An Open Dataset of Labelled Tropical Crops

Jade Chattergoon(✉), Fazeeia Mohammed, Kimberley Gillette, Brittany Peters, and Patrick Hosein

The University of the West Indies, St. Augustine, Trinidad
jadec.projects@gmail.com, mindy.moh@gmail.com, kimg777@gmail.com, brittany.peters1@my.uwi.edu, patrick.hosein@sta.uwi.edu

Abstract. Small Island Developing States (SIDS) are particularly susceptible to the detrimental effects of Climate Change. Issues such as food security have been given increased attention over the last few years and this has sped up with the transportation and logistical issues faced during the COVID pandemic. Farmers must now increase yield to increase food security while battling the effects of climate change. Technologies such as Artificial Intelligence can assist but the data needed to train these algorithms are not available for SIDS and so development in these areas are limited. We received a National Geographic Grant to investigate the application of Artificial Intelligence to Precision Agriculture in SIDS and one of the required steps was the collection of data for training Machine Learning algorithms. This required collection of data (using Unmanned Aerial Vehicles or Drones), processing of the data, labelling of the data and then development of suitable algorithms for problems such as Weed Detection, Water Stress detection and other potential problems. We are making this dataset available to the public on an Open Data platform that we developed and which is located at **data.tt**. This paper describes the data that is being shared, how it can be accessed and examples of how it can be used.

Keywords: Artificial Intelligence · Open Data · Precision Farming · Resilience · Smart agriculture

1 Introduction

The effects of Climate Change on Small Island Developing States (SIDS) are well documented. In addition, food security issues, not only due to Climate Change but also because of transportation and logistics issues, have surfaced because of the Covid pandemic. These issues, for the case of Caribbean SIDS have been documented in publications such as [1].

New technologies, such as Artificial Intelligence (AI) and Robotics, are being used in developed countries to increase crop yield. Unfortunately, the use of these technologies in SIDS is rare. One issue is the lack of data. Artificial Intelligence algorithms require data on which to train and this data must be taken from the same environment in which the algorithm is being applied. Unfortunately, such data is not readily available in many SIDS.

J. Sumantyo et al. (Eds.): ICoSIA 2022, ABSR 29, pp. 24–34, 2023.
https://doi.org/10.2991/978-94-6463-122-7_3

One component of a National Geographic Society Grant that we received was the application of AI to Precision Agriculture for SIDS environments. SIDS have certain limitations such as small farm sizes which limits scaling advantages. Hence, such farmers cannot afford to purchase and use devices such as Unmanned Aerial Vehicles (UAVs). In order to develop AI algorithms for use in SIDS we have collected, processed, labelled and used local agriculture data. We are making this data available to the public using an Open Data repository that we previously developed using the CKAN platform. We describe how this data was collected, how it was labelled and how it can be accessed on this platform which is located at data.tt.

2 Related Work

There have been numerous crop datasets over the years. Toda et. al spoke of using synthetic datasets of barley seeds for training an instance segmentation neural network, namely the Mask R-CNN [2]. These synthetic datasets were created by using a variant of the sim2real approach, which generates synthetic images, with domain randomization, and uses variations of the synthetic images to train the deep neural network [2]. The authors achieved an Average Precision value of .96 on the synthetic dataset and .95 on the real-world dataset [2]. However, at a higher IoU (Intersection over Union) threshold, the Average Precision was .73 on the synthetic dataset and .59 on the real-world dataset [2]. The authors stated that the higher values on the synthetic dataset may be a result of data leak, which occurs when the same images are used in training and testing, but it may also be a result of over-fitting [2]. Additionally, the differences in the synthetic and real-world test dataset are brought to light when the threshold is increased, which suggests that the synthetic dataset is not an accurate representation of the barley seed images.

Jolivot et al. introduced a database consisting of 24 datasets of satellite images of agricultural plots of land [3]. These datasets consist of a variety of crops, ranging from maize, rice, sugarcane, and soybeans. The data was collected over a period of seven years and from seven collection sites located across the world. However, the authors did not apply any machine or deep learning techniques to the data collected. Nevertheless, they made their data publicly available thereby allowing other researchers to apply machine learning techniques to the data.

