Locating Objects Without Bounding Boxes

Javier Ribera, David Güera, Yuhao Chen, Edward J. Delp
Video and Image Processing Laboratory (VIPER), Purdue University

Abstract

Recent advances in convolutional neural networks (CNN) have achieved remarkable results in locating objects in images. In these networks, the training procedure usually requires providing bounding boxes or the maximum number of expected objects. In this paper, we address the task of estimating object locations without annotated bounding boxes which are typically hand-drawn and time consuming to label. We propose a loss function that can be used in any fully convolutional network (FCN) to estimate object locations. This loss function is a modification of the average Hausdorff distance between two unordered sets of points. The proposed method has no notion of bounding boxes, region proposals, or sliding windows. We evaluate our method with three datasets designed to locate people's heads, pupil centers and plant centers. We outperform state-of-the-art generic object detectors and methods fine-tuned for pupil tracking.

1. Introduction

Locating objects in images is an important task in computer vision. A common approach in object detection is to obtain bounding boxes around the objects of interest. In this paper, we are not interested in obtaining bounding boxes. Instead, we define the object localization task as obtaining a single 2D coordinate corresponding to the location of each object. The location of an object can be any key point we are interested in, such as its center. Figure 1 shows an example of localized objects in images. Differently from other keypoint detection problems, we do not know in advance the number of keypoints in the image. To also make the method as generic as possible we do not assume any physical constraint between the points, unlike in cases such as pose estimation. This definition of object localization is more appropriate for applications where objects are very small, or substantially overlap (see the overlapping plants in Figure 1). In these cases, bounding boxes may not be provided by the dataset or they may be infeasible to groundtruth.

Bounding-box annotation is tedious, time-consuming and expensive [37]. For example, annotating ImageNet [43] required 42 seconds per bounding box when crowdsourcing on Amazon's Mechanical Turk using a technique specifically developed for efficient bounding box annotation [50]. In [6], Bell *et al*. introduce a new dataset for material recognition and segmentation. By collecting click location labels in this dataset instead of a full per-pixel segmentation, they reduce the annotation costs an order of magnitude.

Figure 1. Object localization with human heads, eye pupils and plant centers. (Bottom) Heat map and estimations as crosses.

In this paper, we propose a modification of the average Hausdorff distance as a loss function of a CNN to estimate the location of objects. Our method does not require the use of bounding boxes in the training stage, and does not require to know the maximum number of objects when designing the network architecture. For simplicity, we describe our method only for a single class of objects, although it can trivially be extended to multiple object classes. Our method is object-agnostic, thus the discussion in this paper does not include any information about the object characteristics. Our approach maps input images to a set of coordinates, and we validate it with diverse types of objects. We evaluate our method with three datasets. One dataset contains images acquired from a surveillance camera in a shopping mall, and we locate the heads of people. The second dataset contains images of human eyes, and we locate the center of the pupil. The third dataset contains aerial images of a crop field taken

from an Unmanned Aerial Vehicle (UAV), and we locate the centers of highly occluded plants.

Our approach to object localization via keypoint detection is not a universal drop-in replacement for bounding box detection, specially for those tasks that inherently require bounding boxes, such as automated cropping. Also, a limitation of this approach is that bounding box labeling incorporates some sense of scale, while keypoints do not.

The contributions of our work are:

- We propose a loss function for object localization, which we name *weighted Hausdorff distance* (WHD), that overcomes the limitations of pixelwise losses such as L^2 and the Hausdorff distances.
- We develop a method to estimate the location and number of objects in an image, without any notion of bounding boxes or region proposals.
- We formulate the object localization problem as the minimization of distances between points, independently of the model used in the estimation. This allows to use any fully convolutional network architectural design.
- We outperform state-of-the-art generic object detectors and achieve comparable results with crowd counting methods without any domain-specific knowledge, data augmentation, or transfer learning.

2. Related Work

Generic object detectors. Recent advances in deep learning [16, 27] have increased the accuracy of localization tasks such as object or keypoint detection. By generic object detectors, we mean methods that can be trained to detect any object type or types, such as Faster-RCNN [15], Single Shot MultiBox Detector (SSD) [31], or YOLO [40]. In Fast R-CNN, candidate regions or proposals are generated by classical methods such as selective search [59]. Although activations of the network are shared between region proposals, the system cannot be trained end-to-end. Region Proposal Networks (RPNs) in object detectors such as Faster R-CNN [15, 41] allow for end-to-end training of models. Mask R-CNN [18] extends Faster R-CNN by adding a branch for predicting an object mask but it runs in parallel with the existing branch for bounding box recognition. Mask R-CNN can estimate human pose keypoints by generating a segmentation mask with a single class indicating the presence of the keypoint. The loss function in Mask R-CNN is used location by location, making the keypoint detection highly sensitive to alignment of the segmentation mask. SDD provides fixed-sized bounding boxes and scores indicating the presence of an object in the boxes. The described methods either require groundtruthed bounding boxes to train the CNNs or require to set the maximum number of objects in the image being analyzed. In [19], it is observed that generic object detectors such as Faster R-CNN and SSD perform very poorly for small objects.

Counting and locating objects. Counting the number of objects in an image is not a trivial task. In [28], Lempitsky *et al*. estimate a density function whose integral corresponds to the object count. In [47], Shao *et al*. proposed two methods for locating objects. One method first counts and then locates, and the other first locates and then counts.

