture, just to see if any of them result in a significantly higher classification rate.

In general though, we believe our work with non-deep-learning techniques has been useful. We were able to create a decision tree classifier algorithm that was comparatively robust. It had accuracies close to what our deep learning model could obtain (around 50%-60%) and can definitely be improved with more feature engineering and weed specialty knowledge. With more work, this algorithm could potentially be of more use to the agriculturalists, as it would provide an explainable reason for its classification, which is necessary for subsistence farmers who may not have the money to afford the algorithm telling them the wrong species of plant.

A. BROADER IMPACTS & POSSIBLE LIMITATIONS

If the algorithm were to be deployed into an application, then the information would be available to many more farmers than it currently is. While large-scale farmers may have the necessary capital to fund large consultation services to help with weeds, subsistence farmers may not be able to acquire sufficient resources. This algorithm would enable those farmers to have the same capabilities as their large counterparts, in terms of crop protection, at the fraction of the cost as a consulting service would cost.

The application would also need to work quickly with video. Realistically, a farmer could not manually go out into the area of all their fields and classify weeds with their phone. It would require some sort of machinery or possibly drones that could quickly and accurately spray pesticide. This may be more efficient if the algorithm was able to work with the continuous movement of the

robot, instead of having to process individual movements one at a time.

The dataset that we use will mean that the algorithm will be restricted to weeds common to the Northern Australian area. An algorithm that could be used in multiple areas would require a much bigger dataset with more examples of common weeds from different areas.

Apart from the geographical limitations, it is likely that this algorithm will not be able to perform classification at the rates that were shown with ResNet and Inception. An app that is powered by the algorithms that we are building, however, could make it possible for an agriculturalist to have two options - one deep learning method that will have a higher (but not state-of-the-art) accuracy or a lower accuracy model that is able to explain its predictions. The agriculturalist could potentially use both methods and compare the results to get a better idea of the weeds growing in their fields. If the two classifications match, then the farmer can be more sure of the accuracy, and if they differ, then the farmer may be able to get an idea of which one it actually is and decide which algorithm to agree with from there.

REFERENCES

[1] *Impact of Weeds* https://www.environment.gov.au/biodiversity/invasive/ weeds/weeds/why/impact.html (2019).
[2] Olsen, Alex, et al. *DeepWeeds: A multiclass weed species image dataset for deep learning.* Scientific reports 9.1 (2019): 2058.
[3] Yu, Jialin, et al. "Deep learning for image-based weed detection in turfgrass." European journal of agronomy 104 (2019): 78-84.
[4] Barré, Pierre, et al. "LeafNet: A computer vision system for automatic plant species identification." Ecological Informatics 40 (2017): 50-56.