Neural Computing and Applications (2021) 33:4965–4978
https://doi.org/10.1007/s00521-020-05257-z

ORIGINAL ARTICLE

Spatial division networks for weakly supervised detection

Yongsheng Liu[1] · Wenyu Chen[1] · Hong Qu[1] · S. M. Hasan Mahmud[1] · Kebin Miao[2]

Received: 17 January 2020 / Accepted: 27 July 2020 / Published online: 20 August 2020

Abstract
With only global image-level annotations, weakly supervised learning of deep convolutional neural networks has shown enough capacity in classification and localization but lack of ability to present the detection explicitly. In this work, we propose a novel spatial division network, which is applied to detect bounding boxes only with weak supervision. The essence of our model is two innovative differentiable modules, determination network and parameterized division, which perform the spatial division in feature maps of classification networks. After training, the learned parameters of the spatial division would correspond to a set of predicted bounding box coordinates. To demonstrate the effectiveness of our model for multi-label classification and weakly supervised detection, we conduct extensive experiments on the multi-MNIST dataset. Experimental results show our spatial division networks can (1) help improve the accuracy of multi-label classification, (2) implement in an end-to-end way only with the image-level annotations, and (3) output accurate bounding box coordinate, thereby achieving multi-digits detection.

Keywords Deep learning · Learning systems · Convolutional neural networks · Predictive models

1 Introduction

Visual detection with deep convolutional neural networks (DCNNs) has made significant progress in the last decade [19, 32, 33]. These successes are not only due to the efficacious spatial feature extraction capability of DCNNs but the increasing number of large annotated image datasets. Adequate annotation (ground truth bounding boxes) for training is necessary for fully supervised methods to get state-of-art detection results. However, annotating these full supervision labels [42, 43] is labor intensive and time-consuming, motivating us to explore the weakly supervised detection (WSD) method with DCNNs. Compared to the fully supervised method, weakly supervised detection only acquires images with image-level annotations indicating whether an object of a specified category is present in an image or not [22]. Besides, WSD is like the visual system of humans, which first selecting locations of related regions in the "detection" stage and then determining the target in the "identification" stage [20].

Although this learning framework serves to be more economical and interpret, the outcome tends to be somewhat backward compared to fully supervised learning. The fundamental challenge of weakly supervised detection is that the predict bounding boxes have no corresponding ground truth at the same supervision level. Current approaches usually adopt two ideas to address this issue: (1) Instead of predicting bounding boxes, the model selects the highest-scoring candidate from the region proposals as the detection result [5, 38, 39]. (2) The model iteratively generates pseudo-ground truth bounding boxes and learns until it reaches a particular convergence condition [2, 16, 29]. The former could be trained end-to-end but requires region proposals prior. The latter does not require region proposals prior but cannot be trained end-to-end. Therefore, existing models in both cases cannot perform weakly supervised detection in an end-to-end learning manner without region proposals prior.

In this paper, we propose a spatial division network, termed SDN, to perform weakly supervised detection in an end-to-end learning manner. Our proposed WSD framework starts from the feature extraction part of a

✉ Hong Qu
hongqu@uestc.edu.cn

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

[2] China Coal Research Institute, Beijing, China

conventional CNNs and is extended by two differentiable modules to reason about the location and size of the interest object only via the global image-level label without any region proposals prior.

To overcome the problem that there is no corresponding ground truth in the prediction bounding box, we propose a spatial division representation, as shown in Fig. 1. The spatial division separates the entire two-dimensional space of the image into two parts: the space containing the object and the background space. A matrix can represent it. A zero in the matrix indicates the background, and a nonzero value indicates the part that contains the object. In this way, the bounding box regression in full supervision is transformed into mutual constraint learning between different spatial divisions.

Given an input image, our SDN first extracts mid-level image features by convolution and pooling operations. Next, these features are copied and branched into two streams. The first stream handles significant intra-class variations through a multiple instance learning framework. The endpoint of this stream is the category-dependent activation maps (CAMs), which denote confidence score maps to discriminate image regions for classification. Simultaneously, the second stream flows into a differentiable module named determination network, which designed to estimate and output the bounding box parameter. After that, we introduce another differentiable module named parameterized division, which can transform the bounding box parameter into Shadow Activation Maps (SAMs). In Fig. 1, as the name suggests, SAMs are like the

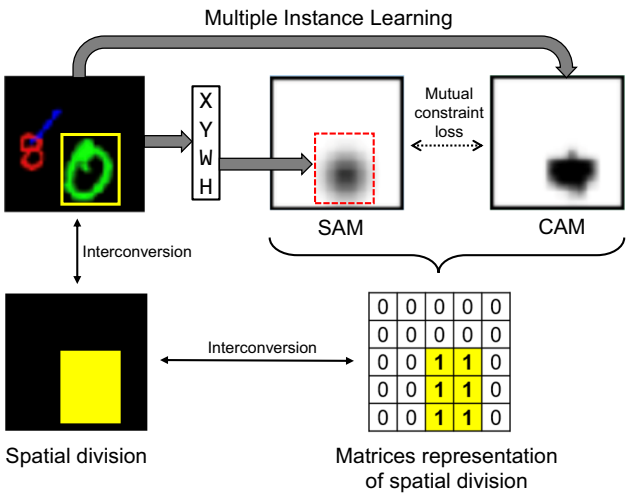

Fig. 1 Bounding box and spatial division and its matrix representation can be converted to each other. Both SAM and CAM meet the definition of spatial division. In this way, our SDN generates these two spatial divisions in two different ways. Through mutual constraint learning between them, the hidden bounding box parameters can be obtained in the middle process. These parameters are used as the final prediction of our model to achieve weakly supervised detection

"shadow" of CAMs, that is, the values at each identical position of them tend to be consistent.

During the training phase, in addition to employing cross-entropy loss for multi-label recognition, we further introduce mutual constraint loss to measure the similarity of SAMs and CAMs. There are two goals for our model training: (1) The classifier placed follow CAMs could achieve excellent results; (2) CAMs and SAMs are as similar as possible.

SDN is first proposed in our ISKE paper [18] and both theory and experiment part is promoted in this full version paper. The main contributions of this paper are summarized as follows:

1. We propose spatial division networks, a new learning framework for weakly supervised detection, which is trained in an end-to-end pipeline and does not rely on the candidate bounding box proposal.
2. We design two differentiable modules (determination network and parameterized spatial division) that can learn and generate bounding box solely through image-level annotations by mutual constraint learning of CAMs and SAMs.
3. Based on the MNIST dataset, we created a new dataset multi-MNIST for multi-label classification and weakly supervised detection.
4. We present a detailed experimental evaluation using multi-MNIST datasets. Our proposed method achieves superior performance over previous competing approaches, which highlights the importance of mutual constraint learning for weakly supervised detection model.

The rest of this paper is organized as follows: Section 2 discusses some work related to our own. In Sect. 3, we introduce our weakly supervised network. Sections 4 and 5 present the experimental results and comparisons with other methods. Finally, this paper concludes in Sect. 6.

2 Related work

In this section, we review the prior work related to the paper, covering the CNNs based on Multiple instance learning, WSDDN and its variants, as well as localization by iteratively learning.

2.1 CNNs based on Multiple Instance Learning (MIL)

Several recent works investigate weakly supervised computer vision tasks by practicing multiple instance learning (MIL) framework. MIL provides a set of bags, and each unit bag includes a collection of instances. For a given

category, the positive bag contains at least one positive instance (target object is present in the image), and the negative bag does not hold any positive instance (there are only noisy backgrounds relative to the target class in the image). In this way, although there are only image-level labels, the fundamental elements of learning are instances (local image region) rather than the entire image so that the learning target can be expanded into more complex tasks. It is straightforward to explain weakly supervised applications to computer vision as a MIL problem. Following this paradigm, CNNs can first extract a set of candidate object regions (including associated target objects or irrelevant backgrounds) from each image. After that, some output modules (such as classifier, object detectors) would be designed to be trained just by the image-level label.

Pioneering work in this direction is the global max pooing MIL strategy proposed in [22]. The authors show that train a CNN network using the global max pooing MIL strategy can localize objects of different sizes and aspect ratios. In [40], Zhou et al. adopt a similar global average pooling layer proposed in [17]. The above two pooling techniques avoid the lost when fully connected layers are used for classification and create a remarkable localizable deep representation. The solution named WELDON proposed in [8] extend the max pooling from the top-1 to the top-k and combine contrary evidence in the prediction layer in the same way as for top instances. Similar to this pooling improvement, there are LSE [30] and GRP [9], all of which seeks a smooth combination of global average pooling and global max pooing, allowing the model to more evenly find the spatial characteristics of the object without undersampling or oversampling. In [7], the authors added a multi-map transfer based on WELDON and proposed a fully convolutional network WILDCAT which jointly aims at getting adequately localized features and fixing object regions for learning spatial coherence.