Paliyam et al. spoke of using Street2Sat, which is a method for generating labelled datasets from geo-tagged images [4]. The proposed method collects images of road-side objects at regular intervals and then transforms them into a set of geo-labelled points with locations. These points can then be used as labels for satellite images. The authors then used YOLOv5 to make predictions on their dataset. The labelled dataset consisted of 296 images of maize and sugarcane. The resultant dataset consisted of 755 bounding boxes of maize and 1795 instances of sugarcane in the training set and 253 instances of maize and 229 instances of sugarcane in the test dataset. The authors achieved a precision of 0.41, and a recall of 0.59, which is poor. The authors stated that it was difficult to draw a bounding box around individual plants in a crop field. This means that labels included many plants and did not cover every crop instance in the image. Moreover, the images collected were mainly from Western Kenya, which means that the models developed

may perform poorly on images that were collected from other countries due to the lack of variability in the dataset.

Zheng et al. introduced a new dataset called CropDeep which consists of 31,147 images of 31 categories of crops such as tomatoes, turnips, pumpkins, scallions and lemons [5]. This dataset was created using IoT cameras, robots, mobile cameras and smartphones. The authors obtained 49,765 bounding boxes from the dataset. Then, the authors compared the performance of image classification and object detection models on the CropDeep dataset, using networks such as VGG16, VGG19, ResNet, Faster R-CNN, YOLOv3 and RFB. For the image classification models, the average accuracy of all models used was over 0.92, which is excellent. For the object detection models, the average Mean Average Precision of all models used was over 0.82, which is good. However, the authors did not state the precision, recall or F1 score of the classification models. It is recommended that more than one evaluation metric be used to evaluate the performance of deep learning models to obtain a better idea of their performance. Additionally, the CropDeep dataset suffered from class imbalance, with the lemon class having fewer samples than other classes. This negatively affected the performance of the models on that class.

Mylonas et al. introduced a database called Eden Library, which consists of images of 15 different crops, 9 weeds and 30 disorders which include pests, diseases and nutrient deficiencies [6]. This database consisted of both proximal and images from an UAV, facilitates image classification and object detection and had an accompanying web application made with the MEAN stack (MongoDB, Express.js, Angular and Node.js). However, this dataset also suffered from class imbalance and the authors did not perform any image classification or object detection tasks on the data.

Pena et al. spoke of utilizing semantic segmentation on a dataset consisting of ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) satellite images of an agricultural region in California [7]. The satellite images captured nine different crops including walnut, almond, alfalfa, tomato, rice, sunflower, safflower and maize. The authors compared the performance of a C4.5 decision tree, a logistic regression model, a support vector machine model and a multi-layer perceptron on the dataset. The authors achieved an accuracy of 0.88 on the multi-layer perceptron and the support vector machine models, an accuracy of 0.86 on the logistic regression model and an accuracy of 0.79 on the C4.5 decision tree model, which, overall, is good. However, the satellite images were captured in 2006 which suggests that the models' performance may falter when tested on satellite images captured in more recent years.

Veeragandham and Santhi introduced a new balanced and multi-class dataset for groundnut crops with 15 different classes of weeds and 24,816 images, known as the groundnut weed dataset [8]. The authors then performed image classification on the dataset by comparing the performance of AlexNet, VGG-16, VGG-19 ResNet-50 and ResNet-101 on the groundnut weed dataset and the existing corn weed dataset. The authors achieved an accuracy of 0.998 and 1 on the groundnut weed dataset and corn weed dataset with the ResNet-50 and RestNet-101 networks. However, the authors did not report the precision, recall and F1 score of the models. It is recommended that more than one evaluation metric be used to evaluate the performance of deep learning models to obtain a better idea of their performance.

