Weighted Hausdorff Distance: A Loss Function For Object Localization

Javier Ribera, David Güera, Yuhao Chen, Edward Delp

Video and Image Processing Laboratory (VIPER),
School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana, USA
{jprat, dgueraco, chen902}@purdue.edu, ace@ecn.purdue.edu

Abstract. Recent advances in Convolutional Neural Networks (CNN) have achieved remarkable results in localizing 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 does not require one to "guess" the maximum number of objects in the image, and 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 report an average precision and recall of 94% for the three datasets, and an average location error of 6 pixels in 256×256 images.

Keywords: Object localization, deep learning, loss function, Hausdorff distance

1 Introduction

Recent advances in deep learning [1,2] have increased the accuracy of localization tasks such as object or keypoint detection. In Fast R-CNN [3], candidate regions or proposals are generated by classical methods such as selective search [4]. 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 [3,5] allow for end-to-end training of models. Mask R-CNN [6] 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 pixel by pixel, making the keypoint detection highly sensitive to alignment of the segmentation mask. The Single Shot

Fig. 1. Results of an object localization task. The red dots show the estimated object locations for human heads, eye pupils and plant centers.

MultiBox Detector (SSD) [7], 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 [8], it is observed that generic object detectors such as Faster R-CNN and SSD perform very poorly for small objects.

Bounding-box annotation is tedious, time-consuming and expensive [9]. For example, annotating ILSVRC, a popular object detection dataset, required 42 seconds per bounding box when crowdsourcing on Amazon's Mechanical Turk [10] using a technique specifically developed for efficient bounding box annotation [11]. In [12], Bell et al. introduce a new dataset for material recognition and segmentation. By collecting click location labels instead of a full per-pixel segmentation, they reduce the annotation costs an order of magnitude.

In this paper, we propose a modification of the Weighted Hausdorff Distance as a loss function of a CNN to estimate the location of objects. This 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. This technique maps input images to a set of coordinates and also provides the number of points. We evaluate or 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 plant centers with high occlusion.

2 Related Work

Counting the number of objects present in an image is not a trivial task. In [13], Lempitsky et al. estimate a density function whose integral corresponds to the object count. In [14], Shao et al. proposed two methods for localizing objects.

One method first counts and then localizes, and the other first localizes and then counts.

Crowd monitoring. Locating and counting people is necessary for many applications such as crowd monitoring in surveillance systems, surveys for new businesses, and emergency management [13,15]. There are multiple studies in the literature, where people in videos of crowds are detected and tracked [16,17]. 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. [18] estimate the density image using a Multi-column CNN (MCNN) that learns features at different scales. In [19], 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 appropiate 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. [21] make use of additional information such as the groundtruthed perspective map.

Eye tracking. Resolving the position of the pupil of each eye is typically required for eye tracking. The development of eye tracker devices has allowed researchers to study human attention and behavior in events such as practicing sports and driving [22]. Commercial applications of eye tracking devices include using eye movements to play video games [23] and for user gaze tracking while using electronic devices or driving [24,25]. In microsurgery, eye tracking can help the surgeon to control the movement of the microscope, the zoom, and the illumination effortlessly [26].

Plant phenotyping. Locating the center of plants in a crop field is a critical step for remote phenotyping. Plant scientists measure physical properties of plants using a set of methodologies known as phenotyping [27]. Such traits are used by agronomists to predict future crop yield [28,29,30] and by plant scientists to breed new plant varieties [31,32]. Traditional phenotyping involves large work crews manually taking measurements during several days. This makes traditional phenotyping costly and slow. Also, plant samples may be damaged by invasive measurement techniques or during their transportation. Unmanned Aerial Vehicles (UAV) and remote imaging techniques provide non-invasive and cost-effective technologies for collecting data and estimating plant traits, such as the center of plants in a crop field. The plant centers can be used to assign estimated phenotypic traits (e.g, number of leaves) to particular plants. Also, the spacing between plants is directly related to crop yield [33,34]. Aich et al. [35] 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 [36]. Modifications of the Hausdorff distance [37] have been used for various multiple tasks, including character recognition [38], face recognition [39] and scene matching [39]. Schutze et al. [40] use the averaged Hausdorff distance to evaluate solutions in multi-objective optimization problems. In [41], Elkhiyari et al. compare features extracted by a CNN according to multiple variants of the Hausdorff distance for the task of face recognition. In [42], 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 [43,44,45,46].