It is worth noting that most existing approaches use CAMs from a global perspective without any constraint apart from image-level supervision, which makes the learning process prone to stuck in a local minimum. In our work, we aim to keep spatial invariance for aligning regions of interest by constraint CAMs by adaptive learning parameters.

2.2 End-to-end weakly supervised detection models

Using CNNs to build a weakly supervised detection end-to-end model is mainly based on WSDDN, and almost all other improvements are based on it.

In WSDDN [5], the authors proposed a weakly supervised deep detection network which consists of two data streams perform region recognition and object detection, respectively. WSDDN is the first to address the problem of WSOD only with end-to-end CNNs. After that, lots of its variants ([6, 13, 14, 38, 39]) are continually developing. For example, Kantorov et al. [14] also use the two-stream network which branches a localization stream in parallel with a classification stream. This improved work proposes context-aware guidance models leverage their surrounding context regions to improve localization. The solution proposed in [6] extend the WSDDN by a new architecture of cascaded networks, which include two multi-stage cascaded networks with different loss functions. In [13], the author introduces the attention mechanism into the WSDDN to further better identify objects of interest from cluttered backgrounds. Zhang et al. [39] adopt the online instance classifier refinement (OICR) method [31] to refine the WSDNN. Zhang et al. [38] propose a zigzag learning strategy based on WSDDN to prevent the model from overfitting initial seeds. Kosugi et al. [15] improve the WSDDN from two instance labeling methods. Besides, WSDDN becomes a base MIL detector network for some more complex WSD models. For example, Zeng et al. [36] propose WSOD2 to extract object boundary information by fusing top-down class confidence scores and bottom-up object evidence. Zeng et al. [35] propose a classification guided attention mechanism to improve localization performance. Shen et al. [28] join weakly supervised object detection and segmentation tasks by Cyclic Guidance Learning, which helps both tasks to complement each other by counterparty patterns.

The key to WSDDN and its variants to effectively implement object detection is the candidate object region (bounding box) proposal mechanism (such as Selective Search Windows [26] and Edge Boxes [41]). In other words, without the prior region proposal, these models will accomplish nothing. In contrast, the structure we present in this paper can be seen as a generalization of differentiable constraint to any bounding box attention without any prior.

It is worth noting that STN [12] is an end-to-end model introducing spatial transformation capabilities to a standard neural network, whose design of differentiable modules is similar to our work. However, STN could only be considered as an attention mechanism rather than a detection network. This means that STN can only specify a category-independent for classification, and our structure can give explicit category-dependent bounding box.

2.3 Localization by iteratively learning

Different from WSDNN to perform object detection in only one single shot, another set of methods ([1–4, 16, 21, 23, 25, 27, 29]) use the iteratively learning paradigm. Although some models only use WSDDN to complete the initial detection, they are still trained

iteratively. Therefore, in this subsection, we emphasize the training process rather than the structure of the model. For example, inspired by curriculum learning Ref. [1, 16] propose self-paced learning, which starts with easy samples, then consider hard ones in training. After that, Ref. [2, 29] proposed a selection of window space by allowing smaller windows, which improves the selection of samples via the confidence of max scoring window in [16]. Also, [27] uses inter-category competition to select samples.

Most of the methods in this paradigm can use either region proposal or other simple initialization strategies (whole image [4, 21], whole image minus a margin [3, 23, 25]) without relying on region proposal. However, the process of iteratively learning is too complicated and requires much human experience to control the iterative switching conditions. In contrast, our model does not rely on region proposal and only requires a simple one-step training process without iteration.

3 Weakly supervised detection

Our weakly supervised object detection model consists of three modules, as depicted in Fig. 2. The first module, named WSL transfer network, generates Class Activation Maps (CAMs) from the input image after feature extraction. These activation proposals describe a set of confident regions available for the subsequent determination network. The second module, named determination network, is a simple bounding box regression network that produces four bounding box parameters from the activation proposals. The third is a parameterized division module, which generates Shadow Activation Maps (SAMs) that mutual constraint with CAMs. In this section, we present our design decisions for each module and describe the mutual constraint learning process and the entire network topology at last.

3.1 Feature extraction and WSL transfer network

We designed two different capacities backbone networks—one small and one large. They are both four layers deep with three regular convolutional layers and one fully

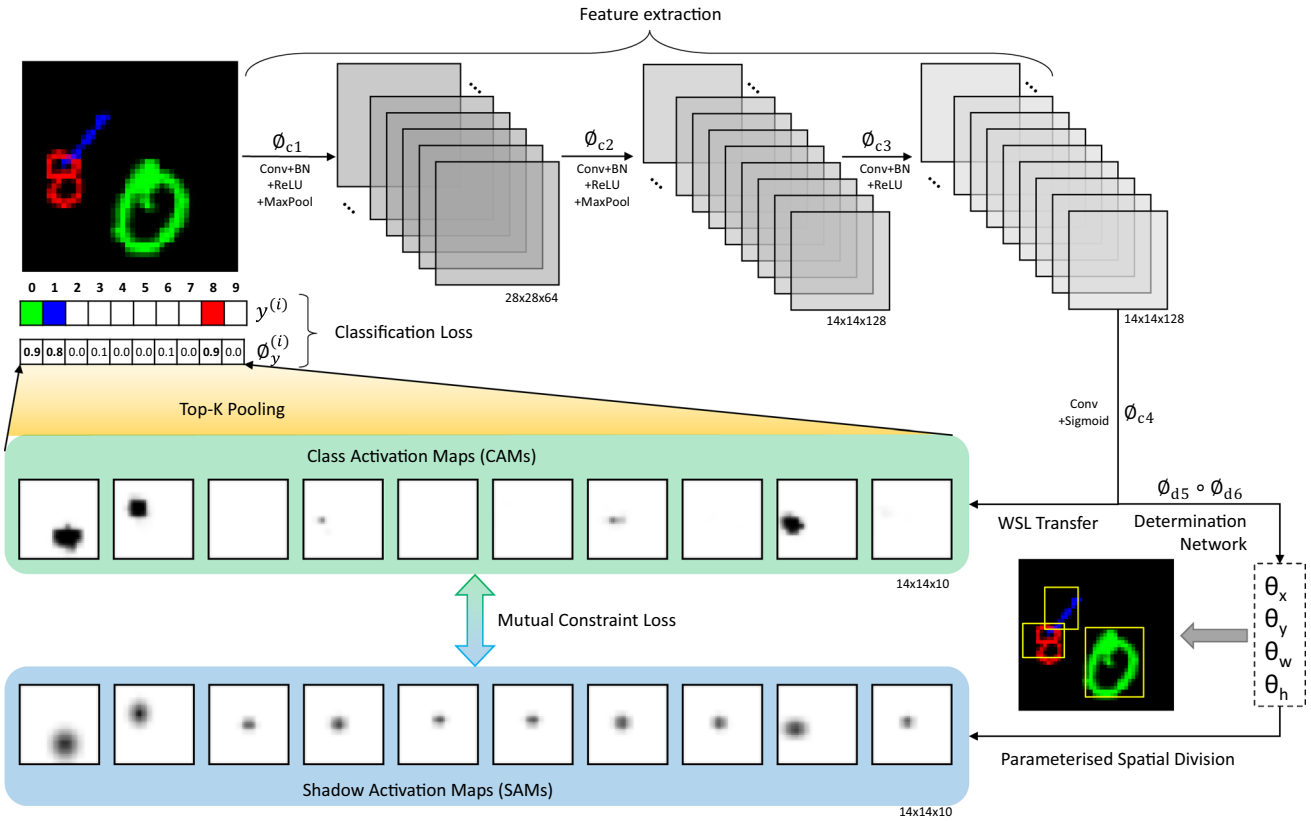

Fig. 2 The architecture of our CNN architecture. First, we use FCN to extract local features from the whole input image. These features are passed to a WSL transfer layer, which output Class Activation Maps. Then, a determination network regresses the spatial division parameters. In addition to image-level labels to WSL, all modules are also learning by a mutual constraint way

convolutional layer. Figure 2 gives an illustration. The convolution stride and padding are both fixed to 1 pixel, which preserves the spatial resolution after convolution. Following some (not all) convolutional layers, we conducted max pooling over a 2×2 pixel window, with stride 2. The activation function (We use ReLU for all those hidden layers in this module) and Batch Normalization (BN) accompany all hidden convolutional layers. A fully convolutional layer with a 3×3 kernel follows this stack of convolutional layers for WSL transfer. Table 1 lists the configurations for these convolutional and pooling layers. The input of our model is a fixed-size 56×56 image with three channels, and the input image passes through a stack of convolutional layers with a 3×3 receptive field to generates Class Activation Maps T, as

$$T = \{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_C\}, \boldsymbol{T}_i \in \mathbb{R}^{H \times W}, \tag{1}$$

where C is the number of categories and $H \times W$ is the shape of the 2-D activation map.

