

CS 153 Project Proposal

Mapping the density of flowers on California buckwheat plants

Alex Hadley
Harvey Mudd College
Claremont, CA
ahadley@hmc.edu

1. Motivation

Honey bees and native bumble bees are important pollinators of both agricultural and wild plants [9]. However, recently the populations of both bees have declined. Donkersley *et al.* [5] have shown that landscape composition is important to the nutrition of honey bees, and Donaldson-Matasci and Dornhaus [4] have shown that species richness and flower density are important variable affecting honey bee communication. Examining these factors involves mapping out the locations of flowers; however, current methods of doing so are either too labor-intensive (*e.g.* ground-based surveys) or lack detail (*e.g.* satellite imaging).

Professor Donaldson-Matasci's Bee Lab at Harvey Mudd College is using high-resolution drone imaging, which provides more detail than satellite imaging but scales more easily than ground surveys. So far, the lab has been able to classify plants by species and generate bounding polygons (see Figure 1). The next step in their pipeline is to identify the density of flowers on these plants. Initially, the lab is interested in the plant *Eriogonum fasciculatum*, the California buckwheat, but they hope to extend flower detection to other plants in the future. The ability to generate detailed maps of flowers and compare them to the nutritional status of bees will help develop models predicting the affects of land use and climate change on pollinators.

2. Methods

The goal of this project is, given a drone image and a bounding polygon (see Figure 1), to return the total number of pixels within the polygon, the number of those pixels corresponding to flowers, and a mask of the flower pixels.

The provided data includes images with corresponding JSON files containing lists of plants in the image, along with a bounding polygon for each. Code to plot these polygons has been provided by Tom Fu.¹ The next step will be



Figure 1. An example of a polygon annotation outlining a California buckwheat plant.

to convert the polygons into binary masks in order to count the total number of pixels in the plant regions, and to isolate the plant regions for flower identification. This will be done using OpenCV's `fillPoly` function.²

The task of labeling pixels in an image according to categories is known as semantic segmentation. A classical method of accomplishing this is color thresholding, which involves tuning lower and upper limits in the RGB or HSV color spaces and labeling pixels within the limits as flowers. Although this method is relatively easy to implement, it does not perform well since threshold values are sensitive to particular lighting conditions and do not take morphological features into account [3]. I intend to use color thresholding as a baseline model; however, I expect the model to incorrectly label the white rocks seen in Figure 1 or have trouble working well in different lighting, such as in Figure 2.

A more promising method involves using a convolutional neural network (CNN). A well-known model that combines semantic segmentation with object detection is Mask R-CNN [6], and it has been used to identify plant

¹https://github.com/tommyfuu/flower_map_new/blob/master/scripts/visualizePolygons.py

²https://docs.opencv.org/3.4/d6/d6e/group_imgproc__draw.html



Figure 2. An example California buckwheat plants in different lighting than Figure 1.

features, such as by Machefer *et al.* [7]. However, use cases for Mask R-CNN typically involve distinguishing individual objects, which is not one of the goals for this project.

One semantic segmentation model that only labels pixels rather than detecting objects is SegNet [1]. SegNet has been used by Palacios *et al.* [8] for identifying grapevine flowers. Using the SegNet architecture with a VGG19 network as the encoder, they were able to achieve an F1 score of $0.930 \pm 3.15\%$ for their segmentation task.

Another promising model used by Dias *et al.* [3] is DeepLab [2]. Like SegNet, DeepLab labels pixels according to different categories, which Dias *et al.* use to detect fruit flowers, achieving F1 scores between 74% and 86% across several species, which greatly outperformed their HSV-based baseline model.

After creating a baseline model based on HSV thresholding, I intend to use either SegNet or DeepLab to increase performance. Dias *et al.* also divide their images into smaller “portraits” and passing those through their CNN [3]. Given that the plant regions in this project have varying sizes, I think this would be a promising method to use. Since the flowers are already small, I do not want to resize the images, so cropping seems like the best option. Furthermore, detecting flowers in one region of the plant should not depend on features from other regions, meaning that examining subregions independently should work well.