3 The Object Localization Task

We investigate the task of localizing objects in images. Figure 1 shows an example of localized objects in images. In this task, we are not interested in obtaining bounding boxes around each object of interest. Instead, we define object localization as the task of obtaining a single 2D coordinate corresponding to the location of each object. The location of an object could be its center, or a keypoint we are interested in. This is a keypoint detection problem where we do not know in advance the number of keypoints in the image. This definition 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. In this paper, we focus on a single class of objects for simplicity, i.e, we know we are only localizing one type of object. Our method is object-agnostic and the discussion in this paper does not include any information about the type of object.

4 Distance Functions Between Sets of Points

In this section, we briefly review the Hausdorff Distance. We then describe the loss function that we use for training and why it is useful in the task of object localization.

4.1 The Average Hausdorff Distance

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, i.e, the Euclidean distance. The sets X and Y do not necessarily have the same number of points. Let Ω 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

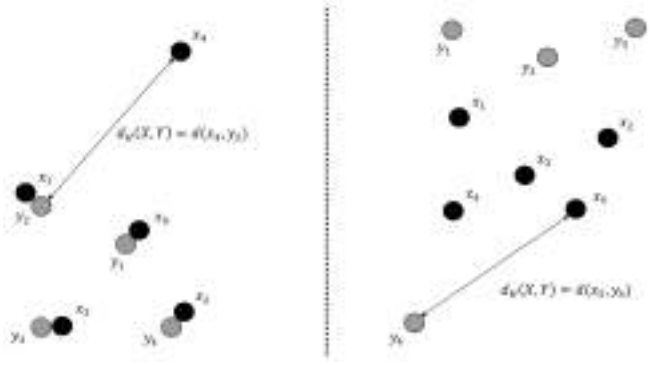

Fig. 2. Illustration of the Hausdorff distance with two different configurations of two finite sets $X = \{x_1, x_2, x_3, x_4, x_5\}$ (solid dots) and $Y = \{y_1, y_2, y_3, y_4\}$ (dashed dots). Despite the differences in both configurations, their respective Hausdorff distances are the same. In both cases, the Hausdorff distance is the distance of the worst outlier.

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

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 [36], the Hausdorff distance is a metric, as $\forall X, Y, Z \subset \Omega$ 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 [47].

A shortcoming of the Hausdorff function is that it is very sensible to outliers [40,43]. Figure 2 shows an illustration of the Hausdorff distance for two sets of finite points with one outlier. To avoid this, the Averaged Hausdorff Distance is more commonly used:

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

where $|X|$ and $|Y|$ are the number of dots in X and Y, respectively. Note that properties (3a), (3c) and (3c) are still true, but (3d) is no longer true. Also, the Averaged Hausdorff Distance is differentiable with respect to any point of 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 impediments when incorporating the Averaged 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 [48] 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.

4.2 The Weigthed Hausdorff Distance

To overcome these two limitations, we modify the Averaged Hausdorff Distance as follows:

$$d_{\mathrm{WH}}(p, Y) = \frac{1}{|\hat{X}| + \epsilon} \sum_{x \in \Omega} p_x \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in \Omega} \frac{d(x, y) + \epsilon}{p_x^{\alpha} + \frac{\epsilon}{d_{max}}}, \tag{5}$$

where

$$|\hat{X}| = \sum_{x \in \Omega} p_x \tag{6}$$

and $\epsilon \approx 10^{-6}$. We call $d_{\mathrm{WH}}(p, Y)$ 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 non-linearity. Note that p does not need to be normalized, i.e, $\sum_{x \in \Omega} p_x = 1$ is not necessary. We justify the modifications applied to Equation (4) to obtain Equation (5) as follows:

1. When $p_x = \{0, 1\}$, $\alpha = 1$, and ignoring the epsilons, the Weighted Hausdorff Distance becomes the Averaged Hausdorff Distance. We can interpret this as the network indicating with complete certainty where the object centers are.
2. The ϵ in the denominator of the first term provides numerical stability when $p_x \approx 0 \quad \forall x \in \Omega$.
3. By multiplying by p_x in the first term, the cost function will penalize high activations in areas of the image where there is no ground truth point y nearby.

