Flowering density estimation from aerial imagery for automated pineapple flower counting

Jennifer Hobbs[1*], **Robert Paull**[2], **Bernard Markowicz**[1], **Greg Rose**[1]
[1]IntelinAir
[2]University of Hawaii at Manoa
{jennifer, bernard, greg}@intelinair.com, paull@hawaii.edu

Abstract

Deep Learning is changing the face of agriculture. Combined with high-resolution aerial imagery, these methods enable farmers to understand and manage their farms with previously unseen precision and efficiency. Beyond reducing costs for an industry already under significant economic stress, these advances have key environmental benefits as well: maximizing production, reducing waste, anticipating disruptions to supply chains, and limiting the use of chemicals and water through targeted application. Our approach uses a U-net based neural network to predict the density of flowering pineapple plants from aerial imagery, enabling farmers to optimize their harvesting schedule.

Figure 1: Our model identifies the density of pineapple flowering across multiple blocks of a pineapple field. The flowering density is depicted as a spectrum from low (yellow) to high (red) and regions of no-flowering shown as transparent. A single field of this size has over 1.5 million plants.

1 Introduction

Specialty crops, such as pineapple, present unique challenges and require sophisticated approaches to maximize productivity. Growers of large area crops such as corn or soybean have access to GPS-based yield maps and precisely apply inputs such as fertilizer and water considering field variability. Specialty crop growers lack access to these data as their crops tend to be hand-harvested. Because of this, specialty growers have been at a disadvantage, having to make decisions without this level of insight.

Growers of these high-value crops make a number of key decisions in every growing cycle. For pineapple, data supporting these decisions are generally limited to visual ground observations. But these observations are from the periphery where spatial and temporal variability, stage of growth, and development cannot be determined or quantified across the entire field. This lack of complete, real-time information about field conditions can lead to poor decisions resulting in too little or too much water, fertilization, pesticides and growth regulators, or poor planning and scheduling of planting and harvest resources, including equipment and labor.

For pineapple, natural flowering affects fruit development and quality, and impacts harvest [Bartholomew *et al.*,]. Pineapple growers use chemicals to induce flowering so that most plants within a field produce fruit of high quality ready to harvest at about the same time. The ideal situation would be for a grower to harvest the entire field in one pass, significantly increasing productivity and eliminating the cost of additional harvests. This ideal single pass occurs when there is little spatial and temporal variation in flowering.

Modifying management practices with data on field conditions goes beyond reducing costs for the farmers. By identifying flowering plants at their earliest stages across entire fields, the application of chemicals can be precisely applied and limited in extent. By monitoring the progression of plant development across the field, harvest times can be optimized so that fruits are picked at their peak development, limiting waste and maximizing return.

Our work leverages aerial imagery and the advances of deep learning to automatically count the number of flowering pineapple plants, which may be in the millions for a single field. We use a counting-by-density-estimation approach to produce a density map of flowers across the field. This approach enables us to determine the number of fruit across all regions of the field and identify areas which are ready for harvest or delayed in development. Our approach produces good results, occasionally better than the human annotations, and is readily amenable to active learning techniques.

*jennifer@intelinair.com

2 Related Work

2.1 Counting Methods

Much of the work in the entity-counting space has grown out of the crowd-counting area [Sindagi and Patel, 2018; Loy *et al.*, 2013]. Most approaches fall under one of three categories: counting by detection, counting by regression, and counting by density prediction [Sindagi and Patel, 2018].

Counting by detection approaches are most applicable when the entities are well separated, occlusions are limited, and the number of entities is small. These may take the form of sliding-window approaches which detect all or part of the entity in question [Dollar *et al.*, 2011; Li *et al.*, 2008] and sum the detections over the entire image. With the success of Deep Learning, many of these traditional approaches have been replaced with neural network-based detection and segmentation algorithms [Ren *et al.*, 2015; He *et al.*, 2017; Redmon *et al.*, 2016], but these new methods still seek to solve the counting problem through the localization of all desired entities in the image. A drawback to these methods is they tend to be computationally heavy, the time complexity often scales with the number of entities detected, and they tend to struggle as occlusion becomes more pronounced.