The main contribution of this paper is the description of a large agriculture dataset that is being made available to the public as open data. This dataset took a significant amount of time to create and we hope that others can benefit from our efforts. We plan to continue expanding this dataset as our research progresses.

3 Data Collection and Processing

Data collection began in October 2021 and finished in April 2022. The data collection is aimed at solving the problems of identifying water stress and weed detection. The methodology of both experiments are outlined below. The farm locations that were chosen were spread over the particular SIDS, the island Trinidad in the Caribbean. These locations are provided in Fig. 1.

For the weed detection, small scale vegetable farmers were contacted for data collection through a local NGO. They were contacted through phone calls and the project was explained to determine the farmer's willingness to participate. The video data of the Capsicum annum mono-cropped field was collected through the use of a DJI Phantom V4 Pro drone. The video data was then split by frames and examined for artefacts and blurriness. This unmanned aerial vehicle was used through the manual flight application DJI pro. It takes images at approximately 3 m off the ground at a speed of 2 m/s. Once video is collected, the frames are then extracted at an interval of 30 frames per second at each data collection site. The data was collected between the hours of 10 am to 12 pm to allow for optimal lighting. Factors like rain and cross wind influenced the uptake of data.

The extracted images are then filtered for duplicates, blurred images and redundant data. The images are scanned and filtered for any inappropriate objects or reflective surfaces.

Fig. 1. Farm locations on the chosen SIDS

Agricultural experts were consulted when designing the water-stress experiment. For this experiment we manipulated the irrigation system at the farm. Additionally, to attain an array of images at different levels of water stress it was suggested that the experiment should be done on a slope so that there would be a gradient of soil water saturation going down a slope which would aide in the robustness of models created. This is an important factor considering the open field in which data was collected has practical implementations for farmers.

For this study we used random sampling for data collection of soil moisture content. The fields were maintained daily by the farming personnel on site. The site was monitored for pests, nutritional deficiencies and invasive weeds. One third of the field was excessively irrigated through flexible irrigation at the roots of the plant. Research suggested that the ideal amount of irrigation during the wet season is approximately 1.054 L and 3.73 L during the dry season. One third of the field was not watered as a control and the remaining one-third was watered moderately. The data and adjusted watering ranges were done during the Week 4 and Week 5 stages of development. Samples were collected once during the wet season and again during the dry season using an RGB camera.

On average a single data collection venture yielded approximately thirty to fifty images this was due to cross winds yielding blurry images and artefacts on site. Image augmentation code was used to determine the similarity between images, and to resize the images to crop out artefacts. This became necessary due to strong winds and poor lighting (due to weather) at the data collection site. Overall, the number of images collected was greater than the data labelled. This was due to time constraints placed on the project. We collected 200 water stress images and 300 images for weed detection.

Weather was a significant factor in data collection. In future iterations of this project, it is proposed that the impact of weather and flooding in small islands as it affects farmers in terms of water-stress should be examined. As such, farmers who experience the devastating effects of flooding and crop destruction would be targeted for data collection efforts in the future.

4 Data Labelling and Sample Usage

The images were labelled using the Image Labelling application found in MATLAB. Using either the super-pixel or assisted freehand tools found in the application, the ground truth data was labelled. The images were marked using recognition of interest (ROI) labels. The healthy crops present in the image were labelled as “crops” and the weeds present in the image were labelled as “weeds”. The crop pixel data was encoded as 0 in the array used to train the model. The weed pixel data was encoded as 1. Similarly, the background pixel data was encoded as 2. The background pixel-data refers to the pixel data of the image which was classified as neither crop nor weed. The crops were labelled in purple and the weeds were labelled in yellow. This was done to ensure that it was easily visible as the chosen colours were contrasting to the natural colours of the crops and weeds respectively.