Locating and counting people is necessary for many applications such as crowd monitoring in surveillance systems, surveys for new businesses, and emergency management [28, 60]. There are multiple studies in the literature, where people in videos of crowds are detected and tracked [2, 7]. These detection methods often use bounding boxes around each human as ground truth. Acquiring bounding boxes for each person in a crowd can be labor intensive and imprecise under conditions where lots of people overlap, such as sports events or rush-hour agglomerations in public transport stations. More modern approaches avoid the need of bounding boxes by estimating a density map whose integral yields the total crowd count. In approaches that involve a density map, the label of the density map is constructed from the labels of the people's heads. This is typically done by centering Gaussian kernels at the location of each head. Zhang *et al*. [62] estimate the density image using a multi-column CNN that learns features at different scales. In [44], Sam *et al*. use multiple independent CNNs to predict the density map at different crowd densities. An additional CNN classifies the density of the crowd scene and relays the input image to the appropriate CNN. Huang *et al*. [20] propose to incorporate information about the body part structure to the conventional density map to reformulate the crowd counting as a multi-task problem. Other works such as Zhang *et al*. [61] use additional information such as the groundtruthed perspective map.

Methods for pupil tracking and precision agriculture are usually domain-specific. In pupil tracking, the center of the pupil must be resolved in images obtained in real-world illumination conditions [13]. A wide range of applications, from commercial applications such as video games [52], driving [48, 17] or microsurgery [14] rely on accurate pupil tracking. In remote precision agriculture, it is critical to locate the center of plants in a crop field. Agronomists use plant traits such as plant spacing to predict future crop yield [56, 51, 57, 12, 8], and plant scientists to breed new plant varieties [3, 35]. In [1], Aich *et al*. count wheat plants by first segmenting plant regions and then counting the number of plants in each segmented patch.

Hausdorff distance. The Hausdorff distance can be used to measure the distance between two sets of points [5]. Modifications of the Hausdorff distance [10] have been used for various multiple tasks, including character recog-

nition [33], face recognition [23] and scene matching [23]. Schutze *et al*. [46] use the average Hausdorff distance to evaluate solutions in multi-objective optimization problems. In [24], Elkhiyari *et al*. compare features extracted by a CNN according to multiple variants of the Hausdorff distance for the task of face recognition. In [11], Fan *et al*. use the Chamfer and Earth Mover's distance, along with a new neural network architecture, for 3D object reconstruction by estimating the location of a fixed number of points. The Hausdorff distance is also a common metric to evaluate the quality of segmentation boundaries in the medical imaging community [54, 63, 30, 55].

3. The Average Hausdorff Distance

Our work is based on the Hausdorff distance which we briefly review in this section. Consider two unordered non-empty sets of points X and Y and a distance metric $d(x, y)$ between two points $x \in X$ and $y \in Y$. The function $d(\cdot, \cdot)$ could be any metric. In our case we use the Euclidean distance. The sets X and Y may have different number of points. Let $\Omega \subset \mathbb{R}^2$ be the space of all possible points. In its general form, the Hausdorff distance between $X \subset \Omega$ and $Y \subset \Omega$ is defined as

$$d_{\mathrm{H}}(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}. \tag{1}$$

When considering a discretized and bounded Ω, such as all the possible pixel coordinates in an image, the suprema and infima are achievable and become maxima and minima, respectively. This bounds the Hausdorff distance as

$$d(X, Y) \leq d_{max} = \max_{x \in \Omega, y \in \Omega} d(x, y), \tag{2}$$

which corresponds to the diagonal of the image when using the Euclidean distance. As shown in [5], the Hausdorff distance is a metric. Thus $\forall X, Y, Z \subset \Omega$ we have the following properties:

$$d_H(X, Y) \geq 0 \tag{3a}$$
$$d_H(X, Y) = 0 \iff X = Y \tag{3b}$$
$$d_H(X, Y) = d_H(Y, X) \tag{3c}$$
$$d_H(X, Y) \leq d_H(X, Z) + d_H(Z, Y) \tag{3d}$$

Equation (3b) follows from X and Y being closed, because in our task the pixel coordinate space Ω is discretized. These properties are very desirable when designing a function to measure how similar X and Y are [4].

A shortcoming of the Hausdorff function is its high sensitivity to outliers [46, 54]. Figure 2 shows an example for two finite sets of points with one outlier. To avoid this, the average Hausdorff distance is more commonly used:

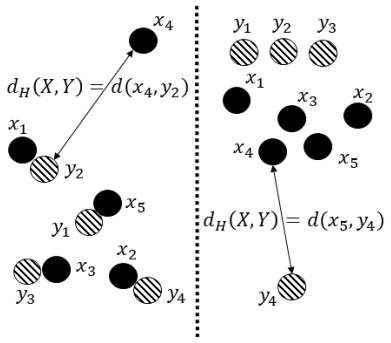

Figure 2. Illustration of two different configurations of point sets $X = \{x_1, ..., x_5\}$ (solid dots) and $Y = \{y_1, ..., y_4\}$ (dashed dots). Despite the clear difference in the distances between points, their Hausdorff distance are equal because the worst outlier is the same.

$$d_{\mathrm{AH}}(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} d(x, y), \tag{4}$$

where $|X|$ and $|Y|$ are the number of points in X and Y, respectively. Note that properties (3a), (3b) and (3c) are still true, but (3d) is not. Also, the average Hausdorff distance is differentiable with respect to any point in X or Y.