In order to obtain an appropriate CAMs which activated confident regions for each category, we extract relevant regions from global top-k pooling. Specifically, based on recent MIL [8], we use global pooling to collect multiple high scoring regions from CAMs. Formally, $g_{w,h,i} \in \{0, 1\}$ is a binary variable denoting the choice of the (w, h) region from the class-wise CAMs, and $T_{w,h,i}$ is the score of the (w, h) region on CAMs for a given class i. We propose the following aggregation strategy $\phi_i(\cdot)$, which picks the k highest scoring regions as follows:

$$\begin{aligned} &\phi_i(\boldsymbol{T}_i) = \frac{1}{k} \max_{\mathbf{g}} \sum_{(w,h) \in \boldsymbol{T}_i} g_{w,h,i} \cdot T_{w,h,i}, \\ &\text{s.t.} \sum_{(w,h) \in \boldsymbol{T}_i} g_{w,h,i} = k, i = 1, 2, \ldots, C, \end{aligned} \tag{2}$$

where $\mathbf{g} = \{g_{w,h,i}\}$, $w \in \{1; W\}$, $h \in \{1; H\}$ and $\boldsymbol{T}_i$ means CAM for a given class i, $i \in \{1; C\}$. We use $\boldsymbol{\phi} = (\phi_1, \phi_2, , \ldots \phi_C)$ to represent the final result of top-k max pooling, which is also the final classification prediction output of our model.

Table 1 Convolutional layers used in our experiments

Layer	Large feature	Small feature	Kernel	Activation	Pool
ϕ_{c1}	64	32	3	ReLU	2
ϕ_{c2}	128	64	3	ReLU	2
ϕ_{c3}	128	64	3	ReLU	N/A
ϕ_{c4}	10	10	3	N/A	N/A
ϕ_{d5}	64	32	3	ReLU	N/A
ϕ_{d6}	40	40	14	Sigmoid	N/A

In the convolutional layers, the stride and padding are both 1. The stride of pooling layers is all 2

From our model design, it is easy to know that given input image, each of output activation maps after the last fully convolutional layer represents a positive region of a specified object. Following [22], we call the output of the WSL transfer network as Class Activation Maps (CAMs).

3.2 Determination network

The determination network takes as input the CAMs $T \in \mathbb{R}^{H \times W \times C}$ with width W, height H, and C channels. By plugging determination network function $f_{\text{det}}(\cdot)$ into the CAMs, we obtain

$$\boldsymbol{\Theta} = (\theta^1, \theta^2, \ldots, \theta^C) = f_{\text{det}}(T), \tag{3}$$

where $\theta^i = (\theta^i_x, \theta^i_y, \theta^i_w, \theta^i_h)$ encodes bounding box positions for one of the C classes, indexed by i.

The determination network is structurally similar to the localization network in STN [12]. However, the $\boldsymbol{\theta}$ output from our network can uniquely describe a bounding box without any transform. This procedure is more like what we see in fully supervised object detection [24], where the model directly outputs the bounding box (four real-valued numbers for each class) and the probability for category predictions in this bounding box.

$f_{\text{det}}(\cdot)$ is a regression network to produce the spatial division parameters $\boldsymbol{\theta}$, which is formulated as:

$$f_{\text{det}}(\cdot) = \phi_{d5} \circ \phi_{d6}(\cdot), \tag{4}$$

where ϕ_{d5} and ϕ_{d6} represents the related calculations of convolutional layer without pooling, and their configuration is listed in Table 1.

3.3 Parameterized division

Object detection demand to divide the entire input image into two parts: class-specific image regions and the background. Here, we construct a parameterized division layer, to be inserted after the determination network to use spatial parameters to perform this division explicitly.

Suppose that we divide the input image into an $H \times W$ grid. For each grid cell contains a pair of coordinate values (x, y), representing the center of the grid cell. We call this general spatial modeling the grid fields. For a particular class, the grid fields also have a nonnegative value less than one represents the likelihood of an object present in the corresponding region. In this way, we can use a matrix D to represent this spatial modeling. The grid fields should depend on the following three assumptions.

Assumption 1 If the grid with the center coordinate (x, y) is all background, then $D_{x,y} = 0$.

Assumption 2 If the grid contains a region related to the category, then $D_{x,y} > 0$ and

$$\begin{aligned} &\forall\ \mathbb{L}[(x_1, y_1)] < \mathbb{L}[(x_2, y_2)], \\ &\quad D_{x_1,y_1} > D_{x_2,y_2}, \end{aligned} \tag{5}$$

where $\mathbb{L}[p]$ is the European distance from point p to center point of the object.

Assumption 3 The gradient can be back-propagated through the of the parameterized division modules, which is all the partial derivatives of D with respect to the input and θ can be computed.

Both CAMs and SAMs are an instantiation of grid fields. In contrast to CAMs supervised by the image-level label, which explores an essential region to generate class-aware attention maps, the spatial division explicitly generates SAMs from spatial parameters.

We gradually get SAMs in three steps (shown in Fig. 3). (1) Calculate the distance of each grid to the center of the predicted object in each direction of the space (our work only takes two directions of the Euclidean plane coordinates). (2) We implement the three assumptions described above in each direction of space. (3) The final SAMs were calculated by combining the results in all directions. Note that, for the convenience of description in this section, we only consider the case of two categories (the interest object and background).

For the first step, let two constant matrices $\boldsymbol{D}_x^{(1)}$ and $\boldsymbol{D}_y^{(1)}$ represent the calibration coordinates of each grid in the horizontal and vertical directions, given by

$$\begin{cases} \boldsymbol{D}_{x,(w,h)}^{(1)} = \dfrac{w - 0.5}{W} & w = 1 \ldots W,\ h = 1 \ldots H, \\ \boldsymbol{D}_{y,(w,h)}^{(1)} = \dfrac{h - 0.5}{H} & w = 1 \ldots W,\ h = 1 \ldots H, \end{cases} \tag{6}$$

where w, h is matrix subscript, H and W represent the number of rows and columns of SAMs. Then, calculate

$$\begin{cases} \boldsymbol{D}_x^{(2)} = |\boldsymbol{D}_x^{(1)} - \theta_x|, \\ \boldsymbol{D}_y^{(2)} = |\boldsymbol{D}_y^{(1)} - \theta_y|, \end{cases} \tag{7}$$

which represents the distance from the center of each grid to the center of the predicted object in the horizontal and vertical. Note that after this step, we only get the geometric distance, which means the unit for measuring distance here is the grid.

