Deep Regression versus Detection for Counting in Robotic Phenotyping*

Adrian Salazar Gomez[1], Erchan Aptoula[2], Simon Parsons[3] and Petra Bosilj[1]

***Abstract*— Work in robotic phenotyping requires computer vision methods that estimate the number of fruit or grains in an image. To decide what to use, we compared three methods for counting fruit and grains, each method representative of a class of approaches from the literature. These are two methods based on density estimation and regression (single and multiple column), and one method based on object detection. We found that when the density of objects in an image is low, the approaches are comparable, but as the density increases, counting by regression becomes steadily more accurate than counting by detection. With more than a hundred objects per image, the error in the count predicted by detection-based methods is up to 5 times higher than when using regression-based ones.**

***Index Terms*— Computer vision, agriculture, phenotyping.**

I. Introduction

Robotics has the potential to transform many industries where productivity is currently low, such as agriculture [1]. Robotic agriculture also offers benefits in terms of sustainability. A key task in agriculture is *phenotyping*, the process of measuring the observable characteristics of plants, called phenotypes. Phenotypes are used by plant researchers to match new cross-breed crops with certain characteristics, to enable breeding crops with higher yield, better endurance to extreme climate events, and minimal environmental impact.

While phenotyping was traditionally performed manually in the field, the current advances in hardware allow constructing high-throughput phenotyping systems capable of imaging hundreds of plants per day. However, the development of accurate and robust automatic methods that can extract phenotypic information from the collected images poses a bottleneck. Spatial scales of interest vary from the microscopic subcellular level to entire fields. At any given scale, we are interested in both structural and physiological traits [2]. Structural traits measure the geometry (e.g. sizes, shapes, angles) of individual units such as cells, fruits, or whole plants. Physiological traits are often expressed as counts of those individual units, or population-wide measures such as biomass or vegetative greenness [3].

In order to measure structural properties, relying on detection or even segmentation methods is necessary to precisely localise every unit of interest – a fruit, flower, or grain – in order to analyse its structure. Population-wide measures are then obtained by summarising the traits of individual units. However, the performance of these methods, and consequently the quality of estimated traits, degrades as the number of individual units per image increases [4]. This is especially true in situations typically encountered in the field where plants are packed closely together, creating highly dense collections of objects with prominent levels of intra-occlusion, noise and, due to biological variation, heterogeneity across the objects (see Fig. 1). In contrast, counting by regression approaches learn to map the image features into density distributions across the whole image, which can be used to create a count. They are typically used for counting crowds in congested scenarios as they do not need to fully identify objects to count them. We compare the performance of *counting by detection* and *counting by regression* for uses in phenotyping [5]. Specifically, we investigate whether it is still possible to estimate accurate population-wide counts in complex scenes, where detection methods fail to identify every unit of interest. This enables measuring a portion of phenotypic information from images at a coarser scale, leading to simpler image acquisition setups.

Fig. 1. Images from phenotyping applications at a range of spatial scales. Clockwise from top left: sorgum, maize, apples, olives, grapes, wheat and strawberries.

We are not the first to make this connection [6], but we go further than previous work in our analysis of the performance of regression methods in agriculture. We investigate the robustness of different counting approaches to varying object density on multiple agricultural datasets with a considerable density disparity. The results show that regression approaches are capable of counting agricultural objects with low error across the available range of object densities. In contrast,

*This work was supported by Lincoln Agri-Robotics as part of the Expanding Excellence in England (E3) Programme.

[1]Adrian Salazar Gomez and Petra Bosilj are with the Lincoln Institute of Agri-food Technology, University of Lincoln, UK. {asalazargomez, pbosilj}@lincoln.ac.uk

[2]Erchan Aptoula is with the Institute of Information Technologies of Gebze Technical University, Turkey. eaptoula@gtu.edu.tr

[3]Simon Parsons is with the Lincoln Centre for Autonomous Systems, University of Lincoln, UK. sparsons@lincoln.ac.uk

counting by detection becomes less accurate as object density increases. Our main contributions are:

- an empirical analysis of the impact of object density on different approaches to counting[1],
- an exhaustive assessment of counting approaches across existing agricultural datasets,
- the comparison of a range of different datasets that can be used for counting and detection in agriculture.