Let Y contain the ground truth pixel coordinates, and X be our estimation. Ideally, we would like to use $d_{\mathrm{AH}}(X, Y)$ as the loss function during the training of our convolutional neural network (CNN). We find two limitations when incorporating the average Hausdorff distance as a loss function. First, CNNs with linear layers implicitly determine the estimated number of points $|X|$ as the size of the last layer. This is a drawback because the actual number of points depends on the content of the image itself. Second, FCNs such as U-Net [42] can indicate the presence of an object center with a higher activation in the output layer, but they do not return the pixel coordinates. In order to learn with backpropagation, the loss function must be differentiable with respect to the network output.

4. The Weighted Hausdorff Distance

To overcome these two limitations, we modify the average Hausdorff distance as follows:

$$d_{\mathrm{WH}}(p, Y) = \frac{1}{\mathcal{S} + \epsilon} \sum_{x \in \Omega} p_x \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \underset{x \in \Omega}{M_\alpha} \left[\, p_x d(x, y) + (1 - p_x) d_{max} \, \right], \tag{5}$$

where

$$\mathcal{S} = \sum_{x \in \Omega} p_x, \tag{6}$$

$$\underset{a\in A}{M_\alpha}\left[f(a)\right] = \left(\frac{1}{|A|}\sum_{a\in A} f^\alpha(a)\right)^{\frac{1}{\alpha}}, \tag{7}$$

is the generalized mean, and ϵ is set to 10^{-6}. We call $d_{\text{WH}}(p, Y)$ the weighted Hausdorff distance (WHD). $p_x \in [0, 1]$ is the single-valued output of the network at pixel coordinate x. The last activation of the network can be bounded between zero and one by using a sigmoid nonlinearity. Note that p does not need to be normalized, i.e., $\sum_{x\in\Omega} p_x = 1$ is not necessary. Note that the generalized mean $M_\alpha[\cdot]$ corresponds to the minimum function when $\alpha = -\infty$. We justify the modifications applied to Equation (4) to obtain Equation (5) as follows:

1. The ϵ in the denominator of the first term provides numerical stability when $p_x \approx 0\ \forall x \in \Omega$.

2. When $p_x = \{0, 1\}$, $\alpha = -\infty$, and $\epsilon = 0$, the weighted Hausdorff distance becomes the average Hausdorff distance. We can interpret this as the network indicating with complete certainty where the object centers are. As $d_{\text{WH}}(p, Y) \geq 0$, the global minimum ($d_{\text{WH}}(p, Y) = 0$) corresponds to $p_x = 1$ if $x \in Y$ and 0 otherwise.

3. In the first term, we multiply by p_x to penalize high activations in areas of the image where there is no ground truth point y nearby. In other words, the loss function penalizes estimated points that should not be there.

4. In the second term, by using the expression $f(\cdot) := p_x d(x, y) + (1 - p_x)d_{max}$ we enforce that

   (a) If $p_{x_0} \approx 1$, then $f(\cdot) \approx d(x_0, y)$. This means the point x_0 will contribute to the loss as in the AHD (Equation (4)).

   (b) If $p_{x_0} \approx 0$, $x_0 \neq y$, then $f(\cdot) \approx d_{max}$. Then, if $\alpha = -\infty$, the point x_0 will not contribute to the loss because the "minimum" $M_{x\in\Omega}[\,\cdot\,]$ will ignore x_0. If another point x_1 closer to y with $p_{x_1} > 0$ exists, x_1 will be "selected" instead by $M[\,\cdot\,]$. Otherwise $M_{x\in\Omega}[\,\cdot\,]$ will be high. This means that low activations around ground truth points will be penalized.

   Note that $f(\cdot)$ is not the only expression that would enforce these two constraints ($f|_{p_x=1} = d(x, y)$ and $f|_{p_x=0} = d_{max}$). We chose a linear function because of its simplicity and numerical stability.

Both terms in the WHD are necessary. If the first term is removed, then the trivial solution is $p_x = 1 \quad \forall x \in \Omega$. If the second term is removed, then the trivial solution is $p_x = 0 \quad \forall x \in \Omega$. These two cases hold for any value of α and the proof can be found in the suplemental material. Ideally, the parameter $\alpha \to -\infty$ so that $M_\alpha(\cdot) = ||\cdot||_{-\infty}$ becomes the minimum operator [26]. However, this would make the second term flat with respect to the output of the network. For a given y, changes in p_{x_0} in a point x_0 that is far from y would be ignored by $M_{-\infty}(\cdot)$, if there is another point x_1 with high activation and closer to y. In practice, this makes training difficult because the minimum is not a smooth function with respect to its inputs. Thus, we approximate the minimum with the generalized mean $M_\alpha(\cdot)$, with $\alpha < 0$. The more negative α is, the more similar to the AHD the WHD becomes, at the expense of becoming less smooth. In our experiments, $\alpha = -1$. There is no need to use $M_\alpha(\cdot)$ in the first term because p_x is not inside the minimum, thus the term is already differentiable with respect to p.

If the input image needs to be resized to be fed into the network, we can normalize the WHD to account for this distortion. Denote the original image size as $(S_o^{(1)}, S_o^{(2)})$ and the resized image size as $(S_r^{(1)}, S_r^{(2)})$. In Equation (5), we compute distances in the original pixel space by replacing $d(x, y)$ with $d(\mathbf{S}x, \mathbf{S}y)$, where $x, y \in \Omega$ and

$$\mathbf{S} = \begin{pmatrix} S_o^{(1)}/S_r^{(1)} & 0 \\ 0 & S_o^{(2)}/S_r^{(2)} \end{pmatrix}. \tag{8}$$

4.1. Advantage Over Pixelwise Losses

A naive alternative is to use a one-hot map as label, defined as $l_x = 1$ for $x \in Y$ and $l_x = 0$ otherwise, and then use a pixelwise loss such as the Mean Squared Error (MSE) or the L^2 norm, where $L^2(l, p) = \sum_{\forall x\in\Omega} |p_x - l_x|^2 \propto \text{MSE}(l, x)$. The issue with pixelwise losses is that they are not informative of how close two points $x \in \Omega$ and $y \in Y$ are unless $x = y$. In other words, it is flat for the vast majority of the pixels, making training unfeasible. This issue is locally mitigated in [58] by using the MSE loss with Gaussians centered at each $x \in Y$. By contrast, the WHD in Equation (5) will decrease the closer x is to y, making the loss function informative outside of the global minimum.