For the second step, we follow **Assumption** 2 by computing $1 - \boldsymbol{D}_x^{(2)}/\theta_w$ and $1 - \boldsymbol{D}_y^{(2)}/\theta_h$. Then, in order to comply with **Assumption** 1, we use $\max(\cdot, 0)$ and plug the previous calculations result into it:

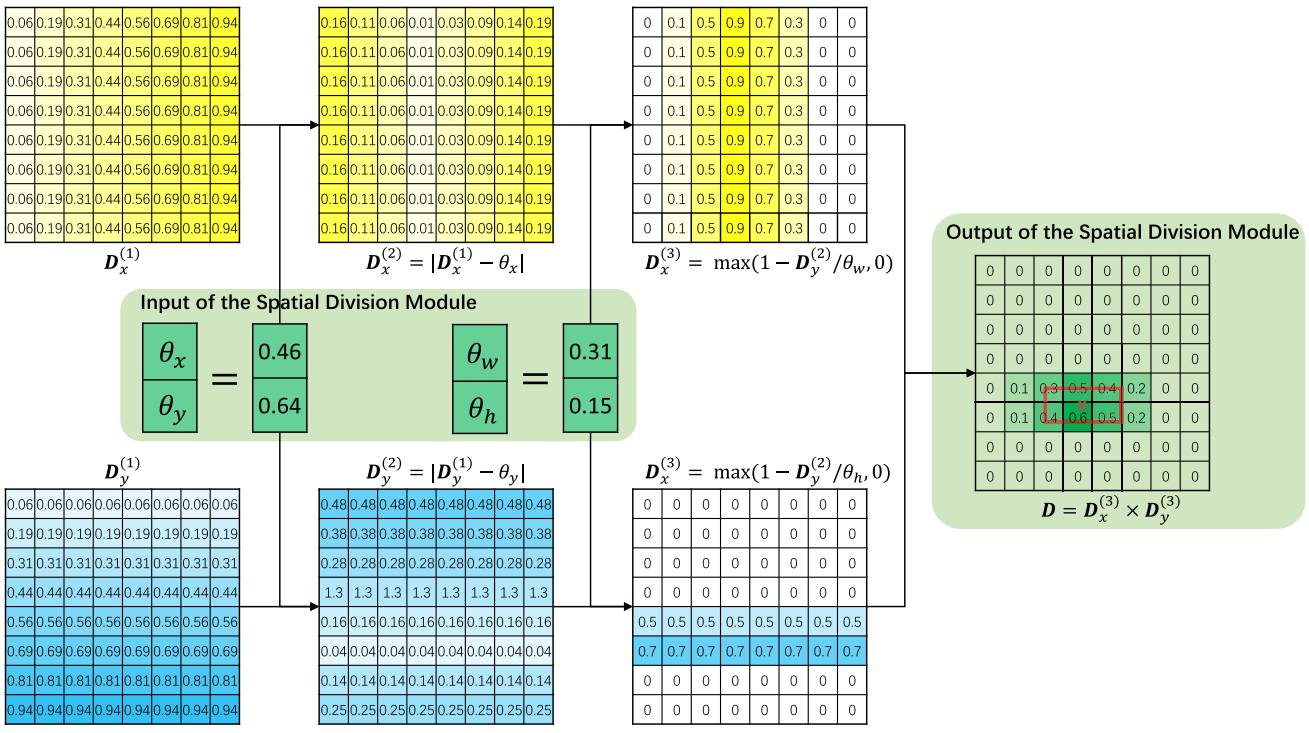

Fig. 3 Example of applying spatial division to get SAMs. In the figure, grids represent the matrices at different steps. The saturation of the color is consistent with the value in the grid, which describes a measure of distance (color figure online)

$$\begin{cases} \boldsymbol{D}_x^{(3)} = \max\left(1 - \boldsymbol{D}_x^{(2)}/\theta_w, 0\right), \\ \boldsymbol{D}_y^{(3)} = \max\left(1 - \boldsymbol{D}_y^{(2)}/\theta_h, 0\right), \end{cases} \tag{8}$$

where $\boldsymbol{D}_x^{(3)}, \boldsymbol{D}_y^{(3)} \in [0,1]^{W \times H}$ and all calculations in this formula are element-wise. The calculation of Eq. (8) first turns the geometric distance into a functional distance. In other words, it is to normalize the geometric distance. The θ_w and θ_h as the denominator means that we can use the width and height predicted from determination network as the boundary for whether the value is zero or not.

For the third step, the calculation needs to determine the multi-directional intersection. Doing so, $\boldsymbol{D}$ is written

$$\boldsymbol{D} = \boldsymbol{D}_x^{(3)} \times \boldsymbol{D}_y^{(3)}, \tag{9}$$

or

$$\boldsymbol{D} = \min(\boldsymbol{D}_x^{(3)}, \boldsymbol{D}_y^{(3)}), \tag{10}$$

where $\boldsymbol{D} \in [0,1]^{W \times H}$ and all calculations in these equations are element-wise. In the second step, we only follow **Assumption** 1 and **Assumption** 2 independently in each direction. In the third step, we let the entire final output jointly follow these two assumptions.

All the equations in this section are consistent with **Assumption** 3. In theory, any spatial distance function can be used, as long as gradients can be defined with $\boldsymbol{\theta}$. For example, we can use

$$\max\left(1 - \sqrt{\left(\frac{\boldsymbol{D}_x^{(1)} - \theta_x}{\theta_w}\right)^2 + \left(\frac{\boldsymbol{D}_y^{(1)} - \theta_y}{\theta_h}\right)^2}, 0\right) \tag{11}$$

directly instead of Eqs. (9)–(11), and in the experimental part, we will compare the differences between different distance functions (Fig. 4).

3.4 Mutual constraint learning

This section describes the procedure for mutual constraint learning. The learning of both CAMs and SAMs is supervised by image-level annotation. Thus, all components can be integrated into single end-to-end training and optimization.

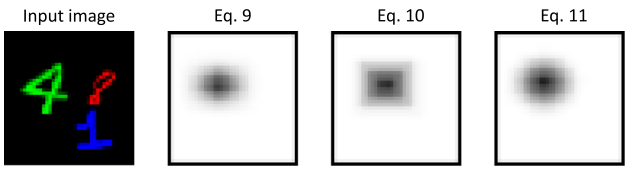

Fig. 4 SAMs generation (digit 4) from different spatial distance functions

The model training objective is derived from the multiple object categories but is extended to mutual constraint between various activation maps for object detection. Specifically, the complete loss consists classification loss ℓ_{class} and mutual constraint loss ℓ_{mutual}. In the training phase, we add the two losses as:

$$\ell = \ell_{\text{class}} + \lambda \ell_{\text{mutual}}. \tag{12}$$

ℓ_{class} is the multi-class cross entropy loss used in [22] which usually measures the probability error in multiple object categories in which each category is independent. ℓ_{mutual} is used to measure the similarity between two feature maps. We hope that through this loss, two feature maps could constrain each other and learn from each other. ℓ_{class} and ℓ_{mutual} are described in detail below.

3.4.1 Classification loss

For C different categories, we simply assume each category is independent, and train the C binary classifiers jointly, using the cross-entropy loss:

$$\begin{aligned} \ell_{\text{class}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\Big[& y_c^{(n)} \log \phi_c^{(n)} \\ & + \left(1 - y_c^{(n)}\right) \log\left(1 - \phi_c^{(n)}\right)\Big] \end{aligned} \tag{13}$$

where $\phi_c^{(n)}$ is the abbreviation for $\phi_c^{(n)}\left(x^{(n)}|\boldsymbol{w}\right)$, $y_c^{(n)}$ represents the ground truth of the n-th sample on class c.

3.4.2 Mutual constraint loss

During training, we also aim at minimizing the mutual constraint loss whose object is to reduce the difference between CAMs and SAMs. The first calculation method we considered in this paper is the $L2$ norm function, as

$$\ell_{\text{mutual}-l2} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C}\sum_{(w,h)=(1,1)}^{W \times H} \mathbb{1}_i^{(n)}\left(T_{w,h,i}^{(n)} - S_{w,h,i}^{(n)}\right)^2 \tag{14}$$

where $\mathbb{1}_i^{(n)}$ denotes if digit i appears in image n.

In order to counteract the standard L2 distance cannot measure the structural similarity of the feature, we also try to use the SSIM structure similarity method to make the last change of the feature and the structural similarity before the change.

The measure between the q-th window of SAMs $S^{(q)}$ and the q-th window of CAMs $T^{(q)}$ of common size 11×11 is:

$$\mathbf{SSIM}(T,S) = \frac{1}{Q}\sum_{q=1}^{Q}\frac{2\mathrm{M}_{T^{(q)}}\mathrm{M}_{S^{(q)}} + \alpha_1}{\mathrm{M}^2_{T^{(q)}} + \mathrm{M}^2_{S^{(q)}} + \alpha_1} \times \frac{2\mathrm{V}_{T^{(q)},S^{(q)}} + \alpha_2}{\mathrm{V}^2_{T^{(q)}} + \mathrm{V}^2_{S^{(q)}} + \alpha_2} \tag{15}$$

where $\mathrm{M}_{T^{(q)}}$ is the average of $T^{(q)}$, $\mathrm{M}_{S^{(q)}}$ is the average of $S^{(q)}$, $\mathrm{V}^2_{T^{(q)}}$ is the variance of $T^{(q)}$, $\mathrm{V}^2_{S^{(q)}}$ is the variance of $S^{(q)}$ and $\mathrm{V}_{T^{(q)},S^{(q)}}$ is the covariance of $T^{(q)}$ and $S^{(q)}$. α_1 and α_2 is set by default as in the original paper [34]. Based on Eq. (15), we introduced mutual-ssim loss, as

$$\ell_{\mathrm{mutual-ssim}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C}\mathbb{1}_i^{(n)}\left(1 - \mathbf{SSIM}\left(T_i^{(n)}, S_i^{(n)}\right)\right) \tag{16}$$

where $\mathbb{1}_i^{(n)}$ denotes if digit i appears in image n.