4. By dividing by p_x in the second term, the distance $d(x, y)$ between a ground truth point y and a pixel coordinate x will not contribute to the loss function if p_x is low, i.e, the network estimates there is not point at x. This is because the minimum over x will ignore $d(x, y)/p_x$ as it will grow large. On the contrary, if $p_x \approx 1$ the cost function will consider $d(x, y)$ as in Equation (4).
5. The ϵ in the second term adds numerical stability to the term when $p_x \approx 0$. At the same time, it enforces that if there is a ground truth point at $y \approx x$ and the map indicates that there is no point, i.e, $p_x \approx 0$, then

$$\frac{d(x, y) + \epsilon}{p_x^\alpha + \frac{\epsilon}{d_{max}}} \approx d_{max}. \tag{7}$$

   This means that low activations around ground truth points will be penalized.
6. The $\alpha > 1$ of the second term is included in order to make the second term larger, because we observed that with $\alpha = 1$, d_{WH} is too conservative when taking new guesses at object locations. According to our observations during training, setting $\alpha = 4$ helps balance precision and recall by increasing the importance of missing estimated points.

From the definition of the WHD in Equation (5), we can make the following observation. If there is a ground truth point at pixel coordinate y, then an estimation of a point at $x \neq y$ will reduce the cost function the closer x is to y. By contrast, pixelwise cost functions (such as L_1 or L_2), between p and a one-hot mask are not informative of how close x and y are unless $x = y$.

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 object locations. We want to emphasize that the network design is not a contribution of this work. Our main contribution is the use of the Weighted Hausdorff Distance (Equation (5)). We adopt the U-Net architecture, as 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 [49] or when using a carefully tuned loss function [48]. We describe the minimal modifications applied to the U-Net [48] architecture to this end. Figure 3 shows the hourglass design of U-Net and its residuals connections, with lateral connections added to regress the number of objects. More precisely, 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.

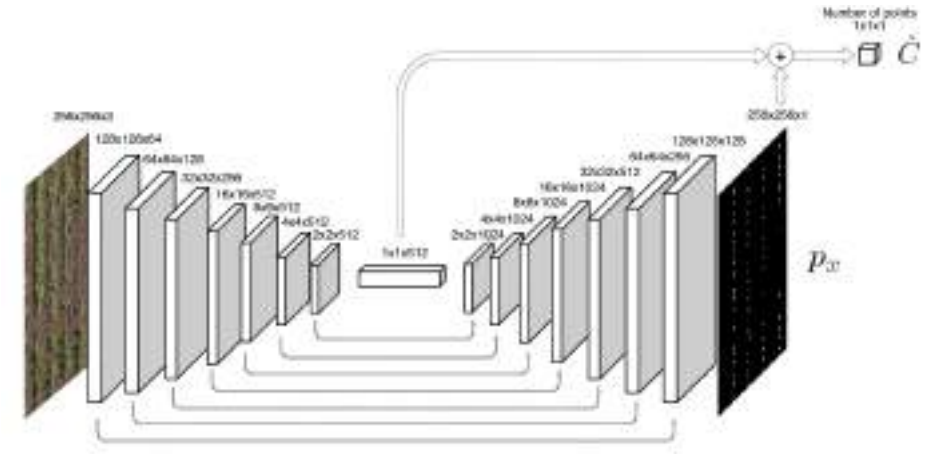

Fig. 3. The FCN architecture used for object localization. Similarly to the U-Net [48] architecture, the features after the convolution operations are concatenated with the features after the upsampling. This characteristic allows the architecture to use original image features in addition to features from previous upsampling layers. We add a linear layer at the deepest level to regress the estimated number of points.

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 1×1 convolution used to map the 128-component feature map to p, the single-channel output of the network.

