International Journal of Computer Vision (2023) 131:407–430
https://doi.org/10.1007/s11263-022-01706-5

Robots Understanding Contextual Information in Human-Centered Environments Using Weakly Supervised Mask Data Distillation

Daniel Dworakowski[1] · **Angus Fung**[1] · **Goldie Nejat**[1]

Received: 23 November 2020 / Accepted: 18 October 2022 / Published online: 11 November 2022

Abstract
Contextual information contained within human environments, such as text on signs, symbols and objects provide important information for robots to use for exploration and navigation. To identify and segment contextual information from images obtained in these environments data-driven methods such as Convolutional Neural Networks (CNNs) can be used. However, these methods require significant amounts of human labeled data which is time-consuming to obtain. In this paper, we present the novel Weakly Supervised Mask Data Distillation (WeSuperMaDD) architecture for autonomously generating pseudo segmentation labels (PSLs) using CNNs not specifically trained for the task of text segmentation, e.g., CNNs alternatively trained for: object classification or image captioning. WeSuperMaDD is uniquely able to generate PSLs using learned image features from datasets that are sparse and with limited diversity, which are common in robot navigation tasks in human-centred environments (i.e., malls, stores). Our proposed architecture uses a new mask refinement system which automatically searches for the PSL with the fewest foreground pixels that satisfies cost constraints. This removes the need for handcrafted heuristic rules. Extensive experiments were conducted to validate the performance of WeSuperMaDD in generating PSLs for datasets containing text of various scales, fonts, orientations, curvatures, and perspectives in several indoor/outdoor environments. A detailed comparison study conducted with existing approaches found a significant improvement in PSL quality. Furthermore, an instance segmentation CNN trained using the WeSuperMaDD architecture achieved measurable improvements in accuracy when compared to an instance segmentation CNN trained with Naïve PSLs. We also found our method to have comparable performance to existing text detection methods.

Keywords Weakly supervised learning for robots · Environment context identification · Segmentation and labeling · Robot navigation and exploration

1 Introduction

Human-centered environments contain an abundance of contextual information such as text on signs, symbols, and objects that are used as landmarks to help guide users with point-to-point navigation in unknown environments (Vilar et al., 2014), and update maps of the environment (Peng et al., 2018). Service robots working in varying human-centered (Dworakowski et al., 2021) environments can exploit these types of contextual information to aid with navigation. For example, robots can use text on aisle signs in grocery stores to determine which aisles to search for a particular item (Thompson et al., 2018). They have also used contextual information for mapping and localization. Namely by using an annotated map of an office with room placards for goal directed navigation (Case et al., 2011). Robots have also created semantic maps using product locations (Cleveland et al., 2017), maps from unique text landmarks identified in images (Wang et al., 2015), and have used salient objects identified from learned features (e.g., edges, contours, etc.) for visual

Communicated by Frederic Jurie.

✉ Daniel Dworakowski
daniel.dworakowski@mail.utoronto.ca

Angus Fung
angus.fung@mail.utoronto.ca

Goldie Nejat
nejat@mie.utoronto.ca

[1] Autonomous Systems and Biomechatronics Laboratory (ASBLab), Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, ON M5S 3G8, Canada

odometry (Liang et al., 2019). These approaches rely specifically on a robot's ability to identify and localize context in an environment.

Recent work in the area of context detection and segmentation has made use of Convolutional Neural Networks (CNNs) to detect the presence of various types of objects or text within images of an environment (He et al., 2017; Zhang, et al., 2018; Radosavovic et al., 2018). However, the amount of human effort required to generate the vast expert labeled datasets required for training these existing networks significantly increases the overall time cost of their use. While some existing datasets do provide the labels necessary to train CNNs for segmentation tasks, they are limited in scope, and only target specific (e.g., people, animals) objects, not necessarily applicable for contextual information for robotic exploration and navigation problems. For example, the widely used PASCAL VOC dataset *only* has 20 object classes, mainly consisting of a limited set of people, animals, vehicles, and some indoor objects (Everingham et al., 2015). Therefore, CNNs trained with this dataset cannot be used to provide the large range of contextual information needed for robots to interpret their environments for exploration and navigation tasks. Other datasets provide larger numbers of classes, however, again only a small portion of examples relative to the number of classes are available. For example, the ADE20k dataset (Zhou et al., 2019) contains 3,169 classes of people, vehicles, and objects in rooms, but only 270 of the classes have more than 100 instances. Training CNNs to accurately segment large numbers of classes with a dataset such as ADE20k which has a long-tail class distribution is an open challenge (Li et al., 2020). On the other hand, datasets such as the Waymo open dataset (Sun et al., 2019), Open Images (Benenson et al., 2019), and ICDAR-15,17 (Karatzas et al., 2015; Nayef et al., 2017) contain several context categories with many examples that can be used by robots, e.g., text, cars, etc., but they are not fully labeled, requiring the classification of each pixel in an image in the dataset prior to training. Creating labels for all context instances within these datasets is significantly time consuming and must be done manually. Past research has found that the time required to manually generate a segmentation label is approximately 54 s per context instance (Jain & Grauman, 2013). Based on this we can estimate that it would take 20 years to segment all 2D objects in the Waymo open dataset! While transfer learning approaches using large-scale synthetic datasets such as SynthText (Gupta et al., 2016) can be used, these approaches still require a fully annotated dataset with segmentation labels from real-world environments. For the Total-Text dataset (Ch'ng and Chan 2017) with 11,459 text instances, this would require over 170 h of manual labeling.

Given the large investment needed to create these labels, several automated methods have been proposed to generate pseudo segmentation labels (PSLs) to segment an input image and to replace human expert labels. In general, Naïve methods have been proposed to avoid the human time cost of per-pixel segmentation by: (1) using bounding box labels (Khoreva et al., 2017), or (2) generating soft labels (i.e., labels between 0 and 1) based on the assumed shape of context instances (Liu et al., 2019). However, these approaches rely on specific assumptions about the shape and structure of context instances that are not always valid, for example, assuming objects do not have holes (e.g., the center of the character 'o').

Recently, semi or weakly supervised learning methods have been proposed to introduce learned features into the label generation process. Semi-supervised methods train a model using a dataset containing fully labeled and unlabeled data to generate pseudo labels for the unlabeled subset (Chapelle et al., 2010). An example of semi-supervised learning is data distillation where the fully labeled subset of a dataset is used to train a CNN. The CNN is then used to generate labels for unlabeled images using an ensemble of predictions from multiple transformed versions of each image (Radosavovic et al., 2018). These methods, however, still require human expert supervision to generate the labeled set. Alternatively, weakly supervised methods generate PSLs using partial information such as class or bounding box labels to completely label a dataset (Zhou, 2017). For example, weakly supervised methods use techniques such as class peak response (Zhou et al., 2018), adversarial erasing (Wei et al., 2017), and Class Activation Maps (CAMs) (Saleh et al., 2018). However, these approaches can significantly restrict the types of CNNs that can be trained due to the use of specialized CNN layers which determine the contribution of pixels in the input image to the CNN's final output. For example, CAMs require an additional global average pooling layer followed by a fully connected layer. Requiring specific layers limits the applicability of such existing weakly supervised segmentation methods since they cannot be generalized to all problems.

To avoid fully training a network, in (Khoreva et al., 2017), unsupervised segmentation methods, such as GrabCut (Rother et al., 2004) were used. However, since this method is class agnostic, it results in significant noise being present in the generated labels, it reducing the performance of the trained models compared to fully supervised models (Khoreva et al., 2017).

As robots must operate in different environments with varying terrain, configurations, and objects, large datasets must be obtained to train a robust segmentation CNN for context detection. The size of these datasets makes generating segmentation labels time consuming and slow. Using weakly supervised methods to generate PSLs is desirable as they do not require fully labeled data and can therefore reduce manual labeling effort. However, weakly supervised methods require

training with large diverse datasets to learn a feature representation that must then be segmented using handcrafted heuristic rules. Moreover, training must be repeated if new data is added to a dataset (Saleh et al., 2018; Wei et al., 2017; Zhou et al., 2018) thus increasing the computational cost of their use. Alternatively, we propose using readily available CNNs that are not trained specifically for text segmentation (but rather object classification, scene classification, etc.) as their convolutional layers can readily be transferred to the segmentation task (Zamir et al., 2018). These CNNs are trained with sparse labels, such as the overall class of an object (chair or desk), or the classes of a sequence of objects (cat and dog), rather than per-pixel image classification (which exact pixels in an image contain a dog). In order to segment contextual information using PSLs, features within these classes (e.g., edges, contours) need to be identified. By extracting the features used by these CNNs and incorporating them into a weakly supervised learning framework we minimize the need for additional training, while providing a source of informative image features to guide label generation.

In this paper, we present a novel autonomous Weakly Supervised Mask Data Distillation pseudo label generation architecture, WeSuperMaDD, for the segmentation of contextual information in varying environments using a partially labeled dataset containing bounding boxes and context labels (e.g., object class labels, image caption labels, etc.). These labels can be used to directly train existing CNN models for robotics tasks such as context identification and recognition, including for robot navigation, exploration, and obstacle avoidance. Our main contributions are:

1) We present the first architecture which uses learned features extracted from existing CNNs not trained specifically for scene text segmentation to generate PSLs for: (a) sequences of disjointed objects (e.g., text characters) regardless of dataset sparsity or diversity, and (b) without the need for gradient-based backpropagation, removing the need also for iterative training.
2) We introduce an autonomous mask refinement system that uniquely searches for PSLs that both have the fewest foreground pixels and satisfy cost constraints as measured by a cost function. This mask refinement system is unique with respect to other existing techniques in that by automating this step, the need for human involvement is eliminated.
3) We perform extensive experiments to successfully validate that our proposed WeSuperMaDD generates PSLs of varying scale, fonts, curvatures and sizes of text in different environments and has a higher segmentation quality than several common and state-of-the-art existing weakly, semi, and fully supervised segmentation methods, while having segmentation accuracy comparable to a state-of-the-art semi-supervised method. Furthermore, with respect to removing the need for additional training, our proposed method is able to robustly and repeatedly generate PSLs on disjointed text and with higher segmentation accuracy than an existing state-of-the-art method.
4) Furthermore, we show that our trained instance segmentation CNN has improved accuracy in generating segmentation outputs than an instance segmentation CNN trained with the standard Naïve PSL generation method and maintains similar performance to several existing state-of-the-art text and segmentation detectors.

By combining these learned features and our new mask refinement process, we remove the costly need for human experts to train a CNN for PSL generation and to create the handcrafted rules for segmentation that are typically required in previous works. The main advantage of our architecture is that it allows for the robust generation of PSLs in partially labeled datasets obtained by robots as they explore their environments, without the need for large diverse datasets. The proposed method has wide applicability for various robotics context detection tasks such as the simultaneous segmentation and detection of text in scenes, classification of terrain types, and grasping and manipulation of objects.

2 Related Work

The literature on semi and weakly supervised learning is discussed in this section with application to both robotics and existing methods which: (1) generate pseudo labels using a single output modality (e.g., segmentation, CAMs, etc.), and (2) train an ensemble of CNN output modalities to generate pseudo labels.

2.1 Semi/Weakly Supervised Learning for Robots

Semi and weakly supervised learning approaches have been used in robotic applications such as: (1) fruit counting for agricultural robots (Bellocchio et al., 2019), (2) ego-motion estimation (Shariati et al., 2020), (3) object manipulation (Singh et al., 2017), (4) location detection for topological localization (Arandjelovic et al., 2016), and (5) 3D point cloud segmentation of a scene (B. H. Wang et al., 2019). Recently, a handful of papers have focused on using semi-supervised learning for semi-automatic labeling of objects from multiple perspectives (Gregorio et al., 2020), weakly supervised learning for segmenting terrain (Barnes et al., 2017; Wellhausen et al., 2019), and detecting surgical tools (Vardazaryan et al., 2018).

In (Gregorio et al., 2020), a semi-supervised method was proposed for generating object labels for industrial robotics. Images of an object were gathered by a camera mounted

on a robot arm from multiple viewpoints while tracking the camera pose. The first image was manually labeled with a bounding box label, and all subsequent images were labeled using a relative transform based on the robot's pose when the image was taken.

In (Wellhausen et al., 2019), a robot was teleoperated through an environment and the robot's footholds were recorded via 2D images taken by an onboard camera and proprioceptive sensors. The footholds in the image frame were labeled manually by terrain type. In (Barnes et al., 2017), images and point cloud data from a 2D camera and LIDAR were recorded while a user drove a vehicle. A traversable path was labeled by registering the path taken into image frames using camera extrinsic parameters and vehicle odometry. Obstacles were detected and labeled in an image using the point cloud data. In both cases, these partially labeled images were used to train a segmentation CNN to predict the labeled path properties.

In (Vardazaryan et al., 2018), a method was proposed for weakly supervised robotic surgical tool detection and localization using image level labels indicating tool presence. During training, an spatial pooling layer was applied to the final feature maps of a fully convolutional segmentation network to produce an output classification vector. The network was trained using a cross entropy loss to predict the presence of a tool. At test time, the position within the feature maps with maximal activation was considered to be the location of a surgical tool.

2.2 Single Modality Methods

In single modality approaches, only one output type is used from the CNN for PSL generation, e.g., per-pixel segmentation, and as a label for self-training to improve the PSLs. Both semi (Baek et al., 2019b; Bonechi et al., 2019; Wang et al., 2021) and weakly supervised (Khoreva et al., 2017; Zhou et al., 2018; Jing et al., 2020; Niu et al., 2019; Zhang et al., 2019; Zhao et al., 2018) methods use this approach to calculate PSLs.