3.5 Network topology

The combination of the determination network, parameterized division, and mutual constraint learning forms our CNN architecture (shown in Fig. 2).The input of our model is the picture to be detected. The output is digits contained in the picture and the corresponding position information of each number. In the process of training, we only use image-level labels, so the whole process is weakly supervised. All the parameters in our model are trained together in an end-to-end manner. Among them, the parameters in the determination network and parameterized division are only learned by the mutual constraint loss. The parameters of other parts of the network are learned by the classification loss and mutual constraint loss together.

It should be noted that the backbone network used to extract image features in our model does not require pretraining. Image feature extraction is not the focus of this paper. So the structure of the network is as simple as possible to ensure its effectiveness. For specific structural details, refer to the data information in Fig. 2.

4 Experiments

In this section, we describe the experiments on an extended dataset multi-MNIST and report the experimental results. In the first two subsections, we introduce the multi-MNIST datasets used in our experiments and the experimental setup, and then the classification and detection experimental results are presented in the last two subsections.

4.1 Multi-MNIST dataset

In order to verify the multi-label classification capability of the model, based on MNIST, we have extended a new dataset named multi-MNIST, in which there are three digits on each image. Like MNIST, the new dataset contains around 60k images (50k for train and validation, 10k for the test), with the task to recognize and detect the presence of digit in each image. Each digit is presented in a separate 56×56 input channel (giving 3-channel inputs), but each digit is transformed independently, with random scale, and translation. Each sample is randomly selected, and the new position in the image is randomly decided after the scale change. The ratio of scale variances is [0.6, 1.5]. The choice of location will ensure that each digit is complete without exceeding the border of the image. Note that these digits in each image in multi-MNIST only guarantee different samples from MNIST, but there is no guarantee that they are not in the same category. That is, two or three identical numbers may appear in one image. Specifically, when constructing multi-MNIST, the data of the training set are all from the training set of MNIST, the same applies to the test set.

4.2 Experimental setup

4.2.1 Evaluation metrics

Average precision (AP) and the mean of AP (mAP) are used in the quantitative evaluation of classification and detection tasks on the testing set of multi-MNIST. For classification, following the previous researches [7–10, 22], we use the same method as standard protocol in [10] to compute and report AP. For detection, we report average precision (AP) at 50% intersection-over-union (IOU) of the detected boxes with the ground truth ones. Additionally, the F1-measure is considered in the classification task, which is commonly used in previous work.

4.2.2 Implementation details

We reproduce all methods in Table 2, and the configuration in Table 1 constructs their backbone network. For all these reproduced baseline methods and our SDN, we employed the same learning configuration. Specifically, we applied Stochastic Gradient Descent (SGD) with momentum 0.9, weight decay 1e-7, and batch size 200. In total, all models were trained for 130 epochs. The learning rate is initially set to 0.01 and reduced by a factor of ten every 30 epochs. We use Xavier [11] initialization to initialize all the convolutional layers. The number of highest scoring regions

Table 2 Classification performances (mAP and F1) on the multi-MNIST test set

Method	MC loss	S Feature	mAP	F1
DeepMIL	–	Small	96.84	95.54
	SSIM	Small	98.09	96.71
WILDCAT	–	Small	96.91	95.40
	SSIM	Small	98.05	97.29
WELDON	–	Small	96.89	94.98
	L2	Small	97.89	97.02
	SSIM	Small	98.33	97.14
	SSIM	Large	**98.81**	**97.63**

The best performance are highlighted in bold

L2 means using L2-norm loss function for mutual constraint learning and SSIM means using SSIM structure similarity method for mutual constraint learning

k is set to 20, which estimated by cross-validation on the training set.

4.3 Multi-MNIST classification

4.3.1 Results and comparisons

Table 2 shows the comparison of classification performance using the mutual constraint learning on the multi-MNIST test set. We compare our method with three state-of-the-art classic or WSL ConvNets, including DeepMIL [22], WELDON [8], and WILDCAT [7, 9]. The main difference between these three methods is the difference between global pooling methods. For comparison, we use these network structures as the basis, and we experimentally show the changes in the classification performance after adding SDN. As demonstrated in Table 2, all models which use SDN are outperforming the base one. Although the methods presented in this paper are not designed for classification, however, due to the mutual learning between CAMs and SAMs, the model can better locate the features favorable for classification, which is in line with common sense: identifying the relevant region from the image first will make the classification more reasonable. The comparison of the improved results of the three models also proves that the previous global pooling method with a better classification effect is also applicable to SDN.

In Table 2, we also show the impact of different mutual constraint loss and feature capacity choices on classification performance. The experimental results show that using SSIM is slightly better than L2, and increasing the network scale will improve classification performance. Figure 5 shows the impact of three critical hyperparameters on the classification effect. The smaller the k and λ, the more accurate the classification, while the learning rate is the opposite. Because the focus of our model is the detection part, we have not adjusted the model parameters based on these trends.

4.3.2 The influence of mutual constraint learning

As mentioned in the previous subsection, mutual constraint learning contributes to the improvement of classification accuracy. As illustrated in Fig. 6, CAMs of WELDON-L2 with mutual constraint learning contain fewer noise points (blocks) compared to CAMs of WELDON. Not only that, but in the first example of Fig. 6, the CAM under mutual constraint is more relevant in terms of scale (compared to CAMs of WELDON, the size of digit 0 and digit 8 in CAMs of WELDON-L2 is closer to the size of the number in the input image). It is important to note that in the second example of Fig. 6, WELDON-L2 locates the digit 1 and WELDON gets nothing. This also proves that better positioning ability of our model does help to improve classification accuracy.

4.4 Multi-MNIST detection

4.4.1 Results and comparisons

We compare our method with DeepMIL, WELDON, and WILDCAT. When these base methods do not use the SDN configuration, they could only locate the approximate position of the digit (the point with the highest score in the CAMs be mapped to the position in the original image) instead of outputting the bounding box directly. After that, we output a square centered at that position as the bounding box output by these methods. As illustrated in Table 3, our proposed CNN architecture outperforms all the compared methods by clear margins under weak annotation settings. These results demonstrate the excellent performance of our mutual constraint learning for weakly supervised object detection. Besides, spatial distance function has little effect, but the model size raise can improve the detection result significantly. For all these experiments, each number in the test image will only output one bounding box, even if the number appears in the input image multiple times. Furthermore, the model will output the bounding box when the classification probability is higher than 0.5. Figure 7 provides examples of visual results for digit detection.

4.4.2 The influence of hyperparameter

In this part, we mainly analyze the impact of k in Eq. 2, λ in Eq. (12), learning rate and epoch of the training. As shown in Fig. 5, when $k = 20$, the detection effect is the best. Although for classification tasks, the smaller k is, the better the effect, for detection, a suitable activation area can more