To estimate the number of objects in the image, we add an additional branch that combines the information from the deepest level features and also from the estimated probability map. This branch is simply a fully-connected layer whose input is the concatenation of both features (the $1 \times 1 \times 512$ feature vector and the 256×256 probability map). The output of this branch is a single feature ($1 \times 1 \times 1$). We denote this feature as ϕ, and rectify it with a SoftPlus non-linearity as

$$\hat{C} = \log\left(1 + \exp(\phi)\right). \tag{8}$$

The SoftPlus ensures that $\hat{C}$, our estimate of the number of objects in the image, is always positive. Without this non-linearity, $\hat{C}$ may be occasionaly negative for some input images, and later steps use $\hat{C}$ as the number of clusters, and thus require it to be positive. A SoftPlus is more desirable than a positive part or "HardPlus" rectifier $[x]^+ = \max\{x, 0\}$ because it is differentiable everywhere. We then round $\hat{C}$ to the closest integer to obtain the final estimate.

Although we use this particular network architecture, any other architecture could be used. The only requirement is that the output 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 [41,42], 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 of the network. To locate an unkown number 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 the weights 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 expanded expression of the training loss is

$$\mathcal{L}(p, Y) = \frac{1}{|\hat{X}| + \epsilon} \sum_{x \in \Omega} p_x \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in \Omega} \frac{d(x, y) + \epsilon}{p_x^{\alpha} + \frac{\epsilon}{d_{max}}} + \mathcal{L}_{\text{reg}}(C - \hat{C}), \tag{9}$$

where Y is the set containing the ground truth coordinates of the objects in the image, $|Y|$ is the number of elements in Y, p_x is the output of the network at coordinate x, and $\hat{C}$ is as defined in Equation (8).
The third term, $\mathcal{L}_{\text{reg}}(\cdot)$ is the regression term, for which we use the smooth L_1 or Huber loss [50], defined as

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

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 in pixel x. Figure 4 shows p in the second column. During evaluation, we want to obtain $\hat{X}$, i. e, the estimates of all object locations. In order to convert p to $\hat{X}$, we threshold p to obtain those locations where $p_x > \tau$. We set τ to be 0.04. The third column of Figure 4 shows the result of thresholding p. Then, we fit a Gaussian Mixture Model to those pixel locations where $p_x > \tau$. This is done using the Expectation Maximization [51] algorithm and the estimated number of plants $\hat{C}$, rounded to the closest integer.
The means of the fitted Gaussians are considered the estimate $\hat{X}$. The forth column of Figure 4 shows the estimated object locations with red dots. Note that even if the map produced by the FCN is of good quality, i.e, there is a cluster on each object location, Expectation Maximization may not yield correct object locations if the count $\hat{C}$ is more than ± 0.5 off. An example can be observed in the first row of Figure 4, where a single head is erroneously estimated as two heads.

6 Experimental Results

We evaluate our method with three datasets.