5. CNN Architecture And Location Estimation

In this section, we describe the architecture of the fully convolutional network (FCN) we use, and how we estimate the final object locations. We want to emphasize that the network design is not a meaningful contribution of this work, thus we have not made any attempt to optimize it. Our main contribution is the use of the weighted Hausdorff distance as the loss function. We adopt the U-Net architecture [42] and modify it minimally for this task. Networks similar to U-Net have been proven to be capable of accurately mapping the input image into an output image, when trained in a conditional adversarial network setting [22] or when using a carefully tuned loss function [42]. Figure 3 shows the

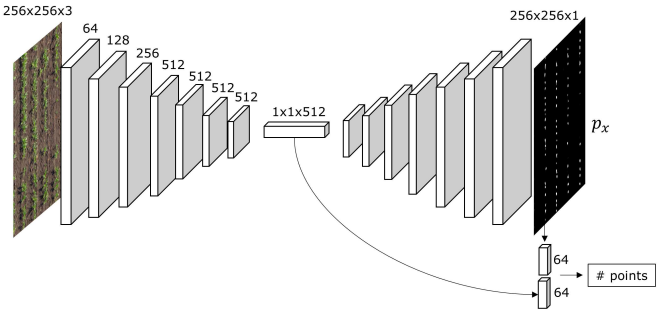

Figure 3. The FCN architecture used for object localization, minimally adapted from the U-Net [42] architecture. We add a small fully-connected layer that combines the deepest features and the estimated probability map to regress the number of points.

hourglass design of U-Net. The residuals connections between each layer in the encoder and its symmetric layer in the decoder are not shown for simplicity.

This FCN has two well differentiated blocks. The first block follows the typical architecture of a CNN. It consists of the repeated application of two 3×3 convolutions (with padding 1), each followed by a batch normalization operation and a Rectified Linear Unit (ReLU). After the ReLU, we apply a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels, starting with 64 channels and using 512 channels for the last 5 layers.

The second block consists of repeated applications of the following elements: a bilinear upsampling, a concatenation with the feature map from the downsampling block, and two 3×3 convolutions, each followed by a batch normalization and a ReLU. The final layer is a convolution layer that maps to the single-channel output of the network, p.

To estimate the number of objects in the image, we add a branch that combines the information from the deepest level features and also from the estimated probability map. This branch combines both features (the $1 \times 1 \times 512$ feature vector and the 256×256 probability map) into a hidden layer, and uses the 128-dimensional feature vector to output a single number. We then apply a ReLU to ensure the output is positive, and round it to the closest integer to obtain our final estimate of the number of objects, $\hat{C}$.

Although we use this particular network architecture, any other architecture could be used. The only requirement is that the output images of the network must be of the same size as the input image. The choice of a FCN arises from the natural interpretation of its output as the weights (p_x) in the WHD (Equation (5)). In previous works [24, 11], variants of the average Haussdorf distance were successfully used with non-FCN networks that estimate the point set directly. However, in those cases the size of the estimated set is fixed by the size of the last layer. To locate an unknown number

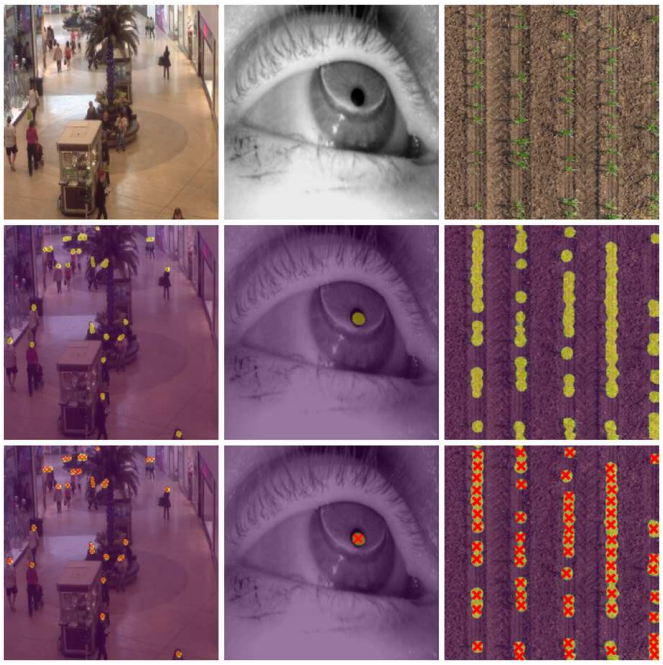

Figure 4. First row: Input image. Second row: Output of the network (p in the text) overlaid onto the input image. This can be considered a saliency map of object locations. Third row: The estimated object locations are marked with a red cross.

of objects, the network must be able to estimate a variable number of object locations. Thus, we could envision the WHD also being used in non-FCN networks as long as the output of the network is used as p in Equation (5).