The process of image labelling required the person labelling to manually zoom into the image at very high magnification rates. This was done to ensure accurate labelling

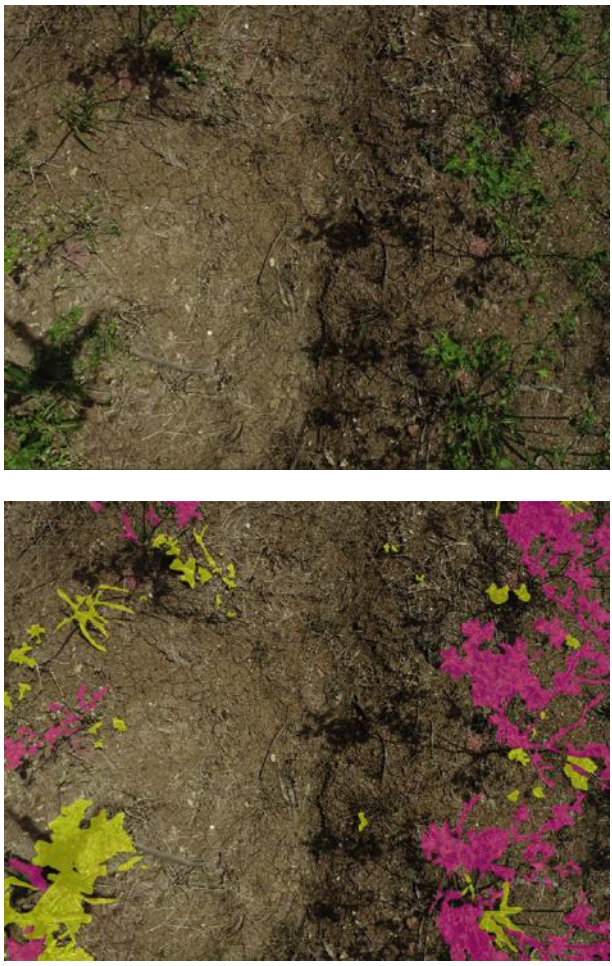

Fig. 2. Example of an original image (top) and the labelled image (bottom)

as it was difficult to decipher between the weeds and crops without performing this step. However, this resulted in the labelling process taking a lengthy time to complete. The model required a large number of images for training and testing to ensure accuracy. 194 of the 200 water stress images were labelled and 180 of the 300 weed detection images were labelled. Each image took approximately one hour to label hence the labelling process was rather time consuming. One limitation was that since the labelling was done manually using MATLAB, human error is likely since the crops and weeds were labelled based on the discretion of the person labelling. Figure 2 contains an example of an original image (top) and the corresponding labelled image (bottom).

The sample images were used to perform weed detection and water stress determination. After labelling was completed, image augmentation such as rotations, zooming, flipping and the addition of filters were applied to increase the number of images in the

data sets and to help create more robust models. The models were then trained using the original and labelled images. The performance of the models was then evaluated. The models displaying a high-performance rate were then saved and used in the creation of the mobile and web applications. The farmers will then use the application of choice to upload images of their crops, which are similar in nature to the original images which were used in the labelling process. Upon being prompted by the application, the farmers would then select either weed detection or water stress as the problem they would like to address. Here, the saved model is loaded by the application and used to predict the labels as either water-stressed or non-water-stressed for the water stress problem and either weed or crop for the weed detection problem. These labels were then assigned a red, green, blue (RGB) value and displayed to the user where it will be used for comparison with the original image uploaded by the farmer.

5 Platform for Data Access

Using the images that were collected and labelled, four data sets were uploaded to the repository **data.tt**. This is an online repository which we developed [9] and is powered by an open source application, CKAN. This site provides open data to the general public for personal or academic use. All data sets available on this site were collected from or are directly related to Trinidad and Tobago. Collections of data sets are published under Organizations and can also be viewed by tags, groups, file formats and licenses.

The links provided in this paper will allow the reader to easily navigate to the data sets collected for the purposes of the Farming Adaptation and Artificial Intelligence for Resilience (FAAIR) project. However, using our repository, the reader can also easily navigate to the data. Figure 3 shows screenshots of the home page of the repository (top) and the page obtained when one searches for the TTLAB organization (bottom).