The rest of the paper is organised as follows: Section II presents related work on counting. Section III details the detection and regression approaches we used for the experiments described in Section IV. Results are discussed in Section V, and conclusions in Section VI.

II. Related work

Counting by regression methods are categorised into two groups: global regression and density estimation. Global regression refers to the straightforward definition of regression methods where handcrafted [7] or learned [8], [9] features are mapped directly into a numerical value. In contrast, density-based methods train a regressor to map the local image features into an object density map. These approaches are typically used for counting in highly congested scenarios such as crowd counting [10] and traffic monitoring [11].

Density-based methods were originally proposed to count cells [5], and relied on handcrafted features. These initial methods were superseded by CNN-based structures with richer feature representations. The first CNN-based method, Counting CNN (CCNN) [11], uses a simple deep convolutional structure to map an image into a density map. Hydra CNNS [11] and Multi-column CNN [12] pioneered the density based approaches with multiple columns, where the outputs of different CCNN branches, also known as "heads", are merged to create a density map. Multi-column approaches can thus learn both multi-scale and multi-perspective features. Recent approaches adaptively weigh the output of the columns to account for variations in the environment; leading to state-of-the-art methods such as Switch-CNN [13] and context aware networks [10] for crowd-counting.

Instead of using multiple heads, single-column approaches use single deeper convolutional structures to compute density maps. These CNNs can learn patterns, scales, and perspectives [14] while reducing the risk that multi-column methods have of learning redundant structures.

In agricultural applications, counting-by-regression has received less attention than counting-by-detection. Detection methods such as Faster R-CNN [15] are widely used to provide counts in plant phenotyping. This includes counting objects in farms from aerial [16] and ground perspectives [17] and counting different kinds of objects such as wheat spikes [16] and fruits [18]. A common drawback of these tools is their inability to detect items in clustered regions.

In contrast, the application of global regression methods in agriculture is currently limited to phenotype extraction from single plant images [19]. Both single and multi-column density-based aproaches have shown recent promise in agriculture for counting sorghum heads [6].

[1]The code to reproduce the experiments can be found here `https://github.com/adrianxsalazar/Deep_Regression_vs_Detection_for_Counting_in_Robotic_Phenotyping`

III. Methodology

The suitability of counting methods for a range of agricultural products has not been analysed as all previous work was focused on a single crop. In addition, the literature points to problems in using counting-by-detection approaches in dense scenarios [17] while density-based methods appear to handle dense images well. However, there is no existing comparative analysis of the impact of density on the counting paradigms, and no data that identifies the admissible levels of density for such methods, and thus no guidance on which methods to choose in a given situation.

To fill this gap, we compared three counting methods, one from each of the main counting paradigms, across multiple agricultural datasets. Single column density approaches, multi-column density approaches, and counting by detection are represented by, respectively, CSRNet [14], context aware network (CAN) [10], and Faster R-CNN [15].

A. Single-column density regression

CSRNet [14] is a single-column density-based CNN with two main parts. The first part uses VGG16 [26], pre-trained on ImageNet, to extract features from the images. The second uses a network made of dilated convolutional layers to up-sample those features into a density map. The loss function is the Euclidean distance between the ground truth and the estimated density map:

$$L(\theta) = \frac{1}{2B}\sum_{i=1}^{B}\left\|D_i^{est} - D_i^{gt}\right\|_2^2, \tag{1}$$

where θ is a set of learnable parameters, B is the batch size, D_i^{gt} is the ground truth density map and D_i^{est} is the estimated density map for the image i.