In contrast, counting by regression approaches eliminate the need to determine precision locations of each entity and seek only to determine the number of entities present [Chen *et al.*, 2012; Ryan *et al.*, 2009; Chan and Vasconcelos, 2009]; these approaches also have benefited tremendously from deep learning based architectures [Wang *et al.*, 2015].

When weak localization in addition to a final count are desired in the presence of high levels of occlusion, density estimation approaches have proven very successful [Lempitsky and Zisserman, 2010; Pham *et al.*, 2015; Xu and Qiu, 2016] especially when combined with deep architectures [Zhang *et al.*, 2016; Boominathan *et al.*, 2016; Onoro-Rubio and López-Sastre, 2016; Sam *et al.*, 2017; Sam *et al.*, 2019]. Many of these leverage fully convolutional neural networks (FCNs) to predict a density [Xie *et al.*, 2018; Ma *et al.*, 2019] across the image; this density can be integrated to provide the count over a region. Because only regional localization is required, these methods tend to outperform detection-based methods in highly occluded scenarios. Additionally, because the output density map is itself a single-channel image, the computational complexity is independent of the number of entities present. Our approach follows these methods as flowers may be occluded by other portions of the plant, and the number of flowers in a given image could be large.

2.2 Counting in Agriculture

Both traditional and deep learning based approaches have been used for a variety of counting-based agricultural applications. The work of [Guo *et al.*, 2018; Ghosal *et al.*, 2019; Malambo *et al.*, 2019] all used detection-based techniques to detect sorghum heads in a field. Similarly [Gené-Mola *et al.*, 2020] used Mask-RCNN to fully identify and segment apples on trees in an orchard. To count palm trees from UAV imagery, [Li *et al.*, 2017] used a CNN-based detection approach. Very recently, [Osco *et al.*, 2020] used an approach very similar to ours to count the number of citrus trees in a grove. Where they sought to count every tree present, in our work we seek to count only those plants who have begun to flower.

3 Methods

3.1 Data Acquisition

Raw imagery was acquired via a DJI Matrice 210 drone equipped with a DJI X3 three band (RGB) camera flown at a height of 200ft above the pineapple fields. Individual images were stitched together using a third party system to produce a single large-scale image for each block. During the stitching process, orthorectification is performed using the RGB image and a digital elevation model (DEM) of the field.

From this full dataset we randomly sampled 866 patches (512×512) across flights over 12 blocks from three fields for annotations. Annotators marked the center of each flower with a point-label, producing 76,659 total point annotations. The data was split such that 650 patches for training and 130 patches for validation were sampled from multiple blocks belonging to an initial set of fields and 106 patches for testing were sampled from blocks belonging to an entirely different set of fields. That is, no field which appeared in the test set appeared in either the training or validation sets.

To produce the target density map, the point labels generated by annotation were blurred using a two-dimensional isotropic Gaussian filter. That is, given an image I_i with pixels p annotated with points $\boldsymbol{P}_i = \{P_1, ..., P_{C(i)}\}\,|P_k \in \mathbf{R}^2$ where $C(i)$ is the total number of points annotated in that image, we define the *ground truth* density map D_i^0 to be a kernel density estimate given by:

$$\forall p \in I_i,\ D_i^0(p) = \sum_{P \in \boldsymbol{P}_i} \mathcal{N}(p; P, \sigma^2 \mathbf{1}_{2\times 2}) \qquad (1)$$

We explored values in $[1, 2, 6, 10, 20]$ for σ, the standard deviation of the Gaussian kernel, and found that $\sigma = 6$ provided the best results both in terms of MSE as well as steps needed for convergence.

3.2 Model

For training, we performed the following augmentation steps: the original sample (and label) was rotated by a random angle and randomly cropped to 256×256. For testing and validation, the original 512×512 patches were split into four non-overlapping 256×256 images.

Our model uses the fully-convolutional encoder-decoder structure of U-net [Ronneberger *et al.*, 2015], taking in the 3 (RGB) input channels and producing a single-channel output corresponding to the flower density (Figure 2). Each convolutional block consisted of 3x3 convolution followed by batch normalization [Ioffe and Szegedy, 2015] and a ReLU nonlinearity. Max Pooling with a 2x2 kernel with a stride of 2 was used in the encoder after every two convolutional blocks. In the decoder, we use a 2x2 transposed convolution for upsampling. We use *same* padding throughout.