2.2.1 Semi-Supervised Methods

Semi-supervised single modality methods train a CNN using a combination of fully and partially labeled data to generate pseudo labels for the partially labeled data using the fully labeled data as training labels (Baek et al., 2019b; Bonechi et al., 2019; Wang et al., 2021). For example, in (Bonechi et al., 2019), a semi-supervised text segmentation CNN used background-foreground segmentation to train a CNN using synthetic images of text. PSLs were generated automatically using the CNN's segmentation output from the text segments cropped from real images of various environments. These labels were then used to train a text segmentation CNN.

In (Baek et al., 2019b), a pipeline was proposed for training a text detection CNN using semi-supervised character segmentation. First, character segmentation was generated using a synthetic dataset containing character region masks and masks defining the connections between adjacent characters. During training, labels for real images were generated by applying a watershed transform to the CNN's character region output to obtain both the inter-character regions and per-character bounding boxes. The CNN was trained with both synthetic and real data concurrently to update labels for the real images throughout training.

In (Wang et al., 2021), a semi-supervised framework was proposed for training a pixel-level scene text segmentation CNN. The CNN had two parallel output branches, the first was used for polygon-level text mask prediction, and the second for pixel-level text mask prediction. The output prediction of each branch was fed back to the other branch so that each task was guided by the learning of the other. As each image only contained either a ground truth pixel-level or polygon-level mask, the prediction from the other branch served as a pseudo-label during training.

2.2.2 Weakly Supervised Methods

Weakly supervised single modality methods use only a partially labeled dataset and self-training to generate PSLs. The training targets used in self-training are generated using various different methods including: (1) using CNNs to predict segmentation of an image (Khoreva et al., 2017; Zhou et al., 2018; Jing et al., 2020; Zhao et al., 2018), (2) using CAMs processed by a Conditional Random Field (CRF) (Zhang et al., 2019), (3) a generative output where a CNN is used to model a joint probability distribution of images and image labels (Niu et al., 2019) or (4) using CNN features from pre-trained models without additional training (Mahendran & Vedaldi, 2016; Simonyan et al., 2014).

Segmentation In (Khoreva et al., 2017), a weakly supervised framework for training segmentation networks was presented. Initial PSLs were generated by using a modified GrabCut (Rother et al., 2004) method. The masks were combined with objectness proposals from multiscale combinatorial grouping (Pont-Tuset et al., 2017) using a union operation. A segmentation network was then trained with these proposals. Labels used for supervision were updated using a set of rules comparing the previous label and segmentation output.

In (Zhou et al., 2018), a weakly supervised object segmentation method consisting of a combination of image level labels and peaks in class response maps (local maxima in a feature map) was proposed. A classification loss was augmented with a peak stimulation function to force the network to focus on discriminative regions. At test time class peaks were detected in the response maps and were refined using

peak back propagation to generate an instance segmentation. The mask representing each peak in image space was ranked and then filtered using non-maximum suppression to obtain final object proposals.

In (Zhao et al., 2018), a weakly supervised framework for training segmentation networks using only bounding box annotations was presented. When training the object segmentation network, a graph-based mask refinement technique was used to combine information from the predicted segmentation probabilities, image texture, and ground truth bounding box to update the pseudo labels used as segmentation targets.

In (Jing et al., 2020), a weakly supervised three-stage framework for the training of a semantic segmentation network using only object labels was proposed. The first stage used a pre-trained unsupervised object segmentation network to generate a coarse initial mask output. The second stage enhanced the generated mask using GrabCut (Rother et al., 2004) to improve foreground–background segmentation. The third stage involved network training, where, for each training example, a target was generated by processing the network's output using the previous stage.

CAMs In (Zhang et al., 2019), a weakly supervised online training procedure was used to train a semantic segmentation CNN based on class labels. The CNN had two parallel output branches, the first was used for CAM and object classification, and the second for per-pixel segmentation. During training, the CAM was filtered using a CRF and converted into a mask containing foreground, background, and unknown information using a heuristic approach to train the segmentation output.

Generative In (Niu et al., 2019), a weakly supervised defect detection procedure using a cyclic generative adversarial CNN and class labels was used. Given both an image with a defect paired and an image without defects the CNN was trained to remove the defect from the image. Similarly, a second CNN was trained to introduce defects into images by using images with defects as ground truth. The outputs were then used as inputs into the opposite generative model and are trained to undo the changes to the image. Segmentation masks were generated by taking the difference between the generated defect free image and the original input and applying heuristic rules.

Pre-trained CNN Models In (Simonyan et al., 2014), two CNN feature visualization techniques were presented, with one extended to generate weakly supervised PSLs. Given an input image, a class label, and a CNN pre-trained for classification, the derivative of the class score with respect to the image was computed using backpropagation and the ground truth label. A saliency mask was then computed using the maximum gradient per image channel for each image pixel. GraphCut color segmentation was then applied. After GraphCut, the largest connected component set of foreground pixels was used as a PSL.

In (Mahendran & Vedaldi, 2016), a weakly supervised method using gradient information and a pre-trained CNN with a parallel reversed architecture were presented to generate PSLs. The reversed architecture was constructed using the gradient functions of individual layers along with per-layer manually specified rules. To generate a saliency mask, the CNN was evaluated until the last layer before softmax. The feature channel for visualization was selected as the channel with the maximally activated neuron. The features and the selected channel were then used within the reversed architecture to generate a per-pixel saliency mask. GrabCut segmentation was then applied. A PSL was generated by selecting the largest connected component set of foreground pixels.

2.3 Ensemble Approaches

Ensemble approaches combine multiple outputs of a CNN for pseudo label generation, taking advantage of multi-task learning and can therefore improve label quality for self-training (Ruder, 2017). The ensemble combinations include: (1) semi-supervised ensembling of multiple predictions from a single CNN using data distillation (Radosavovic et al., 2018), (2) weakly supervised ensemble of multiple segmentation maps (Ibrahim et al., 2018; Wu et al., 2020), (3) a weakly supervised ensemble of an attention and saliency map (Hou et al., 2017), and (4) weakly and semi-supervised ensembles of a CAM and either a saliency map, segmentation mask or intermediate CNN activations (Li et al., 2018; Saleh et al., 2018; Selvaraju et al., 2020; Wang et al., 2017; Wei et al., 2017, 2018).

Data Distillation In (Radosavovic et al., 2018), a pre-trained CNN was used in a semi-supervised data distillation processes to generate labels for unlabeled data. Data distillation involved ensembling predictions using application specific rules from several augmented versions of an input image. Final labels were generated using task specific rules.

Multiple Segmentation Maps In (Ibrahim et al., 2018), a semi-supervised approach that iteratively improves an initial pre-trained segmentation model was presented. Two segmentation models were pre-trained using fully labeled data, the first was trained with bounding boxes and images as input and the second was a self-correction module which corrected segmentation errors made by the first model. Next, a segmentation CNN taking only the image as input was trained, pseudo labels for the weakly supervised part of the dataset were generated by applying the self-correction network to the CNN's predictions.

In (Wu et al., 2020), a semi-supervised text-level segmentation approach that generates polygon-level segmentation masks from bounding box annotations was presented. It

consisted of two components: (1) Bounding Box Supervision (BBS) with Skeleton Attention Segmentation Network (SASN), and (2) Dynamic Self Training (DST). BBS consisted of three stages, whereby the first stage trains SASN with synthetic data with character-level annotations to generate polygon pseudo-labels; the second stage uses bounding box annotations to crop real images which are passed into SASN to generate polygon pseudo-labels; and a third stage which combines pseudo-labels to generate a global pseudo-label. In DST, supervised learning was used to train an initial detector to generate foreground maps on unlabeled data. Background maps were then obtained to filter false negatives using edge detection and distance thresholding. Images were resized to a pre-defined set of scales to generate multi-scale predictions. Training was performed using both foreground predictions of unlabeled data and foreground segmentations of labeled data. This method segments the entire region containing the text as one continuous object, which can introduce background noise pixels. Moreover, it requires handcrafted heuristic rules such as distance thresholding to filter false positives.

Ensembling Saliency and Attention Maps In (Hou et al., 2017), a weakly supervised Expectation Maximization (EM) based method for the generation of segmentation masks using only image labels was presented. An initial estimate for the mask was generated using the per pixel maximum of a class agnostic saliency map and an attention map (Zhang et al., 2018) obtained from pre-existing CNNs via excitation back-propagation. The M-step trained a CNN with a multi-part loss function comparing the posterior and the predicted mask. The E-step was performed by constraining the latent posterior using the image labels, this was used to update target labels at each iteration.

Ensembling of a Combination of Methods In (Wei et al., 2017), a weakly supervised iterative object region erasing approach for object segmentation using class labels was presented. The process first trains a CNN with CAMs to convergence. Using heuristic rules, the CAMs and a saliency map from a pre-trained network were combined to remove aspects of the image discriminative for a particular class. This process was repeated beginning from training until the network no longer converged. Pseudo labels were formed using the removed regions.

In (Saleh et al., 2018), a weakly supervised segmentation CNN trained using class labels was presented. Both intermediate network activations and CAM features were combined to generate a mask label. Intermediate activations taken from layers of the CNN were hand selected based on their apparent discriminative ability. The masks were combined and binarized using heuristic rules and smoothed using a CRF. The predicted mask and class labels were used to train the network's segmentation output.

In (Li et al., 2018), a weakly supervised procedure to refine salient object masks generated from an unsupervised method using a CNN trained with class labels was proposed. A CNN was trained using supervision from the original saliency masks and the class label of the image. Then, the original saliency mask, predicted mask, and the top-3 class CAMs were fused using a CRF to produce an updated annotation for each image. The training and label updating process was repeated to generate the final object labels.

In (Wang et al., 2017), a weakly supervised training procedure was used to train a CNN for salient object segmentation using class labels. A modified CAM layer was used to predict segmentation masks. The network was initially trained for classification, with an L1 penalty on the saliency mask. During training, a saliency map was predicted and refined using a CRF, which was used to train the network with a bootstrapping loss.

In (Wei et al., 2018), a CNN with dilated convolution layers was presented for generating pseudo labels in a semi-supervised setting, where only some images had class level labels. The CNN was trained to predict a CAM, and a saliency map. The outputs were combined using a mask merging procedure to update the segmentation target during training with class level labels. When training with fully labeled images, the segmentation output was trained using a per-pixel per-class loss.

In (Selvaraju et al., 2020), a weakly supervised method for generating a segmentation CNN using a pre-trained classification CNN, Grad-CAM, and class labels was presented. Grad-CAM generated class activation maps by taking the weighted average of feature maps at a particular CNN layer. The weights were calculated using the average gradient of the class score with respect to the feature map of interest. The method used Grad-CAMs within the Seed, Expand, Constrain (Kolesnikov & Lampert, 2016) method to generate PSLs. The latter method trained a segmentation CNN using a combination of a seeding, expansion, and constrain-to-boundary losses. The Seed loss trained the segmentation output to match weak localization landmarks provided by a combination of Grad-CAMs that have been generated from the pre-trained classification CNN and a saliency detection method. The expand and constrain losses were used to refine the initial segmentation seed to full objects.

In (Wan et al., 2018), a weakly supervised minimum entropy based approach was presented using object presence labels for generating bounding box pseudo labels. A CNN was trained with two additional branches: object localization and object discovery. First, region proposals for objects were generated using selective search and features were extracted using a CNN. At each training step these features were multiplied with an object confidence score. The object discovery branch was trained to assign a probability to the presence of an object within the region. In images labeled as not having

any objects, the CNN was trained with a classification loss. When objects were present, labels for proposals were generated using the CNN's object class prediction and spatially grouped region proposals. The object localization branch was trained using the classification loss by assigning objectness labels by comparing the discovery branch object probabilities to an empirically determined threshold.

In (Wan et al., 2019), a weakly supervised approach using a continuation multiple instance learning loss function and image level labels was proposed. A CNN was trained with two branches: continuation instance selection and continuation detector estimation. The branches were trained using a continuation loss function, containing a continuation instance selection loss and continuation detector estimation loss. Selective search was used to generate region proposals for potential objects and these regions were sorted into bags based on their object scores and spatial locations. Objects were assigned into the same bag if their Intersection over Union (IoU) exceeded a certain value. A hinge loss was applied to the bag with the highest average class confidence, using the image level label, the estimated target value, and predictions from the first branch.

2.4 Summary of Recent Work

The aforementioned single modality (Baek et al., 2019b; Bonechi et al., 2019; Jing et al., 2020; Khoreva et al., 2017; Mahendran & Vedaldi, 2016; Niu et al., 2019; Simonyan et al., 2014; Wang et al., 2021; Zhang et al., 2019; Zhao et al., 2018; Zhou et al., 2018) and ensemble (Hou et al., 2017; Ibrahim et al., 2018; Li et al., 2018; Saleh et al., 2018; Selvaraju et al., 2020; Wan et al., 2018, 2019; Wang et al., 2017; Wei et al., 2017, 2018) methods have shown that pseudo labels can be generated for CNN model training. However, they all require large datasets with diverse data to train PSL generation CNNs without overfitting, with the exception of (Mahendran & Vedaldi, 2016; Simonyan et al., 2014). This may not always be feasible for robots obtaining information from their environments, as datasets of particular environments may be sparse and lack diversity (e.g., available only from one environment). Moreover, semi-supervised techniques require synthetic data or an expert user to fully manually label a large varied subset of data in order to generate labels, which can be a time consuming task (Baek et al., 2019b; Bonechi et al., 2019; Radosavovic et al., 2018; Wang et al., 2021; Wei et al., 2018). Weakly supervised single modality and ensemble approaches require handcrafted heuristic rules, either for the binarization of masks (Khoreva et al., 2017; Li et al., 2018; Mahendran & Vedaldi, 2016; Niu et al., 2019; Simonyan et al., 2014; Wei et al., 2017; B. Zhang et al., 2019), for distance thresholding (Wu et al., 2020), or for manual network analysis to determine where to gather information from (Saleh et al., 2018).