B. Multi-column density regression

CAN [10] is a multi-column approach with three parts: a multi-column feature extractor; a weight map extractor, and an upsampling network. The first part consists of the first ten layers of a pre-trained VGG16 which learn to extract a feature map. The multi-scale information is extracted from the feature map with a *Spatial Pyramid Pooling* (SPP) structure [27] with S columns. Each column $j \in \{1, \ldots, S\}$ extracts information at different scales by averaging the feature map into scale-aware blocks with fixed dimensions $d_j \times d_j$. In each column j, the scale-aware blocks are followed by a 1×1 convolutional network. Then, the blocks are up-sampled with bi-linear interpolation to match the dimensions of the feature map, resulting in a scale feature map s_j for each column j:

$$s_j = U_{bi}(\mathcal{F}_j(SPP_{averaging}(f_v, d_j), \theta_j)). \tag{2}$$

f_v represents the feature map, U_{bi} the bi-linear interpolation process, $\mathcal{F}_j$ the 1×1 convolutional layer in each column j with parameters θ_j, and $SPP_{averaging}(f_v, d_j)$ the process of

TABLE I

THE AGRICULTURAL DATASETS USED IN THE EXPERIMENTS. PUBLICLY AVAILABLE DATASETS ARE REFERENCED.

Dataset	Content	Dimensions	Number Images	Number Items	Max objects per picture	Min objects per picture	Average objects	Item deviation	Average objects per pixel (10^5)	Deviation objects per pixel (10^5)
SORGHUM-HEAD [20]	sorghum heads	330×1300	1,309	135,834	142	46	103.7	14.6	24.1	3.4
ACFR ALMONDS [17]	almonds	300×300	620	4,777	37	0	7.7	6.0	8.3	7.3
ACFR APPLES [17]	apples	308×202	1,120	5,765	18	0	5.1	3.3	8.1	5.4
ACFR MANGOES [17]	mangoes	500×500	1,964	7,065	23	0	3.5	3.7	1.4	1.4
ISAR ALMONDS [21]	almonds	300×300	3,060	2,242	25	0	0.7	2.0	0.8	2.1
ISAR OLIVES [21]	olives	606×403	2,807	6,967	65	0	2.4	5.4	0.9	2.1
MTC [22]	maize tassels	3648×2736	361	13,562	120	0	37.5	26.9	1.5	1.02
WGISD [23]	grape clusters	2048×1365	300	4,430	34	2	14.7	5.4	0.5	0.1
GWHD [24]	wheat heads	1024×1024	3,422	147,793	116	0	43.1	20.8	4.1	1.9
STRAWBERRY I	strawberries	4032×3024	179	4,243	71	3	23.7	11.21	0.9	0.47
RISEHOLME130	strawberries	1920×1080	130	3,605	69	7	27.7	12.3	1.7	0.7
MINNEAPPLE [25]	apples	720 × 1280	670	28,182	123	1	42.06	29.05	4.8	3.2

averaging the feature map into a specific scale-aware block of size $d_j \times d_j$.

The second part extracts the importance of each scale's feature map s_j at each spatial location via a weight map w_j for each s_j. Each weight map is learned with a 1×1 convolutional layer $\mathcal{F}_c^j$ with parameters θ_c^j followed by a sigmoid function σ that maps a contrast feature map $c_j = s_j - f_v$ into the weight map. The contrast map provides information about the extent at which s_j changes with respect to the original feature map at each location. The network outputs are the weight maps w_j, calculated using a contextual map: $w_j = \sigma(\mathcal{F}_c^j(c_j, \theta_c^j))$. The final feature map f_I is obtained by concatenating the weighted scale maps with the initial feature map:

$$f_I = \left[f_v \mid \frac{\sum_{j=1}^{S} w_j \odot s_j}{\sum_{j=1}^{S} w_j} \right], \tag{3}$$

where $\odot$ is the element-wise product between the weight map and the corresponding scale feature map s_j, and $[\cdot \mid \cdot]$ is a channel-wise concatenation. Finally, a structure made of dilated convolutions upsamples f_I into a density map.

C. Object detection

Faster R-CNN [15] has two key component. A Region Proposal Network (RPN), which identifies the Regions of Interest (RoI), and a classification network that refines the content of the proposed regions.

An initial feature extractor made of a sequence of convolutional layers learns to extract a feature map from the input images. Then, the RPN network, made of two sibling fully connected regression and classification layers, proposes a series of class agnostic object proposals using the feature map. These proposals are then mapped back into the feature map with attention mechanisms before classification. The classification is based on an R-CNN architecture [28], consisting of fully connected regression and classification layers that refine the class and coordinates of the proposal. Finally, the refined object proposals are accepted and counted if both the probability of the classification layer and the area that overlaps with other detection proposals are over a threshold.