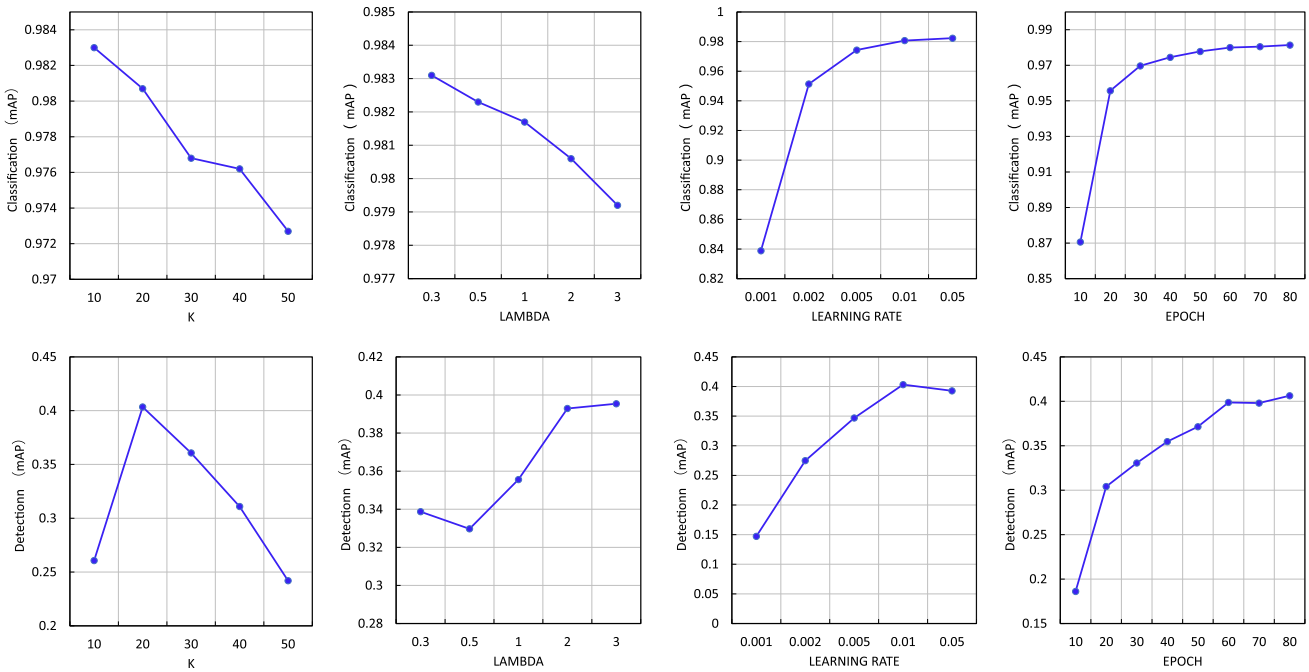

Fig. 5 The effect of different hyperparameters (k in Eq. (2), λ in Eq. (12), learning rate and epoch of the training) on classification and detection results. The basic settings are: $k = 20, \lambda = 2$, learning rate is 0.01 and total 80 epoch. The values in the line chart reflect the results of changing the univariate on the horizontal axis

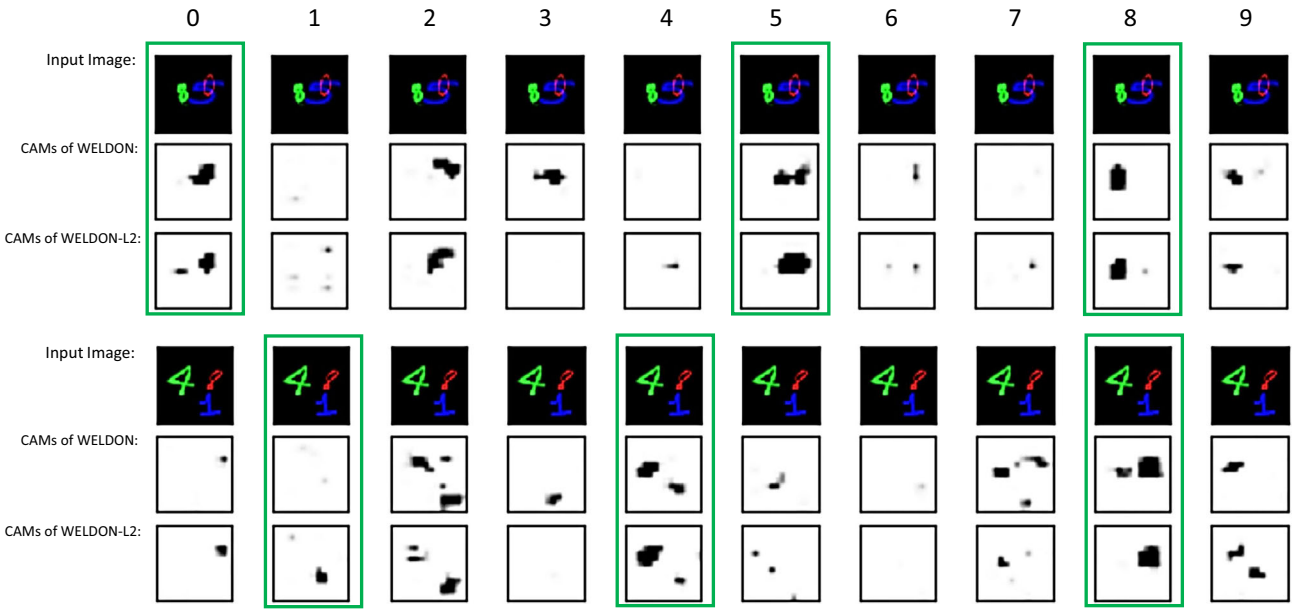

Fig. 6 Two examples of CAMs without/with mutual constraint learning (second/third row of each example). In the green box is the CAM of the ground truth digit. The configuration of mutual constraint learning is WELDON + L2 + Eq. (11) (color figure online)

accurately cover the position of the digit. For λ, it balances the two tasks of classification and detection. Although it seems that the classification effect becomes worse after λ is increased, in fact, it only slows down the convergence rate, and it takes more epoch to achieve the same classification effect. Also, an important point reflected in the third column of Fig. 5 is that the detection is sensitive to the learning rate; that is, too large or too small a learning rate will hurt the detection.

Table 3 Average precision (in %) for different methods on the multi-MNIST test set

Method	MC Loss	Grid Fields	Feature	0	1	2	3	4	5	6	7	8	9	mAP
eepMIL	–	–	Small	10.55	2.27	30.28	23.38	24.45	2.03	8.12	19.72	6.40	15.34	14.25
WELDON	–	–	Small	30.85	2.76	10.97	21.28	17.85	21.21	19.91	10.94	17.243	6.97	16.00
WILDCAT	–	–	Small	32.12	2.05	22.46	21.28	17.43	9.80	20.16	8.93	20.74	8.48	16.35
WELDON	L2	Equation 11	Small	28.10	**49.43**	36.99	45.59	38.10	40.64	38.75	27.52	35.26	**38.43**	37.88
WELDON	SSIM	Equation 9	Small	49.02	38.18	44.02	39.73	38.83	38.34	39.85	32.48	41.85	35.12	39.75
WELDON	SSIM	Equation 10	Small	48.78	38.26	43.31	40.48	37.93	37.89	42.59	32.71	40.83	35.26	39.80
WELDON	SSIM	Equation 11	Small	47.72	37.13	**44.58**	37.44	37.11	**44.22**	44.53	31.73	42.59	35.74	40.28
WELDON	SSIM	Equation 11	Large	**49.19**	40.61	38.66	**46.18**	**41.60**	40.73	**46.22**	**40.87**	**45.59**	37.82	**42.75**

The best performance are highlighted in bold

The upper part shows results using current WSL model. The lower part shows the results of our models with different mutual constraint loss and spatial distance function