Methods that do not require additional training or large datasets for PSL generation (Mahendran & Vedaldi, 2016; Simonyan et al., 2014) are able to only select the largest connected component of foreground pixels and thus, cannot generate PSLs whose foreground pixels are disjointed, which is common in robotics environments (e.g., separate characters in a text PSL, a group of similar objects, etc.). The approaches that have been designed for robotic applications cannot be transferred to different segmentation tasks, as their training process requires inputs related to their specific problem, e.g., terrain information (Barnes et al., 2017; Gregorio et al., 2020; Wellhausen et al., 2019). However, weakly supervised methods can generate PSLs using new datasets that do not need to be fully labeled, thus reducing the human time–cost of generating pseudo labels.

In this paper, we present a novel weakly supervised architecture, WeSuperMaDD, which autonomously generates PSLs. The architecture can use pre-existing networks that are not trained for a segmentation task to determine PSLs without requiring any additional segmentation training in contrast to prior works. Furthermore, our approach does not assume the contextual information being segmented is contained within a single connected component. Our approach allows the use of smaller datasets with limited diversity used in robotics as existing CNN models are used as a basis, where the already learned features can be applied to our segmentation task, regardless of their application. To convert these learned features into pseudo labels, we introduce a new mask refinement system which incorporates an automated parameter search module that uniquely searches for the smallest PSL and also allows for disjointed mask elements to be found. Automating this search eliminates the need for handcrafted heuristic rules to generate PSLs. Therefore, unlike previous methods, we do not require any problem-specific information, thus generalizing our method to wider applications of robot segmentation tasks for obstacle avoidance in cluttered environments, and for grasping and manipulation of objects etc.

3 Weakly Supervised Mask Data Distillation

Our proposed Weakly Supervised Mask Data Distillation architecture, WeSuperMaDD, is presented in Fig. 1 and consists of two sub-systems: (1) Mask Data Generation Sub-System, and (2) Mask Refinement. WeSuperMaDD generates instance level PSLs, M, from images of contextual information in an environment. The procedure takes as input a set of CNNs, $\mathcal{F}$, not specifically trained for the task of context segmentation (e.g., trained for object classification, image captioning, etc.), a maximum number of iterations $t_{\max}$, and a dataset containing a set of images, I. Each image in the

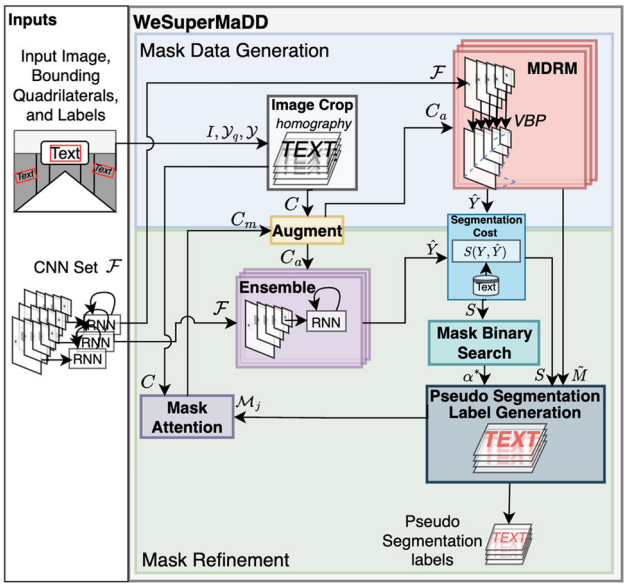

Fig. 1 The WeSuperMaDD architecture

dataset must have a set of bounding quadrilateral labels, and class label $\mathcal{Y}$.

The *Mask Data Generation* sub-system uses the *Image Crop* module to transform each of the context instances within an input image into a standard sized image crop $c \in C$ using the bounding quadrilaterals as reference. A crop, c, is then provided to the *Augment* module to generate an augmented set of crops via random image transformations. This set is used in the *Mask Discriminative Region Mining (MDRM)* module to find segmentation potential masks (SPMs), and the predictions made by the CNNs. SPMs are defined as 2D matrices containing the relative segmentation importance of every pixel in the input image.

Both the SPMs and CNN predictions are provided to the *Mask Refinement* sub-system to iteratively generate and refine PSL candidates. The *Segmentation Cost* module determines the distance of each of the predictions with respect to the ground truth. An average SPM is generated via the weighted average of all the SPMs, where each SPM is weighted by its associated cost. A PSL candidate is then generated in the *Pseudo Segmentation Label Generation* module using the initial parameters provided by the *Mask Binary Search* module and the average SPM. The *Mask Attention* module uses this PSL candidate to remove extraneous information from the initial crop, c. This focused crop is sent to the *Augment* module to generate a new set of augmented crops. The *Ensemble* uses the CNNs to generate predictions of the content of each of the crops. The *Segmentation Cost* module then calculates the cost of the predictions. The average prediction cost is used by the *Mask Binary Search* module to update parameters in the *Pseudo Segmentation Label Generation* module. This process is iterated $t_{\max}$ times and at each iteration the current lowest costing PSL candidate is outputted. The two sub-systems are discussed in more detail below.

Fig. 2 Image Crop module: **a** source image from ICDAR-15 dataset with bounding quadrilateral highlighted in red; and **b** a cropped sample

3.1 Mask Data Generation Sub-System

Given an input image and bounding quadrilateral labels representing the outer boundaries of all context instances, the *Mask Data Generation* sub-system produces SPMs, which contain regions in the image that are determined to have discriminative potential. The overall process for the sub-system is as follows:

3.1.1 Image Crop

The *Image Crop* module takes as input the full image containing contextual information and the bounding quadrilateral labels, and produces images of cropped context instances, Fig. 2. This is achieved using the perspective transformation that changes the context boundary coordinates from the input image into a standard size container, i.e., the input size expected by $\mathcal{F}$. The transform is applied to each of the bounding quadrilaterals to attain the set of crops, C. The output cropped context instances are sent to the *Augment* module.

3.1.2 Augment

The *Augment* module takes as input a single crop, c, and generates new samples representing the same context instance with different viewing perspectives and added noise. The objective is to force the CNNs to not rely on a single set of image features to complete the task they were trained for by randomly altering each image. The image augmentation process is adapted from (Radosavovic et al., 2018), where a single crop, c, is randomly augmented to generate a set of crops C_a and inverse transforms, $\mathcal{T}^{-1}$. Augmentations that change the location or shape of context instances must be invertible so predictions can be merged after the inverse is applied. Figure 3a presents sample crops obtained from this process. The augmented crops and their inverse transforms are provided to the *MDRM* module or the *Ensemble* module in the *Mask Refinement* sub-system.

(a) (b)

Fig. 3 Fig. 2b after processing by: **a** the *Augment* module; and **b** the SPMs generated by VBP from **a**

3.1.3 Mask Discriminative Region Mining (MDRM)

Given the set of augmented crops, the *MDRM* module produces masks representing the discriminative regions of each crop in the form of SPMs. This module uses the features extracted by existing CNNs to generate SPMs for PSL generation. The module begins by performing a forward pass through each of the CNNs in $\mathcal{F}$, and storing their predictions for their respective tasks, $\hat{Y}$, where:

$$\hat{Y} = \left[\hat{y}_{j,k}\right] \forall \left(c_j,\ f_k\right) \in C_a \times \mathcal{F},\ \hat{y}_{j,k} = f_k\left(c_j\right). \tag{1}$$

We use VisualBackProp (VBP) (Bojarski et al., 2018), which examines activations in intermediate CNN layers to determine the discriminative regions in an input crop to form an SPM. The regions estimate the relative contribution of each of the pixels to the network output $\widehat{y}_{j,k}$. This value is used to estimate the relative importance of each pixel for segmentation. VBP has a fast runtime, high visualization quality, and can be easily applied to common network backbones e.g., ResNets (He et al., 2016), VGG (Simonyan & Zisserman, 2014), etc. Sample SPMs obtained using VBP are shown in Fig. 3b.

After obtaining the SPMs from VBP, the inverse transform corresponding to the crop that was used to generate the mask is applied to attain the set of masks, $\widetilde{M}$, where:

$$\begin{aligned} \widetilde{M} &= \left[m_{j,k}\right] \forall \left(c_j,\ f_k\right) \in C_a \times \mathcal{F}, \\ m_{j,k} &= \mathcal{T}_j^{-1}\left(VBP\left(c_j,\ f_k\right)\right) \end{aligned} \tag{2}$$

and sent to the *Pseudo Segmentation Label Generation* module in the *Mask Refinement* sub-system. Additionally, the predictions, $\widehat{Y}$, are passed to the *Segmentation Cost* module.

3.1.4 Segmentation Cost

The *Segmentation Cost* module takes as input both a task prediction, $\widehat{y}$, and a ground truth label, y, and assigns a segmentation accuracy cost to each prediction. In particular, we use a cost function, $s = \mathcal{S}(y, \widehat{y})$, to compare the predicted output of a network with the ground truth. The output of this module is either passed to the *Pseudo Segmentation Label Generation* module for mask weighting or the *Mask Binary Search* module to update parameters controlling the creation of PSL candidates, both within the *Mask Refinement* sub-system.

3.2 Mask Refinement Sub-System

The *Mask Refinement* sub-system takes the SPMs, $\widetilde{M}$, the maximum number of search steps, $t_{\max}$, and the costs, $\mathcal{S}\left(\widehat{Y},\ y\right)$, of each of the predictions made from the augmented data to generate and refine PSL candidate using the set of CNNs, $\mathcal{F}$, where:

$$\mathcal{S}\left(\hat{Y},\ y\right) = \left[s_{j,k}\right] \forall \hat{y}_{j,k} \in \hat{Y},\ s_{j,k} = \mathcal{S}\left(\hat{y}_{j,k},\ y\right). \tag{3}$$

The process for the *Mask Refinement* sub-system is as follows:

3.2.1 Ensemble

The *Ensemble* module uses the ensemble technique in (Radosavovic et al., 2018) to reduce variance in the SPMs. It takes as input a set of crops provided by the *Augment* module, and the CNN set, $\mathcal{F}$, and generates predictions, $\widehat{Y}$, made by its members to measure the quality of a PSL candidate. Given the set of crops received from the *Augment* module, a prediction set is created, $\widehat{Y} = [\widehat{y}_{j,k}]$. The output of this module is sent to the *Segmentation Cost* module to inform the *Mask Binary Search* module.

3.2.2 Mask Binary Search

The *Mask Binary Search* module uniquely uses the current average prediction cost of the *Ensemble*, $\mathcal{S}\left(\widehat{Y},\ y\right)$, calculated from the cropped images from the *Mask Attention* module. This module autonomously selects the per-image segmentation control parameters. Namely, it updates the current gain, $\alpha^{(t)}$, for the *Pseudo Segmentation Label Generation* module to find the PSL candidate with fewest foreground pixels where the ensemble attains, on average, a cost less than the threshold $s_{\mathbb{1}}$. The initial bounds $u^{(0)} = \max(\widetilde{m})/\text{Otsu}(\widetilde{m})$, and $l^{(0)} = 0$, of the search are set such that the segmentation thresholds can reach the highest and lowest pixel activation of the average mask $\widetilde{m}$. Otsu$(\cdot)$ (Otsu, 1979) is a function that returns a value representing the binarization threshold which minimizes the intra-class variance between the foreground and background classes. The initial gain, $\alpha^{(0)} = 1$, is set such that the search process initialization is unbiased, however, the value can be user selected to incorporate a prior to the search process.

The module tracks and updates the best-known gain α^* and cost s^* according to:

$$\alpha^* \leftarrow \mathbb{1}_s * \alpha^{(t)} + (1 - \mathbb{1}_s *)\alpha^*, \tag{4}$$

Fig. 4 Candidate PSLs: **a** obtained by the *Pseudo Segmentation Label Generation* module; and **b** Mask Attention applied to each input crop using the candidate PSLs after each parameter update

$$s^* \leftarrow \mathbb{1}_s * s^{(t)} + (1 - \mathbb{1}_s *) s^*, \tag{5}$$

where $\mathbb{1}_s * = \mathbb{1}\left(s^{(t)} < s^*\right)$, and $\mathbb{1}(\cdot)$ is the indicator function. After finding a PSL candidate that satisfies $s^* < s_1$, the optimization procedure is updated to be:

$$\alpha^* \leftarrow \begin{cases} \alpha^{(t)}, \text{ if } s^{(t)} < s_{\mathbb{1}} \wedge \alpha^{(t)} > \alpha^* \\ \alpha^*, \text{ else} \end{cases}, \tag{6}$$

in order to store the largest gain adhering to the cost function.