For this experiment, the implemented feature extractor consists of a ResNet101 [29] based on a feature pyramid network (FPN) [30] pretrained on 37 COCO [31] epochs.

D. Evaluation metrics

Conventional metrics [11]–[13] to evaluate counting methodologies are *mean absolute error* (MAE) and *root mean squared error* (RMSE) which are defined as:

$$MAE = \frac{1}{N} \sum_{n=1}^{N} |Ci - Ci^{gt}|, \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (C_i - C_i^{gt})^2}, \tag{5}$$

where *N* is the number of images, C_i^{gt} is the ground truth count and C_i is the estimated count for image i. In density based methods, C_i is obtained by integrating the pixels in the predicted density map D_i^{est}:

$$C_i = \sum_{l=1}^{L} \sum_{w=1}^{W} D_i^{est}(p_{l,w}|I_i) \tag{6}$$

where L and W represents the length and width of the density maps obtained from the image I_i and $p_{l,w}$ is the pixel of the density map at location (l, w).

IV. EXPERIMENTS

A. Datasets

We assess the performance of counting methodologies on 12 agricultural datasets. Table I summarises their characteristics in terms of the crop imaged, number and size of images, and the distribution of objects in the dataset. Out of the 12 datasets, ten are publicly available while STRAWBERRY I and RISEHOLME130 are internal datasets collected at the University of Lincoln, UK. Most of these datasets were annotated and used for object detection, while only MTC [22] was previously used for counting.

The 12 datasets cover nine different crops and vary in the number of images they contain, from 130 to 3060; the size of the images, from $308\,\text{px} \times 202\,\text{px}$ to $4032\,\text{px} \times 3024\,\text{px}$; and in spatial scale. The average distribution metrics also vary from 103.7 objects per image and 24.1×10^5 objects per px in the densest SORGHUM-HEAD dataset [20] to 0.7 objects per image and 0.8×10^5 objects per px in the least dense ISAR ALMONDS [21] dataset. This allows us to analyse the impact of reduced dataset sizes, varying image sizes and object densities, which are common in agricultural applications, on different approaches for counting.

B. Data preparation

All the datasets contain bounding box annotations. In density-based methods, bounding boxes' centres were used as object locations x_i when generating the ground truth. Each dataset is randomly divided into training, testing and validation sets containing 70%, 20%, and 10% of dataset images respectively. The same sets are used in all experiments.

C. Ground truth generation

Applying density-based methods starts from a set of annotated pixels x_i indicating the location of the objects. To obtain the density map, each annotation pixel x_i is blurred with Gaussian kernels normalised to 1 so that kernel integration returns the number of objects, as per Eq. (6). We use a Gaussian geometry-adaptive kernel G_σ that considers the size distortion caused by imaging 3D scenes, and thus approximating the real object sizes in the density map [12]. To do so, it is assumed that the objects that are close together are evenly distributed. Then, the kernel uses the average distance between the item x_i and its k nearest neighbours to estimate the geometric distortion and modify the kernel size accordingly. Fig. 2 shows an example of the generation of a density map with and without the proposed kernel. The implemented kernel is defined as:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), \text{ with } \sigma_i = \beta \bar{d}_i. \quad (7)$$

For each annotation pixel x_i, the Gaussian geometry-adaptive kernel G_σ convolves the ground truth $\delta(x - x_i)$ with a standard deviation of σ_i. Where x is the position of the item in the image and δ a Dirac Delta function representing the object area. The kernel σ_i depends on $\bar{d}_i$ which is the average distance of the item i to the k closest items and a hyperparameter β that ranges from 0 to 1 and regulates the impact of the neighbours on the size distortion of the objects.

D. Parameters

For the creation of the ground truth density maps, we follow the original configuration for Gaussian geometry-adaptive kernels [12] with $\beta = 0.3$ and $k = 3$.