The data sets for Weed Detection are labelled as “Unlabeled Weed Detection Images for Hot Peppers” and “Labelled Weed Detection Images for Hot Peppers”. Similarly, for the Bok Choy data sets. Each data set is accompanied by a short description which tells the user about the species of the crop as well as the usefulness and the context of using the data. On this page, there are two sections: “Data and Resources” as well as “Additional Info”. The “Additional Information” section provides information such as the last date that the data set was updated, the date of creation, the licence and other important metadata. The user can navigate to the data sets by clicking the name of the data set in the “Data and Resources” section and then clicking on the url provided at the top of the resulting page.

Each url provided on the site will direct the user to a folder on a Google drive containing the images for the selected data set. This folder can then be downloaded to the user’s local machine. The user should note that the labelled images are given the same name as their corresponding unlabelled versions in the original data set. A GitHub link containing a step-by-step tutorial on displaying and utilising the pixel data from the original images and the labels from the labelled images is provided to the user. The images used in the tutorial will be available in the linked Github repository as well as the link provided to the data sets. In the near future the user will be given the added option to preview and use the images directly from the site.

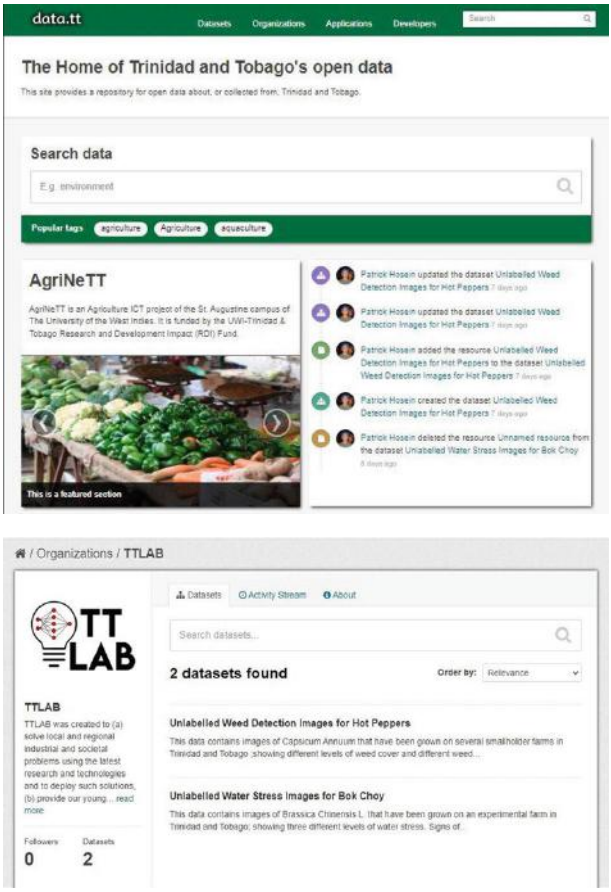

Fig. 3. Screenshots of Home page and TTLAB Organization page on **data.tt**

Detailed instructions on using images from both file paths and urls can be found in the Jupyter notebook file in the Github repository [10]. Here we provide some things to note:

- The tutorial is contained in a Jupyter Notebook file that is openly accessible on Github [10] and uses python version 3.10. Other than Jupyter, this file can be used in PyCharm, Visual Studio Code or any other editor that supports notebook files. If preferred, notebook files (.ipynb) can also be converted to python files (.py).
- Downloaded images can be loaded into a python program by using the *cv2 imread* function. This function requires the file path of the image on the user's local machine. In most cases, Windows requires the use of forward-facing slashes as opposed to back facing slashes in the file path. If the user faces an invalid string error, the path can be edited accordingly. The user should ensure that the unlabelled and labelled images remain in separate folders and that their file paths are assigned to the appropriate file path variables. The image file name variable should be an integer. The Label prefix at

Fig. 4. Sample of images generated with Github code (original on top and labelled below)

the end of the labelled image file path is only applicable in the tutorial for purposes of helping the user differentiate between the images. The file formats used in the data sets (.png or.jpg) must be defined as a string variable. The user should ensure that the file format used in the notebook aligns with that of the images in the required data set.