The gain is updated every iteration using:

$$\alpha^{(t+1)} = \mathbb{1}_{s_{\mathbb{1}}}\left(\alpha^{(t)} + u^{(t)}\right)/2 + \left(1 - \mathbb{1}_{s_{\mathbb{1}}}\right)\left(\alpha^{(t)} + l^{(t)}\right)/2, \tag{7}$$

where $\mathbb{1}_{s_{\mathbb{1}}} = \mathbb{1}\left(s^{(t)} > s_1\right)$. The upper, $u^{(t)}$, and lower, $l^{(t)}$, limits are also updated based on the prediction quality:

$$l^{(t+1)} = \left(1 - \mathbb{1}_{s_{\mathbb{1}}}\right)\alpha^{(t)} + \mathbb{1}_{s_{\mathbb{1}}} l^{(t)}, \tag{8}$$

$$u^{(t+1)} = \mathbb{1}_{s_{\mathbb{1}}}\alpha^{(t)} + \left(1 - \mathbb{1}_{s_{\mathbb{1}}}\right)u^{(t)}. \tag{9}$$

The increase or decrease in the gain corresponds to increasing or decreasing the number of foreground pixels in the generated PSL. Figure 4a shows the evolution of PSL candidates generated using the parameters provided by the *Mask Binary Search* module. When the cost given the current PSL candidate is less than $s_{\mathbb{1}}$, the number of foreground pixels in the PSL is decreased by increasing the gain to remove extraneous pixels. When the cost given the current PSL candidate is greater than $s_{\mathbb{1}}$, the gain is decreased to increase the number of foreground pixels in the PSL candidate and incorporate additional image information. The module provides to the *Pseudo Segmentation Label Generation* module either the current calculated gain, $^{(t+1)}$, or the best gain, *, when $t = t_{\max}$ number of iterations have elapsed.

3.2.3 Pseudo Segmentation Label Generation

The *Pseudo Segmentation Label Generation* module takes the SPMs, $\widetilde{M}$, and determines a candidate PSL. We first ensemble the potential masks to produce a single SPM for each crop using a weighted average using the prediction

Fig. 5 **a** The weighted average of the SPMs from the *MDRM* module; and **b** Pseudo label generated after $t_{\max}$ steps based on **a**

cost. This reduces the influence of predictions based on non-discriminative regions on the mask ensemble. Given that lower costs indicate a better match we invert the cost and then apply the SoftMax function, $\sigma(\cdot)$, to obtain the relative weighting of each mask:

$$w_{j,k} = \sigma\left(\max\left(\mathcal{S}\left(\hat{Y}, y\right)\right) - \mathcal{S}\left(\hat{Y}, y\right)\right)_{j,k}. \tag{10}$$

A weighted average is used to combine all of the masks into an average SPM:

$$\tilde{m}^{(r,s)} = \sum_{j \in [1, |C_a|], k \in [1, |\mathcal{F}|]} w_{j,k} \tilde{M}^{(r,s)}_{j,k}, \tag{11}$$

where (r, s) is a pixel coordinate. Figure 5a presents a mask representing the weighted average of the SPMs in Fig. 3b.

We make the assumption that the regions of the image that contain the context instance are within the pixels of the SPM whose activation is greater than some value. We use Otsu's method (Otsu, 1979) as the basis for the segmentation thresholds since the mask would be properly segmented if the two classes were clustered in activation. We use the gain parameter, $\alpha^{(t)}$, from the *Mask Binary Search* module to control mask binarization. Therefore, the threshold is set to be:

$$\epsilon^{(t)} = \alpha^{(t)} \cdot \text{Otsu}(\tilde{m}). \tag{12}$$

We binarize the SPM given the current threshold $\epsilon^{(t)}$, using a threshold operation at each pixel in $\tilde{m}$:

$$\overline{m}^{(t,r,s)} = \mathbb{1}\left(\tilde{m}^{(r,s)} > \epsilon^{(t)}\right). \tag{13}$$

We then use the GrabCut algorithm (Rother et al., 2004) to clean the boundary between the foreground and background. GrabCut uses the input crop image, a Gaussian Mixture Model (GMM), and color to predict the labels of unknown regions of space given an initial foreground and background segmentation. Using the binarized mask, $\tilde{m}^{(t)}$, we generate the unknown region using erosion, $\text{er}(\cdot)$, and dilation, $\text{dl}(\cdot)$, operations. Specifically, we generate our known foreground and background as:

$$m^{(t)}_{\text{fg}} = \text{er}\left(\overline{m}^{(t)}\right), \tag{14}$$

$$m_{\text{bg}}^{(t)} = \neg \text{dl}\left(\overline{m}^{(t)}\right). \tag{15}$$

Lastly the probable foreground and background regions are:

$$m_{\text{pfg}}^{(t)} = m^{(t)} \wedge \neg m_{\text{fg}}^{(t)}, \tag{16}$$

$$m_{\text{pbg}}^{(t)} = \neg m_{\text{bg}}^{(t)} \wedge \neg m_{\text{fg}}^{(t)}. \tag{17}$$

We apply the GrabCut algorithm using these masks and with an additional set of masks where the probable foreground also includes the foreground mask, selecting the latter if a minimum number of pixels are detected as foreground. We obtain the PSL candidate, $m^{(t)}$, that predicts the class of the pixels in the boundary region shown in Fig. 5b. The process stops when $t = t_{\max}$, then the mask is provided as the output PSL, m. If $t < t_{\max}$, then the PSL candidate is passed to the *Mask Attention* module, continuing the iterative process of PSL generation and evaluation.

3.2.4 Mask Attention

The *Mask Attention* module is a hard attention operation, that removes pixels of the input crop that are not estimated to be part of the foreground class. In particular, for crop $c^{(t)}$ the PSL is applied as a mask to the input crop as:

$$c^{(t,r,s)} = c^{(r,s)} m^{(t,r,s)} + \mu\left(1 - m^{(t,r,s)}\right), \tag{18}$$

thus, leaving only information from discriminative regions by setting the background to a default color μ. Figure 4b shows the evolution of the selected image region using subsequent PSL candidates from the *Pseudo Segmentation Label Generation* module. The masked crops C_m are sent to the *Augment* module to generate samples to evaluate the quality of the PSL candidate.

4 WeSuperMaDD Algorithm

The overall architecture is summarized within Algorithm 1 and the *Pseudo Segmentation Label Generation* module (the *PSLGEN* function) is detailed in Algorithm 2:

Algorithm 1 WeSuperMaDD procedure for a single image.

inputs:
$t_{\max}$: the number of search iterations, $\alpha^{(0)}$: the initial gain, $\text{fg}_{\min}$: the minimum number of foreground pixels in a PSL, $\mathcal{F}$: the set of pre-trained CNNs, $\mathcal{I}$: image, task labels and bounding quadrilaterals.
output:
$(I, \mathcal{Y}, \mathcal{Y}_q) = \mathcal{I}$ #Image, task label and bounding quadrilateral label
$\mathcal{M}_{\mathcal{I}} = \emptyset$ #Empty set to hold the new labels for an image
for $y \in \mathcal{Y}, g_q \in \mathcal{Y}_q$ **do** #For all context instances in the image
  $h_{c_q}^{g_q} = H(g_q, c_q)$ #Calculate homography
  $c = \text{Interpolate}(I, h_{c_q}^{g_q})$ #Interpolate I to perform crop
  $(C_a, \mathcal{T}^{-1}) = \text{Augment}(c)$ #Generate augmentation samples
  ***#MDRM* module**
  $\hat{Y} = [\hat{y}_{j,k}] \forall (c_j, f_k) \in C_a \times \mathcal{F}, \hat{y}_{j,k} = f_k(c_j)$ #CNN predictions
  $\widetilde{M} = [m_{j,k}] \forall (c_j, f_k) \in C_a \times \mathcal{F}, m_{j,k} = \mathcal{T}_j^{-1}(\text{VBP}(c_j, f_k))$ #VBP masks
  $S = [s_{j,k}] \forall y_{j,k} \in \hat{Y}, s_{j,k} = \mathcal{S}(\hat{y}, y)$ #Cost of the predictions
  $\widetilde{m} = \sum_{j \in [1,|C_a|], k \in [1,|\mathcal{F}|]} \widetilde{M}_{j,k} \cdot \sigma(\max(S) - S)_{j,k}$ #Weighted average mask
  $l^{(0)} = 0$ # Initialize search parameters
  $u^{(0)} = \max(\widetilde{m})/\text{Otsu}(\widetilde{m})$
  $s^* = -\infty$
  for $t \in [0, t_{\max}]$ **do** #Mask binary search loop
    $m^{(t)} = PSLGEN(c, \alpha^{(t)}, \widetilde{m}, \text{fg}_{\min})$
    $c^{(t)} = c \cdot m^{(t)} + \mu \cdot (1 - m^{(t)})$ #Attention
    $(C_a, \_) = \text{Augment}(c^{(t)})$
    $\hat{Y} = [\hat{y}_{j,k}], \hat{y}_{j,k} = f_k(c_j) \forall (c_j, f_k) \in C_a \times \mathcal{F}$ #Ensemble
    $s^{(t)} = \mathbb{E}[S(\hat{Y}, y)]$ #Segmentation cost
    #Binary search based on average prediction cost
    $\alpha^{(t+1)} = \mathbb{1}_{s_{\text{l}}}(\alpha^{(t)} + u^{(t)})/2 + (1 - \mathbb{1}_{s_{\text{l}}})(\alpha^{(t)} + l^{(t)})/2$
    $l^{(t+1)} = (1 - \mathbb{1}_{s_{\text{l}}})\alpha^{(t)} + \mathbb{1}_{s_{\text{l}}} l^{(t)}$
    $u^{(t+1)} = \mathbb{1}_{s_{\text{l}}}\alpha^{(t)} + (1 - \mathbb{1}_{s_{\text{l}}})u^{(t)}$
    $s^* \leftarrow \mathbb{1}_{s^*} s^{(t)} + (1 - \mathbb{1}_{s^*}) s^*$
    if $s^* < s_{\text{l}}$ **then**
      $\alpha^* \leftarrow \begin{cases} \alpha^{(t)}, \text{if } s^{(t)} < s_{\text{l}} \wedge \alpha^{(t)} > \alpha^* \\ \alpha^*, \text{else} \end{cases}$
    else
      $\alpha^* \leftarrow \mathbb{1}_{s^*}\alpha^{(t)} + (1 - \mathbb{1}_{s^*})\alpha^*$
    end if
  end for
  $m = PSLGEN(c, \alpha^*, \widetilde{m}, \text{fg}_{\min})$ #Final PSL
  $\mathcal{M}_{\mathcal{I}} \leftarrow \mathcal{M}_{\mathcal{I}} \cup \{(m, y, g_q)\}$ #Update the dataset
  end for
return $\mathcal{M}_{\mathcal{I}}$

Algorithm 2 The *Pseudo Segmentation Label Generation* module (PSLGEN).

inputs: c: image crop, α: current gain, $\widetilde{m}$: weighted average of the SPMs, $\text{fg}_{\min}$: minimum number of foreground pixels in a PSL.
output:
$\epsilon = \alpha \cdot \text{Otsu}(\widetilde{m})$ #Calculate segmentation threshold
$\overline{m} = \mathbb{1}(\widetilde{m} > \epsilon)$ #Binarize mask with the threshold
$m_{\text{fg}} = \text{er}(\overline{m})$ #Known foreground
$m_{\text{bg}} = \neg \text{dl}(\overline{m})$ #Known background
$m_{\text{pfg}} = m \wedge \neg m_{\text{fg}}$ #Probable foreground
$m_{\text{pbg}} = \neg m_{\text{bg}} \wedge \neg m_{\text{fg}}$ #Probable background
#Perform GrabCut optimization to generate PSL candidate *m*, where 0 indicates a mask with only 0's
$m = \text{GrabCut}(c, \mathbf{0}, m_{\text{bg}}, m_{\text{pfg}} \vee m_{\text{fg}}, m_{\text{pbg}})$
if $\sum m^{(\text{r,s})} < \text{fg}_{\min}$ **then** #Check number of foreground pixels
  $m = \text{GrabCut}(c, m_{\text{fg}}, m_{\text{bg}}, m_{\text{pfg}}, m_{\text{pbg}})$
end if
return m

5 Experiments

Our experiments focus on the Optical Character Recognition (OCR) task of the simultaneous detection and segmentation of text in environments due to: (1) its applicability in robotics applications for the detection of text signs to aid for exploration and navigation in unknown cluttered structured environments, and (2) the limitation of existing weakly supervised PSL generation methods as they are not generalizable to text segmentation since they require the inclusion of additional problem specific NN layers. The weakly supervised generation of text PSLs thus represents a challenging problem not well explored in the existing literature.

The majority of existing publicly available text detection datasets do not have segmentation labels, therefore, to train an instance segmentation CNN for the simultaneous detection and segmentation of text would require segmentation labels to be manually generated by a human expert or by an autonomous PSL generation method. In these experiments, we investigate the performance of our WeSuperMaDD method in autonomously generating the needed PSLs when only bounding quadrilateral and class label data are available for training.

Herein, we perform three experiments: (1) an Ablation Study, (2) a comparison study of WeSuperMaDD's performance in the generation of PSLs versus other standard methods, and (3) a detailed investigation of instance segmentation of various context images using numerous datasets. The performance metrics used in these experiments are defined as: (1) Precision (P), (2) Recall (R), and (3) F_1 score. Experiments were conducted on a server with a Titan V GPU, an AMD 2990WX CPU, and 128 GB of memory.