For all three methods, we used standard training parameters and architectures from the literature [10], [14], [15]. For CSRNet, the dilation rate in upsampling is set to 2. There is no data augmentation. Stochastic gradient descent (SGD) is used with a learning rate of 10^{-7}, batch size 1, a decay of 5×10^{-4}, and a momentum of 0.95. For CAN we set $S = 4$, with $d_j \in \{1, 2, 3, 6\}$, and used the loss function in Eq. (1). Learning uses SGD, with a learning rate of 10^{-4}, batch size 1, and a decay of 5×10^{-4}. For Faster R-CNN we used the standard parameter set from [15]. This was a learning rate of 2.5×10^{-3}, a weight decay of 10^{-4}, and a momentum of 0.9. We set the number of proposed bounding boxes per image to $N_p = 300$, and the probability threshold to 0.5. For two additional parameters — the set of anchors and the threshold value used for non-maximum suppression (see, for example, [32]) (NMS) — we tuned the values for individual datasets on the relevant training data. For NMS we performed a gridsearch and for the anchor set we followed the approach in [33]. We then conducted our experiments using both the original values from [15] and the values that we established through tuning. We found that Faster R-CNN performed best on all datasets using tuned values of the NMS threshold and the anchor set from [15], and these are the values that we report below.

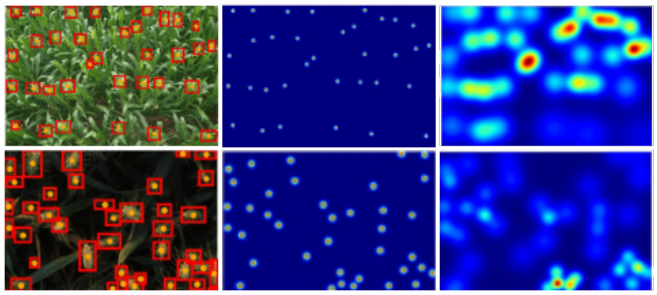

Fig. 2. A visualization of the process of generating the ground-truth density maps with and without a geometry-adaptive kernel. The top row shows the process on a image of the MINNEAPPLE [25] dataset and the second row an image from the GWHD dataset [24]. From left to right, the raw images with the bounding boxes and pixel annotations, density maps made with a non geometry-adaptive kernel, density maps made with a geometry-adaptive kernel.

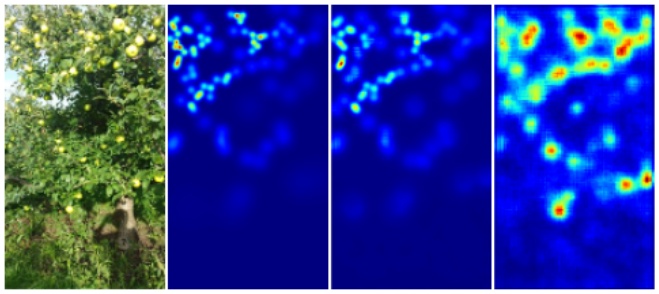

Fig. 3. Sample of the predicted density maps. From left to right, the figures shows the original image from the MINNEAPLE dataset [25], the ground truth density map, the CAN density map, and the CSRNet density map

V. RESULTS

Table II shows the results of the proposed methods across all the datasets. CAN performs best in seven of the 12 datasets for both metrics, showing the capability of multi-column regression networks in agricultural settings. Faster R-CNN performs best on five datasets, indicating that detection approaches can be competitive under specific conditions.

CRSNet is the worst performing method. We believe that this is because deeper architectures like CSRNet suffer from a larger degree of overfitting on the available modestly-sized datasets [14][2]. Fig. 3 shows the output of the single-column approach over-estimating the density of the objects w.r.t. the ground truth density map.

[2]This is an inherent feature of the available agricultural image datasets at this point in time — they are simply several orders of magnitude smaller than many other publicly available datasets. If you want to work in agriculture, you have to find ways to deal with this limitation.

TABLE II

COUNTING PERFORMANCES IN TERMS OF MAE AND RMSE. THE BEST PERFORMANCES ARE HIGHLIGHTED.