The final layer consists of a 1D convolution followed by ReLU activation: this ensures that every point in output layer is positive, which is required by our density prediction task.

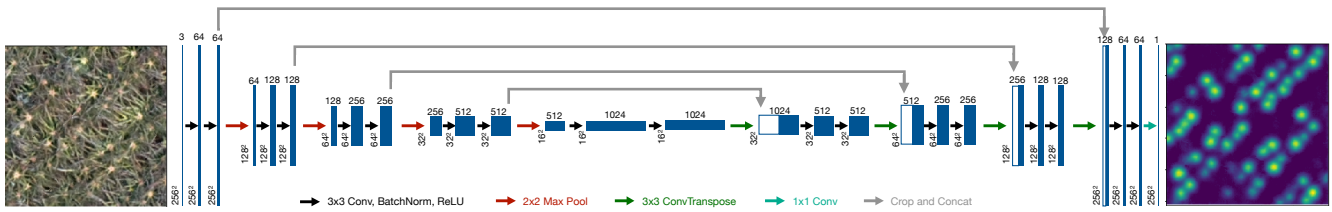

Figure 2: Our architecture follows the encoder-decoder structure of U-Net where the input is an RGB image and the output is a density map.

Note, that the output density is *not* required to be $[0, 1]$, but only positive; if many flowers are located closely together, their densities could add to greater than 1 in some places. In practice, we do not see this occur and therefore a final sigmoid activation could be used in place of the ReLU to enforce a range of $[0, 1]$. However, we find that the final ReLU activation outperforms these alternatives.

We use MSE between the target and predicted density maps as our loss function and is given by

$$MSE_i = \frac{1}{p}\sum^{p} \|D_i(p) - D_i^0(p)\|_2^2$$

where we have abused the notation for p to indicate each pixel in the image I. Adam Optimizer was used with a learning rate of 0.001, $\beta_1 = 0.9, \beta_2 = 0.99$, and weight decay of $1e^{-5}$. The model was trained with a batch size of 10 on a machine equipped with a Tesla P4 for up to 1000 steps; the final model was halted using early stopping after 630 steps.

3.3 Counting

The output of the U-net is a single channel density map of the flowering plants across the field. To get the total count of flowers in a particular region, in this case the sample window, we integrate over the density map to produce the final count.

In some applications, we may desire to extract a discrete location of points from this final density map. We first threshold the image so regions of low density, below γ, are removed. Next, we use a 2D local-max finding algorithm common to most image processing tool-kits to identify peaks requiring a minimum distance of δ between peaks. We find that $\gamma = 0.05$ and $\delta = 4$ work well in practice. The output of this post-processing step can be seen in the top row, right column of Figure 3. Note that because of the filtering applied during this process, the sum over these peaks will always be *less* than the overall predicted count obtained by integrating over the density map.

4 Results and Discussion

Results from our approach are shown in Figure 3. The per-pixel MSE for both validation and test sets was 0.004.

Qualitatively we see the predicted density maps closely resemble the target maps. In certain cases, particularly when the flowers are redder in appearance (corresponding to earlier stages of growth), the outputs of the model occasionally appear more correct than the initial labels. While perhaps seemingly simple on the surface, this is a non-trivial labeling task