5.1 Ablation Study

We performed an ablation study to examine the relative importance of the hyper-parameters used by WeSuperMaDD with respect to segmentation F_1 scores. Namely, we evaluate the performance of our segmentation method by comparing the class of each pixel of the predicted PSLs and the ground truth masks available in the ICDAR-13 dataset (Karatzas et al., 2013).Within the dataset, pixels overlapping text characters are labeled as foreground, and those not overlapping text characters are labeled as background. We use the ICDAR-13 dataset (Karatzas et al., 2013) as it is the only real-world dataset with a per-character text segmentation ground truth with quadrilateral labels for straight text. Here we define a positive detection as a predicted PSL pixel matching a ground truth mask pixel (e.g., $m^{(r,s)} = m_{\text{gt}}^{(r,s)}$). The following subsections provide the details on the "Ensemble of SPMs", the "Segmentation cost function", and the "Training datasets" and "Testing dataset" used in the ablation study, respectively.

5.1.1 Ensemble of SPMs

To validate the ability of the WeSuperMaDD approach to ensemble SPMs from multiple sources, we generate our CNN set, $\mathcal{F}$, with the following text recognition CNNs which identify the text string contained in an image: (1) a Character Recognition Neural Network (CRNN) (Shi et al., 2016) for straight text, and (2) a case-insensitive Thin Plate Spline (TPS) CNN with bidirectional long short-term memory and attention (Baek et al., 2019a) for both straight and curved text. These CNNs were selected as they use standard structures and are known to be top performers in the text recognition task. To ensemble the SPM predictions we must invert the TPS layer prior to using it in the *Mask Generation* module.

5.1.2 Segmentation Cost Function

The cost function, $\mathcal{S}(\widehat{y},\ y)$, is modeled using the Edit Distance (ED) between the string predicted by the ensemble and the ground truth string. ED is defined as the minimum number of elementary string operations required to transform one string into the other. We set the threshold, $s_{\mathbb{1}}$ to 1. A cost below this value indicates the CNN was able to identify the text contained within an input image.

5.1.3 Training Datasets

We train the CRNN with a union of synthetic data from the MJSynth dataset (Jaderberg et al., 2014) and cropped ground truth regions from the 800,000 images of the Synth-Text dataset (Gupta et al., 2016), with a combined total of approximately 14 million synthetic English text instances. We also use the union of both the IIIT5k (Mishra et al., 2012) and cropped ground truth text instances from the ICDAR-15 (Karatzas et al., 2015) dataset, containing approximately 6600 text instances from 1500 images of scene text. For the TPS model, we use an available pre-trained model ("What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis" 2020), which was trained using both the MJSynth (Jaderberg et al., 2014) and SynthText (Gupta et al., 2016) datasets.

5.1.4 Testing Dataset

Performance was evaluated on the ICDAR-13 dataset (Karatzas et al., 2013). The dataset contains 462 images of focused real scene text, with a total of 1,944 text instances. All generated PSLs are resized and compared individually to maintain equal weighting between large and small text instances.

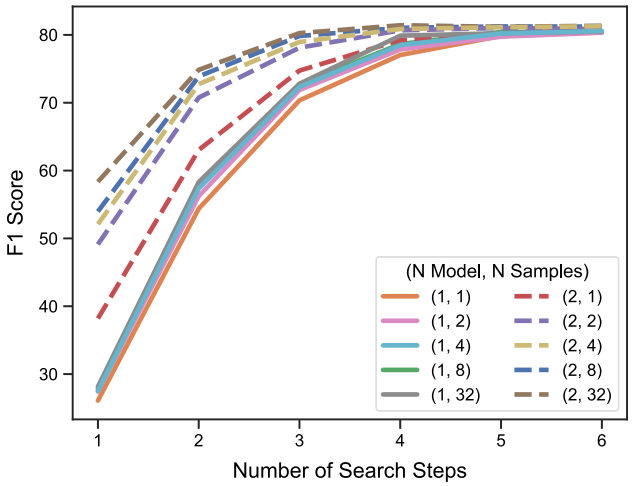

Fig. 6 Ablation of number of models, augmentation samples, and search steps

5.1.5 Procedure

To perform the ablation study, we used the ICDAR-13 training set and recorded the F_1 score obtained using each set of parameters. In particular, we varied: 1) the number of samples generated by the *Augment module,* i.e., $|C_a|= \{1, 2, 4, 8, 32\}$, 2) the number of models used in the ensemble, i.e., $|\mathcal{F}|= \{1, 2\}$, with only the CRNN ($|\mathcal{F}|= 1$), or both CRNN and TPS ($|\mathcal{F}|= 2$), and 3) the number of binary search iterations, i.e., $t_{max} = \{1, 2, 3, 4, 5, 6\}$, where $t_{max} = 1$ corresponds to no search steps performed, i.e., only the mask generation step is implemented (with no mask refinements).

The results of the ablation study are summarized in Fig. 6. In general, increasing the value of any of the hyperparameters increases the overall F_1 score. As can be seen with one search step including 2 models and increasing the number of samples to 32 significantly increases the F_1 score from 38 to 58. Increasing either of these parameters improves PSL generation performance as the activation value of text regions is increased in the SPMs. Thus, the increased activation improves the likelihood that the feature locations will be included in the PSLs.

When increasing the number of search steps to 6 (i.e., five iterations of mask refinement) from 1 step (i.e., no mask refinement), the F_1 score performance improves by 40% when also using 2 models and 32 samples, and 210% when using 1 model and 1 sample. Performance gains are between these two values for all other combinations of the number of models and samples. The increased number of search steps improves the ability of the mask binary search to remove extraneous features from the PSLs as more segmentation parameters are tested. Based on the overall trend, the number of search steps is the most significant determining factor for PSL generation performance.

However, it can be seen that the performance gain decreases as the number of samples increases. For example, only small performance gains are observed with increasing the number of search steps beyond 4 while also increasing the number of samples or models. As can be seen in Fig. 6, when using both 32 samples and 2 models, increasing the number of search steps past 4 to 5 or 6 no longer provides performance gains after reaching a peak F_1 score of approximately 81 at 4 steps and plateauing. After 4 search steps the CNNs can no longer reduce the size of the segmentation masks while satisfying the cost constraints. Using the knowledge gained from these results, we use 4 steps (mask generation step and 3 iterations of the mask refinement step), 32 samples, and 2 models, to reduce the computational cost of generating PSLs in our next experiments.

5.2 PSL Generation Experiments

We compare the performance of the WeSuperMaDD method against several standard methods. Performance is calculated similarly to the ablation study where the class of each pixel of the PSLs predicted by each method is compared to the ground truth masks in the ICDAR-13 test dataset (Karatzas et al., 2013) for straight text, and in the Total-Text dataset (Ch'ng and Chan 2017) for curved text. The Total-Text dataset consists of 1,555 images of real scene text containing a total of 11,459 text instances. The overall evaluation procedure for individual PSLs is the same as in the ablation study in Sect. "Ablation Study".

5.2.1 Methods for Comparison

We compare our method with PSLs generated using the following common weakly supervised techniques: (1) GrabCut (Rother et al., 2004), (2) Pyramid (Liu et al., 2019), and (3) Naïve (Ibrahim et al., 2018). These methods were chosen since they represent typical gradient free methods that: (1) can be directly applied to an OCR task without the need for training, and (2) have similar label requirements as our proposed method. The methods are applied to generate a PSL for all the context instances for straight text from the ICDAR-13 dataset and curved text from the Total-Text dataset. The GrabCut generation method (Rother et al., 2004) takes the full image and a bounding box representing the outer edges of a text instance and trains a GMM to segment the background colors from the unknown region inside the bounding quadrilateral which is used as the PSL. The Pyramid generation method generates PSLs using a soft label in the form of a pyramid, the pseudo label peaks at a value of one in the center of a ground truth quadrilateral and decays to a label of zero at the edges (Liu et al., 2019). In the Naïve generation method, PSLs are generated by labeling the interior of a

ground truth quadrilateral text region as foreground (Ibrahim et al., 2018).

5.2.2 State-of-the-Art Techniques

We additionally compare against the current state-of-the-art semi-supervised Supervision Generation Procedure (SGP) method, whose results are reported in (Bonechi et al., 2019), and weakly supervised Simple Does It (SDI) method (Khoreva et al., 2017). For the state-of-the-art methods which do not require additional training, we compare against Saliency Maps with GrabCut (SMG) (Simonyan et al., 2014). We also compare against the *fully supervised* Character Attention Fully Connected Network (CA-FCN) method (Liao, Zhang, et al., 2018), and the *semi-supervised* Bounding Box Supervision (BBS) method (Wu et al., 2020). In (Bonechi et al., 2019) a segmentation CNN was trained using text segmentation masks generated using a synthetic dataset generation procedure. The segmentation CNN was then applied to cropped images of real text from the ICDAR-13 dataset in order to generate PSLs. For SDI, we ourselves pre-train a segmentation CNN on the SynthText dataset. For fine-tuning, we use the ICDAR-15 dataset with initial masks generated from GrabCut. Furthermore, we also fine-tune with masks generated from our first stage Mask Data Generation which we name SDI* herein. The CNN was recursively trained by applying the model over the training set as segmentation labels for the subsequent training rounds. PSLs were generated by applying the final trained model on the ICDAR-13 dataset.

The CA-FCN method trains a segmentation CNN to predict segmentation masks representing the locations and classes of each character in a word. The method consists of a label generation pre-processing step in which per-character ground truth bounding boxes are converted into Naïve masks of half the size. To generate PSLs from CA-FCN, we use the optimal masks expected from the network by evaluating directly on the ground truth per-character bounding boxes. The BBS method trains a segmentation CNN to generate PSLs using word-level mask labels. Similar to CA-FCN, we use the optimal masks by evaluating directly on the word-level masks. The SMG method takes the full image, class label, and a pre-trained CNN to generate a class saliency map through backpropagation. We use the same CRNN model in WeSuperMaDD as the pre-trained CNN. To adapt the method for instance segmentation, we backpropagate the largest predicted correct class for each vector in the extracted feature sequence to create a sequence of saliency maps corresponding to each extracted feature. The responses from each saliency map are pooled together using logical OR to produce a final binary saliency map. Regions in the saliency map corresponding to text are cropped and then segmented by GrabCut.

5.2.3 Segmentation Results

We generate PSLs for the images in both the ICDAR-13 and Total-Text test set using each of the GrabCut, Pyramid, Naïve, and our WeSuperMaDD approaches and determine the F_1 score for each PSL. The average scores for all four methods are summarized in Table 1. For the ICDAR-13 dataset, a non-parametric Kruskal–Wallis test, $n = 1095$, showed a statistically significant difference in F_1 between all methods ($H = 1817$, $p < 0.001$). Posthoc Dunn tests were conducted between our WeSuperMaDD method and the other three methods and showed that our method has a statistically significant higher F_1 with respect to the GrabCut, ($Z = 7.68$, $p < 0.001$), Pyramid ($Z = 26.22$, $p < 0.001$), and Naïve ($Z = 18.54$, $p < 0.001$) methods, respectively. For the Total-Text dataset, a non-parametric Kruksal-Wallis test, $n = 2{,}543$, showed a statistically significant difference in F_1 between all methods ($H = 4429$, $p < 0.001$). Posthoc Dunn tests were conducted between our WeSuperMaDD method and the other three methods and showed that our method has a statistically significant higher F_1 with respect to the GrabCut, ($Z = 20.31$, $p < 0.001$), Pyramid ($Z = 35.18$, $p < 0.001$), and Naïve ($Z = 39.11$, $p < 0.001$), methods, respectively.

Sample PSLs (in red) obtained from the methods are also presented in Fig. 7 on the ICDAR-13 dataset, and in Fig. 8 on the Total-Text dataset, where each image is overlaid atop of the ground truth segmentation label. Our WeSuperMaDD approach, Fig. 7a and Fig. 8a, is able to find all characters with some small amounts of background information incorporated into the label. The GrabCut result in Fig. 7b and Fig. 8b highlights the problem with using an unsupervised approach, as it only finds a few of the characters rather than considering groups of characters concurrently. On the other hand, the pyramid PSL in Fig. 7c and Fig. 8c is conservative, selecting a fairly small amount of the image as foreground, particularly near the edges of the crop, which significantly reduces the overall score. In the case of the Naïve approach, Fig. 7d and Fig. 8d, all foreground pixels are included in the PSL, resulting in a 1.0 R, however, significantly more background pixels were labeled as the foreground class giving it one of the lowest P scores (Table 1). The inclusion of significant amounts of background pixels is typical to this approach, as it assumes all pixels are part of a foreground object.