The training loss we use to train the network is a combination of Equation (5) and a smooth L_1 loss for the regression of the object count. The final training loss is

$$\mathcal{L}(p, Y) = d_{WH}(p, Y) + \mathcal{L}_{\text{reg}}(C - \hat{C}(p)), \qquad (9)$$

where Y is the set containing the ground truth coordinates of the objects in the image, p is the output of the network, $C = |Y|$, and $\hat{C}(p)$ is the estimated number of objects. $\mathcal{L}_{\text{reg}}(\cdot)$ is the regression term, for which we use the smooth L_1 or Huber loss [21], defined as

$$\mathcal{L}_{\text{reg}}(x) = \begin{cases} 0.5x^2, & \text{for} |x| < 1 \\ |x| - 0.5, & \text{for} |x| \geq 1 \end{cases} \qquad (10)$$

This loss is robust to outliers when the regression error is high, and at the same time is differentiable at the origin.

The network outputs a saliency map p indicating with $p_x \in [0, 1]$ the confidence that there is an object at pixel x. Figure 4 shows p in the second row. During evaluation, our ultimate goal is to obtain $\hat{Y}$, i. e., the estimate of all object locations. In order to convert p to $\hat{Y}$, we threshold p to obtain the pixels $T = \{x \in \Omega \mid p_x > \tau\}$. We can use three different methods to decide which τ to use:

1. Use a constant τ for all images.

2. Use Otsu thresholding [36] to find an adaptive τ different for every image.

3. Use a Beta mixture model-based thresholding (BMM). This method fits a mixture of two Beta distributions to the values of p using the algorithm described in [45], and then takes the mean value of the distribution with highest mean as τ.

Figure 4 shows in the third row an example of the result of thresholding the saliency map p. Then, we fit a Gaussian mixture model to the points T. This is done using the expectation maximization (EM) [34] algorithm and the estimated number of plants $\hat{C}$.

The means of the fitted Gaussians are considered the final estimate $\hat{Y}$. The third row of Figure 4 shows the estimated object locations with red crosses. Note that even if the map produced by the FCN is of good quality, i.e., there is a cluster on each object location, EM may not yield the correct object locations if $|\hat{C} - C| > 0.5$. An example can be observed in the first column of Figure 4, where a single head is erroneously estimated as two heads.

6. Experimental Results

We evaluate our method with three datasets.

The first dataset consists of 2,000 images acquired from a surveillance camera in a shopping mall. It contains annotated locations of the heads of the crowd. This dataset is publicly available at `http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html` [32]. 80%, 10% and 10% of the images were randomly assinged to the training, validation, and testing datasets, respectively.

The second dataset is presented in [13] with the roman letter V and publicly available at `http://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html`. It contains 2,135 images with a single eye, and the goal is to detect the center of the pupil. It was also randomly split into training, validation and testing datasets as 80/10/10 %, respectively.

The third dataset consists of aerial images of a crop field taken from a UAV flying at an altitude of 40 m. The images were stitched together to generate a $6,000 \times 12,000$ orthoimage of 0.75 cm/pixel resolution shown in Figure 5. The location of the center of all plants in this image was groundtruthed, resulting in a total of 15,208 unique plant centers. This mosaic image was split, and the left 80% area was used for training, the middle 10% for validation, and the right 10% for testing. Within each region, random image crops were generated. These random crops have a uniformly distributed height and width between 100 and 600 pixels. We extracted 50,000 random image crops in the training region, $5,000$ in the validation region, and $5,000$ in the testing region. Note that some of these crops may highly overlap. We are making the third dataset publicly available at `https://engineering.purdue.edu/~sorghum/dataset-plant-centers-2016`. We believe this dataset will be valuable for the community, as it poses a challenge due to the high occlusion between plants.

Figure 5. An orthorectified image of a crop field with 15,208 plants. The red region was used for training, the region in green for validation, and the region in blue for testing.

All the images were resized to 256×256 because that is the minimum size our architecture allows. The groundtruthed object locations were also scaled accordingly. As for data augmentation, we only use random horizontal flip. For the plant dataset, we also flipped the images vertically. We set $\alpha = -1$ in Equation (7). We have also experimented with $\alpha = -2$ with no apparent improvement, but we did not attempt to find an optimal value. We retrain the network for every dataset, i.e., we do not use pretrained weights. For the mall and plant dataset, we used a batch size of 32 and Adam optimizer [25, 39] with a learning rate of 10^{-4} and momentum of 0.9. For the pupil dataset, we reduced the size of the network by removing the five central layers, we used a batch size of 64, and stochastic gradient descent with a learning rate of 10^{-3} and momentum of 0.9. At the end of each epoch, we evaluate the average Hausdorf distance (AHD) in Equation (4) over the validation set, and select the epoch with lowest AHD on validation.

As metrics, we report Precision, Recall, F-score, AHD, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percent Error (MAPE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|e_i|, \quad \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|e_i|^2} \tag{11}$$

$$\text{MAPE} = 100\frac{1}{N}\sum_{\substack{i=1 \\ C_i \neq 0}}^{N}\frac{|e_i|}{C_i} \tag{12}$$

where $e_i = \hat{C}_i - C_i$, N is the number of images, C_i is the true object count in the i-th image, and $\hat{C}_i$ is our estimate.

A true positive is counted if an estimated location is at most at distance r from a ground truth point. A false positive is counted if an estimated location does not have any

ground truth point at a distance at most r. A false negative is counted if a true location does have any estimated location at a distance at most r. Precision is the proportion of our estimated points that are close enough to a true point. Recall is the proportion of the true points that we are able to detect. The F-score is the harmonic mean of precision and recall. Note that one can achieve a precision and recall of 100% even if we estimate more than one object location per ground truth point. This would not be an ideal localization. To take this into account, we also report metrics (MAE, RMSE and MAPE) that indicate if the number of objects is incorrect. The AHD can be interpreted as the average location error in pixels.