Given a plant region, I will take a bounding rectangle and divide this into squares. The squares will have some overlap in order to account for the potential of features’ being underweighted near the edges. I plan to train and run the CNN on these subregions, and then take a union of the resulting masks to create an overall flower mask for the plant. Finally, I will discard flower pixels outside of the plant’s bounding polygon.

In order to train the model (either SegNet or DeepLab), I plan to use PyTorch and implement a similar method to Dias *et al.* [3], which involves using weights pre-trained on the COCO-Stuff dataset and fine-tuning the model to rec-

ognize the specific flowers in question. This dataset already includes labels for common features such as flowers and grass, so the pre-trained model should be a good starting point.

In order to train and test the model, I will require the data to be annotated with pixels corresponding to flowers. I will then divide the regions and annotations into square subregions as described above, and use those as input to the CNN.

3. Measuring success

I plan to measure success quantitatively based on precision, recall, and F1 scores of the proportion of correctly predicted pixels within the plant regions. I will also qualitatively measure the success of the models by comparing the output flower masks to the ground truth annotations. I will also compare the performance of the CNN model to the baseline HSV model to ensure that the extra computation is worth it.

4. Challenges

One challenge is that the plant regions are not square or even rectangular and are not consistently sized, so they cannot be passed directly into a CNN. I plan to address this using the method described above, where I take a bounding rectangle of the region and divide it into partially overlapping squares, using the squares as independent input to the CNN. This also has the advantage of not needing to rescale the images, which potentially loses important detail for small flowers.

Another challenge lies in annotating the data. There are 888 images containing 11,981 plants. It will be a challenge to annotate enough plants to train and test a model well. There is no need to annotate all of the data, but it is important to acquire enough sample data to train on. One solution will be to share the work of annotation with other CS 153 students working on this project. It will also be important to sample images from the different batches provided in order to produce a robust model, since they were taken in different regions and in different lighting conditions.

Another challenge will be actually training these CNN models since I do not have much experience doing so. A solution will be examining how this is done in the papers, reading PyTorch tutorials online, and getting help from Professor Wloka or other students working on the project. Also, I have identified two potential models, SegNet and DeepLab, so that if I have trouble implementing one, I will hopefully be able to use the other.

Acknowledgements

I discussed this project with Thuy-Linh Le, another CS 153 student working on this project.

References

- [1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. [2](#)
- [2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. [2](#)
- [3] Philipe A. Dias, Amy Tabb, and Henry Medeiros. Multi-species fruit flower detection using a refined semantic segmentation network. *IEEE Robot. Autom. Lett.*, 3(4):3003–3010, 2018. [1](#), [2](#)
- [4] Matina C. Donaldson-Matasci and Anna Dornhaus. How habitat affects the benefits of communication in collectively foraging honey bees. *Behav. Ecol. Sociobiol.*, 66(4):583–592, 2012. [1](#)
- [5] Philip Donkersley, Glenn Rhodes, Roger W. Pickup, Kevin C. Jones, and Kenneth Wilson. Honeybee nutrition is linked to landscape composition. *Ecol. Evol.*, 4(21):4195–4206, 2014. [1](#)
- [6] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 234–778, 2017. [1](#)
- [7] Méliissande Machefer, François Lemarchand, Virginie Bonnefond, Alasdair Hitchins, and Panagiotis Sidiropoulos. Mask R-CNN refitting strategy for plant counting and sizing in UAV imagery. *Remote Sensing*, 12(18), 2020. [2](#)
- [8] Fernando Palacios, Gloria Bueno, Jesús Salido, Maria P. Diago, Inés Hernández, and Javier Tardaguila. Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. *Comput. Electron. Agric.*, 178:105796, 2020. [2](#)
- [9] Simon G. Potts, Jacobus C. Biesmeijer, Claire Kremen, Peter Neumann, Oliver Schweiger, and William E. Kunin. Global pollinator declines: trends, impacts and drivers. *Trends Ecol. Evol.*, 25(6):345–353, 2010. [1](#)