5.2.4 Comparison with State-of-the-Art Techniques

Additionally, we compared the F_1 scores of the PSLs generated by WeSuperMaDD with the F_1 scores from the semi-supervised SGP trained with text segmentation labels published in (Bonechi et al., 2019) using images and the same evaluation procedure from the ICDAR-13 test set. The evaluation procedure provides a weighted F_1 score for each mask

Table 1 Comparison of segmentation methods on the ICDAR-13 and Total-Text test sets

Method	Dataset					
	ICDAR-13			Total-text		
	P	R	F_1	P	R	F_1
GrabCut	52.71	72.62	57.25	27.49	43.26	33.62
Pyramid	54.28	35.76	43.12	40.01	36.20	38.04
Naïve	50.88	100.00	67.44	32.71	100.00	47.74
Simple does it (round 1)	50.10	98.37	66.39	33.54	85.49	48.17
Simple does it (round 2)	50.27	99.12	66.71	33.97	83.00	48.21
Simple does it (round 3)	50.53	99.33	66.98	33.41	86.59	48.22
Simple does it* (round 1)	57.46	89.92	69.13	33.06	88.16	48.09
Simple does it* (round 2)	57.70	92.93	69.74	33.14	88.13	48.17
Simple does it* (round 3)	57.32	93.77	69.25	33.39	88.04	48.42
CA-FCN	60.98	35.76	45.08	–	–	–
Bounding box supervision	50.88	100.00	67.44	41.80	82.82	55.56
Saliency maps with grabcut	33.88	29.69	31.65	11.00	19.80	14.15
WeSuperMaDD (ours)	73.91	88.83	80.69	54.67	76.09	63.62

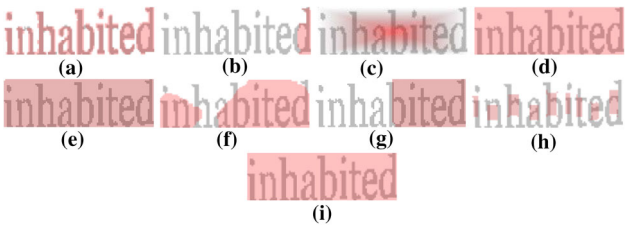

Fig. 7 Sample PSLs overlaid in red on an ICDAR-13 ground truth sample using the following weakly-supervised segmentation methods: **a** WeSuperMaDD; **b** GrabCut; **c** Pyramid; **d** Naïve ; **e** SDI; **f** SDI*; **g** SMG; **h** CA-FCN; and **i** BBS

Fig. 8 Sample PSLs overlaid in red on a Total-Text ground truth sample using the following weakly-supervised segmentation methods: **a** WeSuperMaDD; **b** GrabCut; **c** Pyramid; **d** Naïve ; **e** SDI; **f** SDI*; **g** SMG; and **h** BBS

based on the size of the original input image. Our WeSuperMaDD obtained P, R, and F_1 scores of 71.58, 88.15, and 79.00, while the SGP had published results of P, R, and F_1 scores of 89.10, 70.74, and 78.87, respectively. SGP had a higher P, while WeSuperMaDD has a higher R. However, the overall F_1 scores are comparable. We postulate that this performance difference in P and R could be related to the training procedures of the CNNs used within each method. In SGP the training data for the segmentation CNN can have more background pixels than foreground pixels in their segmentation labels, therefore, the CNN can be biased towards predicting a pixel as background in the presence of uncertainty (Kotsiantis et al., 2005). The bias results in SGP predicting fewer pixels as foreground, focusing only on obvious true positives as reflected in the results, i.e., higher precision, lower recall. In contrast, WeSuperMaDD is designed to predict pixels as foreground rather than the background as the CNNs incorporated are trained for a text recognition task, where the CNNs must identify all characters in an image to recognize text contained within it. Therefore, the features extracted by the CNNs used in WeSuperMaDD must be from the full character sequence. Consequently, the combination of the *Mask Refinement* and *Mask Binary Search* modules would likely create PSLs that assign more pixels as foreground resulting in a higher recall than precision. This would help to ensure CNNs trained with WeSuperMaDD identify full character sequences. Overall, the average quality measured using the F_1 score of the PSLs generated by WeSuperMaDD and the SGP (Bonechi et al., 2019), are comparable as a result. Our weakly supervised method, however, has the clear advantage of not requiring any segmentation labels for training and can be thus directly applied to a wide variety of datasets without any human time-effort.

We compare the F_1 scores of the PSLs generated by WeSuperMaDD with the F_1 scores from rounds 1, 2, and 3 of SDI and SDI* on both the ICDAR-13 and Total-Text datasets, Table 1. The F_1 scores of both SDI and SDI* improve marginally with the number of training rounds. SDI has a lower F_1 score compared to both the Naïve method and our proposed WeSuperMaDD method. SDI also achieves

a higher R but lower P compared to our method. This is observed in the generated PSLs, as SDI is unable to create masks of individual characters and instead creates one continuous mask for the entire text, Figs. 7e and 8e. Thus, it greedily predicts both foreground and background pixels as foreground, trading P for R. In subsequent training rounds SDI uses the predicted CNN labels as supervision for the next round. Thus, the PSLs become increasingly biased towards entire text groupings as one continuous object, and cannot handle disjoint objects. SDI*, initialized using the first stage masks from WeSuperMaDD, provides an improvement with respect to SDI with higher P and F_1 score, but still greedily predicts many background pixels as foreground (Fig. 7f), and has difficulty segmenting disjointed letters (e.g., is not able to segment the upper hole in the letter 'R' in Fig. 8f). Overall, our proposed method achieves a higher F_1 score compared to both SDI and SDI* without requiring iterative training.

We compare the P, R, and F_1 scores of the PSLs generated by CA-FCN, Table 1. We only report results on the ICDAR-13 test set, as the Total-Text dataset does not provide per-character ground truth bounding boxes required by this method's label generation pre-processing step. CA-FCN has a higher F_1 score compared to the Naïve method, but a lower score when compared to the others. This is observed in the generated PSLs in Fig. 7h, as the reduction in mask size during the pre-processing step results in masks which only capture a small portion of each character.

Additionally, we compare the P, R, and F_1 scores of the PSLs generated by BBS, Table 1. On the ICDAR-13 dataset with straight text, BBS generates P, R, and F_1 scores which are identical to the Naïve method. This is due to the generated PSLs which are word-level masks being the same as the rectangular crops used for evaluation for straight text, Fig. 7i. However, on the Total-Text dataset with curved text, the F_1 score for BBS is better than the Naïve method, but still lower than the F_1 score for our WeSuperMaDD method. This is due to BBS considering background pixels surrounding the text as foreground, Fig. 8h.

We also compare the P, R, and F_1 scores of the PSLs generated by SMG, Table 1. SMG performs considerably worse compared to the other methods. In some cases, SMG finds only a few of the text characters from each text crop, Fig. 7g, as saliency maps can only capture the most discriminative parts of the object rather than the entire object (Simonyan et al., 2014). Moreover, after GrabCut, the largest connected component set of foreground pixels are chosen as a PSL. In particular, in Fig. 8g, SMG greedily segments the entire text groupings as one continuous object. Thus, SMG is unable to segment disjointed elements. In contrary, our method does not assume the contextual information is a single connected component which allows for disjointed elements to be found.

For the ICDAR-13 dataset, a non-parametric Kruskal–Wallis test, $n = 1095$, showed a statistically significant difference in F_1 between these weakly-supervised methods ($H = 1871$, $p < 0.001$). Posthoc Dunn tests were conducted between our WeSuperMaDD method and the other five methods and showed that our method has a statistically significant higher F_1 with respect to the SMG, ($Z = 42.69$, $p < 0.001$), SDI ($Z = 16.97$, $p < 0.001$), SDI* ($Z = 16.45$, $p < 0.001$), CA-FCN ($Z = 15.14$, $p < 0.001$), and BBS ($Z = 42.51$, $p < 0.001$) methods, respectively. For the Total-Text dataset, a non-parametric Kruskal–Wallis test, $n = 2{,}543$, also showed a statistically significant difference in F_1 between these weakly-supervised methods ($H = 3496$, $p < 0.001$). Posthoc Dunn tests conducted between our WeSuperMaDD method and the other four methods and showed that our method has a statistically significant higher F_1 with respect to the SMG, ($Z = 60.03$, $p < 0.001$), SDI ($Z = 19.72$, $p < 0.001$), SDI* ($Z = 18.81$, $p < 0.001$), and CA-FCN ($Z = 21.92$, $p < 0.001$) methods, respectively.

5.3 Context Detection and Segmentation Experiments

The objective of the Context Detection and Segmentation Experiments is to evaluate the full text detection and segmentation performance of our WeSuperMaDD architecture.

5.3.1 Training Datasets

We used the SynthText (Gupta et al., 2016) for pre-training due to its large size and data diversity. The following datasets were then used for fine tuning: (1) ICDAR-13 (Karatzas et al., 2013), (2) ICDAR-15 (Karatzas et al., 2015), and (3) ICDAR-17 multi-language text (Nayef et al., 2017). The ICDAR-17 dataset contains 9,000 training/validation images of text from nine languages and six different scripts.

5.3.2 Testing Datasets

We use: (1) ICDAR-13 (Karatzas et al., 2013), (2) ICDAR-15 (Karatzas et al., 2015), (3) ICDAR-17 multi-language text (Nayef et al., 2017) datasets, and (4) our own grocery dataset for performance evaluation. The grocery dataset contains 2226 images that were collected using our mobile interactive Blueberry robot navigating aisles in a real grocery store and focuses on the context detection task of determining real text on aisle signs in the environment. Both the detection and segmentation results in Table 2 are reported using a procedure similar to the ICDAR-15 evaluation procedure (Karatzas et al., 2015). In this case, a positive detection is defined as a predicted quadrilateral with IoU greater than 0.5

Table 2 Comparison of detection and segmentation methods on the test datasets.

Method	Dataset															
	ICDAR-13				ICDAR-15				ICDAR-17				Grocery			
	P	*R*	F_1	IoU	*P*	*R*	F_1	IoU	*P*	*R*	F_1	IoU	*P*	*R*	F_1	IoU
Detection results																
WCNN*	90.72	88.44	89.56	78.97	91.05	84.69	87.75	69.43	78.94	61.05	68.85	78.29	82.34	92.89	87.30	68.73
NCNN*	90.05	88.84	89.44	78.61	89.36	83.34	86.25	69.41	75.77	62.60	68.56	78.03	82.64	91.94	87.04	64.21
Segmentation results																
WCNN*	85.41	81.97	83.65	75.60	81.42	88.21	84.68	62.83	79.64	57.98	67.10	70.38	80.21	90.94	85.24	63.64
NCNN*	70.79	71.26	71.03	65.33	66.78	66.97	66.88	60.44	45.76	50.62	41.76	58.34	49.70	77.86	60.68	54.50
Semi-supervised detection results																
CRAFT† (Y. Baek et al., 2019b)	97.40	93.10	95.20	–	89.80	84.30	86.90	–	80.60	68.20	73.90	–	–	–	–	–
Mask TS** (Lyu, Liao, et al., 2018a)	95.00	88.60	91.70	–	91.60	81.00	86.00	–	–	–	–	–	–	–	–	–
One/Two-stage fully supervised detection results																
Lyu et al.** (Lyu, Yao, et al. 2018)	93.30	79.40	85.80	–	94.10	70.70	80.70	–	83.80	55.60	66.80	–	–	–	–	–
CRPN** (Linjie Deng et al., 2019b)	92.10	84.00	87.90	–	88.70	80.70	84.50	–	–	–	–	–	–	–	–	–
FOTS** (X. Liu et al., 2018)	–	–	88.30	–	91.00	85.17	87.99	–	80.95	57.51	67.25	–	–	–	–	–
Textboxes + + * (Liao et al., 2018a)	88.00	74.00	81.00	–	87.20	76.70	81.70	–	–	–	–	–	–	–	–	–
STELA* (L. Deng et al., 2019a)	93.30	85.10	89.00	–	88.70	78.60	84.50	–	78.70	65.50	71.50	–	–	–	–	–

All missing values above (denoted '–') were not reported
* Indicates a one-stage detector, ** indicates a multi-stage, † indicates a segmentation-based method

with a ground truth quadrilateral, with the additional restriction that a ground truth quadrilateral must be associated with at most one positive detection. In particular, to obtain the segmentation results, an additional refinement step, discussed in "Testing Procedure", is used to convert predicted segmentation masks to a quadrilateral. The P,R, F_1, and IoU scores are then computed between the predicted and ground truth quadrilaterals.

The datasets provide quadrilateral labels of the locations of text instances and text labels for text contained within the images. These datasets provide a variety of real text data in several environments e.g., malls, roads, etc., with varying fonts, scales, and languages, therefore being a useful representation of what a robot may encounter in human-centered environments.

5.3.3 Text Instance Segmentation Model

WeSuperMaDD uses a modified RetinaNet (Lin et al., 2017) architecture with a ResNeXt-50 (Xie et al., 2017) backbone with pyramid levels P_2 to P_6 for features were used for text detection and segmentation. The selection of a small CNN backbone and a one-stage detection framework reflects the common need of using CNNs for robotics context detection tasks, as small CNNs allow for shorter training time, and faster inference. We modify the RetinaNet architecture to include additional horizontal prior boxes to improve multi-line text detection performance (Liao et al., 2018a). The bounding quadrilaterals are formed through the prediction of the parameters of a homography matrix that transforms the prior box to a predicted text instance location. The text mask detection branch was shared across all feature maps (He et al., 2017) and uses the rotated RoI-Align module to retrieve features from the backbone (Huang et al., 2018).