Figure 8 shows the F-score as a function of r. Note that r is only an evaluation parameter. It is not needed during training or testing. MAE, RMSE, and MAPE are shown in Table 1. Note that we are using the same architecture for all tasks, except for the pupil dataset, where we removed intermediate layers. Also, in the case of the pupil detection, we know that there is always one object in the image. Thus, regression is not necessary and we can remove the regression term in Equation (9) and fix $\hat{C}_i = C_i = 1 \; \forall i$.

A naive alternative approach to object localization would be to use generic object detectors such as Faster R-CNN [41]. One can train these detectors by constructing bounding boxes with fixed size centered at each labeled point. Then the center of each bounding box can be taken as the estimated location. We used bounding boxes of size 20×20 (the approximate average head and pupil size) and anchor sizes of 16×16 and 32×32. Note that these parameters may be suboptimal even though they were selected to match the type of object. The threshold we used for the softmax scores was 0.5 and for the intersection over union it was 0.4, because they minimize the AHD over the validation set. We used the VGG-16 architecture [49] and trained it using stochastic gradient descent with learning rate of 10^{-3} and momentum of 0.9. For the pupil dataset, we always selected the bounding box with the highest score. We experimentally observed that Faster R-CNN struggles with detecting very small objects that are very close to each other. Tables 2-4 show the results of Faster R-CNN results on the mall, pupil, and plant datasets. Note that the mall and plant datasets, with many small and highly overlapping objects, are the most challenging for Faster R-CNN. This behaviour is consistent with the observations in [19], where, all generic object detectors perform very poorly and Faster R-CNN yields a mean Average Precision (mAP) of 5% in the best case.

We also experimented using mean shift [9] instead of Gaussian mixtures (GM) to detect the local maxima. However, mean shift is prone to detect multiple local maxima, and GMs are more robust against outliers. In our experiments, we observed that precision and recall were substantially worse than using GM. More importantly, using Mean Shift slowed down validation an order of magnitude. The average time for the Mean Shift algorithm to run on one of our images was 12 seconds, while fitting GM using expectation maximization took around 0.5 seconds, when using the scikit-learn implementations [38].

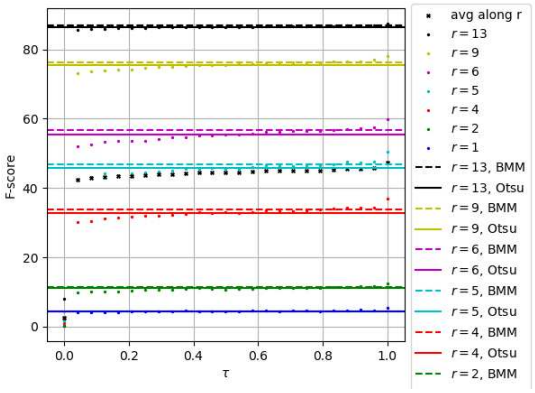

Figure 6. Effect on the F-score of the threshold τ.

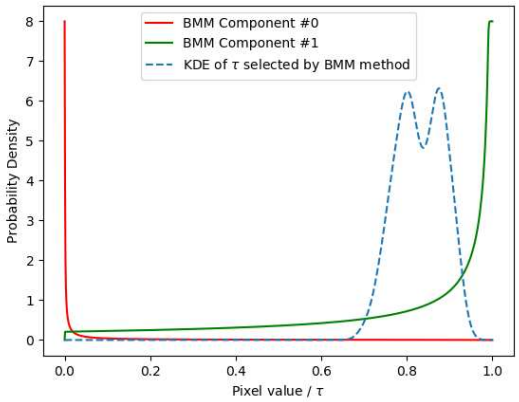

Figure 7. Beta mixture model fitted on the values of p_x, and the thresholds τ used by the BMM method.

We also investigated the effect of the parameter τ, and the three methods to select it presented in Section 5. One may think that this parameter could be a trade-off between some metrics, and that it should be cross-validated. In practice, we observed that τ does not balance precision and recall, thus a precision-recall curve is not meaningful. Instead, we plot the F-score as a function of r in Figure 8. Also, cross-validating τ would imply fixing an "optimal" value for all images. Figure 6 shows that we can do better with adaptive thresholding methods (Otsu or BMM). Note that BMM thresholding (dashed lines) always outperforms Otsu (solid lines), and most of fixed τ. To justify the appropriateness of the BMM method, note that in Figure 4 most of the values in the estimated map are very high or very low. This makes a Beta distribution a better fit than a Normal distribution (as used in Otsu's method) to model p_x. Figure 7 shows the fitted BMM and a kernel density estimation of the values of τ adaptively selected by the BMM method.

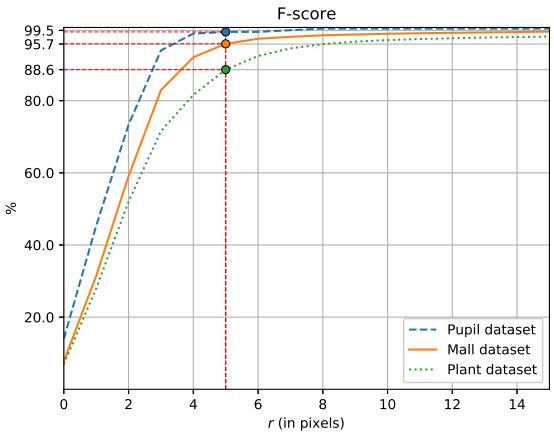

Figure 8. F-score as a function of r, the maximum distance between a true and an estimated object location to consider it correct or incorrect. A higher r makes correctly locating an object easier.