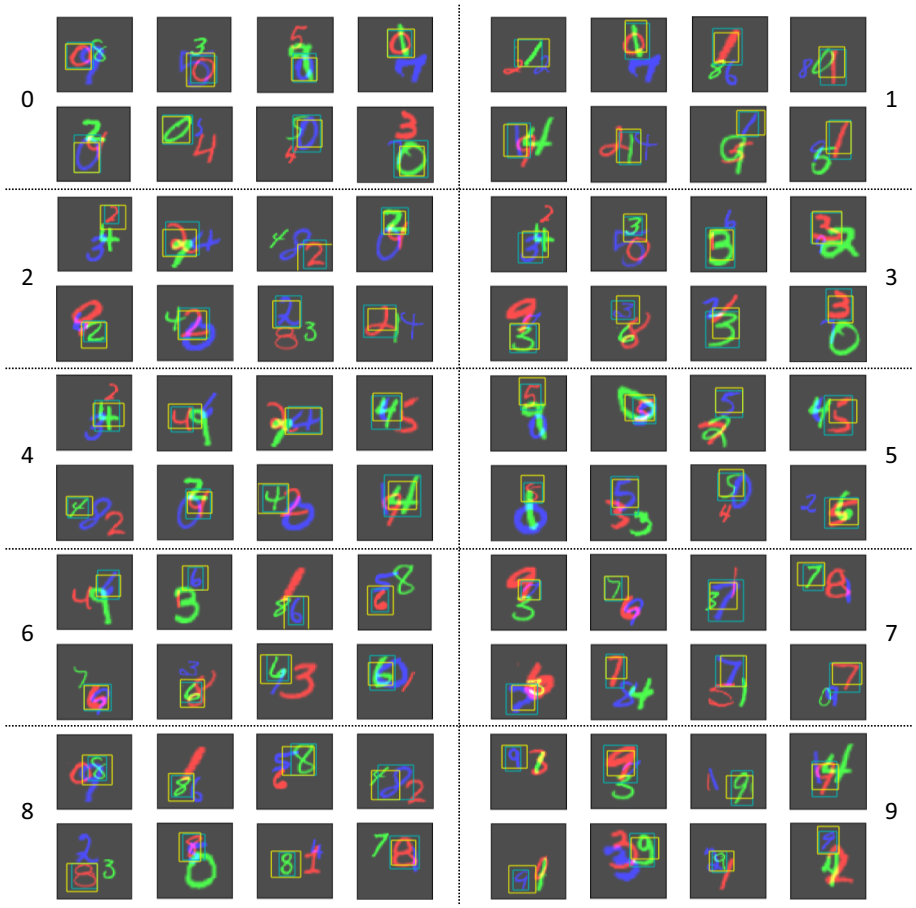

Fig. 7 Qualitative detection results of our method (WELDON + SSIM + Eq. (11)). Yellow bounding boxes indicate objects detected by our method, while cyan ones correspond to ground truth (color figure online)

4.4.3 The influence of mutual constraint loss

From the results in Table 3, we observed that SSIM works better for digits with close aspect ratios (for example, 0, 4, 5). This phenomenon is also seen in Fig. 8. When L2 is used, the aspect ratio of the region corresponding to the digit "1" in SAM is more realistic than the counterpart of using SSIM. The above shows that although the SSIM can better locate the position of the object, the ability to express the shape of the object is insufficient. Besides, Fig. 8 shows the fundamental difference between the two kinds of loss functions in the generation process of SAMs. Using SSIM loss, SAM covers the entire area as much as possible at the beginning of training, and then the coverage area gradually

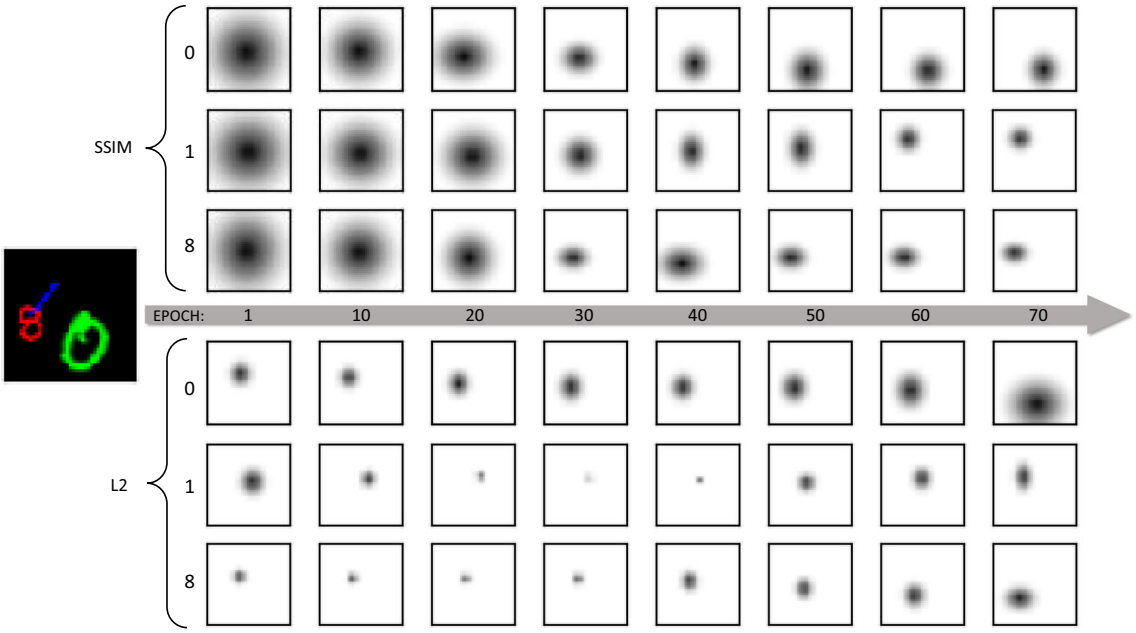

Fig. 8 Visualization of SAMs during learning. The upper part uses SSIM loss, and the lower part uses L2 loss. The three lines of each part correspond to the SAMs with the digit 0, 1, 8, respectively

decreases and converges to the correct position. Using L2 loss, SAM has a small coverage area at the beginning of training. During the learning process, it gradually explores the correct position and gradually converges from small to large to the correct shape and size. We have observed a large number of SAMs in the experiment and found that using L2 loss requires a longer learning time than using SSIM loss because the SAM coverage area is too small in the initial training. This makes the generation of CAMs and SAMs asynchronous, and their mutual constraints are not sufficient. Therefore, the performance of using L2 loss is not optimal, either for classification or detection.

4.5 Discussion

The section will provide some discussion of the convergence and robustness of our proposed method. It mainly involves the limitations of the model found during our experiments.

The changes in classification and detection results as epoch increases during training are shown in the last column of Fig. 5. The classification results converge more smoothly than object detection. In fact, there is a possibility of non-convergence when using L2 as MC Loss. The specific manifestation is that the detection of few numbers will fail. This may be due to the fact that in this case, the area covered by the nonzero values of the SAM at the beginning of the training is too small, as shown in Fig. 8.

During the detection process, when the same number appears multiple times in the input image, the detection often fails. Especially when they are next to each other or overlapped, the model cannot distinguish between them. This is due to the limitations of our approach. Our SDN only outputs a single predicted bounding box for the same number, which is not sufficient for multiple occurrences.

5 Additional experiments

In this section, we further show that the proposed model gets competitive results across a realistic dataset. To this end, we report results on the task of beverage detection in a benchmark dataset for smart unmanned vending machines (UVM) [37]. The UVM dataset contains a total of 34,052 images containing beverages (10 categories in total). We selected the images with no more than three objects to form a subset of the data suitable for our model. The number of available images is 17, 579. We randomly selected 80% of the images to form the training set and used the remaining images for testing.

We reused the best architecture (WELDON + SSIM + Large feature) in Sect. 4 for beverage detection. Our network was not pre-trained with any other natural image dataset. We report the results by mAP and compared with fully supervised methods to demonstrate the ability of weakly supervised detection methods to be applied in a realistic environment.

Table 4 shows the obtained results. In fact, the table comparison is not rigorous. The YOLOv3 result from [37] in the table uses the entire dataset rather than a subset

Table 4 Comparison of results between our approach and the fully supervised approach

Method	mAP	Weakly/Fully
YOLOv3 [37]	91.81	–
WELDON	48.95	53%
WELDON+SSIM+Large	54.45	59%

In the top half of the table, YOLOv3 is a fully supervised detection model with results from [37]. WELDON is the base baseline in the bottom half of the table, and WELDON+SSIM+Large is the weakly supervised method we propose. The percentage in the third column indicates how much the weakly supervised method achieves the fully supervised method in terms of detection results

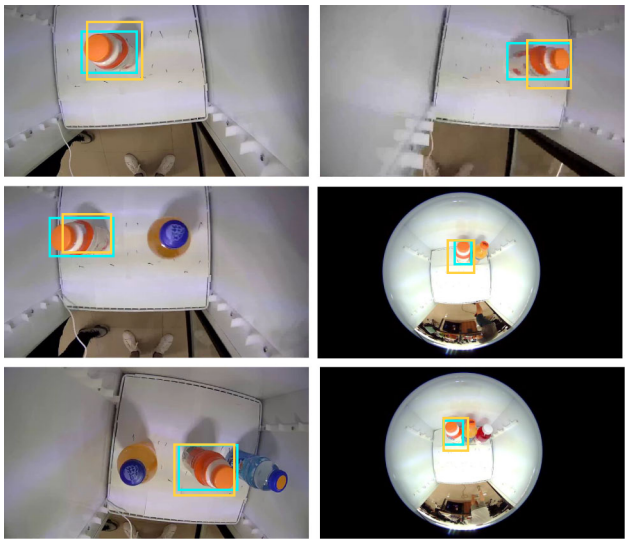

Fig. 9 Examples of visual results for beverage detection in category "D." Yellow bounding boxes indicate beverage detected by our method, while cyan ones correspond to ground truth. From top to bottom, each row represents the case where the input image contains one, two, and three objects (color figure online)

of the data as we define it. However, the comparison of the cases in the table still shows the gap in detection results between the weakly supervised and fully supervised methods. There are three possible reasons for the gap: first, the feature extraction part of our network is not pre-trained using other datasets; second, our backbone network structure is relatively simple and has limited feature extraction capabilities; and third, the limitations of weak supervision itself. Figure 9 provides examples of visual results for beverage detection.