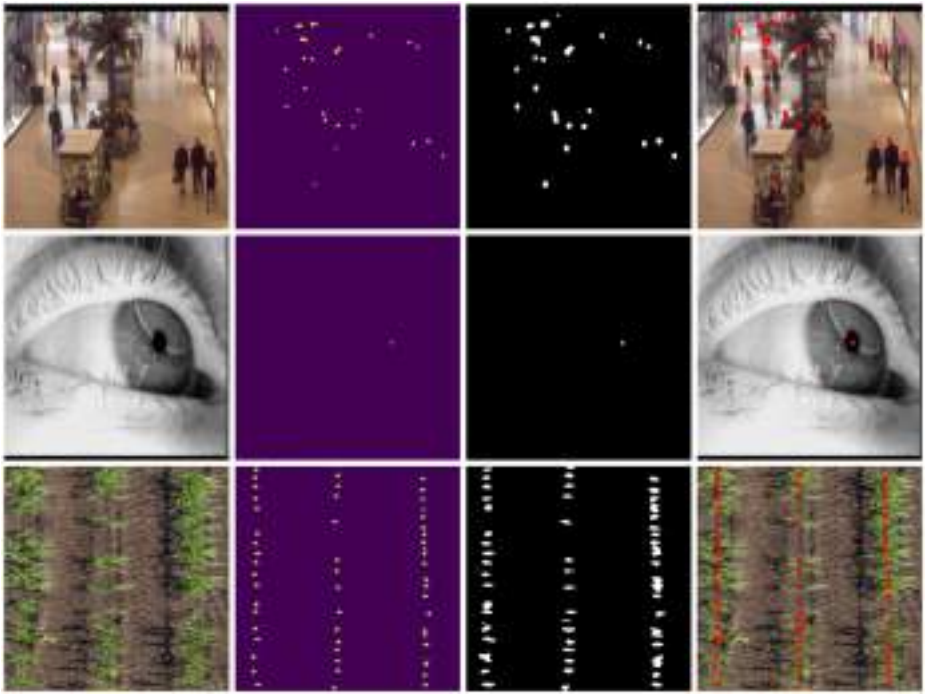

Fig. 4. First column: Input image fed to the Fully Convolutional Network (FCN). Second column: Output of the FCN (p in the text). This can be considered a saliency map of object locations. Third column: Result of thresholding the output of the FCN. This is a binary image. Forth column: The estimated object locations are marked with a red dot.

The first dataset consists of 2,000 images acquired from a surveillance camera in a shopping mall. It contains the 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` [52]. The images were randomly split with 80%, 10% and 10% of the images assigned to the training, validation and testing sets, respectively.

The second dataset is publicly available at `http://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html` [22]. It consists of images containing a single eye, and the goal is to detect the center of the pupil. We use the dataset denoted in [22] with the roman letter V as our second dataset. This dataset contains a total of 2,135 eye images with their annotated pupil centers. Then, the images were randomly split with 80%, 10% and 10% of the images for training, validation, and testing sets, respectively.

The third dataset consists of aerial images taken from an Unmaned Aerial Vehicle (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

Fig. 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.

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 the following way. The left 80% area of the image was used for training, the middle 10% area was used for validation and the right 10% area was used 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 images may highly overlap. These images constitute the training, validation, and testing sets, respectively. We are making the third dataset is publicly available at `https://engineering.purdue.edu/~sorghum/dataset-plant-centers-2016`. We believe this dataset will be valuable for the community because it poses a challenge due to the high occlusion between plants.

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. Each of the three color channels of the input image were normalized to have approximately zero mean and unitary standard deviation. This is done by subtracting 0.5 and dividing by 0.5. As data augmentation technique, we only flip the input images horizontally with a probability of 0.5. For the plant dataset, we also flipped the images vertically with a probability of 0.5.
We set $\alpha = 4$ in Equation (5). The batch size we use is 48. We retrain the network for every dataset, i.e, we do not use pretrained weights. We use Adam for the first 100 epochs, and then switch to Stochastic Gradient Descent with a learning rate of 1×10^{-4} and momentum of 0.9. At the end of each epoch, we evaluate the Average Haussdorf Distance (AHD) in Equation (4) over the entire validation set. We select the epoch at which the model returns the lowest AHD on the validation set. The machine we use for computation is an Intel i7-6900K and three Nvidia Titan Xp.

We use the following metrics to evaluate our method:

$$\text{Precision} = 100\frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{Recall} = 100\frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{C}_i - C_i| \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left|\hat{C}_i - C_i\right|^2} \tag{14}$$

$$\text{MAPE} = 100\frac{1}{N}\sum_{\substack{i=1 \\ C_i \neq 0}}^{N} \frac{\left|\hat{C}_i - C_i\right|}{C_i} \tag{15}$$

N is the number of images in the dataset, C_i is the true number of objects in the i-th image, and $\hat{C}_i$ is our estimate, defined in Equation (8). MAE, RMSE, and MAPE stand for Mean Absolute Error, Root Mean Squared Error, and Mean Absolute Percent Error, respectively.

A true positive (TP) is counted if an estimated location is at most at distance r from a ground truth point. A false positive (FP) is counted if an estimated location does not have any ground truth point at a distance at most r. A false negative (FN) is counted if a true location does have any estimated location at a distance at most r. This definition of precision represents the proportion of our estimated points that are close enough to a real point. The definition of recall represents the proportion of the points that we are able to detect. We are aware that we 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 Average Hausdorff Distance can be interpreted as the average location error in pixels.

Precision and recall as a function of r for the three datasets is shown in Figure 6. MAE, RMSE, and MAPE are shown in Table 1. Note that we are using the exact same architecture for the three tasks. The only difference is that 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 $\hat{C}_i = C_i = 1 \quad \forall i$.