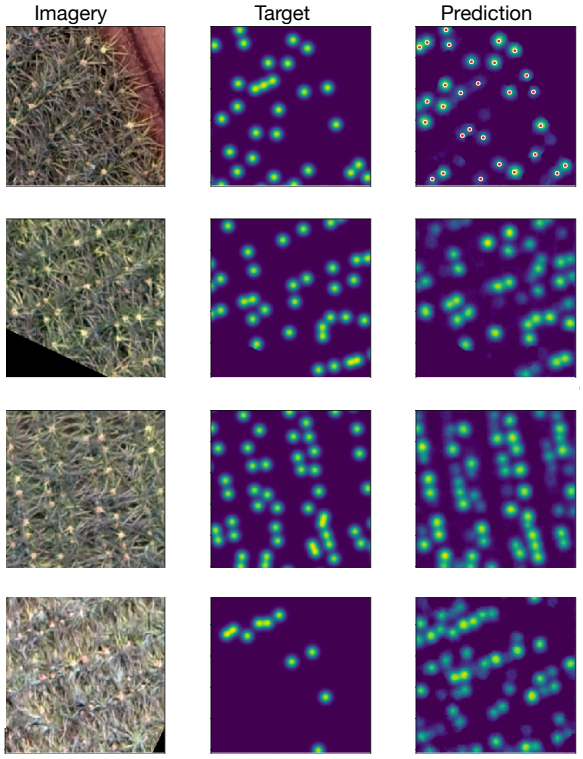

Figure 3: (Left) Input RGB image. (Middle) Target density maps generated from the human point annotations and smoothed with a Gaussian kernel with $\sigma = 6$. (Right) Predicted density map. The output density maps can be filtered to discard regions of very low density and then a peak-finding algorithm applied to determine discrete locations occupied by flowering plants (shown as red circles in the top row only for clarity). Particularly when the flowers are less well defined, the model can be seen to outperform the human annotations (bottom row).

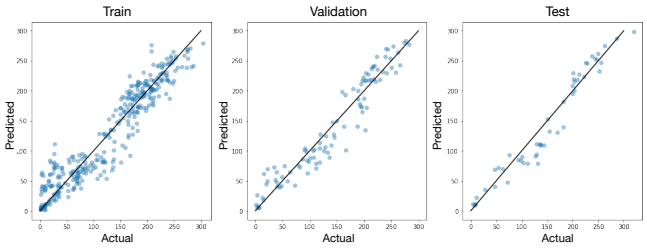

Figure 4: The actual vs. predicted number of flowers shown for each sample in the training and test sets. The black line corresponds to x=y.

given the variety of flower appearances, stages of flower development, density of plant overlap and occlusion. The ability to convert the density map to discrete peak outputs allows these results to be re-proposed to human annotators as part of a Human-in-the-Loop system [Xin *et al.*, 2018]; this reduces the burden on human annotators and enables the model to continue to improve as it is provided additional and more challenging examples. Optimization and impact of this approach is the focus of future work.

Integration of the predicted density maps over the entire image provides us with a prediction of the total flowers. For each original image we compare the actual number of flowers to the number predicted by the model as seen in Figure 4. Because U-net is a fully convolutional network, it is amenable to figures of variable sizes so long as the pooling operations result in integer dimensions. So for this analysis, we inferenced the original 512×512 images without any augmentation (i.e. rotation or cropping) in the training, validation, and test sets. We see that in all three splits, the data falls close to the x=y line with a mean absolute error (MAE) of 13.9.

The computational efficiency of this approach also offers key advantages. At inference, a single sample can be run in under 0.04sec on a single P4 GPU. Especially with appropriate compilation steps which would even further increase efficiency, this speed would enable the model to be run in real-time, potentially allowing for on-the-fly decision making.

5 Conclusion

By leveraging a U-net style deep learning model, we are able to count the number of flowering pineapple plants in a field, which may number in the millions, in a matter of seconds. This automated approach will enable growers to more effectively coordinate their harvesting efforts thereby improving yield efficiency and reducing waste. This in turn provides financial benefits to an industry already under economic stress while also addressing the global food crisis.

Already this approach has yielded solid results with data taken from a relatively small number of blocks. Importantly, the ability to generate discrete point labels from the density map and the already good performance of the model will allow us to rapidly obtain significantly more and better data in a rapid manner. This application is a prime candidate for active learning approaches, which is the focus of ongoing work; as the model is trained on more fields with a wider range of appearances and stages of flower development, we expect the performance to only continue to improve.

Acknowledgements

This material is based on work supported by the USDA SBIR Grant Award 2019-33610-29755. The authors greatly appreciate the collaboration with Dole Hawaii where this research was carried out. The technical help provided by Ms. Gail Uruu is also recognized.