6 Conclusion

We proposed a mutual constraint learning approach for object detection in a weakly supervised scenario, aiming at generating a predicted bounding box through object localization maps directly. This work paves a simple yet entirely new way to mine object regions only with a classification network.

Using the proposed approach, we achieved improved results on a multi-MNIST dataset we created ourselves based on MNIST. Finally, we presented a thorough analysis of the main components of our proposed approach, showing the effect of our design choices and allowing other authors to build on our method, possibly choosing those components which best fit with other application.

Acknowledgements This work was supported by National Key R&D Program of China under Grant 2018YFC0808304, and in part by the National Science Foundation of China under Grant 61976043 and Grant 61573081.

Compliance with ethical standards

Conflict of interest The authors confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

1. Achutti A, Achutti VR (2009) Curriculum learning. In: International conference on machine learning (ICML). ACM, Montreal, pp 41–48. https://doi.org/10.1017/s1047951100000925
2. Bilen H, Namboodiri VP, Van Gool LJ (2014) Object and action classification with latent window parameters. Int J Comput Vis 106(3):237–251
3. Bilen H, Pedersoli M, Namboodiri VP, Tuytelaars T, Van Gool L (2014) Object classification with adaptable regions. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 3662–3669
4. Bilen H, Pedersoli M, Tuytelaars T (2014) Weakly supervised detection with posterior regularization. In: British machine vision conference, Nottingham, pp 1–12
5. Bilen H, Vedaldi A (2016) Weakly supervised deep detection networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, Nevada, pp 2846–2854
6. Diba A, Sharma V, Pazandeh A, Pirsiavash H, Van Gool L, Leuven K (2017) Weakly supervised cascaded convolutional networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, Hawaii, pp 914–922
7. Durand T, Mordan, T, Thome N, Cord M (2017) WILDCAT: weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: The IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, Hawaii, vol 2, pp 5957–5966
8. Durand T, Thome N, Cord M (2016) WELDON: Weakly supervised learning of deep convolutional neural networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, Nevada, pp 4743–4752
9. Durand T, Thome N, Cord M (2018) Exploiting negative evidence for deep latent structured models. IEEE Trans Pattern Anal Mach Intell 41:337–351
10. Everingham M, Winn J (2011) The PASCAL visual object classes challenge 2012 (VOC2012) development kit, Pattern Analysis, Statistical Modelling and Computational Learning. Tech Rep 1(1):1–32

11. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, Sardinia, pp 249–256
12. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. In: Advances in neural information processing systems (NIPS), Montreal, pp 2017–2015
13. Jiang W, Zhao Z, Su F (2018) Weakly supervised detection with decoupled attention-based deep representation. Multimed Tools Appl 77(3):3261–3277
14. Kantorov V, Oquab M, Cho M, Laptev I (2016) ContextLocNet: context-aware deep network models for weakly supervised localization. In: European conference on computer vision (ECCV), pp 350–365. https://doi.org/10.1007/978-3-319-46448-0
15. Kosugi S, Yamasaki T, Aizawa K (2019) Object-aware instance labeling for weakly supervised object detection. In: The IEEE conference on computer vision and pattern recognition (CVPR), Long Beach, CA, pp 6064–6072
16. Kumar MP, Packer B, Koller D (2010) Self-paced learning for latent variable models M. In: Advances in neural information processing systems (NIPS), Vancouver, pp 1189–1197
17. Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint p. arXiv:1312.4400
18. Liu Y, Chen W, Mahmud SMH, Qu H (2019) Mutual constraint learning for weakly supervised object detection. In: IEEE 14th international conference on intelligent systems and knowledge engineering
19. Murtza I, Khan A, Akhtar N (2019) Object detection using hybridization of static and dynamic feature spaces and its exploitation by ensemble classification. Neural Comput Appl 31(2):347–361
20. Neri P, Heeger DJ (2002) Spatiotemporal mechanisms for detecting and identifying image features in human vision. Nat Neurosci 5(8):812–816
21. Nguyen MH, Torresani L, de la Torre F, Carsten (2009) Weakly supervised discriminative localization and classification: a joint learning approach. In: IEEE international conference on computer vision, Kyoto, pp 925–1932
22. Oquab M, Bottou L, Laptev I, Sivic J (2015) Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), Boston, Massachusetts, pp 685–694
23. Pandey M, Lazebnik S (2011) Scene recognition and weakly supervised object localization with deformable part-based models megha pandey and svetlana lazebnik. In: The IEEE conference on computer vision and pattern recognition (CVPR), Colorado Springs, pp 1307–1314
24. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: The IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, Hawaii, pp 7263–7271
25. Russakovsky O, Lin Y, Yu K, Fei-Fei L (2012) Object-centric spatial pooling for image classification. In: European conference on computer vision (ECCV), Florence, pp 1–15
26. Sande KVD (2011) Segmentation as selective search for object recognition. In: The IEEE international conference on computer vision (ICCV), vol 1, p 7. Colorado Springs. https://doi.org/10.1109/ICCV.2011.6126456
27. Sangineto E, Nabi M, Culibrk D, Sebe N (2018) Self paced deep learning for weakly supervised object detection. IEEE Trans Pattern Anal Mach Intell 41(3):712–725
28. Shen Y, Ji R, Wang Y, Wu Y, Cao L (2019) Cyclic guidance for weakly supervised joint detection and segmentation. In: The IEEE conference on computer vision and pattern recognition (CVPR), Long Beach, CA, pp 697–707
29. Shi Z, Yang Y, Hospedales TM, Xiang T (2014) Weakly supervised learning of objects, attributes and their associations. In: European conference on computer vision (ECCV), Springer, pp 472–487
30. Sun C, Paluri M, Collobert R, Nevatia R, Bourdev L (2016) ProNet: learning to propose object-specific boxes for cascaded neural networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, Nevada, pp 3485–3493
31. Tang P, Wang X, Bai X, Liu W (2017) Multiple instance detection network with online instance classifier refinement. In: The IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, Hawaii, vol 1, pp 2843–2851. https://doi.org/10.1109/CVPR.2017.326
32. Vo T, Nguyen T, Le CT (2019) A hybrid framework for smile detection in class imbalance scenarios. Neural Comput Appl 31(12):8583–8592
33. Wang J, Wang N, Li L, Ren Z (2020) Real-time behavior detection and judgment of egg breeders based on YOLO v3. Neural Comput Appl 32(10):5471–5481
34. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
35. Yang K, Li D, Dou Y (2019) Towards precise end-to-end weakly supervised object detection network. In: Proceedings of the IEEE international conference on computer vision (ICCV), Seoul, pp 8372–8381
36. Zeng Z, Liu B, Fu J, Chao H, Zhang L (2019) WSOD2: learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), Seoul, pp 8292–8300
37. Zhang H, Li D, Ji Y, Zhou H, Wu W, Liu K (2019) Towards new retail: a benchmark dataset for smart unmanned vending machines. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2019.2954956
38. Zhang X, Feng J, Xiong H, Tian Q (2018) Zigzag learning for weakly supervised object detection. In: The IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, Utah, pp 4262–4270
39. Zhang Y, Bai Y, Ding M, Li Y, Ghanem B (2018) W2F: a weakly-supervised to fully-supervised framework for object detection. In: The IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, Utah, pp 928–936
40. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: The IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, Nevada, pp 2921–2929
41. Zitnick CL, Dollár P (2014) Edge boxes: locating object proposals from edges. In: European conference on computer vision (ECCV), Springer, Zurich, pp 391–405
42. Zhang M, Luo X, Chen Y, Wu J, Belatreche A, Pan Z, Qu H, Li H (2020) An efficient threshold-driven aggregate-label learning algorithm for multimodal information processing. IEEE J Sel Top Signal Process 14(3):592–602
43. Zhang M, Qu H, Belatreche A, Chen Y, Zhang Y (2018) A highly effective and robust membrane potential-driven supervised learning method for spiking neuron. IEEE Trans Neural Netw Learn Syst 30(1):123–137