Table 1. Results of our method for object localization, using $r = 5$. Metrics are defined in Equations (4), (11)-(12). Regression metrics for the pupil dataset are not shown because there is always a single pupil ($\hat{C} = C = 1$). Figure 8 shows the F-score for other r values.

Metric	Mall dataset	Pupil dataset	Plant dataset	Average
Precision	95.2%	99.5%	88.1%	94.4%
Recall	96.2%	99.5%	89.2%	95.0%
F-score	95.7%	99.5%	88.6%	94.6%
AHD	4.5 px	2.5 px	7.1 px	4.7 px
MAE	1.4	-	1.9	1.7
RMSE	1.8	-	2.7	2.3
MAPE	4.4%	-	4.2%	4.3 %

Lastly, as our method locates and counts objects simultaneously, it could be used as a counting technique. We also evaluated our technique in the task of crowd counting using the ShanghaiTech Part B dataset presented in [62], and achieve a MAE of 19.9. Even though we do not outperform state of the art methods that are specifically fine-tuned for crowd counting [29], we can achieve comparable results with our generic method. We expect future improvements such as architectural changes or using transfer learning to further increase the performance.

A PyTorch implementation of the weighted Hausdorff distance loss and trained models are available at `https://github.com/javiribera/locating-objects-without-bboxes`.

7. Conclusion

We have presented a loss function for the task of locating objects in images that does not need bounding boxes. This loss function is a modification of the average Hausdorff distance (AHD), which measures the similarity between two unordered sets of points. To make the AHD differentiable with respect to the network output, we have considered the certainty of the network when estimating an object location. The output of the network is a saliency map of object locations and the estimated number of objects. Our method is not restricted to a maximum number of objects in the image, does not require bounding boxes, and does not use region proposals or sliding windows. This approach can be used in tasks where bounding boxes are not available, or the small size of objects makes the labeling of bounding boxes impractical. We have evaluated our approach with three different datasets, and outperform generic object detectors and task-specific techniques. Future work will include developing a multi-class object location estimator in a single network, and evaluating more modern CNN architectures.

Table 2. Head location results using the mall dataset, using $r = 5$.

Metric	Faster-RCNN	Ours
Precision	81.1%	**95.2 %**
Recall	76.7%	**96.2 %**
F-score	78.8 %	**95.7 %**
AHD	7.6 px	**4.5 px**
MAE	4.7	**1.4**
RMSE	5.6	**1.8**
MAPE	14.8%	**4.4 %**

Table 3. Pupil detection results, using $r = 5$. Precision and recall are equal because there is only one estimated and one true object.

Method	Precision	Recall	AHD
Swirski [53]	77 %	77 %	-
ExCuSe [13]	77 %	77 %	-
Faster-RCNN	99.5 %	99.5 %	2.7 px
Ours	**99.5** %	**99.5** %	**2.5 px**

Table 4. Plant location results using the plant dataset, using $r = 5$.

Metric	Faster-RCNN	Ours
Precision	86.6 %	**88.1 %**
Recall	78.3 %	**89.2 %**
F-score	82.2 %	**88.6 %**
AHD	9.0 px	**7.1 px**
MAE	9.4	**1.9**
RMSE	13.4	**2.7**
MAPE	17.7 %	**4.2 %**

Acknowledgements: This work was funded by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000593. The views and opinions of the authors expressed herein do not necessarily reflect those of the U.S. Government or any agency thereof. We thank Professor Ayman Habib for the orthophotos used in this paper. Contact information: Edward J. Delp, `ace@ecn.purdue.edu`

References

[1] S. Aich, I. Ahmed, I. Obsyannikov, I. Stavness, A. Josuttes, K. Strueby, H. Duddu, C. Pozniak, and S. Shirtliffe. Deepwheat: Estimating phenotypic traits from crop images with deep learning. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, March 2018. Stateline, NV.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. Anchorage, AK.

[3] J. L. Araus and J. E. Cairns. Field high-throughput phenotyping: the new crop breeding frontier. *Trends in Plant Science*, 19(1):52–61, January 2014.

[4] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3), March 1991.

[5] H. Attouch, R. Lucchetti, and R. J. B. Wets. The topology of the ρ-Hausdorff distance. *Annali di Matematica Pura ed Applicata*, 160(1):303–320, December 1991.

[6] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database (supplemental material). *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. Boston, MA.

[7] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, 2011.

[8] B. S. Chauhan and D. E. Johnson. Row spacing and weed control timing affect yield of aerobic rice. *Field Crops Research*, 121(2):226–231, March 2001.

[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[10] M.-P. Dubuisson and A. K. Jain. A modified Hausdorff distance for object matching. *Pattern Recognition*, pages 566–568, October 1994.

[11] H. Fan, H. Su, and L. Guibas. A point set generation network for 3D object reconstruction from a single image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2463–2471, July 2017. Honolulu, HI.

[12] D. E. Farnham. Row spacing, plant density, and hybrid effects on corn grain yield and moisture. *Agronomy Journal*, 93:1049–1053, September 2001.

[13] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. ExCuSe: Robust pupil detection in real-world scenarios. *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pages 39–51, September 2015. Valletta, Malta.