A naïve alternative approach to object localization would be to use generic object detectors such as Faster R-CNN [5]. 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. Table 2 and Table 3 show the results of Faster R-CNN on the mall dataset and the pupil dataset, respectively. We use bounding boxes of size 20×20 (the approximate average head and pupil size), anchor sizes of 16×16 and 32×32 and

an Intersection Over Union threshold of 0.4. We use the VGG-16 architecture and train for 150 epochs using Stochastic Gradient Descent with learning rate of 1×10^{-3} and momentum of 0.9. We experimentally observed that Faster R-CNN struggles with detecting very small objects that are very close to each other. This is consistent with the observations in [8]. In that study, all generic object detections perform very poorly, with Faster R-CNN yielding the highest mean Average Precision (mAP) of 5%.

We also experimented using Mean Shift [53] instead of Gaussian Mixtures (GM) to detect the local maximima. 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 training an order of mangnitude. 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 their scikit-learn implementations [54].

Lastly, as our method locates and counts objects simultaneously, we can use it as a counting technique. We evaluate the performance of our technique in the task of crowd counting using the challenging Shanghaitech dataset presented in [18]. Table 4 shows the performance of our method against state-of-the-art techniques. While we do not claim to outperform the state-of-the-art methods that are specifically desinged for crowd counting, we achieve comparable performance with a single-class keypoint detector.

A PyTorch implementation of the Weighted Hausdorff Distance is available at `https://github.com/javiribera/weighted-hausdorff-loss`.

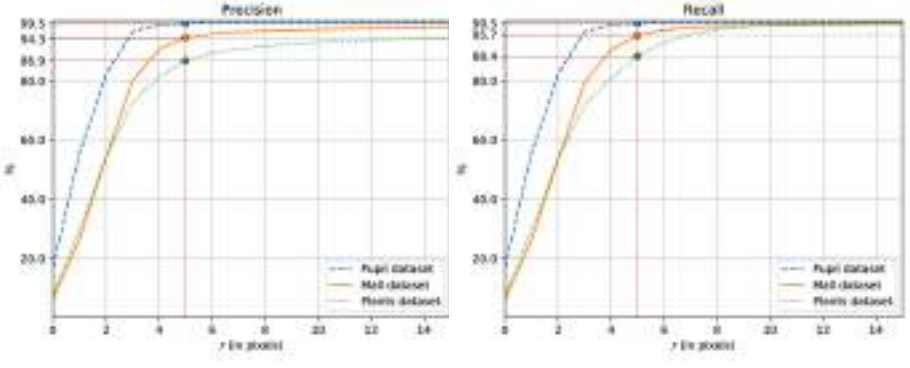

Fig. 6. Precision and recall as a function of r for the three datasets. r is the maximum distance between a ground truth point and an estimated object location to consider a correct or missing detection. As expected, the higher r the easier it is to achieve a higher precision and recall. Note that r is only an evaluation parameter. It is not needed during training or testing.

Table 1. Results of object localization and counting for the three datasets using our method. The metrics are defined in Equations (11)-(15). Figure 6 shows precision and recall for other r values. The Average Hausdorff Distance (AHD) is defined in Equation (4). We do not provide regression metrics for the pupil dataset because there is always a single pupil in the image. Thus, there is no need for regression and we fix the object count as $\hat{C} = C = 1$.

Metric	**Mall dataset**	**Pupil dataset**	**Plant dataset**	**Average**
Precision ($r = 5$)	94.7%	99.5%	86.9%	**93.7%**
Recall ($r = 5$)	95.4%	99.5%	88.4%	**94.4%**
MAE	1.8	-	2.0	**1.9**
RMSE	2.3	-	2.8	**2.6**
MAPE	5.7%	-	4.3%	**5.0%**
AHD	4.9 px	2.5 px	9.3 px	**5.6 px**

Table 2. Comparison of the performance of different methods for head location using the mall dataset.

Metric	**Faster-RCNN**	**Ours**
Precision ($r = 5$)	81.1%	**94.7 %**
Recall ($r = 5$)	76.7%	**95.4%**
MAE	4.7	**1.8**
RMSE	5.6	**2.3**
MAPE	14.8%	**5.7 %**
AHD	7.6 px	**4.9 px**

Table 3. Comparison of the performance of different methods for pupil detection. Precision and recall are equal because there is only one estimated object and one true object.