5.3.4 Methods for Comparison

We compare a text instance segmentation CNN trained with PSLs generated by WeSuperMaDD, referred to as WCNN herein, against a text instance segmentation CNN trained with PSLs generated by the aforementioned Naïve approach, referred to as NCNN. The Naïve approach was selected as the weakly supervised alternative as it is the PSL method designed for text (Wu et al., 2020) with the highest F_1 score after WeSuperMaDD in Table 1. Both WCNN and NCNN are based on the aforementioned *Text Instance Segmentation Model.*

5.3.5 State-of-the-Art-Methods

Furthermore, we also compare our detection performance against the results reported in the literature for the following methods: (1) Lyu et al. (Lyu, Yao, et al. 2018b), (2) CRPN (Linjie Deng et al., 2019b), (3) FOTS (Liu et al., 2018), (4) Textboxes ++ (Liao et al., 2018a), (5) STELA (L. Deng et al., 2019a), (6) Mask TS (Lyu, Liao, et al., 2018a), and (7) CRAFT (Y. Baek et al., 2019b). Methods 1–5 are typical text detection CNNs that have been trained using *fully* supervised data, while methods 6 and 7 are *semi-supervised* text detection methods. Furthermore, methods 1–3 and 6 are two-stage methods, and methods 4 and 5 are one-stage methods. CRAFT is a text region segmentation-based method. One-stage methods directly predict bounding quadrilaterals from an image. Two-stage methods extend the one-stage formulation, by including a sub-network that refines the first stage's text proposals. The segmentation method examines a text segmentation output to generate a text bounding quadrilateral. We have selected these methods to investigate the detection performance of WCNN with respect to existing text detection techniques, while uniquely being the only such CNN to provide character level segmentation.

5.3.6 Training Procedure

Both WCNN and NCNN were trained to output a set of bounding quadrilaterals representing the locations of text within an input image, and a mask highlighting the location of text within this quadrilateral. We used a weighted three-part loss, L, containing a bounding box regression loss, a classification loss, and a mask loss, L_m:

$$L = \left(\mathrm{CE}\left(z_g,\, z_p\right) + \alpha_H L_H\left(\widehat{h}_g,\, \widehat{h}_p\right)\right)/n_{\mathrm{pos}} + L_m, \tag{19}$$

where n_{pos} is the number of matched prior boxes, and $\mathrm{CE}\left(z_g,\, z_p\right)$ measures the classification loss of the predicted confidence logits z_p, and the matching label z_g. We rank the prior boxes using the Huber loss, L_H, between each prior box and the ground truth quadrilaterals and select the top-k, $k = 20$ boxes with lowest loss as positive matches in the label tensor z_g. This selection method balances the number of matched prior boxes per ground truth quadrilateral. Out of the remaining prior boxes we select the top-$3n_{\mathrm{pos}}$ with the highest classification loss as the negative examples within z_g(Fu et al., 2019) to ensure that negative samples are not overrepresented in the loss. Prior boxes matched with ground truth marked as "illegible" within the dataset are not included in the loss calculation.

The α_H hyper-parameter is used to reduce the influence of $L_H\left(\widehat{h}_g,\, \widehat{h}_p\right)$ on the overall loss (Ren et al., 2016). The network is trained to predict the homography that transforms a prior box to a ground truth target. The ground truth homography, $\widehat{h}_g$, is the homography of a unit square prior box, u, centered at (0,0) to a normalized and centered ground truth quadrilateral, $\widehat{g}_g$, to maintain position invariance, $\widehat{h}_g = H\left(u,\, \widehat{g}_g\right)$. Here $H(\cdot,\, \cdot)$ represents the solution to the set of linear equations describing the homography

between two quadrilaterals. To attain $\widehat{g}_g$, we normalize each of the coordinates of g_q as:

$$\hat{g}_g = \left(\frac{g_{q_{x_i}} - b_{c_x}}{b_w \sigma^2_{b_w}}, \frac{g_{q_{y_i}} - b_{c_y}}{b_h \sigma^2_{b_h}} \right), \forall i \in [1, 4], \tag{20}$$

using the center (b_{cx}, b_{cy}), side lengths (b_w, b_h), and length variances $\left(\sigma^2_{b_w}, \sigma^2_{b_h}\right)$ of the matched prior boxes.

Instance segmentation is trained using the mask loss (He et al., 2017):

$$L_m = \frac{1}{k_s} \sum_{k=1}^{k_s} CE\left(M_g^{(k)}, M_p^{(k)}\right) / |M_g|, \tag{21}$$

where $\mathrm{CE}\left(M_g^{(k)}, M_p^{(k)}\right)$ is a cross entropy loss between each predicted mask pixel and the PSLs. The loss function is applied to the masks generated from the top-k_s, $k_s = 50$ bounding quadrilateral predictions (Fu et al., 2019).

Datasets with similar image distributions are grouped for fine-tuning and testing similar to (Baek et al., 2019b). When testing on ICDAR-13, and ICDAR-17 we fine-tune using their training sets. When testing on the ICDAR-15, and our own real grocery store dataset we fine tune using the ICDAR-15 training set. We apply random color, perspective, rescaling, and cropping perturbations during training and PSL generation. Before training either WCNN or NCNN we generate PSLs by applying WeSuperMaDD or the Naïve approach for each image in the dataset.

5.3.7 Testing Procedure

During testing, we resize the longer edges of images from ICDAR-13, 15, 17, to 960, 1,920, and 1,600, respectively. For our own grocery dataset, the image size of 1280 × 720 pixels is used to compare both the WCNN and NCNN to measure their detection and segmentation performance in a real robot context detection task.

As the ICDAR-15, 17 and our grocery datasets do not provide instance segmentation labels, we compared the ability to localize text with the masks generated by WCNN and NCNN using a Proposal Refinement Procedure (PRP) that we have developed. In particular, we generate text detection quadrilateral predictions from segmentation masks using the minimum rotated rectangle surrounding the mask. The procedure converts a quadrilateral text region proposal into a rotated rectangle and uses the text mask detection branch to segment the text contained within it. We iteratively move each edge of the region proposal outwards, i.e., increasing the distance from the proposal centroid to the edge center by 1%, until at least n of the pixels on that edge are foreground. After a specified number of iterations, we generate the final proposal using the minimum rotated rectangle (minAreaRect function from OpenCV) surrounding the segmentation mask. The procedure tests whether the bounding rotated rectangle proposal can be recreated using the predicted mask. Given a predicted quadrilateral, if the quadrilateral obtained from the procedure is larger than the corresponding ground truth quadrilateral, then background information was incorporated into the mask. If the quadrilateral is smaller, a part of the text instance was not detected. This process examines the ability of the CNN to (1) find the characters within a region, (2) to differentiate individual text instances, and (3) to detect the ability of the CNN to localize text. An example of this process applied to both methods is shown in Fig. 9, where initial masks, Fig. 9a and c, are refined using the described procedure to grow the initial mask proposals as shown in Fig. 9b and d.

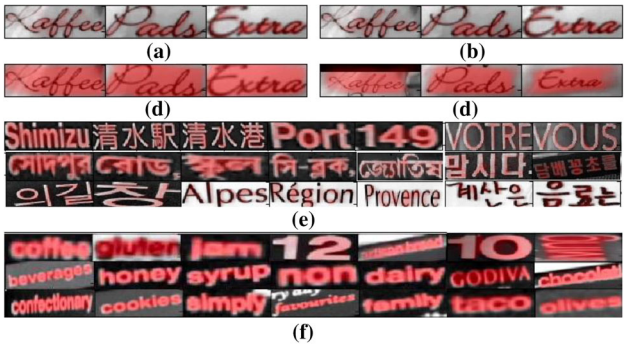

Fig. 9 Sample crops from ICDAR-13 dataset overlaid with: **a** masks generated from WCNN; **b** masks generated from NCNN; **c** masks from **a** after refinement; **d** masks generated from **b** after refinement; **e** randomly selected segmentation samples generated by WCNN overlaid on ICDAR-17 validation images; and **f** varying sample crops from our grocery dataset overlaid with WCNN masks

5.3.8 Text Detection Results

The text detection results for WCNN and NCNN are presented in Table 2. For each of the ICDAR-13, 15, 17, and grocery datasets, we observe an F_1 score ranging from 68.85 to 89.56 for WCNN and an F_1 score ranging from 68.56 to 89.44 for NCNN, respectively. The results show WCNN has only a slight improvement in F_1 scores between 0.12 and 1.50 higher than those achieved by NCNN for each of the datasets. The results also show that WCNN has an improvement in IoU scores between 0.02 and 4.52 over NCNN for each of the datasets. The performance improvements in F_1 and IoU are likely due to WCNN learning more robust features for instance segmentation due to the higher quality WeSuperMaDD masks used (compared to the Naïve masks used by NCNN), despite using the same ground truth detection labels to train both CNNs. Since the WeSuperMaDD PSLs provide more accurate localization information for text, they have

better text segmentation output quality for localization than NCNN, as is explored in the following section.

5.3.9 Text Segmentation Results

We also compare the segmentation performance of both WCNN and NCNN based on their final F_1 scores after applying the PRP ("Testing Procedure"). The results are summarized in Table 2. Since only WCNN and NCNN provide a text instance segmentation output, we, therefore, cannot compare against the existing state-of-the-art techniques using this procedure. For each of the ICDAR-13, 15, 17, and grocery datasets we observed an F_1 score ranging from 67.10 to 85.24 for WCNN and an F_1 score ranging from 41.76 to 71.03 for NCNN, respectively. We note that WeSuperMaDD PSLs provide a gain of between 42 and 83% in terms of F_1 when compared to the Naïve PSLs. We also observed an IoU score ranging from 62.83 to 75.60 for WCNN, and an IoU score ranging from 54.50 to 65.33 for NCNN, for these four datasets, respectively. The lower F_1 and IoU scores obtained by NCNN can be attributed to the label quality of the Naïve PSLs. Namely, since the Naïve method mislabels background pixels near the edges of ground truth bounding quadrilaterals, the edges of text segmentation masks will also be inaccurate. In contrast, WeSuperMaDD PSLs are designed to only label character font as foreground, reducing the ambiguity of the location of the edge of a text instance thereby improving the F_1 score.

We further conducted a qualitative comparison of WCNN and NCNN. Figure 9a and b show a segmentation proposal from WCNN before and after mask refinement. We note that there is minimal change to the bounding quadrilateral proposals since the initial segmentations were within the boundary of the proposals. This contrasts to the masks generated by the NCNN which incorporates background pixels surrounding text as seen in Fig. 9c and results in the refinement procedure increasing the size of the bounding quadrilateral in Fig. 9d to the point that they no longer satisfy the IoU criteria for a positive detection. Therefore, WCNN is able to produce instance segmentation masks that are more informative of the location and size of text than those generated by NCNN. WeSuperMaDD is able to generate PSLs that only segment characters within the text bounding boxes, providing higher precision guidance for WCNN to localize text boundaries.

We provide several examples of text instance segmentation masks generated by WCNN on the ICDAR-17 validation set within Fig. 9e and on our grocery dataset in Fig. 9f for further qualitative analysis. It can be seen that WCNN provides segmentation proposals (red) for multiple scripts and languages that overlap the text in the source image while incorporating minimal background pixels. It is interesting to note that this is despite using CNNs that have only been trained on English text and Latin script to generate the PSLs. The ability of WCNN to detect and segment important features in text in varying languages and scripts highlights the generalizability of the WeSuperMaDD pseudo label generation process.

5.3.10 Comparison to State-of-the-Art Detection Techniques

We compared the performance of our one-stage WCNN to seven other approaches presented in the literature with the ICDAR-13, 15, and 17 datasets, Table 2. Our WCNN outperforms all existing one-stage detectors on the ICDAR-13 and 15 datasets with an F_1 score increase of 0.56 and 3.25. However, on the ICDAR-17 dataset, it had an F_1 score 2.85 which is lower than STELA. We postulate that the reason that WCNN outperformed the other one-stage detectors on the ICDAR-13 and 15 datasets was due to our unique use of the text instance segmentation training. This segmentation training helps the network distinguish text regions from background regions due to the inherent per-pixel detail of PSLs. In contrast, the other one-stage methods only use bounding quadrilaterals during training which only provide broad guidance of what image regions contain text. We hypothesize that the reason why the STELA had better performance on the ICDAR-17 dataset was due to it directly learning the shape and location of the prior boxes used during the detection task via regression which can improve the number of prior boxes that satisfy the ground truth bounding quadrilateral matching criteria. This is important for large sized bounding quadrilaterals since they typically only match with one prior box and are thereby underrepresented in training resulting in lower detection performance. This was evident for the ICDAR-17 dataset since it contains several text instances that are very large in area.

When compared to the remaining two-stage and segmentation methods on the ICDAR-13, 15, and 17 datasets, WCNN had slightly lower F_1 scores of 5.64, 0.24, and 5.05 on each dataset, respectively. This is not unexpected given that two-stage models are known to outperform one-stage models (Fu et al., 2019), especially in the case of semi-supervised methods that are trained using a fully labeled subset of data. However, the clear advantage of our method with respect to all these two-stage methods is that WCNN can provide a segmentation output without the need for labeled data and with more accurate localization information, as it provides per-pixel segmentation, while other methods rely purely on using bounding quadrilaterals for text localization. Semi-supervised methods cannot be generalized to all datasets due to the need for a fully labeled subset of the dataset. Overall, it can be seen that WCNN has comparable performance to existing detection models since it outperforms all

of the alternative methods on ***at least*** one of the three datasets tested.