[14] W. Fuhl, T. Santini, C. Reichert, D. Claus, A. Herkommer, H. Bahmani, K. Rifai, S. Wahl, and E. Kasneci. Non-intrusive practitioner pupil detection for unmodified microscope oculars. *Computers in Biology and Medicine*, 79:36–44, December 2016.

[15] R. Girshick. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, December 2015.

[16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, November 2016.

[17] J. Gu, X. Yang, S. De Mello, and J. Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1548–1557, July 2017. Honolulu, HI.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *arXiv:1703.06870*, April 2017.

[19] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. Honolulu, HI.

[20] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3):1049–1059, March 2018.

[21] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. Honolulu, HI.

[23] K. L. K. Lin and W. Siu. Spatially eigen-weighted Hausdorff distances for human face recognition. *Pattern Recognition*, 36(8):1827–1834, August 2003.

[24] H. E. Khiyari and H. Wechsler. Age invariant face recognition using convolutional neural networks and set distances. *Journal of Information Security*, 8(3):174–185, July 2017.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of the International Conference for Learning Representations*, abs/1412.6980, April 2015. San Diego, CA.

[26] C. S. Kubrusly. Banach spaces L^p. In *Essentials of Measure Theory*, page 83. Springer, Cham, 2005.

[27] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015.

[28] V. Lempitsky and A. Zisserman. Learning to count objects in images. *Proceedings of the Advances in Neural Information Processing Systems*, pages 1324–1332, December 2010. Vancouver, Canada.

[29] Y. Li, X. Zhang, and D. Chen. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, June 2018.

[30] S. Liao, Y. Gao, A. Oto, and D. Shen. Representation learning: A unified deep learning framework for automatic prostate mr segmentation. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pages 254–261, September 2013. Nagoya, Japan.

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. SSD: Single shot multibox detector. *Proceedings of the European Conference on Computer Vision*, pages 21–37, October 2016. Amsterdam, The Netherlands.

[32] C. C. Loy, K. Chen, S. Gong, and T. Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, pages 347–382. Springer, October 2013.

[33] Y. Lu, C. L. Tan, W. Huang, and L. Fan. An approach to word image matching based on weighted Hausdorff distance. *Proceedings of International Conference on Document Analysis and Recognition*, pages 921–925, September 2001.

[34] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, November 1996.

[35] E. H. Neilson, A. M. Edwards, C. K. Blomstedt, B. Berger, B. L. Mller, and R. M. Gleadow. Utilization of a high-throughput shoot imaging system to examine the dynamic phenotypic responses of a C_4 cereal crop plant to nitrogen and water deficiency over time. *Journal of Experimental Botany*, 66(7):1817–1832, 2015.

[36] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979.

[37] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 854–863, June 2016. Las Vegas, NV.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *Proceedings of the International Conference on Learning Representations*, April 2018. Vancouver, Canada.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, June 2016. Las Vegas, NV.

[41] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1137–1149, June 2017.

[42] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, October 2015. Munich, Germany.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 11(3):211–252, December 2015.

[44] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4031–4039, July 2017.

[45] C. Schröder. A hybrid parameter estimation algorithm for beta mixtures and applications to methylation state classification. *Algorithms for Molecular Biology*, 12(21):62–66, August 2017.

[46] O. Schutze, X. Esquivel, A. Lara, and C. A. C. Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective opti-

mization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, August 2012.

[47] J. Shao, D. Wang, X. Xue, and Z. Zhang. Learning to point and count. *arXiv:1512.02326*, December 2015.

[48] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2107–2116, July 2017. Honolulu, HI.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations*, May 2015. San Diego, CA.

[50] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. *Proceedings of the Association for the Advancement of Artificial Intelligence Human Computation Workshop*, WS-12-08:40–46, July 2012. Toronto, Canada.

[51] R. Sui, B. E. Hartley, J. M. Gibson, C. Yang, J. A. Thomasson, and S. W. Searcy. High-biomass sorghum yield estimate with aerial imagery. *Journal of Applied Remote Sensing*, 5(1):053523, January 2011.

[52] V. Sundstedt. *Gazing at Games: An Introduction to Eye Tracking Control*, volume 5. Morgan & Claypool Publishers, San Rafael, CA, 2012.

[53] L. Świrski, A. Bulling, and N. Dodgson. Robust real-time pupil tracking in highly off-axis images. *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 173–176, March 2012. Santa Barbara, CA.

[54] A. A. Taha and A. Hanbury. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, August 2015.

[55] P. Teikari, M. Santos, C. Poon, and K. Hynynen. Deep learning convolutional networks for multiphoton microscopy vasculature segmentation. *arXiv:1606.02382*, June 2016.

[56] J. H. M. Thornley. Crop yield and planting density. *Annals of Botany*, 52(2):257–259, August 1983.

[57] I. Tokatlidis and S. D. Koutroubas. A review of maize hybrids' dependence on high plant populations and its implications for crop yield stability. *Field Crops Research*, 88(2):103–114, August 2004.

[58] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, June 2015. Boston, MA.

[59] J. R. R. Uijlings, , K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, September 2013.

[60] F. Xiong, X. Shi, and D. Yeung. Spatiotemporal modeling for crowd counting in videos. *Proceedings of the IEEE International Conference on Computer Vision*, pages 5151–5159, October 2017. Venice, Italy.

[61] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, June 2015. Boston, MA.

[62] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, June 2016. Las Vegas, NV.

[63] S. K. Zhou, H. Greenspan, and D. Shen. *Deep Learning for Medical Image Analysis*. Academic Press, London, United Kingdom, 2017.