- The above method is modified when images are being read from a url. The code for this method is included in the tutorial and can be used by following instructions to replace certain commented blocks of code with others that are relevant to the above method and currently not commented in the notebook. If the preferred method of loading the file is with an image url, the user must ensure that the url leads to a web page that only displays the desired image. The tutorial currently includes the urls to two sample images from the Weed Detection data set.
- Since cv2 reads the image using the BGR (blue-green-red) pixel arrangement format, the user can use the *cv2 cvtColor* function with the parameter which converts the pixel arrangement format from BGR to RGB if this pixel arrangement better suits their requirement. This step is not necessary when the image is loaded from the url.
- In order to access the image's pixel data, the image object can be converted to a numpy array using the numpy *asarray* function. This numpy array will contain each pixel's red, green and blue pixel values. If a file url was used to load the images, additional steps are necessary to obtain the required numpy array. These are outlined in a similar manner using commented code. To display the labelled image data, the user can utilise

the numpy *where* function to replace each label with a pixel value. For the tutorial the background, crop and weed labels were assigned the colours black, grey, white respectively using the appropriate pixel values.
- The user can use the *GridSpec* library to view both the labelled and unlabelled images side by side using the unlabelled image's pixel array and the modified labelled image's pixel array. The *GridSpec* variable controls the layout of the display and can be altered if desired. Figure 4 shows how the unlabelled and labelled images were displayed in the tutorial using GridSpec.
- The unlabelled image (as a numpy array) can then be augmented as needed. The arrays can be converted back into images using the pil *fromarray* function. The resulting image objects can then be saved to the user's local machine using the pil *save* function while specifying the file path and the user's desired image format.
- The Github step-by-step tutorial and sample images can be found at [10].

6 Conclusions

We described a dataset that was collected for research in the application of Artificial Intelligence to Precision Agriculture in a SIDS. The dataset is being made available to the public and this paper is meant to serve as a source of information on the dataset, how it was collected, how it can be accessed and how it can be used. Any data that is collected in the future for this project will also be added to the repository which is located at **data.tt**.

Acknowledgments. This research was supported through a research grant from The National Geographic Society. Through this grant Cloud Computing resources were provided by Microsoft Corporation. Local computing resources were supported through a hardware grant from NVIDIA.

References

1. Lincoln Lenderking H, Robinson S a and Carlson G 2021 International Journal of Sustainable Development & World Ecology 28 238–245
2. Toda Y, Okura F, Ito J, Okada S, Kinoshita T, Tsuji H and SaishoD 2020 Communications biology 3 1–12
3. Jolivot A, Lebourgeois V, Leroux L, Ameline M, Andriamanga V, Bell´on B, Castets M, Crespin-Boucaud A, Defourny P, Diaz S et al. 2021 Earth System Science Data 13 5951–5967
4. Paliyam M, Nakalembe C, Liu K, Nyiawung R and Kerner H 2021 Tackling Climate Change with Machine Learning Workshop at the International Conference on Machine Learning
5. Zheng Y Y, Kong J L, Jin X B, Wang X Y, Su T L and ZuoM 2019 Sensors 19 1058
6. Mylonas N, Malounas I, Mouseti S, Vali E, Espejo-Garcia B and Fountas S 2022 Smart Agricultural Technology 2 100028
7. Pena J M, Guti´errez P A, Herv´as-Mart´ınez C, Six J, Plant R E and L´opez-Granados F 2014 Remote sensing 6 5019–5041
8. Veeragandham S and Santhi H 2022 Computers and Electrical Engineering 103 108315
9. Lutchman S and Hosein P 2015 Journal of ICT Standardization 289–302
10. Chattergoon J 2022 GitHub. URL https://github.com/Jade98-afk/FAAIR_OpenData_Tutorial