References

[Bartholomew *et al.*,] DP Bartholomew, KG Rohrbach, and DO Evans. Pineapple cultivation in Hawaii. University of Hawaii at Manoa, College of Tropical Agriculture and Human Resources, Cooperative Extension Service, Fruits and Nut Series FN-7.

[Boominathan *et al.*, 2016] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 640–644, 2016.

[Chan and Vasconcelos, 2009] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *2009 IEEE 12th international conference on computer vision*, pages 545–551. IEEE, 2009.

[Chen *et al.*, 2012] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3, 2012.

[Dollar *et al.*, 2011] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2011.

[Gené-Mola *et al.*, 2020] Jordi Gené-Mola, Ricardo Sanz-Cortiella, Joan R Rosell-Polo, Josep-Ramon Morros, Javier Ruiz-Hidalgo, Verónica Vilaplana, and Eduard Gregorio. Fruit detection and 3d location using instance segmentation neural networks and structure-from-motion photogrammetry. *Computers and Electronics in Agriculture*, 169:105165, 2020.

[Ghosal *et al.*, 2019] Sambuddha Ghosal, Bangyou Zheng, Scott C Chapman, Andries B Potgieter, David R Jordan, Xuemin Wang, Asheesh K Singh, Arti Singh, Masayuki Hirafuji, Seishi Ninomiya, et al. A weakly supervised deep learning framework for sorghum head detection and counting. *Plant Phenomics*, 2019:1525874, 2019.

[Guo *et al.*, 2018] Wei Guo, Bangyou Zheng, Andries B Potgieter, Julien Diot, Kakeru Watanabe, Koji Noshita, David R Jordan, Xuemin Wang, James Watson, Seishi Ninomiya, et al. Aerial imagery analysis–quantifying appearance and number of sorghum heads for applications in breeding and agronomy. *Frontiers in plant science*, 9:1544, 2018.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[Lempitsky and Zisserman, 2010] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1324–1332. Curran Associates, Inc., 2010.

[Li *et al.*, 2008] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[Li *et al.*, 2017] Weijia Li, Haohuan Fu, Le Yu, and Arthur Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9(1):22, 2017.

[Loy *et al.*, 2013] Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, simulation and visual analysis of crowds*, pages 347–382. Springer, 2013.

[Ma *et al.*, 2019] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6142–6151, 2019.

[Malambo *et al.*, 2019] Lonesome Malambo, Sorin Popescu, Nian-Wei Ku, William Rooney, Tan Zhou, and Samuel Moore. A deep learning semantic segmentation-based approach for field-level sorghum panicle counting. *Remote Sensing*, 11(24):2939, 2019.

[Onoro-Rubio and López-Sastre, 2016] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016.

[Osco *et al.*, 2020] Lucas Prado Osco, Mauro dos Santos de Arruda, José Marcato Junior, Neemias Buceli da Silva, Ana Paula Marques Ramos, Érika Akemi Saito Moryia, Nilton Nobuhiro Imai, Danillo Roberto Pereira, José Eduardo Creste, Edson Takashi Matsubara, et al. A convolutional neural network approach for counting and geolocating citrus-trees in uav multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160:97–106, 2020.

[Pham *et al.*, 2015] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261, 2015.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Ryan *et al.*, 2009] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Crowd counting using multiple local features. In *2009 Digital Image Computing: Techniques and Applications*, pages 81–88. IEEE, 2009.

[Sam *et al.*, 2017] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4031–4039. IEEE, 2017.

[Sam *et al.*, 2019] Deepak Babu Sam, Neeraj N Sajjan, Himanshu Maurya, and R Venkatesh Babu. Almost unsupervised learning for dense crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8868–8875, 2019.

[Sindagi and Patel, 2018] Vishwanath A Sindagi and Vishal M Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16, 2018.

[Wang *et al.*, 2015] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302, 2015.

[Xie *et al.*, 2018] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3):283–292, 2018.

[Xin *et al.*, 2018] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. Accelerating human-in-the-loop machine learning: challenges and opportunities. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*, pages 1–4, 2018.

[Xu and Qiu, 2016] Bolei Xu and Guoping Qiu. Crowd density estimation based on rich features and random projection forest. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[Zhang *et al.*, 2016] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016.