Dataset	CSRNet		CAN		Faster R-CNN	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
SORGHUM-HEAD	3.88	5.07	**2.96**	**4.026**	8.76	12.44
ACFR ALMONDS	2.31	3.90	**2.15**	**3.07**	2.16	3.43
ACFR APPLES	1.36	3.08	**1.01**	**1.41**	1.22	1.79
ACFR MANGOES	2.46	3.33	0.74	1.19	**0.63**	**1.17**
ISAR ALMOND	0.94	1.76	0.34	8.27	**0.33**	**0.90**
ISAR OLIVES	2.70	4.80	**1.09**	**2.78**	2.34	5.51
MTC	20.18	24.15	**3.70**	**5.23**	4.06	6.61
WGSID	4.3	4.7	1.93	**2.42**	**1.85**	2.57
GWHD	7.28	10.22	**2.38**	**3.36**	3.08	4.26
STRAWBERRY I	6.05	8.53	3.04	3.62	**1.34**	**1.96**
RISEHOLME130	7.04	7.6	4.97	5.96	**3.19**	**4.27**
MINNEAPPLE	13.36	16.67	**3.57**	**5.74**	5.79	9.27

The results for Faster R-CNN are more variable across datasets compared with CAN. The datasets on which Faster R-CNN outperforms the other methods are identified in Table I as the ones with the lowest average number of items per image. Conversely, in the datasets with higher average number of objects, the counting performance of Faster R-CNN drastically drops. In the dataset with the highest average of number of objects per image, SORGHUM-HEAD, the MAE and RMSE for Faster-RCNN are much higher than for CAN and CRSNet.

There are many factors that might result in the varying performance of each counting method across datasets shown in Table II. However, plotting MAE against the average number of objects in an image from a given dataset, as in Fig. 4, suggests that the number of objects being counted is significant. Indeed, the performance of all the methods decreases as the number of objects increases. However the average numbers of objects in Fig. 4 does not reveal the extent of the dependence between performance and number of objects in an image. Since the variance in the number of objects in an image is high (Table I) in many of the datasets, plotting error against the average number of objects in an image from a given dataset does not tell us much about how the error depends on the number of objects in an image.

To better understand the relationship between error and number of objects, we divided the test set of each dataset into subsets grouping images with a similar number of objects together. In addition to evaluating all the trained models on each dataset as a whole, we also report the evaluation results of our models for each separate subset of each dataset where the number of objects in an image falls into a fixed interval (shown in Fig. 5).

All methods perform similarly well when the number of objects per image is small, and the error rate increases with the number of objects. The impact is moderate on the performance of both CAN (ranging between 0.22 to 7.56 MAE) and CSRNet (ranging between 1.19 to 15.37 MAE). In contrast, when the number of objects per image exceeds 95, Faster R-CNN's ability to count objects decreases drastically (reaching the MAE of 41 for > 140 objects per image). This shows that flexibility of density-based methods is particularly effective in images with large

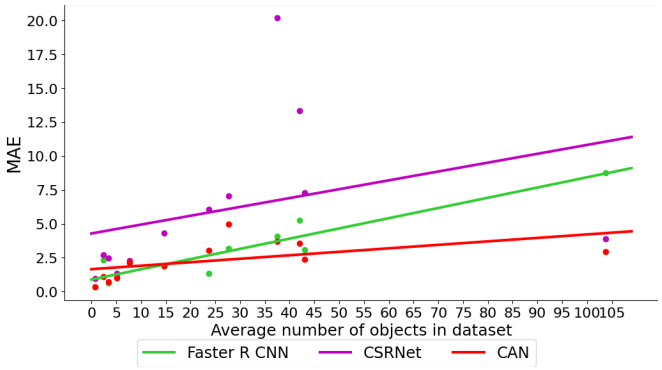

Fig. 4. MAE plotted against the average number of objects per image in each of the datasets. Points give the performance of each approach on a dataset against the average number of objects per image. Lines are a regression on the points.