6 Conclusion and Future Work

In this paper, we present the novel WeSuperMaDD method for the weakly supervised generation of PSLs of contextual information datasets collected in various environments. The novelty of the method is that it can autonomously generate PSLs using pre-existing CNNs not specifically trained for the segmentation task. Our method uses learned image features to directly generate these labels for small and non-diverse datasets typically present in robotic environments including grocery stores, malls, roads, etc. A new mask refinement system is introduced to find the PSL with fewest foreground pixels that satisfies constraints as measured by a cost function. The mask refinement system removes the need for handcrafted heuristic rules typically needed by existing PSL generation methods. Experiments validating the performance of WeSuperMaDD were conducted using OCR datasets due to the wide applicability of detecting scene text for robotic applications such as navigating unknown environments, annotating maps, etc. The datasets contained images of text with varying languages, scripts, curves, and scales obtained from various environments including parks, border crossings, and shopping malls. The experiments showed that PSLs generated by WeSuperMaDD had: (1) significantly higher F_1 scores for segmentation when compared to other weakly supervised (GrabCut, Pyramid, Naïve , SDI, SMG), semi-supervised (BBS), and fully supervised (CA-FCN) methods, and (2) comparable F_1 scores to the current state-of-the-art semi-supervised PSL generation method (SGP). We further validated our overall architecture for instance segmentation and detection of real text in varying indoor and outdoor environments. We found our CNN trained with WeSuperMaDD PSLs has a higher segmentation F_1 scores than a context segmentation CNN trained with Naïve PSLs and higher detection F_1 scores than existing state-of-the-art one-stage models.

To further increase the applicability of our method to a variety of robotics tasks, future work will extend our approach for datasets with only bounding boxes or class labels to reduce the burden of manual labeling. Future work will also include extending our experimental analysis to generic object segmentation such as on the MS COCO dataset through the selection of an ensemble of object classification models such as EfficientNets, the extension of VBP to support these additional specialized layers, and the design of an appropriate object segmentation cost function.

Funding This work was supported by AGE-WELL Inc., the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chairs program (CRC), the Vector Institute Scholarship in Artificial Intelligence, the NVIDIA GPU grant, and Longo Brothers Fruit Markets Inc.

Data Availability The ICDAR-13, 15, and 17 datasets are available at: ICDAR-13 ("Overview—Focused Scene Text—Robust Reading Competition" n.d.) .ICDAR-15 ("Overview—Incidental Scene Text—Robust Reading Competition" n.d.). ICDAR-17 ("Overview—ICDAR2017 Competition on Multi-lingual scene text detection and script identification—Robust Reading Competition" 2017). Total-Text ("Total-Text-Dataset (Official site)", 2017).

Declarations

Conflict of interest We have no known conflicts of interest.

References

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5297–5307).

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., et al. (2019a). What is wrong with scene text recognition model comparisons? Dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4715–4723).

Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019b). Character Region Awareness for Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9365–9374).

Barnes, D., Maddern, W., & Posner, I. (2017). Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. In 2017 *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 203–210).

Bellocchio, E., Ciarfuglia, T. A., Costante, G., & Valigi, P. (2019). Weakly supervised fruit counting for yield estimation using spatial consistency. *IEEE Robotics and Automation Letters, 4*(3), 2348–2355.

Benenson, R., Popov, S., & Ferrari, V. (2019). Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11700–11709).

Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U., & Zieba, K. (2018). VisualBackProp: efficient visualization of CNNs. In *2018 IEEE International Conference on Robotics and Automation* (ICRA) (pp. 4701–4708).

Bonechi, S., Andreini, P., Bianchini, M., & Scarselli, F. (2019). COCO_TS Dataset: Pixel–level annotations based on weak supervision for scene text segmentation. In *International Conference on Artificial Neural Networks and Machine Learning* (pp. 238–250). Cham: Springer.

Case, C., Suresh, B., Coates, A.,& Ng, A. Y., (2011). Autonomous sign reading for semantic mapping. In *2011 IEEE international Conference on Robotics and Automation* (pp. 3297–3303).

Chapelle, O., Schlkopf, B., & Zien, A. (2010). *Semi-supervised learning* (1st ed.). The MIT Press.

Ch'ng, C. K., & Chan, C. S. (2017). Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)* (pp. 935–942).

Cleveland, J., Thakur, D., Dames, P., Phillips, C., Kientz, T., Daniilidis, K., et al. (2017). Automated system for semantic object labeling with soft-object recognition and dynamic programming segmentation. *IEEE Transactions on Automation Science and Engineering, 14*(2), 820–833.

Deng, L., Gong, Y., Lin, Y., Shuai, J., Tu, X., Zhang, Y., et al. (2019b). Detecting multi-oriented text with corner-based region proposals. *Neurocomputing, 334*, 134–142.

Deng, L., Gong, Y., Lu, X., Lin, Y., Ma, Z., & Xie, M. (2019a). STELA: A real-time scene text detector with learned anchor. *IEEE Access, 7*, 153400–153407.

Dworakowski, D., Thompson, C., Pham-Hung, M., & Nejat, G. (2021). A robot architecture using contextSLAM to find products in unknown crowded retail environments. *Robotics, 10*(4), 110.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision, 111*(1), 98–136.

Fu, C.-Y., Shvets, M., & Berg, A. C. (2019). RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arxiv*.

Gregorio, D. D., Tonioni, A., Palli, G., & Stefano, L. D. (2020). Semiautomatic labeling for deep learning in robotics. *IEEE Transactions on Automation Science and Engineering, 17*(2), 611–620.

Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2315–2324).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern RecognitioN* (pp. 770–778).

Hou, Q., Massiceti, D., Dokania, P. K., Wei, Y., Cheng, M.-M., & Torr, P. H. (2017). Bottom-up top-down cues for weakly-supervised semantic segmentation. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (pp. 263–277). Springer.

Huang, J., Sivakumar, V., Mnatsakanyan, M., & Pang, G. (2018). Improving rotated text detection with rotation region proposal networks. *arxiv.*

Ibrahim, M. S., Vahdat, A., & Macready, W. G. (2018). Weakly supervised semantic image segmentation with self-correcting networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12715–12725).

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Synthetic data and artificial neural networks for natural scene text recognition. *arxiv*.

Jain, S. D., & Grauman, K. (2013). Predicting sufficient annotation strength for interactive foreground segmentation. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1313–1320).

Jing, L., Chen, Y., & Tian, Y. (2020). Coarse-to-fine semantic segmentation from image-level labels. *IEEE Transactions on Image Processing, 29*, 225–236.

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., et al. (2015). ICDAR 2015 competition on robust reading. In *13th International Conference on Document Analysis and Recognition* (pp. 1156–1160).

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L. G. i, Mestre, S. R., et al. (2013). ICDAR 2013 robust reading competition. In *12th International Conference on Document Analysis and Recognition* (pp. 1484–1493).

Khoreva, A., Benenson, R., Hosang, J., Hein, M., & Schiele, B. (2017). Simple does it: weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 876–885).

Kolesnikov, A., & Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV* (pp. 695–711).

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2005). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering, 30*, 25–36.

Li, G., Xie, Y., & Lin, L. (2018). Weakly supervised salient object detection using image labels. In *AAAI Conf. on Artificial Intelligence* (pp. 7024–7031).

Li, Y., Wang, T., Kang, B., Tang, S., Wang, C., Li, J., & Feng, J. (2020). Overcoming Classifier Imbalance for Long-Tail Object Detection With Balanced Group Softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10991–11000).

Liang, H., Sanket, N. J., Fermüller, C., & Aloimonos, Y. (2019). SalientDSO: Bringing attention to direct sparse odometry. *IEEE Transactions on Automation Science and Engineering, 16*(4), 1619–1626.

Liao, M., Shi, B., & Bai, X. (2018a). Textboxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Transactions on Image Processing, 27*(8), 3676–3690.

Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., et al. (2018b). Scene text recognition from two-dimensional perspective, *arXiv*.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., et al. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2980–2988).

Liu, J., Liu, X., Sheng, J., Liang, D., Li, X., & Liu, Q. (2019). Pyramid mask text detector.

Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., & Yan, J. (2018). FOTS fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5676–5685).

Lyu, P., Liao, M., Yao, C., Wu, W., & Bai, X. (2018a). Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lyu, P., Yao, C., Wu, W., Yan, S., & Bai, X. (2018b). Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7553–7563)

Mahendran, A., & Vedaldi, A. (2016). Salient deconvolutional networks. *Computer vision—ECCV 2016* (pp. 120–135). Springer.

Mishra, A., Alahari, K., & Jawahar, C. V. (2012). Scene text recognition using higher order language priors. In *British Machine Vision Conference* (p. 127.1–127.11).

Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., et al. (2017). ICDAR2017 Robust reading challenge on multi-lingual scene text detection and script identification—RRC-MLT. In *2017 14th IAPR International Conference on Document Analysis and Recognition* (pp. 1454–1459).

Niu, S., Lin, H., Niu, T., Li, B., & Wang, X. (2019). DefectGAN: Weakly-supervised defect detection using generative adversarial network. In *IEEE International Conference on Automation Science and Engineering* (pp. 127–132).

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

Overview—Focused Scene Text - Robust Reading Competition. (n.d.). *Robust Reading Competition*. https://rrc.cvc.uab.es/?ch=2. Accessed 20 November 2020

Overview—ICDAR2017 Competition on Multi-lingual scene text detection and script identification - Robust Reading Competition. (2017, January 4). *Robust Reading Competition*. https://rrc.cvc.uab.es/?ch=8. Accessed 20 November 2020

Overview—Incidental scene text - robust reading competition. (n.d.). *Robust Reading Competition*. https://rrc.cvc.uab.es/?ch=4. Accessed 20 November 2020

Peng, Z., Gao, S., Xiao, B., Guo, S., & Yang, Y. (2018). CrowdGIS: Updating digital maps via mobile crowdsensing. *IEEE Transactions on Automation Science and Engineering, 15*(1), 369–380.

Pont-Tuset, J., Arbeláez, P., Barron, J. T., Marques, F., & Malik, J. (2017). Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*, 128–140.

Radosavovic, I., Dollár, P., Girshick, R., Gkioxari, G., & He, K. (2018). Data distillation: Towards omni-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4119–4128).

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149.

Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics, 23*(3), 309–314.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arxiv*.

Saleh, F. S., Aliakbarian, M. S., Salzmann, M., Petersson, L., Alvarez, J. M., & Gould, S. (2018). Incorporating network built-in priors in weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*, 1382–1396.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision, 128*(2), 336–359.

Shariati, A., Holz, C., & Sinha, S. (2020). Towards privacy-preserving ego-motion estimation using an extremely low-resolution camera. *IEEE Robotics and Automation Letters, 5*(2), 1222–1229.

Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(11), 2298–2304.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arxiv*.

Singh, A., Yang, L., & Levine, S. (2017). GPLAC: Generalizing vision-based robotic skills using weakly labeled images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5851–5860).

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., et al. (2019). Scalability in perception for autonomous driving: waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2446–2454).

Thompson, C., Khan, H., Dworakowski, D., Harrigan, K., & Nejat, G. (2018). An autonomous shopping assistance robot for grocery stores. In *IEEE/RSJ Proceedings of the Workshop on Robotic Coworkers 4.0*.

Vardazaryan, A., Mutter, D., Marescaux, J., & Padoy, N., et al. (2018). Weakly-supervised learning for tool localization in laparoscopic videos. In D. Stoyanov, Z. Taylor, S. Balocco, R. Sznitman, A. Martel, & L. Maier-Hein (Eds.), *Intravascular imaging and computer assisted stenting and large-scale annotation of biomedical data and expert label synthesis* (pp. 169–179). Springer.

Vilar, E., Rebelo, F., & Noriega, P. (2014). Indoor human wayfinding performance using vertical and horizontal signage in virtual reality. *Human Factors and Ergonomics in Manufacturing & Service Industries, 24*(6), 601–615.

Wan, F., Liu, C., Ke, W., Ji, X., Jiao, J., & Ye, Q. (2019). C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*

Wan, F., Wei, P., Jiao, J., Han, Z., & Ye, Q. (2018). Min-entropy latent model for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, B. H., Chao, W., Wang, Y., Hariharan, B., Weinberger, K. Q., & Campbell, M. (2019). LDLS: 3-D object segmentation through label diffusion from 2-D images. *IEEE Robotics and Automation Letters, 4*(3), 2902–2909.

Wang, C., Zhao, S., Zhu, L., Luo, K., Guo, Y., Wang, J., & Liu, S. (2021). Semi-supervised pixel-level scene text segmentation by mutually guided network. *IEEE Transactions on Image Processing, 30*, 8212–8221.

Wang, H., Finn, C., Paull, L., Kaess, M., Rosenholtz, R., Teller, S., & Leonard, J. (2015). Bridging text spotting and SLAM with junction features. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3701–3708).

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., & Ruan, X. (2017). Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 136–145).

Wei, Y., Feng, J., Liang, X., Cheng, M.-M., Zhao, Y., & Yan, S. (2017). Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1568–1576).

Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., & Huang, T. S. (2018). Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7268–7277).

What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. (2020). https://github.com/clovaai/deep-text-recognition-benchmark. Accessed 6 June 2020

Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., & Hutter, M. (2019). Where should i walk? Predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters, 4*(2), 1509–1516.

Wu, W., Xie, E., Zhang, R., Wang, W., Pang, G., Li, Z., et al. (2020). SelfText beyond polygon: Unconstrained text detection with box supervision and dynamic self-training, *arXiv*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., test, & tst. (2017). Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1492–1500).

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., & Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3712–3722).

Zhang, B., Xiao, J., Wei, Y., Sun, M., & Huang, K. (2019). Reliability does matter: An End-to-end weakly supervised semantic segmentation approach. *arxiv*.

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., & Sclaroff, S. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision, 126*(10), 1084–1102.

Zhao, X., Liang, S., & Wei, Y. (2018). Pseudo mask augmented object detection. In *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition* (pp. 4061–4070).

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2019). Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision, 127*(3), 302–321.

Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., & Jiao, J. (2018). Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3791–3800).

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44–53.