Metric	**Swirski [55]**	**ExCuSe [22]**	**Faster-RCNN**	**Ours**
Precision ($r = 5$)	77 %	77 %	99.5 %	**99.5 %**
Recall ($r = 5$)	77 %	77 %	99.5 %	**99.5 %**
AHD	-	-	2.7 px	**2.5 px**

Table 4. Comparison of the performance of methods specifically fine-tuned for crowd counting using the Shanghaitech (part B) dataset.

	MAE	**RMSE**
LBP + RR [20]	59.1	81.7
Zhang [21]	32.0	49.8
MCNN [18]	26.4	41.3
Crowd CNN [21]	32.0	49.8
Huang [20]	20.2	35.6
Switch-CNN [19]	21.6	33.4
CP-CNN [56]	20.1	30.1
Ours	29.0	48.5

7 Conclusion

We have presented a loss function for the task of localizing objects in images. 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 have estimated head locations, pupil centers, and plant centers. We have compared our method against generic object detectors and task-specific techniques. Future work will include developing a multi-class object location estimator in a single network.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (November 2016)
2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521** (May 2015) 436–444
3. Girshick, R.: Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (December 2015) 1440–1448
4. Uijlings, J.R.R., , Van De Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. International Journal of Computer Vision **104**(2) (September 2013) 154–171
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(6) (June 2017) 1137–1149
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. arXiv:1703.06870 (April 2017)
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C.: SSD: Single shot multibox detector. Proceedings of the European Conference on Computer Vision (October 2016) 21–37 Amsterdam, The Netherlands.
8. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (July 2017) Honolulu, HI.
9. Papadopoulos, D.P., Uijlings, J.R.R., Keller, F., Ferrari, V.: 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 (June 2016) 854–863 Las Vegas, NV.
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **11**(3) (December 2015) 211–252
11. Su, H., Deng, J., Fei-Fei, L.: Crowdsourcing annotations for visual object detection (July 2012) Toronto, Canada.
12. Bell, S., Upchurch, P., Snavely, N., Bala, K.: 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.
13. Lempitsky, V., Zisserman, A.: Learning to count objects in images. Proceedings of the Advances in Neural Information Processing Systems (December 2010) 1324–1332 Vancouver, Canada.
14. Shao, J., Wang, D., Xue, X., Zhang, Z.: Learning to point and count. arXiv preprint arXiv:1512.02326 (December 2015)
15. Xiong, F., Shi, X., Yeung, D.: Spatiotemporal modeling for crowd counting in videos. Proceedings of the International Conference on Computer Vision (October 2017) 5151–5159 Venice, Italy.
16. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern REcognition (June 2008) Anchorage, AK.
17. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE transactions on pattern analysis and machine intelligence **33**(9) (2011) 1820–1833

18. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. (June 2016) 589–597 Las Vegas, NV.
19. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (July 2017) 4031–4039
20. Huang, S., Li, X., Zhang, Z., Wu, F., Gao, S., Ji, R., Han, J.: Body structure aware deep crowd counting. IEEE Transactions on Image Processing **27**(3) (March 2018) 1049–1059
21. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2015) 833–841 Boston, MA.
22. Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., Kasneci, E.: ExCuSe: Robust pupil detection in real-world scenarios. Proceedings of the International Conference on Computer Analysis of Images and Patterns (September 2015) 39–51 Valletta, Malta.
23. Sundstedt, V.: Gazing at Games An Introduction to Eye Tracking Control. Morgan & Claypool Publishers, San Rafael, California (2012)
24. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (July 2017) 2107–2116 Honolulu, HI.
25. Gu, J., Yang, X., De Mello, S., Kautz, J.: Dynamic facial analysis: From bayesian filtering to recurrent neural network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (July 2017) 1548–1557 Honolulu, HI.
26. Fuhl, W., Santini, T., Reichert, C., Claus, D., Herkommer, A., Bahmani, H., Rifai, K., Wahl, S., Kasneci, E.: Non-intrusive practitioner pupil detection for unmodified microscope oculars. Computers in Biology and Medicine **79** (December 2016) 36–44
27. Panguluri, S.K., Kumar, A.A.: Phenotyping for Plant Breeding. Springer New York, New York, NY (2013)
28. Thornley, J.H.M.: Crop yield and planting density. Annals of Botany **52**(2) (August 1983) 257–259
29. Sui, R., Hartley, B.E., Gibson, J.M., Yang, C., Thomasson, J.A., Searcy, S.W.: High-biomass sorghum yield estimate with aerial imagery. Journal of Applied Remote Sensing **5**(1) (January 2011) 053523
30. Tokatlidis, I., Koutroubas, S.D.: A review of maize hybrids' dependence on high plant populations and its implications for crop yield stability. Field Crops Research **88**(2) (August 2004) 103–114
31. Araus, J.L., Cairns, J.E.: Field high-throughput phenotyping: the new crop breeding frontier. Trends in Plant Science **19**(1) (January 2014) 52–61
32. Neilson, E.H., Edwards, A.M., Blomstedt, C.K., Berger, B., Mller, B.L., Gleadow, R.M.: 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) (2015) 1817–1832
33. Farnham, D.E.: Row spacing, plant density, and hybrid effects on corn grain yield and moisture. Agronomy Journal **93** (September 2001) 1049–1053
34. Chauhan, B.S., Johnson, D.E.: Row spacing and weed control timing affect yield of aerobic rice. Field Crops Research **121**(2) (March 2001) 226–231
35. Aich, S., Ahmed, I., Obsyannikov, I., Stavness, I., Josuttes, A., Strueby, K., Duddu, H., Pozniak, C., Shirtliffe, S.: 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.
36. Attouch, H., Lucchetti, R., Wets, R.J.B.: The topology of the ρ-Hausdorff distance. Annali di Matematica Pura ed Applicata **160**(1) (December 1991) 303–320
37. Dubuisson, M.P., Jain, A.K.: A modified Hausdorff distance for object matching. Pattern Recognition (October 1994) 566–568
38. Lu, Y., Tan, C.L., Huang, W., Fan, L.: An approach to word image matching based on weighted Hausdorff distance. Proceedings of International Conference on Document Analysis and Recognition (September 2001) 921–925
39. K. Lin, K.L., Siu, W.: Spatially eigen-weighted Hausdorff distances for human face recognition. Pattern Recognition **36**(8) (August 2003) 1827–1834
40. Schutze, O., Esquivel, X., Lara, A., Coello, C.A.C.: Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation **16**(4) (August 2012) 504–522
41. Khiyari, H.E., Wechsler, H.: Age invariant face recognition using convolutional neural networks and set distances. Journal of Information Security **8**(3) (July 2017) 174–185
42. Fan, H., Su, H., Guibas, L.: A point set generation network for 3d object reconstruction from a single image. (July 2017) 2463–2471 Honolulu, HI.
43. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. BMC Medical Imaging **15**(1) (August 2015) 29
44. Zhou, S.K., Greenspan, H., Shen, D.: Deep Learning for Medical Image Analysis. Academic Press, London, United Kingdom (2017)
45. Liao, S., Gao, Y., Oto, A., Shen, D.: Representation learning: A unified deep learning framework for automatic prostate mr segmentation. Proceedings of the Medical Image Computing and Computer-Assisted Intervention (September 2013) 254–261 Nagoya, Japan.
46. Teikari, P., Santos, M., Poon, C., Hynynen, K.: Deep learning convolutional networks for multiphoton microscopy vasculature segmentation. arXiv:1606.02382 (June 2016)
47. Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S.: An efficiently computable metric for comparing polygonal shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(3) (March 1991)
48. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (October 2015) 234–241 Munich, Germany.
49. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (July 2017) Honolulu, HI.
50. Huber, P.J.: Robust estimation of a location parameter. The annals of mathematical statistics (1964) 73–101
51. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine **13**(6) (November 1996) 47–60
52. Loy, C.C., Chen, K., Gong, S., Xiang, T.: Crowd counting and profiling: Methodology and evaluation. In: Modeling, Simulation and Visual Analysis of Crowds. Springer (October 2013) 347–382
53. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence **24**(5) (2002) 603–619

54. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12** (2011) 2825–2830
55. Świrski, L., Bulling, A., Dodgson, N.: Robust real-time pupil tracking in highly off-axis images. Proceedings of the Symposium on Eye Tracking Research and Applications (March 2012) 173–176 Santa Barbara, CA.
56. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid CNNs. IEEE International Conference on Computer Vision (October 2017) 1879–1888 Venice, Italy.