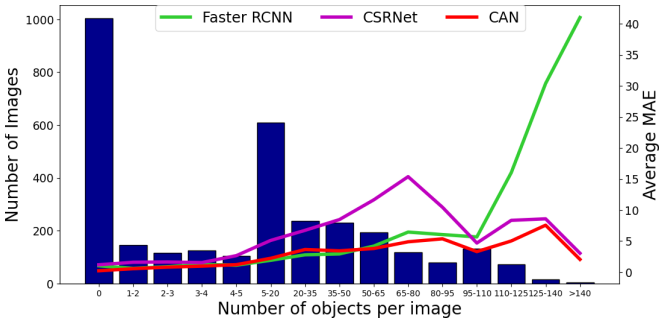

Fig. 5. MAE plotted against number of objects per image.

number of objects where we expect high levels of occlusion and a lack of object heterogeneity.

Since different datasets have different sized images, the number of objects per image might not be an independent factor, but a function of image size. This raises the question as to how much the *density* of objects, expressed as the number of objects adjusted for image size, impacts performance. As there is no established measure of density for this analysis, we used the number of objects *per pixel*.

Using this measure of density, we repeated the previous analysis, obtaining Figs. 6 and 7. The headline result is similar to that for the analysis based on number of objects: Faster R-CNN performs worse than the density-based methods when the object density exceeds 20×10^{-5} objects per pixel. This reaffirms the ability of density-based methods to count objects despite high levels of occlusion and variability.

VI. CONCLUSION

This paper studied the comparative performance of different approaches to counting objects on tasks related to plant phenotyping, laying the groundwork for the development of robotic phenotyping platforms. There are three main contributions. First, we assembled and compared the features of a range of datasets that can be used for experiments on counting in an agricultural setting. Second, we provided an exhaustive assessment of counting approaches across these datasets. Third, we gave an empirical analysis of the impact of object density on the approaches. This analysis shows that

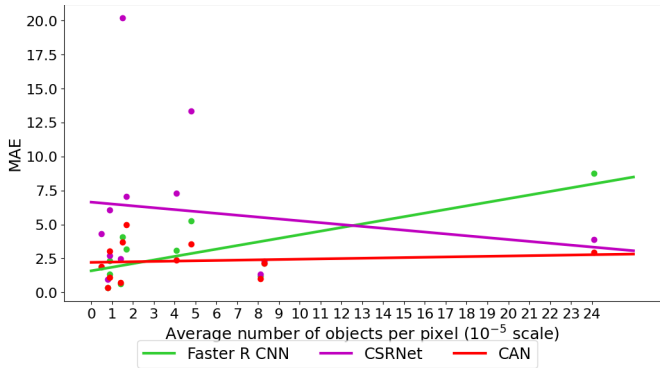

Fig. 6. MAE plotted against the average number of objects per pixel across the images in each dataset. Points give the performance of each approach on a dataset against the average number of objects per image. Lines are a regression on the points.

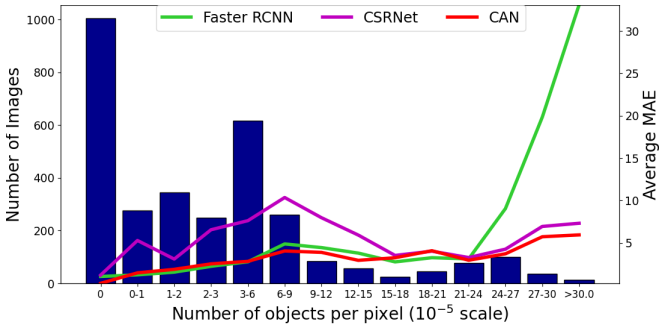

Fig. 7. MAE plotted against number of objects per pixel.

counting-by-regression outperforms counting-by-detection at high object densities, specifically when images contain more than around 100 objects or more than 20×10^{-5} objects per pixel, while giving comparable performance at lower densities. This leads us to conclude that regression-based methods are a good approach for accurate counting across a wide range of object densities, and thus have a role in obtaining count-based population-wide measures in phenotyping.

ACKNOWLEDGMENT

This work was partially supported by the Lincoln Agri-Robotics grant from Research England.

REFERENCES

[1] T. Duckett, S. Pearson, S. Blackmore, and B. Grieve, "Agricultural robotics: The future of robotic agriculture," *arXiv:1806.06762*, 2018.

[2] M. Minervini, H. Scharr, and S. A. Tsaftaris, "Image analysis: the new bottleneck in plant phenotyping [applications corner]," *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. 126–131, 2015.

[3] N. Fahlgren, M. A. Gehan, and I. Baxter, "Lights, camera, action: high-throughput plant phenotyping is ready for a close-up," *Current Opinion in Plant Biology*, vol. 24, pp. 93–99, 2015.

[4] N. Häni, P. Roy, and V. Isler, "Apple counting using convolutional neural networks," in *IEEE IROS*, 2018, pp. 2559–2565.

[5] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in NIPS*, 2010, pp. 1324–1332.

[6] M.-h. Oh, P. Olsen, and K. N. Ramamurthy, "Counting and segmenting sorghum heads," *arXiv:1905.13291*, 2019.

[8] S. Seguí, O. Pujol, and J. Vitria, "Learning to count with deep object features," in *IEEE CVPR*, 2015, pp. 90–96.

[7] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *2009 Digital Image Computing: Techniques and Applications*, 2009, pp. 81–88.

[9] E. Aptoula and S. Ariman, "Hierarchical spatial-spectral features for the chlorophyll-a estimation of Lake Balik, Turkey," *IEEE GRSL*, 2020.

[10] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *IEEE CVPR*, 2019, pp. 5099–5108.

[11] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *IEEE ECCV*, 2016, pp. 615–629.

[12] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *IEEE CVPR*, 2016, pp. 589–597.

[13] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *IEEE CVPR*, 2017, pp. 4031–4039.

[14] Y. Li, X. Zhang, and D. Chen, "CSRnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *IEEE CVPR*, 2018, pp. 1091–1100.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in NIPS*, 2015, pp. 91–99.

[16] P. Sadeghi-Tehran, N. Virlet, E. M. Ampe, P. Reyns, and M. J. Hawkesford, "DeepCount: In-field automatic quantification of wheat spikes using simple linear iterative clustering and deep convolutional neural networks," *Frontiers in Plant Science*, vol. 10, p. 1176, 2019.

[17] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *IEEE ICRA*, 2017, pp. 3626–3633.

[18] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE RAL*, vol. 2, no. 2, pp. 781–788, 2017.

[19] J. Ubbens, M. Cieslak, P. Prusinkiewicz, and I. Stavness, "The use of plant models in deep learning: an application to leaf counting in rosette plants," *Plant Methods*, vol. 14, no. 1, p. 6, 2018.

[20] W. Guo, B. Zheng, A. B. Potgieter, J. Diot, K. Watanabe, K. Noshita, D. R. Jordan, X. Wang, J. Watson, S. Ninomiya *et al.*, "Aerial imagery analysis–quantifying appearance and number of sorghum heads for applications in breeding and agronomy," *Frontiers in Plant Science*, vol. 9, p. 1544, 2018.

[21] E. Bellocchio, T. A. Ciarfuglia, G. Costante, and P. Valigi, "Weakly supervised fruit counting for yield estimation using spatial consistency," *IEEE RAL*, vol. 4, no. 3, pp. 2348–2355, 2019.

[22] H. Lu, Z. Cao, Y. Xiao, Z. Fang, and Y. Zhu, "Towards fine-grained maize tassel flowering status recognition: dataset, theory and practice," *Applied Soft Computing*, vol. 56, pp. 34–45, 2017.

[23] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Comp. Elec. in Agri*, vol. 170, 2020.

[24] E. David *et al.*, "Global wheat head detection (GWHD) dataset: a large and diverse dataset of high resolution RGB labelled images to develop and benchmark wheat head detection methods," *arXiv:2005.02162*, 2020.

[25] N. Häni, P. Roy, and V. Isler, "Minneapple: A benchmark dataset for apple detection and segmentation," *IEEE RAL*, vol. 5, no. 2, pp. 852–858, 2020.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE PAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[28] R. Girshick, "Fast R-CNN," in *IEEE ICCV*, 2015, pp. 1440–1448.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv:1602.07261*, 2016.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE CVPR*, 2017, pp. 2117–2125.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the IEEE ECCV*, 2014, pp. 740–755.

[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE PAMI*, vol. 32, no. 9, pp. 1627–1645, 2009.

[33] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *IEEE CVPR*, 2017, pp. 7263–7271.